
Nama : Muhammad Alfarezzi Fallevi

Kelas : 4IA17

NPM : 50421905

Laporan Akhir 2

Praktikum Rekayasa Perangkat Lunak 2

1. Jelaskan konsep utama dalam object-oriented programming(OOP) seperti encapsulation, inheritance, dan polymorphism. Berikan contoh penerapan setiap konsep ini dalam program java

2. buatlah sebuah program yang memanfaatkan konsep-konsep OOP dan berikan penjelesannya.

1.- Encapsulation

Encapsulation biasa dikenal dengan information hiding adalah kombinasi antara data dan method pada sebuah class yang ditunjukan untuk menyembunyikan detail dari user (pengakses class) terhadap suatu object.

Dalam OOP, encapsulation dapat menggunakan acces modifier sebagai berikut:

- Private : hanya dapat diakses oleh class itu sendiri
- Public : dapat diakses dari class manapun
- Protected : hanya dapat diakses dari package (folder) dan subclass

Contoh Encapsulation

```
10  - //
11  class Mahasiswa {
12      private String nama;
13      private int nim;
14
15      public Mahasiswa(String nama, int nim) {
16          this.nama = nama;
17          this.nim = nim;
18      }
19
20      public String getNama() {
21          return nama;
22      }
23
24      public void setNama(String nama) {
25          this.nama = nama;
26      }
27
28      public int getNim() {
29          return nim;
30      }
31
32      public void setNim(int nim) {
33          this.nim = nim;
34      }
35  }
36
37  public class Main {
38      public static void main(String[] args) {
39          Mahasiswa mhs = new Mahasiswa("Firman", 50421905);
40          System.out.println("Nama: " + mhs.getNama());
41          System.out.println("NIM: " + mhs.getNim());
42
43          mhs.setNama("Levi");
44          System.out.println("Nama Baru: " + mhs.getNama());
45      }
46  }
```

- Inheritance

Inheritance adalah suatu proses dimana suatu class yang bisa disebut super class dapat mewarisi sifatnya pada class turunannya yaitu subclass. Superclass akan mewarisi nilai dari attribute atau behaviour pada kelas turunannya. Sistem penurunan sifat ini jika dilakukan secara hirarki atau bertingkat-tingkat, maka semakin kebawah, akan semakin detail.

Dengan menggunakan kata kunci `extends` setelah deklarasi nama sub class kemudian di ikuti dengan deklarasi nama super class-nya.

Jika pada super class terdapat sebuah method, maka method tersebut secara otomatis dapat diwariskan kepada class turunannya atau subclass. Sehingga hanya perlu sekali saja menulis method pada super class dan bisa digunakan oleh semua class turunannya atau subclass.

Contoh Inheritance

```
8 |
9 | * @author WishperingWhite
10 | */
11 |
12 | class Person {
13 |     protected String nama;
14 |
15 |     public Person(String nama) {
16 |         this.nama = nama;
17 |     }
18 |
19 |     public void perkenalan() {
20 |         System.out.println("Halo, nama saya " + nama);
21 |     }
22 | }
23 |
24 | class Dosen extends Person {
25 |     private String nidn;
26 |
27 |     public Dosen(String nama, String nidn) {
28 |         super(nama);
29 |         this.nidn = nidn;
30 |     }
31 |
32 |     public void tampilkanNIDN() {
33 |         System.out.println("NIDN: " + nidn);
34 |     }
35 | }
36 |
37 | public class Main {
38 |     public static void main(String[] args) {
39 |         Dosen dosen = new Dosen("Firman", "123456789");
40 |         dosen.perkenalan();
41 |         dosen.tampilkanNIDN();
42 |     }
43 | }
```

-Polymorphism

Konsep ini memungkinkan user menggunakan suatu interface yang sama untuk memerintah object agar melakukan aksi atau tindakan yang mungkin secara prinsip sama namun secara proses berbeda.

Polymorphism sering dikaitkan dengan penggunaan lebih dari satu method dengan nama yang sama. Penggunaan dengan nama yang sama dapat diterapkan dengan method overloading dan overriding.

Contoh Polymorphism

```
13  class Hewan {
14      public void suara() {
15          System.out.println("Hewan bersuara...");
16      }
17  }
18
19  class Kucing extends Hewan {
20      @Override
21      public void suara() {
22          System.out.println("Meong...");
23      }
24  }
25
26  class Anjing extends Hewan {
27      @Override
28      public void suara() {
29          System.out.println("Guk guk...");
30      }
31  }
32
33  public class Main {
34      public static void main(String[] args) {
35          Hewan hewan1 = new Kucing();
36          Hewan hewan2 = new Anjing();
37
38          hewan1.suara();
39          hewan2.suara();
40      }
41  }
```

2.

```
5 package pertemuan2_50421905;
6
7 /**
8  *
9  * @author WishperingWhite
10 */
11 import java.util.ArrayList;
12 import java.util.List;
13
14 // Kelas utama Buku (Encapsulation)
15 class Buku {
16     private String judul;
17     private String penulis;
18     private int tahunTerbit;
19
20     public Buku(String judul, String penulis, int tahunTerbit) {
21         this.judul = judul;
22         this.penulis = penulis;
23         this.tahunTerbit = tahunTerbit;
24     }
25
26     public String getJudul() {
27         return judul;
28     }
29
30     public String getPenulis() {
31         return penulis;
32     }
33
34     public int getTahunTerbit() {
35         return tahunTerbit;
36     }
37
38     public void tampilkanInfo() {
39         System.out.println("Judul: " + judul);
40         System.out.println("Penulis: " + penulis);
41         System.out.println("Tahun Terbit: " + tahunTerbit);
42     }
43 }
```

```

43     }
44
45     class Majalah extends Buku {
46         private int volume;
47
48         public Majalah(String judul, String penulis, int tahunTerbit, int volume) {
49             super(judul, penulis, tahunTerbit);
50             this.volume = volume;
51         }
52
53         public int getVolume() {
54             return volume;
55         }
56
57         @Override
58         public void tampilkanInfo() {
59             super.tampilkanInfo();
60             System.out.println("Volume: " + volume);
61         }
62     }
63
64     class Novel extends Buku {
65         private String genre;
66
67         public Novel(String judul, String penulis, int tahunTerbit, String genre) {
68             super(judul, penulis, tahunTerbit);
69             this.genre = genre;
70         }
71
72         public String getGenre() {
73             return genre;
74         }
75
76         @Override
77         public void tampilkanInfo() {
78             super.tampilkanInfo();
79             System.out.println("Genre: " + genre);
80         }
81     }

```

```

81     }
82
83     class PengelolaPerpustakaan {
84         private List<Buku> koleksiBuku = new ArrayList<>();
85
86         public void tambahBuku(Buku buku) {
87             koleksiBuku.add(buku);
88         }
89
90         public void tampilkanKoleksi() {
91             System.out.println("Koleksi Buku di Perpustakaan:");
92             for (Buku buku : koleksiBuku) {
93                 buku.tampilkanInfo();
94                 System.out.println("-----");
95             }
96         }
97     }
98
99     public class Main {
100         public static void main(String[] args) {
101             PengelolaPerpustakaan perpustakaan = new PengelolaPerpustakaan();
102
103             Buku bukul = new Buku("Dasar Pemrograman", "John Doe", 2020);
104             Majalah majalah1 = new Majalah("Majalah Teknologi", "Jane Doe", 2021, 15);
105             Novel novell = new Novel("Petualangan di Negeri Ajaib", "Alex Roe", 2019, "Fantasy");
106
107             perpustakaan.tambahBuku(bukul);
108             perpustakaan.tambahBuku(majalah1);
109             perpustakaan.tambahBuku(novell);
110
111             perpustakaan.tampilkanKoleksi();
112         }
113     }
114
115
116

```

Output - Run (pertemuan2_50421905)

```

-----[ jar ]-----
--- exec-maven-plugin:3.0.0:exec (default-cli)
Koleksi Buku di Perpustakaan:
Judul: Dasar Pemrograman
Penulis: John Doe
Tahun Terbit: 2020
-----
Judul: Majalah Teknologi
Penulis: Jane Doe
Tahun Terbit: 2021
Volume: 15
-----
Judul: Petualangan di Negeri Ajaib
Penulis: Alex Roe
Tahun Terbit: 2019
Genre: Fantasy
-----

BUILD SUCCESS

-----
Total time: 1.350 s

```

Encapsulation:

Atribut judul, penulis, dan tahunTerbit di kelas Buku dienkapsulasi dengan akses modifier private. Atribut-atribut ini hanya dapat diakses melalui getter (misalnya, getJudul(), getPenulis(), dan getTahunTerbit()).

Kelas PengelolaPerpustakaan juga menggunakan encapsulation untuk mengelola daftar buku dalam List<Buku> koleksiBuku yang disimpan secara private.

Inheritance:

Majalah dan Novel adalah kelas turunan (child class) dari kelas Buku. Mereka mewarisi atribut dan metode dari kelas Buku, sehingga tidak perlu mendeklarasikan atribut judul, penulis, dan tahunTerbit lagi.

Majalah dan Novel memiliki atribut tambahan (volume untuk Majalah dan genre untuk Novel), yang merupakan karakteristik spesifik masing-masing kelas.

Polymorphism:

Metode tampilkanInfo di kelas Buku di-override dalam kelas Majalah dan Novel. Hal ini memungkinkan masing-masing kelas memiliki implementasi khusus untuk menampilkan informasi buku. Saat tampilkanKoleksi dipanggil di PengelolaPerpustakaan, metode tampilkanInfo() yang tepat akan dipanggil berdasarkan tipe objek (buku, majalah, atau novel).