

**LAPORAN PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK
MODUL 6**



**NAMA : M TAQIYYUDDIN F
NIM : 105219039**

**PROGRAM STUDI ILMU KOMPUTER
FAKULTAS SAINS DAN ILMU KOMPUTER
UNIVERSITAS PERTAMINA 2021**

1. Pendahuluan

a. Inti Sari

Pada praktikum di modul ini, dibuat suatu program dengan menerapkan teknik polymorphism. Polymorphism adalah cara kerja suatu objek yang dapat berubah bentuk ke class lain. Metode yang digunakan pada praktikum saat ini ialah menggunakan cara dinamis (overriding) dan downcasting. Overriding adalah suatu cara dimana mengubah suatu method yang diturunkan oleh parent class, diubah di bagian child class yang mengubahnya. Sedangkan downcasting adalah, mengubah suatu objek dari parent class, menjadi objek dari child class.

b. Rumusan Masalah

- Bagaimana class overriding bekerja ?
- Bagaimana class downcasting bekerja ?

2. Pembahasan

```
public class Biome1 {  
    protected int chunkSize;  
    protected double[] orePercentage = new double[6]; // (coal, iron, copper, gold, lapis, diamond)
```

Pada percobaan ini, dibuat suatu class parent yang akan menggunakan teknik downcasting, bernama Biome1. Dengan attribute chunkSize sebagai jumlah chunk yang digunakan, serta orePercentage yang mendistribusikan ore yang terdapat pada biome tersebut, yang berupa array dari 6 jenis ore dari coal hingga diamond.

```
void biomeGenerate() {  
    System.out.println("Biome in this area is generating....");  
}  
  
void oreGenerate(double[] percentage) {  
    System.out.println("Ore in this biome generating....");  
    int i = 0;  
    for(double item : percentage) {  
        orePercentage[i] = item;  
        i = i + 1;  
    }  
}
```

Pada percobaan ini, dibuat suatu method biomeGenerate, berfungsi untuk menampilkan output yang berupa biome tersebut sedang dibentuk, dan method oreGenerate yang menampilkan output bahwa ore juga sedang dibentuk, dengan mengambil nilai argument yang diberikan sebagai persentase ore di objek biome tersebut.

```
public class Taiga extends Biome1 {
    private int totalVillage;
    private int totalOre;
    private String woodType = "Spruce";
    private int totalTrees;
    private int totalCaves;
}
```

Pada percobaan ini, dibuat suatu class child Taiga, dengan inheritance dari parent class Biome1. Pada class ini dibuat suatu attribute berupa totalVillage, totalOre, totalTrees, dan totalCaves, yang kegunaannya menghitung total berturut-turut sebuah desa villager, ore pada chunk, pohon, dan gua. Dan juga, dibuat suatu attribute woodType sebagai jenis pohon yang dominan tumbuh pada biome taiga, dimana pohon tersebut berupa spruce.

```
public Taiga(int chunkSize, int totalVillage) {
    this.chunkSize = chunkSize;
    this.totalVillage = totalVillage;
}
```

Pada percobaan ini, dibentuk suatu constructor dengan argument chunkSize dan totalVillage, yang memberi nilai berturut-turut attribute dari parent yaitu chunkSize, dan attribute dari Taiga yaitu totalVillage.

```
@Override
public void biomeGenerate() {
    System.out.println("Taiga biome is generating...");
    this.totalOre = (int) (this.chunkSize * 17000 * 0.014); // 1 chunk kisaran 17000 blok darat, 1 chunk kisaran 230 ore
    this.totalTrees = this.chunkSize * 4;
    this.totalCaves = (int) (this.chunkSize * 0.25) + 1;
}

@Override
void oreGenerate(double[] percentage) {
    System.out.println("Ore in Taiga biome generating...");
    int i = 0;
    for(double item : percentage) {
        orePercentage[i] = item;
        i = i + 1;
    }
}
```

Pada percobaan ini, digunakan teknik override kepada dua method dari parent class tersebut dengan proses maupun isi dari method yang didalam child class berubah. Method biomeGenerate merubah output serta menambahkan kode yang memberikan nilai pada atribut child, yaitu totalOre, totalTrees, dan totalCaves, dengan perhitungan masing-masing. Sedangkan pada method oreGenerate, yang diubah hanyalah output didalam method tersebut.

```

public void biomeStats() {
    System.out.println("Currently biome stats :");
    System.out.println("Biome name : Taiga");
    System.out.println("Chunk size : " + this.chunkSize);
    System.out.println("    Total non-air block : " + this.chunkSize * 17000);
    System.out.println("Dominant types of tree is " + this.woodType + ", total Tree : " + this.totalTrees);
    System.out.println("Total caves : " + this.totalCaves);
    System.out.println("Total village : " + this.totalVillage);
    System.out.println("    Total villager : " + (this.totalVillage * 6 + (chunkSize % 4)) + "\n");
}

public void oreStats() {
    System.out.println("Currently ore in taiga biome stats :");
    System.out.println("Total Ore : " + this.totalOre);
    System.out.println("    1. Coal : " + (int) (this.totalOre * this.orePercentage[0]));
    System.out.println("    2. Iron : " + (int) (this.totalOre * this.orePercentage[1]));
    System.out.println("    3. Copper : " + (int) (this.totalOre * this.orePercentage[2]));
    System.out.println("    4. Gold : " + (int) (this.totalOre * this.orePercentage[3]));
    System.out.println("    5. Lapis : " + (int) (this.totalOre * this.orePercentage[4]));
    System.out.println("    6. Diamond : " + ((int) (this.totalOre * this.orePercentage[5])) + "\n");
}

```

Pada percobaan ini, dibuat suatu method pada child class Taiga yaitu biomeStats yang memberikan deskripsi mengenai biome tersebut, mengenai penjelasan total dari gua, chunk dan total blok, villager dan village, dan lainnya. Beserta oreStats yang memberikan deskripsi mengenai jumlah ore masing-masing yang terdapat pada biome tersebut, baik dari coal hingga diamond (terdiri dari 6 ore).

```

public class Biome2 {
    protected int chunkSize;
    protected int totalOre;
    protected int totalCaves;
    protected double[] orePercentage = new double[6]; // (coal, iron, copper, gold, lapis, diamond)
}

```

Pada percobaan ini, dibuat suatu class sebagai parent class yang menerapkan metode downcasting, bernama Biome2. Class tersebut memiliki atribut berupa chunkSize, totalOre, totalCaves, dan orePercentage. Penggunaan totalOre dan totalCaves pada parent class untuk method yang digunakan pada class ini.

```

public void setChunkSize(int chunkSize) {this.chunkSize = chunkSize;}

```

Pada percobaan ini, dibuat setter dari chunkSize untuk mendapatkan nilai dari atribut class Biome2 chunkSize.

```

void biomeGenerate() {
    System.out.println("Biome in this area is generating....");
    this.totalOre = (int) (chunkSize * 17000 * 0.014); // 1 chunk kisaran 17000 blok darat, 1 chunk kisaran 230 ore
    this.totalCaves = (int) (chunkSize * 0.25) + 1;
}

void oreGenerate(double[] percentage) {
    System.out.println("Ore in this biome generating....");
    int i = 0;
    for(double item : percentage) {
        orePercentage[i] = item;
        i = i + 1;
    }
}

```

Pada percobaan ini, dibuat suatu method berupa biomeGenerate yang memberikan output penjelasan biome sedang dibentuk, dimana totalOre dan totalCaves dikalkulasi berdasarkan chunkSize yang didapatkan. Dan juga, dibuat method oreGenerate yang menerima argument array persentasi untuk pendistribusian tiap-tiap ore yang dibentuk pada biome tersebut.

```
public class Darkwood extends Biome2 {
    private String woodType = "Dark Oak";
```

Pada percobaan ini, dibuat suatu class Darkwood, yang memiliki parent class berupa Biome2, dengan attribute yang dibentuk adalah woodType, dengan nilai atribut berupa Dark Oak.

```
public void biomeStats() {
    System.out.println("Currently biome stats :");
    System.out.println("Biome name : Darkwood");
    System.out.println("Chunk size : " + this.chunkSize);
    System.out.println("    Total non-air block : " + this.chunkSize * 17000);
    System.out.println("Dominant types of tree is " + this.woodType + ", total Tree : " + this.chunkSize * 4);
    System.out.println("Total caves : " + this.totalCaves + "\n");
}

public void oreStats() {
    System.out.println("Currently ore in darkwood biome stats :");
    System.out.println("Total Ore : " + this.totalOre);
    System.out.println("    1. Coal : " + (int) (this.totalOre * this.orePercentage[0]));
    System.out.println("    2. Iron : " + (int) (this.totalOre * this.orePercentage[1]));
    System.out.println("    3. Copper : " + (int) (this.totalOre * this.orePercentage[2]));
    System.out.println("    4. Gold : " + (int) (this.totalOre * this.orePercentage[3]));
    System.out.println("    5. Lapis : " + (int) (this.totalOre * this.orePercentage[4]));
    System.out.println("    6. Diamond : " + ((int) (this.totalOre * this.orePercentage[5])) + "\n");
}
```

Pada percobaan ini, dibuat suatu method yaitu biomeStats, dengan perhitungan jumlah pohon dihitung didalam method tersebut, beserta method oreStats.

```
public static void main(String[] args) {
    Taiga taiga1 = new Taiga(25, 2);
    taiga1.biomeGenerate();
    double[] percentage = {0.25, 0.15, 0.15, 0.15, 0.2, 0.1}; // (coal, iron, copper, gold, lapis, diamond)
    taiga1.oreGenerate(percentage);
}
```

Pada percobaan ini, dibuat class yang menjalankan keseluruhan program, yang dimulai oleh pembuatan objek taiga1 dengan class taiga yang menerima argument sizeChunk 25 dan totalVillage 2. Lalu, dipanggil method biomeGenerate dari objek taiga1 yang akan membentuk biome tersebut. Setelah itu, dibuat suatu array yang berisi persentase distribusi ore yang akan digunakan pada saat pemanggilan method oreGenerate oleh objek taiga1. Method biomeGenerate dan oreGenerate tersebut didapatkan dari cara override yang dibentuk class Taiga.

```
taiga1.biomeStats();
taiga1.oreStats();
```

Pada percobaan ini, objek taiga1 menggunakan method biomeStats() dan oreStats() untuk menampilkan detail deskripsi mengenai biome pada objek tersebut.

```
Biome2 biome = new Darkwood();
biome.setChunkSize(17);
double[] percentage2 = {0.22, 0.16, 0.16, 0.16, 0.23, 0.07}; // (coal, iron, copper, gold, lapis, diamond)
biome.biomeGenerate();
biome.oreGenerate(percentage2);
```

Pada percobaan ini, dibuat suatu objek yaitu biome, sebagai objek parent dengan teknik upcasting dari class child darkwood dengan parent class Biome2. Setelah itu, dipanggil method setChunkSize() untuk mendapatkan nilai chunkSize objek biome.

Selanjutnya dibuat suatu array distribusi ore dengan nilai berbeda, yang akan digunakan setelah pemanggilan method `biomeGenerate()` untuk pembentukan biome, pada method `oreGenerate` dari objek biome dalam pembentukan ore.

```
Darkwood darkwood1 = (Darkwood)biome;  
  
System.out.println("=====");  
darkwood1.biomeStats();  
darkwood1.oreStats();
```

Pada percobaan ini, dibuat suatu objek `darkwood1`, menggunakan teknik downcasting eksplisit dari objek parent biome. Nilai maupun proses dari atribut dan method parent objek akan dimiliki oleh child objek `darkwood1`. Setelah itu, data dari objek `darkwood1` akan dideskripsikan menggunakan method `biomeStats()` dan `oreStats` yang dipanggil.

3. Kesimpulan

- Teknik polymorphism dengan cara override merupakan suatu cara yang mengubah bentuk method yang diberikan oleh parent class kepada child class dengan penggunaan yang unik berdasarkan proses atau hasil yang diperlukan oleh child class tersebut. Seperti pada parent class Biome1 dengan child class Taiga dimana objek dari class Taiga override keseluruhan method pada class Biome1 menjadi method berdasarkan kebutuhan class Taiga.
- Teknik polymorphism dengan cara downcasting merupakan suatu cara yang mengubah objek dari suatu parent class berubah menjadi child objek, dengan atribut serta method yang dibawa dari parent objek itu sendiri.

4. Tes Akhir

```
d:\1. Tugas\Coding\1. Semester 4\Praktikum\Praktikum6 - VS Code Console
Taiga biome is generating....
Ore in Taiga biome generating....
=====
Currently biome stats :
Biome name : Taiga
Chunk size : 25
    Total non-air block : 425000
Dominant types of tree is Spruce, total Tree : 100
Total caves : 7
Total village : 2
    Total villager :13

Currently ore in taiga biome stats :
Total Ore : 5950
    1. Coal : 1487
    2. Iron : 892
    3. Copper : 892
    4. Gold : 892
    5. Lapis : 1190
    6. Diamond : 595

=====
=
=====
Biome in this area is generating....
Ore in this biome generating....
=====
Currently biome stats :
Biome name : Darkwood
Chunk size : 17
    Total non-air block : 289000
Dominant types of tree is Dark Oak, total Tree : 68
Total caves : 5

Currently ore in darkwood biome stats :
Total Ore : 4046
    1. Coal : 890
    2. Iron : 647
    3. Copper : 647
    4. Gold : 647
    5. Lapis : 930
    6. Diamond : 283

Press any key to continue . . . .
```

Gambar 4.1. Hasil Output Dari Program Tersebut

Folder package terdapat pada file terkompresi.