

PEMROGRAMAN BERORIENTASI OBJEK

Resume Pembelajaran PBO

Dosen pengampu :
Andhik Ampuh Yunanto S.Kom., M.Kom.



Disusun oleh:

Aaron Febrian Prakoso (3122500060)

2 D3 Teknik Informatika B

PROGRAM STUDI D3 TEKNIK INFORMATIKA

POLITEKNIK ELEKTRONIKA NEGERI SURABAYA

2023 / 2024

Rangkuman Singkat Pembelajaran Mata Kuliah PBO Selama 1 Semester

Apa itu Pemrograman Berorientasi Objek?

Pemrograman berorientasi objek (PBO) atau dalam bahasa Inggris disebut Objek Oriented Programming (OOP) adalah paradigma pemrograman yang berfokus pada konsep objek, yang mengorganisir data dan fungsi ke dalam entitas yang disebut objek. Dalam PBO, suatu sistem atau program diorganisir sebagai kumpulan objek yang berinteraksi satu sama lain. Setiap objek dapat memiliki atribut (variabel) dan method (fungsi) sesuai kebutuhannya. OOP berlandaskan pada beberapa prinsip utama, yaitu:

1. Enkapsulasi

Enkapsulasi adalah mekanisme yang menggabungkan data dan metode menjadi satu unit yang disebut kelas.

Manfaat enkapsulasi:

- ❖ Meningkatkan keamanan data
- ❖ Meningkatkan kemudahan pemeliharaan program
- ❖ Meningkatkan keterbacaan program

Cara menerapkan enkapsulasi:

- ❖ Menggunakan modifier access (private, protected, default, public)
- ❖ Menggunakan method getter dan setter untuk mengakses data private

2. Inheritance

Inheritance adalah mekanisme pewarisan sifat dan perilaku dari kelas induk (superclass) ke kelas turunan (subclass). Inheritance memfasilitasi reusabilitas kode, memungkinkan untuk membangun hierarki kelas sehingga mengurangi duplikasi kode.

Manfaat inheritance:

- ❖ Meningkatkan modularitas program
- ❖ Meningkatkan kemudahan pengembangan program
- ❖ Meningkatkan kemudahan dokumentasi program

Cara menerapkan inheritance:

- ❖ Menggunakan kata kunci "extends"

3. Overriding dan Overloading

Overriding adalah mekanisme menimpa metode yang dimiliki kelas induk di kelas turunan. Overriding ini hanya terjadi antara parent class atau interface dengan child class. Syarat overriding adalah nama, return type, dan parameter method harus sama serta method yang melakukan overriding harus memiliki baris kode yang berbeda. Sedangkan Overloading adalah mekanisme memberikan nama method yang sama untuk beberapa kelas namun berbeda parameter dan tipe data. Overloading bisa terjadi antara parent class dengan child class ataupun di antar method dalam satu class.

Manfaat overriding:

- ❖ Meningkatkan fleksibilitas program
- ❖ Meningkatkan kemudahan pengembangan program

Manfaat overloading:

- ❖ Meningkatkan kemudahan penggunaan program
- ❖ Meningkatkan kemudahan pemeliharaan program

Cara menerapkan overriding:

- ❖ Menggunakan kata kunci "override"

Cara menerapkan overloading:

- ❖ Menggunakan nama method yang sama
- ❖ Menentukan parameter atau tipe data yang berbeda

4. Polimorfisme

Polimorfisme adalah kemampuan suatu objek untuk berperilaku berbeda-beda tergantung pada konteksnya. Polimorfisme dapat dicapai melalui overriding metode atau menggunakan konsep seperti interface dan abstract class.

Manfaat polimorfisme:

- ❖ Meningkatkan fleksibilitas program
- ❖ Meningkatkan kemudahan pengembangan program

Cara menerapkan polimorfisme:

- ❖ Menggunakan overriding metode
- ❖ Menggunakan interface
- ❖ Menggunakan abstract class

5. Abstract Class, Interface, Inner Class

Abstract class adalah kelas yang tidak dapat diinstansiasi. Abstract class hanya dapat digunakan sebagai induk dari kelas turunan. Interface adalah kumpulan method yang harus diimplementasikan oleh kelas yang mengimplementasikannya. Inner class adalah kelas yang didefinisikan di dalam kelas lain.

Manfaat abstract class:

- ❖ Meningkatkan modularitas program
- ❖ Meningkatkan kemudahan pengembangan program

Manfaat interface:

- ❖ Meningkatkan modularitas program
- ❖ Meningkatkan kemudahan pengembangan program

Manfaat inner class:

- ❖ Meningkatkan modularitas program
- ❖ Meningkatkan keamanan program

Cara menerapkan abstract class:

- ❖ Menggunakan kata kunci "abstract"

Cara menerapkan interface:

- ❖ Menggunakan kata kunci "interface"

Cara menerapkan inner class:

- ❖ Menggunakan kata kunci "inner"

6. Composition, Aggregation, Association, Dependency

Composition adalah hubungan antara dua objek yang mana satu objek memiliki bagian dari objek lainnya. Composition biasanya menggunakan kata kunci "new" untuk membuat objek baru. Aggregation adalah hubungan antara dua objek yang mana satu objek memiliki keterkaitan dengan objek lainnya (hubungan ini biasanya digambarkan dengan "has-a"). Association adalah hubungan antara dua objek yang mana kedua objek tersebut memiliki keterkaitan. Dependency adalah hubungan antara dua objek yang mana satu objek bergantung pada objek lainnya.

7. Class Diagram

Class diagram adalah diagram yang menggambarkan hubungan antara class-class dalam sebuah sistem.

8. Coupling

Coupling adalah keterkaitan antara dua objek. Coupling yang tinggi dapat menyebabkan program menjadi sulit untuk dikembangkan dan dipelihara. Untuk mengurangi coupling, dapat dilakukan beberapa hal, yaitu:

- ❖ Menggunakan interface
- ❖ Menggunakan composition
- ❖ Menggunakan aggregation