

OBJECT ORIENTED PROGRAMMING



By :

Sri Kresna Maha Dewa

2241720244

STUDY PROGRAM D-IV INFORMATIC ENGINEERING

INFORMATION TECHNOLOGY DEPARTMENT

MALANG STATE POLYTECHNIC

Soekarno Hatta Street No.9, Jatimulyo, Lowokwaru District, Malang City, East Java

65141

PERTANYAAN PERCOBAAN 1

1. Method setter dan getter adalah sebuah method yang digunakan untuk mengakses attribute yang telah di enkapsulasi.
2. Konstruktor dengan parameter dapat kita isi value nya saat kita instansiasi, tetapi konstruktor tanpa parameter proses assign value nya dilakukan setelah instansiasi.
3. Proc adalah attribute yang bertipe object dari class Processor
4. Pada method setProc, laptop bisa dianggap berelasi dengan class Processor
5. Proc.info digunakan untuk menampilkan Informasi attribute dari object Proc
6. P adalah object instansiasi dari Class Processor, saat kode diubah hasil program tidak menunjukkan perubahan saat dijalankan.

PERTANYAAN PERCOBAAN 2

1. Pada attribute object Mobil dan Sopir serta method getter-setter dari kedua attribute tersebut sudah menunjukkan relasi antar class dari ketiga class.
2. Karena biaya akan dihitung per hari untuk setiap mobil dan sopir yang disewa
3. Kedua method tersebut digunakan untuk menghitung jumlah yang harus dibayar dari menyewa mobil atau sopir untuk n hari.
4. Kedua sintaks tersebut digunakan untuk setter atau mengassign attribute object mobil dan sopir pada object Pelanggan (p)
5. Method hitungBiayaTotal() digunakan untuk menghitung total biaya yang harus dibayar pelanggan selama menyewa atau menggunakan layanan baik itu mobil dan sopir
6. Sintaks tersebut digunakan untuk mengakses object mobil kemudian mengakses method getter dari object Mobil tersebut.

PERTANYAAN PERCOBAAN 3

1. Kedua baris tersebut digunakan untuk mengakses method info() dari kedua object yang dimaksud.



```
oop_java > week/ > MainPertanyaan.java > ...
1 package oop_java.week7;
2
3 public class MainPertanyaan {
4     public static void main(String[] args) {
5         Pegawai masinis = new Pegawai(nip:"1234", nama:"Spongebob Squarepants");
6         KeretaApi keretaApi = new KeretaApi(nama:"Gaya Baru", kelas:"Bisnis",
7             masinis);
8         System.out.println(keretaApi.info());
9     }
10 }
11
12
13
```

Exception in thread "main" java.lang.NullPointerException: Cannot invoke "oop_java.week7.Pegawai.info()" because "this.asisten" is null

at oop_java.week7.KeretaApi.info(KeretaApi.java:62)
at oop_java.week7.MainPertanyaan.main(MainPertanyaan.java:9)

PS D:\remote\code-for-edu>

3.

Hal ini terjadi karena method info() mencoba memanggil method getter dari Pegawai, tetapi Pegawai tidak di assign saat instansiasi object sehingga terjadi error null pointer exception karena masih kosong.

4. Hal yang dapat dilakukan adalah membuat method baru Bernama infoMasinis() yang hanya memanggil method getter info dari object masinis saja

```
66     public String infoMasinis() {
67         String info = "";
68         info += "Nama: " + nama + "\n";
69         info += "Kelas: " + kelas + "\n";
70         info += "Masinis: " + masinis.info() + "\n";
71         return info;
72     }
```

```
Nama: Gaya Baru
Kelas: Bisnis
Masinis: NIP: 1234
Nama: Spongebob Squarepants

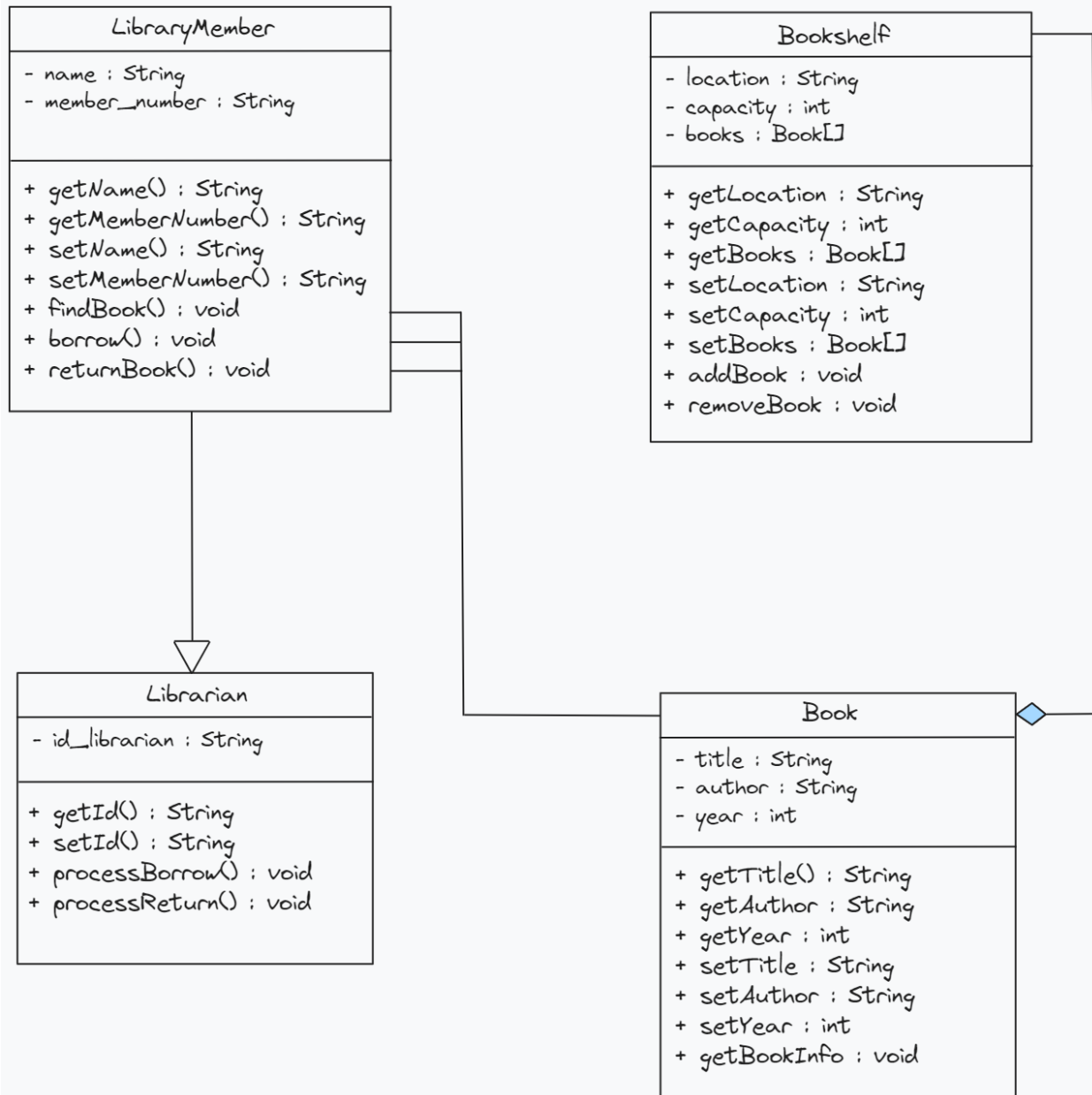
PS D:\remote\code-for-edu>
```

PERTANYAAN PERCOBAAN 4

1. Jumlah kursinya adalah 10, karena value pada konstruktor akan di assign pada array size kursi.
2. Ketika attribute object penumpang sudah diisi atau tidak kosong maka info akan dapat menyimpan Informasi dari object penumpang
3. Karena index sebuah array dimulai dari nol, dimana jika ingin mengakses indeks ke 3 dengan input angka 4 maka harus dikurangi 1
4. Budi tidak akan mendapat kursi, karena kursi nomor satu sudah diisi oleh penumpang pertama

```
5.     public void setPenumpang(Penumpang penumpang, int nomor) {
        if (this.arrayKursi[nomor - 1].getPenumpang() == null) {
            this.arrayKursi[nomor - 1].setPenumpang(penumpang);
        } else {
            System.out.println("Kursi sudah terisi!");
        }
    }
```

TUGAS



CLASS :

```
1  package oop_java.week7;
2
3
4  public class Book {
5      private String title, author;
6      private int year;
7
8      public Book(String title, String author, int year) {
9          this.title = title;
10         this.author = author;
11         this.year = year;
12     }
13
14     public String getTitle() {
15         return title;
16     }
17
18     public String getAuthor() {
19         return author;
20     }
21
22     public int getYear() {
23         return year;
24     }
25
26     public String setTitle(String title) {
27         return this.title = title;
28     }
29
30     public String setAuthor(String author) {
31         return this.author = author;
32     }
33
34     public int setYear(int year) {
35         return this.year = year;
36     }
37
38     public void getBookInfo() {
39         System.out.println("Title: " + title);
40         System.out.println("Author: " + author);
41         System.out.println("Year: " + year);
42     }
43 }
44
```

```
1  package oop_java.week7;
2
3  public class LibraryMember {
4      String name, member_number;
5
6      public LibraryMember(String name, String member_number){
7          this.name = name;
8          this.member_number = member_number;
9      }
10
11     public String getName(){
12         return name;
13     }
14
15     public String getMemberNumber(){
16         return member_number;
17     }
18
19     public String setName(String name){
20         return this.name = name;
21     }
22
23     public String setMemberNumber(String member_number){
24         return this.member_number = member_number;
25     }
26
27     public void findBook(Librarian Librarian, Book book){
28         System.out.println("Library member want to find book " + book.getTitle());
29     }
30
31     public void borrow(Librarian Librarian, Book book){
32         System.out.println("Library member want to borrow book " + book.getTitle());
33     }
34
35     public void returnBook(Librarian Librarian, Book book){
36         System.out.println("Library member want to return book " + book.getTitle());
37     }
38 }
39
40
```

```
1 package oop_java.week7;
2
3
4 public class Bookshelf {
5     private String location;
6     private int capacity;
7     private Book[] books;
8
9     public Bookshelf(String location, int capacity, Book[] books) {
10         this.location = location;
11         this.capacity = capacity;
12         this.books = books;
13     }
14
15     public String getLocation() {
16         return location;
17     }
18
19     public int getCapacity() {
20         return capacity;
21     }
22
23     public Book[] getBooks() {
24         return books;
25     }
26
27     public String setLocation(String location) {
28         return this.location = location;
29     }
30
31     public int setCapacity(int capacity) {
32         return this.capacity = capacity;
33     }
34
35     public Book[] setBooks(Book[] books) {
36         return this.books = books;
37     }
38
39     public void addBook(Book book) {
40         if (books == null) {
41             books = new Book[capacity];
42         }
43         for (int i = 0; i < books.length; i++) {
44             if (books[i] == null) {
45                 books[i] = book;
46                 break;
47             }
48         }
49     }
50
51     public void removeBook(Book book) {
52         if (books == null) {
53             return;
54         }
55         for (int i = 0; i < books.length; i++) {
56             if (books[i] == book) {
57                 books[i] = null;
58                 break;
59             }
60         }
61     }
62
63 }
64
```

```
1 package oop_java.week7;
2
3
4 public class Librarian extends LibraryMember {
5     private String id_librarian;
6
7     public Librarian(String name, String member_number, String id_Librarian){
8         super(name, member_number);
9         this.name = name;
10        this.id_librarian = id_Librarian;
11    }
12
13
14    public String getId(){
15        return id_librarian;
16    }
17
18    public String setId(String id){
19        return this.id_librarian = id;
20    }
21
22    public void processBorrow(LibraryMember member, Book book){
23        System.out.println("Book " + book.getTitle() + " has been borrowed by " + member.getName());
24    }
25
26    public void processReturn(LibraryMember member, Book book){
27        System.out.println("Book " + book.getTitle() + " has been returned by " + member.getName());
28    }
29 }
```


MAIN :

```
1 package oop_java.week7;
2
3 public class LibraryIntClass {
4     public static void main(String[] args) {
5         Book book1 = new Book("The Lord of The Rings", "J.R.R Tolkien", 1954);
6         Book book2 = new Book("Harry Potter and The Philosopher's Stone", "J.K Rowling", 1997);
7         Book book3 = new Book("The Hobbit", "J.R.R Tolkien", 1937);
8
9         LibraryMember sucipto = new LibraryMember("Sucipto", "123456");
10        LibraryMember budi = new LibraryMember("Budi", "654321");
11
12        Librarian pakde = new Librarian("Pakde", "123456789", "001");
13
14        Bookshelf bookshelf1 = new Bookshelf("A1", 10, null);
15        Bookshelf bookshelf2 = new Bookshelf("B2", 10, null);
16
17        bookshelf1.addBook(book1);
18        bookshelf1.addBook(book2);
19        bookshelf2.addBook(book3);
20
21        sucipto.findBook(pakde, book3);
22        for (int i = 0; i < bookshelf1.getBooks().length; i++) {
23            if (bookshelf1.getBooks()[i] == book3) {
24                System.out.println("Book " + book3.getTitle() + " is found in bookshelf " + bookshelf1.getLocation());
25                break;
26            } else if (bookshelf2.getBooks()[i] == book3) {
27                System.out.println("Book " + book3.getTitle() + " is found in bookshelf " + bookshelf2.getLocation());
28                break;
29            } else {
30                System.out.println("Book " + book3.getTitle() + " is not found");
31                break;
32            }
33        }
34
35        System.out.println("=====");
36        sucipto.borrow(pakde, book1);
37        pakde.processBorrow(sucipto, book1);
38
39        System.out.println("=====");
40        sucipto.returnBook(pakde, book1);
41        pakde.processReturn(sucipto, book1);
42
43    }
44 }
```