



Name : Faatihurrizki Prasojo

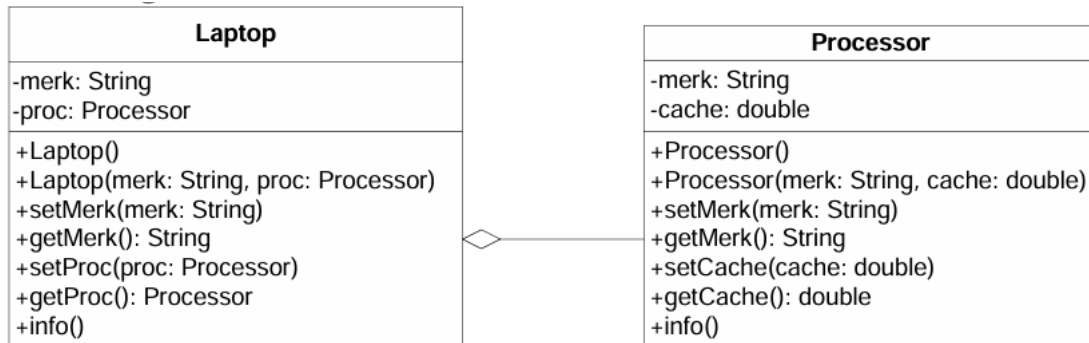
NIM : 244107020142

Class : TI\_1H

## Laporan Jobsheet 4

### Pemrograman Berorientasi Objek (PBO)

#### Percobaan 1



1. Buat class sesuai diagram class diatas

#### Class Laptop

```
public class Laptop {
    private String merk;
    private Processor proc;

    public Laptop() {
    }

    public Laptop(String merk, Processor proc) {
        this.merk = merk;
        this.proc = proc;
    }

    public void setMerk(String merk) {
        this.merk = merk;
    }

    public void setProc(Processor proc) {
        this.proc = proc;
    }

    public String getMerk() {
        return merk;
    }

    public Processor getProc() {
        return proc;
    }
}
```



Name : Faatihurrizki Prasojo

NIM : 244107020142

Class : TI\_1H

```
public void info() {  
    System.out.println("Merk Laptop: " + this.merk);  
    this.proc.info();  
}  
}
```

### Class Processor

```
public class Processor {  
    private String merk;  
    private double cache;  
    public Processor() {  
    }  
    public Processor(String merk, double cache) {  
        this.merk = merk;  
        this.cache = cache;  
    }  
    public void setMerk(String merk) {  
        this.merk = merk;  
    }  
    public void setCache(double cache) {  
        this.cache = cache;  
    }  
    public String getMerk() {  
        return merk;  
    }  
    public double getCache() {  
        return cache;  
    }  
    public void info() {  
        System.out.println("Merk Prosesor: " + this.merk);  
        System.out.println("Cache Prosesor: " + this.cache + " GB");  
    }  
}
```



Name : Faatihurrizki Prasajo

NIM : 244107020142

Class : TI\_1H

### Pertanyaan

1. Di dalam class Processor dan class Laptop , terdapat method setter dan getter untuk masing-masing atributnya. Apakah gunanya method setter dan getter tersebut ?  
= **Method Setter** digunakan untuk mengatur nilai dari sebuah atribut. **Method Getter** digunakan untuk mendapatkan nilai dari sebuah atribut
2. Di dalam class Processor dan class Laptop, masing-masing terdapat konstruktor default dan konstruktor berparameter. Bagaimanakah beda penggunaan dari kedua jenis konstruktor tersebut ?
  - Konstruktor Default : Konstruktor ini tidak memiliki parameter. Gunanya adalah untuk membuat objek tanpa langsung memberikan nilai awal pada atribut-atributnya.  
Contoh penggunaan : `Laptop L1 = new Laptop();`
  - Konstruktor Berparameter : Konstruktor ini memiliki parameter yang digunakan untuk menginisialisasi atribut saat objek dibuat. Ini memungkinkan Anda membuat objek dan langsung memberikan nilai pada atribut didalam baris kode.  
Contoh penggunaan : `Laptop L1 = new Laptop("Thinkpad", p);`
3. Perhatikan class Laptop, di antara 2 atribut yang dimiliki (merk dan proc), atribut manakah yang bertipe object ?  
= pada kelas Laptop, atribut yang bertipe objek adalah proc, tipe datanya adalah Processor, Tipe datanya adalah Processor, yang merupakan kelas lain yang didefinisikan. Atribut merek bertipe data String, yang juga merupakan objek tetapi sering dianggap sebagai tipe data dasar
4. Perhatikan class Laptop, pada baris manakah yang menunjukkan bahwa class Laptop memiliki relasi dengan class Processor ?  
= `private Processor proc;`
5. Perhatikan pada class Laptop , Apakah guna dari sintaks proc.info() ?  
= Memanggil method info() dari objek Processor yang menjadi atribut di dalam objek Laptop
6. Pada class MainPercobaan1, terdapat baris kode: `Laptop l = new Laptop("Thinkpad", p);`. Apakah p tersebut ? Dan apakah yang terjadi jika baris kode tersebut diubah menjadi:  
= Variabel p adalah objek Processor yang sudah dibuat. Mengubah kode menjadi `Laptop l = new Laptop("Thinkpad", new Processor("Intel i5", 3));` akan memberikan hasil yang sama. Bedanya, cara pertama membuat objek Processor di variabel p lalu menggunakannya, sedangkan cara kedua membuat objek Processor baru langsung di dalam konstruktor Laptop tanpa menyimpannya di variabel terpisah.

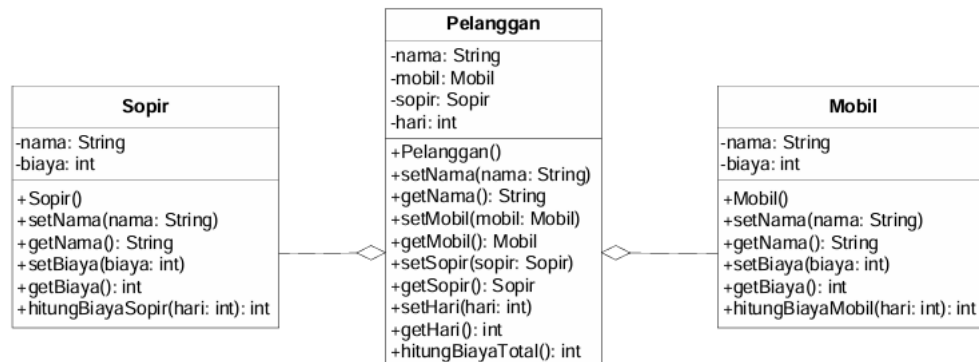


Name : Faatihurrizki Prasajo

NIM : 244107020142

Class : TI\_1H

## Percobaan 2



1. Buat class sesuai diagram class diatas

### Class Mobil

```
public class Mobil {
    private String nama;
    private int biaya;

    public Mobil() {
    }

    public void setNama(String nama) {
        this.nama = nama;
    }

    public String getNama() {
        return nama;
    }

    public void setBiaya(int biaya) {
        this.biaya = biaya;
    }

    public int getBiaya() {
        return biaya;
    }

    public int hitungBiayaMobil(int hari) {
        return biaya * hari;
    }
}
```



Name : Faatihurrizki Prasojo

NIM : 244107020142

Class : TI\_1H

### **Class Sopir**

```
public class Sopir {  
    private String nama;  
    private int biaya;  
    public Sopir(){  
    }  
    public void setNama(String nama){  
        this.nama = nama;  
    }  
    public String getNama(){  
        return nama;  
    }  
    public void setBiaya(int biaya){  
        this.biaya = biaya;  
    }  
    public int getBiaya(){  
        return biaya;  
    }  
    public int hitungBiayaSopir(int hari){  
        return biaya * hari;  
    }  
}
```

### **Class Pelanggan**

```
public class Pelanggan {  
    private String nama;  
    private Mobil mobil;  
    private Sopir sopir;  
    private int hari;
```



Name : Faatihurrizki Prasajo

NIM : 244107020142

Class : TI\_1H

```
public Pelanggan() {  
    }  
    public void setName(String nama) {  
        this.nama = nama;  
    }  
    public String getName() {  
        return nama;  
    }  
    public void setMobil(Mobil mobil) {  
        this.mobil = mobil;  
    }  
    public Mobil getMobil() {  
        return mobil;  
    }  
    public void setSopir(Sopir sopir) {  
        this.sopir = sopir;  
    }  
    public Sopir getSopir() {  
        return sopir;  
    }  
    public void setHari(int hari) {  
        this.hari = hari;  
    }  
    public int getHari() {  
        return hari;  
    }  
    public int hitungBiayaTotal() {  
        return mobil.hitungBiayaMobil(hari) +  
            sopir.hitungBiayaSopir(hari);  
    }  
}
```

Biaya Total = 1100000

PS C:\Users\fdlpr\Polinema\



Name : Faatihurrizki Prasajo

NIM : 244107020142

Class : TI\_1H

### Pertanyaan

1. Perhatikan class Pelanggan. Pada baris program manakah yang menunjukkan bahwa class Pelanggan memiliki relasi dengan class Mobil dan class Sopir ?
  - private Mobil mobil;
  - private Sopir sopir;
2. Perhatikan method hitungBiayaSopir pada class Sopir, serta method hitungBiayaMobil pada class Mobil. Mengapa menurut Anda method tersebut harus memiliki argument hari ?  
= biaya sewa atau jasa dihitung **berdasarkan jumlah hari**
3. Perhatikan kode dari class Pelanggan. Untuk apakah perintah mobil.hitungBiayaMobil(hari) dan sopir.hitungBiayaSopir(hari) ?  
= untuk menghitung total biaya yang harus dibayar pelanggan
4. Perhatikan class MainPercobaan2. Untuk apakah sintaks p.setMobil(m) dan p.setSopir(s) ?  
= **menghubungkan objek** Mobil (m) dan Sopir (s) dengan objek Pelanggan (p)
5. Perhatikan class MainPercobaan2. Untuk apakah proses p.hitungBiayaTotal() tersebut ?  
= **menghitung dan mendapatkan total biaya** dari seluruh transaksi sewa
6. Perhatikan class MainPercobaan2, coba tambahkan pada baris terakhir dari method main dan amati perubahan saat di-run!  
`System.out.println(p.getMobil().getMerk());`  
Jadi untuk apakah sintaks `p.getMobil().getMerk()` yang ada di dalam method *main* tersebut?
  - **p.getMobil()**: Perintah ini mengambil objek Mobil yang terhubung dengan objek Pelanggan (p).
  - **getMerk()**: Setelah mendapatkan objek Mobil, sintaks ini mencoba memanggil method `getMerk()`. Namun, berdasarkan kode Mobil.java yang Anda berikan, tidak ada method bernama `getMerk()`. Method yang ada adalah `getNama()`.

### Percobaan 3

#### Class Pegawai

```
public class Pegawai {  
    private String nip;  
    private String nama;  
    public Pegawai(String nip, String nama){  
        this.nip = nip;  
        this.nama = nama;  
    }  
}
```



Name : Faatihurrizki Prasojo

NIM : 244107020142

Class : TI\_1H

```
public void setNip(String nip){
    this.nip = nip;
}
public String getNip(){
    return nip;
}
public void setNama(String nama){
    this.nama = nama;
}
public String getNama(){
    return nama;
}
public String info(){
    String info = "";
    info += "NIP\t: " + this.nip + "\n";
    info += "Nama\t: " + this.nama + "\n";
    return info;
}
}
```

### **Class KeretaApi**

```
public class KeretaApi {
    private String nama;
    private String kelas;
    private Pegawai masinis;
    private Pegawai asisten;
```





Name : Faatihurrizki Prasojo

NIM : 244107020142

Class : TI\_1H

```
/*  
  
public KeretaApi(String nama, String kelas, Pegawai masinis) {  
    this.nama = nama;  
    this.kelas = kelas;  
    this.masinis = masinis;  
}  
*/  
  
public KeretaApi(String nama, String kelas, Pegawai masinis, Pegawai  
asisten) {  
    this.nama = nama;  
    this.kelas = kelas;  
    this.masinis = masinis;  
    this.asisten = asisten;  
}  
  
public void setNama(String nama){  
    this.nama = nama;  
}  
  
public String getNama(){  
    return nama;  
}  
  
public void setKelas(String kelas){  
    this.kelas = kelas;  
}  
  
public String getKelas(){  
    return kelas;  
}  
  
public void setMasinis(Pegawai masinis){  
    this.masinis = masinis;  
}  
  
public Pegawai getMasinis(){  
    return masinis;  
}  
  
public void setAsisten(Pegawai asisten){  
    this.asisten = asisten;  
}
```



Name : Faatihurrizki Prasajo

NIM : 244107020142

Class : TI\_1H

```
public String info() {  
    String info = "";  
    info += "Nama\t: " + this.nama + "\n";  
    info += "Kelas\t: " + this.kelas + "\n";  
    info += "-----\n";  
    info += "=====--Masinis-----\n" + this.masinis.info();  
    info += "=====--Asisten-----\n" + this.asisten.info();  
    info += "-----";  
    return info;  
}  
}
```

### Class MainPercobaan3

```
package Prasajo.relasticlass.percobaan3;  
  
public class MainPercobaan3 {  
    public static void main(String[] args) {  
        Pegawai masinis = new Pegawai("1234", "Spongebob Squarepants");  
        Pegawai asisten = new Pegawai("4567", "Patrick Star");  
        KeretaApi keretaApi = new KeretaApi("Gaya Baru", "Bisnis", masinis,  
asisten);  
  
        System.out.println(keretaApi.info());  
    }  
}
```

### Pertanyaan

1. Di dalam method info() pada class KeretaApi, baris this.masinis.info() dan this.asisten.info() digunakan untuk apa ?  
= untuk memanggil metode **info()** dari objek **Pegawai**



Name : Faatihurrizki Prasajo

NIM : 244107020142

Class : TI\_1H

2. Buatlah main program baru dengan nama class MainPertanyaan pada package yang sama. Tambahkan kode berikut pada method main() !

```
Pegawai masinis = new Pegawai("1234", "Spongebob  
Squarepants");  
KeretaApi keretaApi = new KeretaApi("Gaya Baru", "Bisnis",  
masinis);
```

```
System.out.println(keretaApi.info());
```

=

```
public class MainPertanyaan {  
    Run | Debug  
    public static void main(String[] args) {  
        Pegawai masinis = new Pegawai("1234", "Spongebob Squarepants");  
        KeretaApi keretaApi = new KeretaApi("Gaya Baru", "Bisnis", masinis);  
        System.out.println(keretaApi.info());  
    }  
}
```

3. Apa hasil output dari main program tersebut ? Mengapa hal tersebut dapat terjadi ?

```
Exception in thread "main" java.lang.NullPointerException: Cannot invoke "P  
rasajo.relasticlass.percobaan3.Pegawai.info()" because "this.asisten" is nul  
l  
    at Prasajo.relasticlass.percobaan3.KeretaApi.info(KeretaApi.java:61)  
    at Prasajo.relasticlass.percobaan3.MainPertanyaan.main(MainPertanyaa  
n.java:8)  
PS C:\Users\fdlpr\Polinema\Semester 3\PB0\Praktikum\Week 4>
```

Penyebabnya adalah **konstruktor** yang dipanggil di kelas MainPertanyaan (new KeretaApi("Gaya Baru", "Bisnis", masinis)) tidak cocok dengan konstruktor yang tersedia di kelas KeretaApi.java.

- Di kelas MainPertanyaan, kita mencoba memanggil konstruktor dengan **tiga argumen**: nama (String), kelas (String), dan masinis (objek Pegawai).
- Namun, di kelas KeretaApi.java, konstruktor yang ada **membutuhkan empat argumen**: nama (String), kelas (String), masinis (objek Pegawai), dan **asisten (objek Pegawai)**.

Oleh karena itu, Java tidak dapat menemukan konstruktor yang sesuai untuk membuat objek keretaApi, sehingga terjadi error.

4. Perbaiki class KeretaApi sehingga program dapat berjalan !

```
public class MainPertanyaan {  
    public static void main(String[] args) {  
        Pegawai masinis = new Pegawai("1234", "Spongebob Squarepants");  
        Pegawai asisten = new Pegawai("4567", "Prasajo");  
        KeretaApi keretaApi = new KeretaApi("Gaya Baru", "Bisnis", masinis, asisten);  
        System.out.println(keretaApi.info());  
    }  
}
```



Name : Faatihurrizki Prasojo

NIM : 244107020142

Class : TI\_1H

## Percobaan 4

### Class Penumpang

```
public class Penumpang {  
    private String ktp;  
    private String nama;  
  
    public Penumpang(String ktp, String nama) {  
        this.ktp = ktp;  
        this.nama = nama;  
    }  
    public void setKtp(String ktp) {  
        this.ktp = ktp;  
    }  
    public String getKtp() {  
        return ktp;  
    }  
    public void setName(String nama) {  
        this.nama = nama;  
    }  
    public String getName() {  
        return nama;  
    }  
    public String info() {  
        String info = "";  
        info += "Ktp: " + ktp + "\n";  
        info += "Nama: " + nama + "\n";  
        return info;  
    }  
}
```



Name : Faatihurrizki Prasojo

NIM : 244107020142

Class : TI\_1H

### Class Kursi

```
public class Kursi {  
    private String nomor;  
    private Penumpang penumpang;  
  
    public Kursi(String nomor) {  
        this.nomor = nomor;  
    }  
    public void setNomor(String nomor) {  
        this.nomor = nomor;  
    }  
    public String getNomor() {  
        return nomor;  
    }  
    public void setPenumpang(Penumpang penumpang) {  
        this.penumpang = penumpang;  
    }  
    public Penumpang getPenumpang() {  
        return penumpang;  
    }  
    public String info() {  
        String info = "";  
        info += "Nomor: " + nomor + "\n";  
        if (this.penumpang != null) {  
            info += "Penumpang: " + penumpang.info() + "\n";  
        }  
        return info;  
    }  
}
```



Name : Faatihurrizki Prasajo

NIM : 244107020142

Class : TI\_1H

### Class Gerbong

```
public class Gerbong {  
    private String kode;  
    private Kursi[] arrayKursi;  
    public Gerbong(String kode, int jumlah) {  
        this.kode = kode;  
        this.arrayKursi = new Kursi[jumlah];  
        this.initKursi();  
    }  
    private void initKursi() {  
        for (int i = 0; i < arrayKursi.length; i++) {  
            this.arrayKursi[i] = new Kursi(String.valueOf(i + 1));  
        }  
    }  
    public void setKode(String kode) {  
        this.kode = kode;  
    }  
    public String getKode() {  
        return kode;  
    }  
    public void setPenumpang(Penumpang penumpang, int nomor) {  
        this.arrayKursi[nomor - 1].setPenumpang(penumpang);  
    }  
    public Kursi[] getArrayKursi() {  
        return arrayKursi;  
    }  
    public String info() {  
        String info = "";  
        info += "Kode: " + kode + "\n";  
        for (Kursi kursi : arrayKursi) {  
            info += kursi.info();  
        }  
        return info;  
    }  
}
```



Name : Faatihurrizki Prasajo

NIM : 244107020142

Class : TI\_1H

### Pertanyaan

1. Pada main program dalam class MainPercobaan4, berapakah jumlah kursi dalam Gerbong A ?  
= 10 kursi

2. Perhatikan potongan kode pada method info() dalam class Kursi. Apa maksud kode tersebut ?

```
...
if (this.penumpang != null) {
    info += "Penumpang: " + penumpang.info() + "\n";
} ...
```

= maksud dari kode ini adalah melakukan pengecekan apakah sebuah kursi sudah memiliki penumpang atau belum.

- Jika `this.penumpang` tidak `null`, artinya kursi tersebut sudah diduduki, maka informasi penumpang akan ditambahkan ke dalam string `info` dengan memanggil method `penumpang.info()`.
- Jika `this.penumpang` adalah `null`, artinya kursi tersebut kosong, maka blok `if` akan dilewati dan informasi penumpang tidak akan dicetak. Ini mencegah terjadinya `NullPointerException` saat mencoba memanggil method dari objek yang belum ada (`null`).

3. Mengapa pada method `setPenumpang()` dalam class `Gerbong`, nilai nomor dikurangi dengan angka 1 ?  
= karena indeks awal array nya adalah 0, maka harus dikurangi 1 untuk mendapatkan nilai array yang pertama
4. Instansiasi objek baru budi dengan tipe `Penumpang`, kemudian masukkan objek baru tersebut pada gerbong dengan `gerbong.setPenumpang(budi, 1)`. Apakah yang Page 9 of 10 terjadi ?

```
// Menjawab pertanyaan no 4
Penumpang budi = new Penumpang(ktp:"67890", nama:"Budi");
gerbong.setPenumpang(budi, nomor:1);
```

```
Kode: A
Nomor: 1
Penumpang: Ktp: 67890
Nama: Budi
```

= Yang terjadi adalah **data penumpang pada kursi nomor 1 akan ditimpa**. Awalnya, kursi nomor 1 diduduki oleh "Mr. Krab". Ketika `gerbong.setPenumpang(budi, 1)` ditambahkan, objek penumpang di dalam objek `Kursi` pada indeks 0 akan diganti dari objek p ("Mr. Krab") menjadi objek budi. Hasil akhirnya, kursi nomor 1 akan ditempati oleh "Budi"



Name : Faatihurrizki Prasajo

NIM : 244107020142

Class : TI\_1H

5. Modifikasi program sehingga tidak diperkenankan untuk menduduki kursi yang sudah ada penumpang lain !

```
// Method setPenumpang sebelum modifikasi
// public void setPenumpang(Penumpang penumpang, int nomor) {
//     this.arrayKursi[nomor - 1].setPenumpang(penumpang);
// }

// Method setPenumpang setelah modifikasi (Jawaban Pertanyaan 5)
public void setPenumpang(Penumpang penumpang, int nomor) {
    if (this.arrayKursi[nomor - 1].getPenumpang() != null) {
        System.out.println("kursi nomor " + nomor + " sudah ditempati oleh penumpang lain.");
    } else {
        this.arrayKursi[nomor - 1].setPenumpang(penumpang);
    }
}
```

```
Kursi nomor 1 sudah ditempati oleh penumpang lain.
Kode: A
Nomor: 1
Penumpang: Ktp: 12345
Nama: Mr. Krab
```

## Tugas

Sistem ini akan mengelola buku, anggota, dan proses peminjaman di sebuah perpustakaan.

- Buku: Kelas untuk merepresentasikan data sebuah buku.
  - Atribut: judul, penulis, penerbit.
  - Method: Constructor, setter, getter, info().
- Anggota: Kelas untuk merepresentasikan data anggota perpustakaan.
  - Atribut: idAnggota, nama.
  - Method: Constructor, setter, getter, info().
- Peminjaman: Kelas yang merepresentasikan satu transaksi peminjaman. Kelas ini memiliki relasi *has-a* dengan Anggota dan Buku.
  - Atribut: anggota (tipe Anggota), buku (tipe Buku), tglPinjam, tglKembali.
  - Method: Constructor, setter, getter, info().
- Perpustakaan: Kelas utama yang mengelola daftar buku, anggota, dan transaksi peminjaman.
  - Atribut: listBuku (array Buku), listAnggota (array Anggota), listPeminjaman (array Peminjaman).
  - Method: tambahBuku(), tambahAnggota(), prosesPeminjaman(), tampilkanDataPeminjaman().





Name : Faatihurrizki Prasojo

NIM : 244107020142

Class : TI\_1H

## Kode Program

```
Tugas 2 - Anggota.java - Anggota
1  package Tugas;
2
3  public class Anggota {
4      private String idAnggota;
5      private String nama;
6
7      public Anggota(String idAnggota, String nama) {
8          this.idAnggota = idAnggota;
9          this.nama = nama;
10     }
11
12     public String getIdAnggota() {
13         return idAnggota;
14     }
15
16     public String getNama() {
17         return nama;
18     }
19
20     public String info() {
21         return String.format(format:"ID: %s\nNama: %s", idAnggota, nama);
22     }
23 }
```

```
1  package Tugas;
2
3  public class Buku {
4      private String judul;
5      private String penulis;
6      private String penerbit;
7
8      public Buku(String judul, String penulis, String penerbit) {
9          this.judul = judul;
10         this.penulis = penulis;
11         this.penerbit = penerbit;
12     }
13
14     public String info() {
15         return String.format(format:"Judul\t: %s\nPenulis\t: %s\nPenerbit: %s", judul, penulis, penerbit);
16     }
17 }
```



Name : Faatihurrizki Prasajo

NIM : 244107020142

Class : TI\_1H

```
1 package Tugas;
2
3 public class Peminjaman {
4     private Anggota anggota;
5     private Buku buku;
6     private String tglPinjam;
7     private String tglKembali;
8
9     public Peminjaman(Anggota anggota, Buku buku, String tglPinjam, String tglKembali) {
10         this.anggota = anggota;
11         this.buku = buku;
12         this.tglPinjam = tglPinjam;
13         this.tglKembali = tglKembali;
14     }
15
16     public void info() {
17         System.out.println(x:"--- Detail Peminjaman ---");
18         System.out.println("Peminjam\t: " + anggota.getNama());
19         System.out.println("Buku yang Dipinjam: \n" + buku.info());
20         System.out.println("Tanggal Pinjam\t: " + tglPinjam);
21         System.out.println("Tanggal Kembali\t: " + tglKembali);
22         System.out.println(x:"-----");
23     }
24 }
25
```

```
1 package Tugas;
2
3 public class Perpustakaan {
4     private Buku[] listBuku = new Buku[100];
5     private Anggota[] listAnggota = new Anggota[100];
6     private Peminjaman[] listPeminjaman = new Peminjaman[100];
7     private int jmlBuku = 0;
8     private int jmlAnggota = 0;
9     private int jmlPeminjaman = 0;
10
11     public void tambahBuku(Buku buku) {
12         if (jmlBuku < listBuku.length) {
13             listBuku[jmlBuku++] = buku;
14         }
15     }
16
17     public void tambahAnggota(Anggota anggota) {
18         if (jmlAnggota < listAnggota.length) {
19             listAnggota[jmlAnggota++] = anggota;
20         }
21     }
22
23     public void prosesPeminjaman(Anggota anggota, Buku buku, String tglPinjam, String tglKembali) {
24         if (jmlPeminjaman < listPeminjaman.length) {
25             listPeminjaman[jmlPeminjaman++] = new Peminjaman(anggota, buku, tglPinjam, tglKembali);
26         }
27     }
28
29     public void tampilkanDataPeminjaman() {
30         System.out.println(x:"===== DATA PEMINJAMAN PERPUSTAKAAN =====");
31         for (int i = 0; i < jmlPeminjaman; i++) {
32             listPeminjaman[i].info();
33         }
34         System.out.println(x:"=====");
35     }
36 }
```



Name : Faatihurrizki Prasojo

NIM : 244107020142

Class : TI\_1H

```
1 package Tugas;
2
3 public class MainTugas {
4     Run | Debug
5     public static void main(String[] args) {
6         Perpustakaan perpustakaan = new Perpustakaan();
7
8         Buku buku1 = new Buku(judul:"Pemrograman Berorientasi Objek", penulis:"John Doe", penerbit:"Penerbit Informatika");
9         Buku buku2 = new Buku(judul:"Dasar-dasar Jaringan Komputer", penulis:"Jane Smith", penerbit:"Penerbit Andi");
10
11         perpustakaan.tambahBuku(buku1);
12         perpustakaan.tambahBuku(buku2);
13
14         Anggota anggota1 = new Anggota(idAnggota:"A001", nama:"Budi Santoso");
15         Anggota anggota2 = new Anggota(idAnggota:"A002", nama:"Citra Lestari");
16
17         perpustakaan.tambahAnggota(anggota1);
18         perpustakaan.tambahAnggota(anggota2);
19
20         perpustakaan.prosesPeminjaman(anggota1, buku1, tglPinjam:"18-09-2025", tglKembali:"25-09-2025");
21         perpustakaan.prosesPeminjaman(anggota2, buku2, tglPinjam:"19-09-2025", tglKembali:"26-09-2025");
22
23         perpustakaan.tampilkanDataPeminjaman();
24     }
25 }
26 }
```

===== DATA PEMINJAMAN PERPUSTAKAAN =====

--- Detail Peminjaman ---

Peminjam : Budi Santoso

Buku yang Dipinjam:

Judul : Pemrograman Berorientasi Objek

Penulis : John Doe

Penerbit: Penerbit Informatika

Tanggal Pinjam : 18-09-2025

Tanggal Kembali : 25-09-2025

-----

--- Detail Peminjaman ---

Peminjam : Citra Lestari

Buku yang Dipinjam:

Judul : Dasar-dasar Jaringan Komputer

Penulis : Jane Smith

Penerbit: Penerbit Andi

Tanggal Pinjam : 19-09-2025

Tanggal Kembali : 26-09-2025

-----

=====

PS C:\Users\fdlpr\Polinema\Semester 3\PB0\Praktikum\We