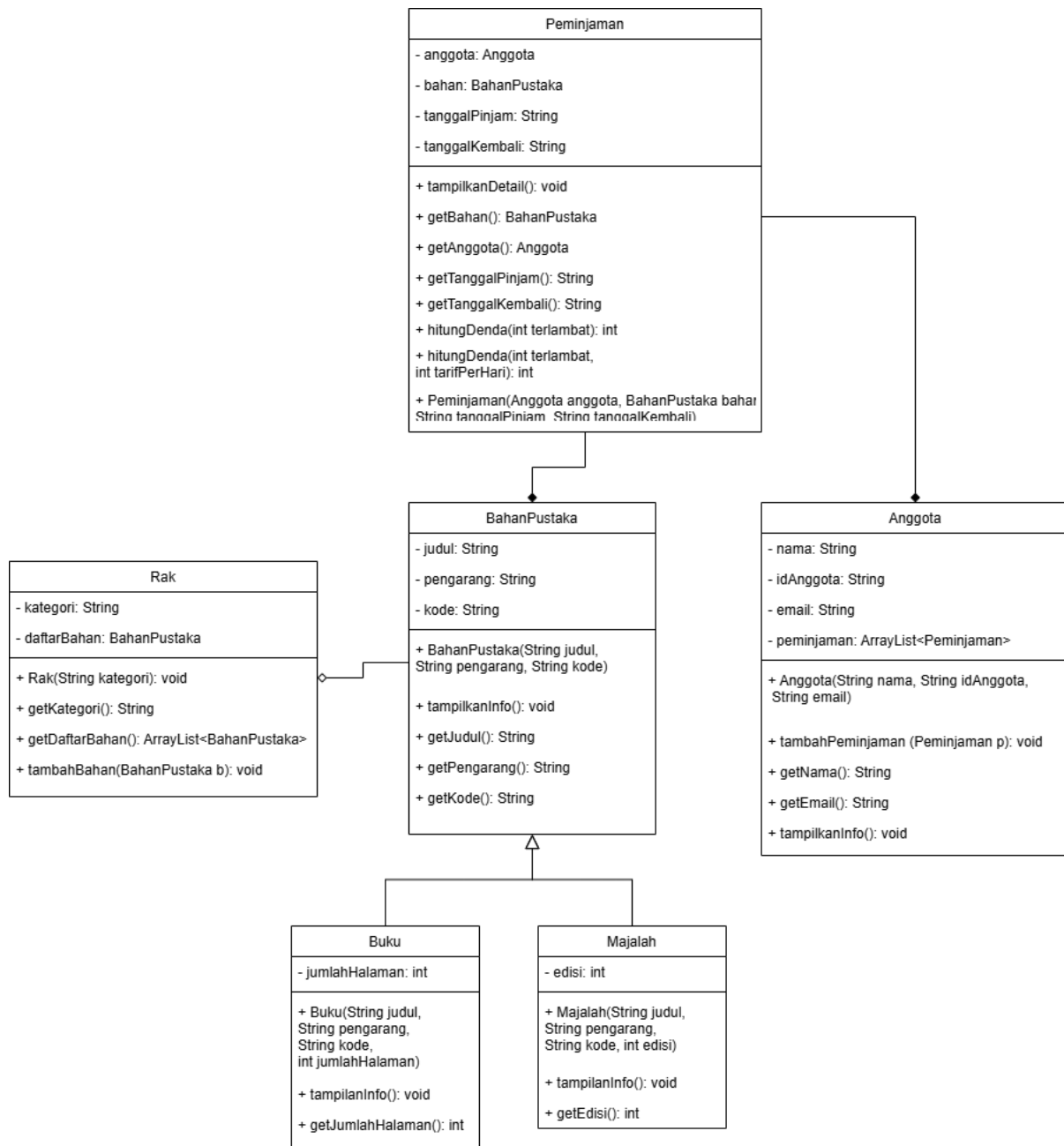


UTS
Praktikum Pemrograman Berbasis Objek



244107060144
Maria Savira
D-IV Sistem Informasi Bisnis / 2C

A. Perancangan



Metode yang akan dioverride :

- Method tampilkanInfo pada parent BahanPustaka beserta method tampilkanInfo dalam child Buku dan Majalah.

Beberapa keputusan saya :

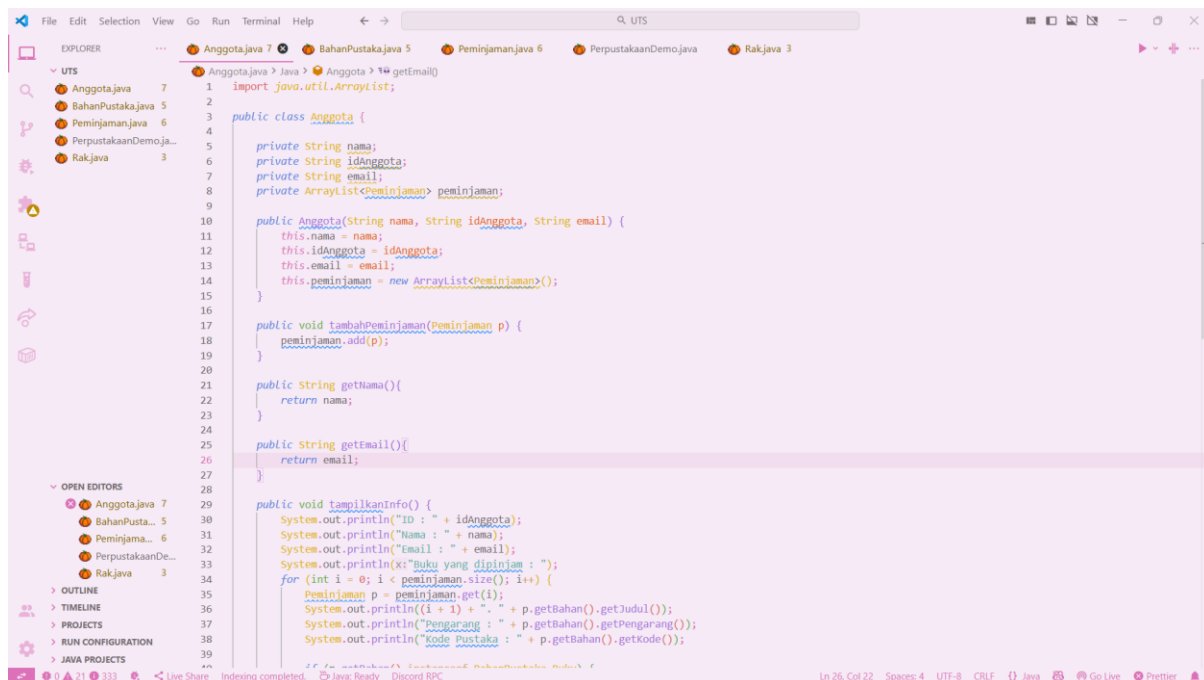
Semua atribut dibuat private untuk menjaga prinsip enkapsulasi. Dengan cara ini, data internal class (seperti nama, judul, kode, dll) tidak bisa diakses langsung dari luar class, melainkan harus melalui getter dan setter. Ini mencegah perubahan data sembarangan dan menjaga integritas data objek.

Selain itu, tidak semua atribut diberikan getter/setter. Hanya atribut yang memang perlu diakses dari luar class saja yang disediakan, misalnya `getJudul()` dan `getKode()`.

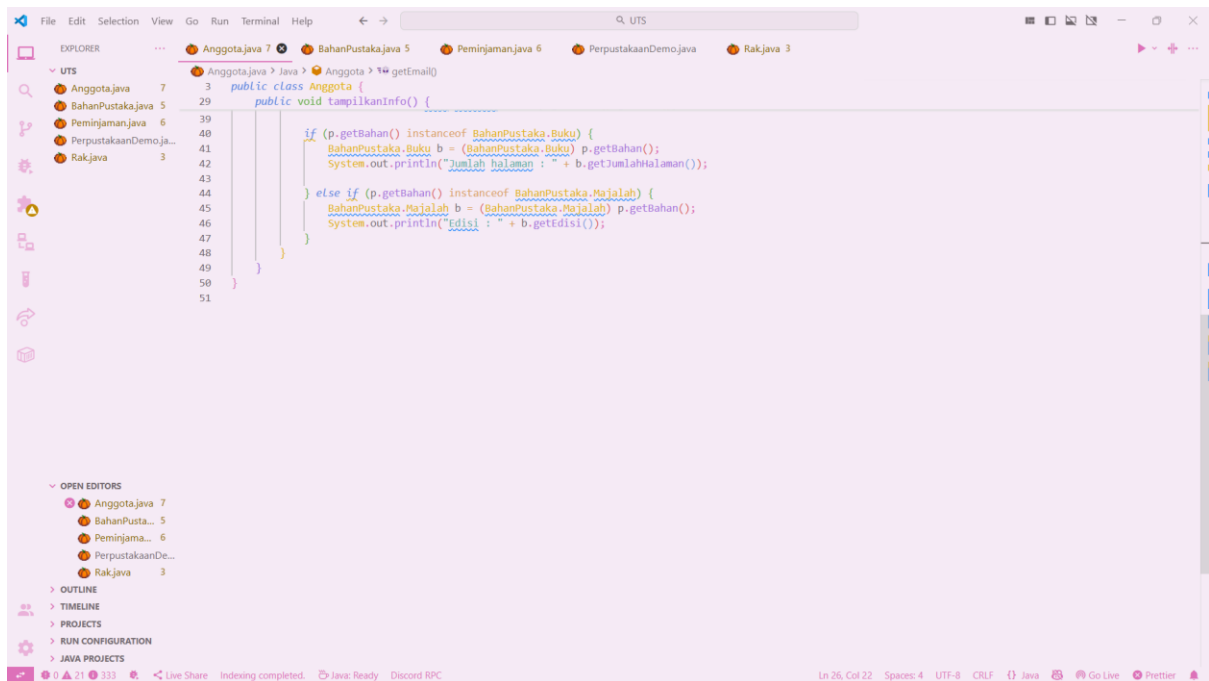
Terakhir, Class `BahanPustaka` dijadikan abstract class agar bisa menjadi dasar bagi jenis pustaka lain (seperti `Buku` dan `Majalah`). Pewarisan ini membantu menghindari duplikasi atribut dan method yang sama, serta memungkinkan penggunaan polimorfisme

B. Implementasi

Kode Program – Anggota.java



```
1 import java.util.ArrayList;
2
3 public class Anggota {
4
5     private String nama;
6     private String idAnggota;
7     private String email;
8     private ArrayList<Peminjaman> peminjaman;
9
10    public Anggota(String nama, String idAnggota, String email) {
11        this.nama = nama;
12        this.idAnggota = idAnggota;
13        this.email = email;
14        this.peminjaman = new ArrayList<Peminjaman>();
15    }
16
17    public void tambahPeminjaman(Peminjaman p) {
18        peminjaman.add(p);
19    }
20
21    public String getName(){
22        return nama;
23    }
24
25    public String getEmail(){
26        return email;
27    }
28
29    public void tampilkanInfo() {
30        System.out.println("ID : " + idAnggota);
31        System.out.println("Nama : " + nama);
32        System.out.println("Email : " + email);
33        System.out.println("Buku yang dipinjam : ");
34        for (int i = 0; i < peminjaman.size(); i++) {
35            Peminjaman p = peminjaman.get(i);
36            System.out.println((i + 1) + " - " + p.getBahan().getJudul());
37            System.out.println("Pengarang : " + p.getBahan().getPengarang());
38            System.out.println("Kode Pustaka : " + p.getBahan().getKode());
39        }
40    }
41}
```



```
File Edit Selection View Go Run Terminal Help
Anggota.java 7
BahanPustaka.java 5
Peminjaman.java 6
PerpustakaanDemo.java
Rak.java 3

public class Anggota {
    public void getEmail() {
    }

    public void tampilInfo() {
        if (p.getBahan() instanceof BahanPustaka.Buku) {
            BahanPustaka.Buku b = (BahanPustaka.Buku) p.getBahan();
            System.out.println("Jumlah halaman : " + b.getJumlahHalaman());
        } else if (p.getBahan() instanceof BahanPustaka.Majalah) {
            BahanPustaka.Majalah b = (BahanPustaka.Majalah) p.getBahan();
            System.out.println("Edisi : " + b.getEdisi());
        }
    }
}
```

BahanPustaka.java



```
File Edit Selection View Go Run Terminal Help
Anggota.java 7
BahanPustaka.java 5
Peminjaman.java 6
PerpustakaanDemo.java
Rak.java 3

public class BahanPustaka {
    private String judul;
    private String pengarang;
    private String kode;

    public BahanPustaka(String judul, String pengarang, String kode) {
        this.judul = judul;
        this.pengarang = pengarang;
        this.kode = kode;
    }

    public void tampilInfo() {
        System.out.println("Judul : " + judul);
        System.out.println("Pengarang : " + pengarang);
        System.out.println("Kode : " + kode);
    }

    public String getJudul() {
        return judul;
    }

    public String getPengarang() {
        return pengarang;
    }

    public String getKode() {
        return kode;
    }

    public static class Buku extends BahanPustaka {
        private int jumlahHalaman;

        public Buku(String judul, String pengarang, String kode, int jumlahHalaman) {
            super(judul, pengarang, kode);
            this.jumlahHalaman = jumlahHalaman;
        }

        public int getJumlahHalaman() { return jumlahHalaman; }
    }
}
```

```
File Edit Selection View Go Run Terminal Help
BahanPustaka.java
1 public class BahanPustaka {
2     public static class Buku extends BahanPustaka {
3
4     }
5
6     @Override
7     public void tampilkanInfo() {
8         super.tampilkanInfo();
9         System.out.println("Jumlah Halaman : " + jumlahHalaman);
10    }
11
12    public static class Majalah extends BahanPustaka {
13        private int edisi;
14
15        public Majalah(String judul, String pengarang, String kode, int edisi) {
16            super(judul, pengarang, kode);
17            this.edisi = edisi;
18        }
19
20        public int getEdisi() { return edisi; }
21
22        @Override
23        public void tampilkanInfo() {
24            super.tampilkanInfo();
25            System.out.println("Edisi : " + edisi);
26        }
27    }
28 }
```

Peminjaman.java

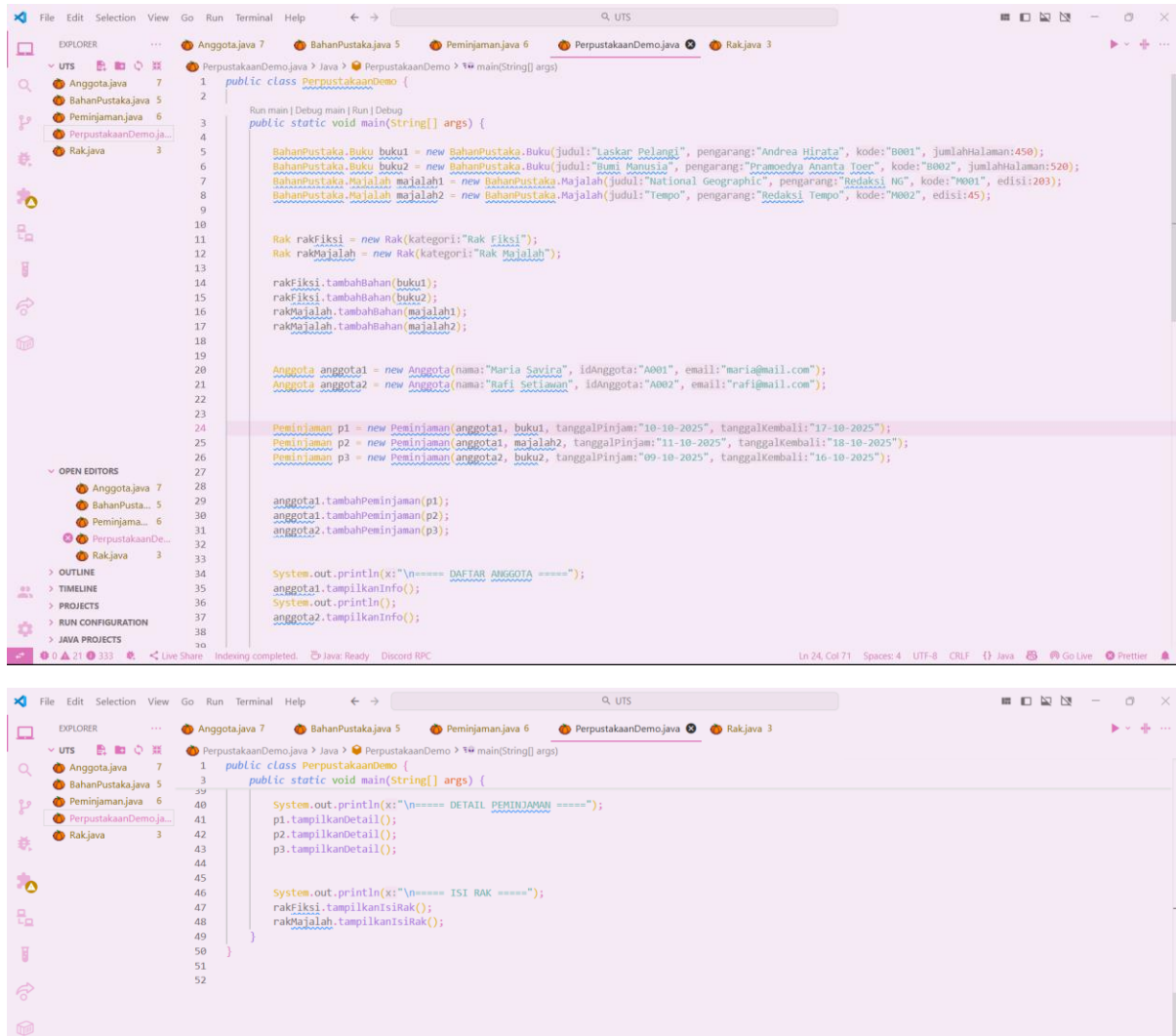
```
File Edit Selection View Go Run Terminal Help
Peminjaman.java
1 public class Peminjaman {
2
3     private Anggota anggota;
4     private BahanPustaka bahan;
5     private String tanggalPinjam;
6     private String tanggalKembali;
7
8
9     public Peminjaman(Anggota anggota, BahanPustaka bahan, String tanggalPinjam, String tanggalKembali) {
10         this.anggota = anggota;
11         this.bahan = bahan;
12         this.tanggalPinjam = tanggalPinjam;
13         this.tanggalKembali = tanggalKembali;
14     }
15
16     public BahanPustaka getBahan() {
17         return bahan;
18     }
19
20     public Anggota getAnggota() {
21         return anggota;
22     }
23
24     public String getTanggalPinjam() {
25         return tanggalPinjam;
26     }
27
28     public String getTanggalKembali() {
29         return tanggalKembali;
30     }
31
32     public int hitungDenda(int hariTerlambat) {
33         return hariTerlambat * 1000;
34     }
35
36     public int hitungDenda(int hariTerlambat, int tarifPerHari) {
37         return hariTerlambat * tarifPerHari;
38     }
39 }
```

```
1 public class Peminjaman {
2     public void hitungDenda(int hariTerlambat) {
3
4     }
5
6     public void tampilkanDetail() {
7         System.out.println("==== Detail Peminjaman =====");
8         System.out.println("Nama Anggota : " + anggota.getNama());
9         System.out.println("Email Anggota : " + anggota.getEmail());
10        System.out.println("Bahan Pustaka : " + bahan.getJudul());
11        System.out.println("Kode Pustaka : " + bahan.getKode());
12        System.out.println("Tanggal Pinjam : " + tanggalPinjam);
13        System.out.println("Tanggal Kembali : " + tanggalKembali);
14
15        if (bahan instanceof BahanPustaka.Buku) {
16            BahanPustaka.Buku b = (BahanPustaka.Buku) bahan;
17            System.out.println("Jenis : " + b.getJenis());
18            System.out.println("Jumlah Halaman : " + b.getJumlahHalaman());
19            System.out.println();
20        } else if (bahan instanceof BahanPustaka.Majalah) {
21            BahanPustaka.Majalah m = (BahanPustaka.Majalah) bahan;
22            System.out.println("Jenis : " + m.getJenis());
23            System.out.println("Edisi : " + m.getEdisi());
24            System.out.println();
25        }
26
27        int denda1 = hitungDenda(hariTerlambat);
28        int denda2 = hitungDenda(hariTerlambat, tarifPe);
29
30        System.out.println("Denda (tarif default): Rp" + denda1);
31        System.out.println("Denda (tarif custom) : Rp" + denda2);
32    }
33 }
```

Rak.java

```
1 import java.util.ArrayList;
2
3 public class Rak {
4     private String kategori;
5     private ArrayList<BahanPustaka> daftarBahan;
6
7     public Rak(String kategori) {
8         this.kategori = kategori;
9         this.daftarBahan = new ArrayList<BahanPustaka>();
10    }
11
12    public void tambahBahan(BahanPustaka b) {
13        daftarBahan.add(b);
14    }
15
16    public void tampilkanRak() {
17        System.out.println();
18        System.out.println("Rak Kategori: " + kategori);
19        for (int i = 0; i < daftarBahan.size(); i++) {
20            BahanPustaka b = daftarBahan.get(i);
21            System.out.println((i + 1) + ". " + b.getJudul() + " (" + b.getPengarang() + ")");
22        }
23    }
24
25    public String getKategori() {
26        return kategori;
27    }
28
29    public ArrayList<BahanPustaka> getDaftarBahan() {
30        return daftarBahan;
31    }
32 }
33
```

PerpustakaanDemo.java



```
1 public class PerpustakaanDemo {
2
3     public static void main(String[] args) {
4
5         BahanPustaka.Buku buku1 = new BahanPustaka.Buku(judul:"Laskar Pelangi", pengarang:"Andrea Hirata", kode:"B001", jumlahHalaman:450);
6         BahanPustaka.Buku buku2 = new BahanPustaka.Buku(judul:"Bumi Manusia", pengarang:"Pramoedya Ananta Toer", kode:"B002", jumlahHalaman:520);
7         BahanPustaka.Majalah majalah1 = new BahanPustaka.Majalah(judul:"National Geographic", pengarang:"Redaksi NG", kode:"M001", edisi:203);
8         BahanPustaka.Majalah majalah2 = new BahanPustaka.Majalah(judul:"Tempo", pengarang:"Redaksi Tempo", kode:"M002", edisi:145);
9
10
11         Rak rakFiksi = new Rak(kategori:"Rak Fiksi");
12         Rak rakMajalah = new Rak(kategori:"Rak Majalah");
13
14         rakFiksi.tambahBahan(buku1);
15         rakFiksi.tambahBahan(buku2);
16         rakMajalah.tambahBahan(majalah1);
17         rakMajalah.tambahBahan(majalah2);
18
19
20         Anggota anggota1 = new Anggota(nama:"Maria Savira", idAnggota:"A001", email:"maria@mail.com");
21         Anggota anggota2 = new Anggota(nama:"Rafi Setiawan", idAnggota:"A002", email:"rafi@mail.com");
22
23
24         Peminjaman p1 = new Peminjaman(anggota1, buku1, tanggalPinjam:"10-10-2025", tanggalKembali:"17-10-2025");
25         Peminjaman p2 = new Peminjaman(anggota1, majalah2, tanggalPinjam:"11-10-2025", tanggalKembali:"18-10-2025");
26         Peminjaman p3 = new Peminjaman(anggota2, buku2, tanggalPinjam:"09-10-2025", tanggalKembali:"16-10-2025");
27
28
29         anggota1.tambahPeminjaman(p1);
30         anggota1.tambahPeminjaman(p2);
31         anggota2.tambahPeminjaman(p3);
32
33
34         System.out.println(x:"\n===== DAFTAR ANGGOTA =====");
35         anggota1.tampilkanInfo();
36         System.out.println();
37         anggota2.tampilkanInfo();
38
39
40         System.out.println(x:"\n===== DETAIL PEMINJAMAN =====");
41         p1.tampilkanDetail();
42         p2.tampilkanDetail();
43         p3.tampilkanDetail();
44
45
46         System.out.println(x:"\n===== ISI RAK =====");
47         rakFiksi.tampilkanIsiRak();
48         rakMajalah.tampilkanIsiRak();
49
50     }
51
52 }
```

Output

<p>===== DAFTAR ANGGOTA =====</p> <p>ID : A001 Nama : Maria Savira Email : maria@mail.com Buku yang dipinjam : 1. Laskar Pelangi Pengarang : Andrea Hirata Kode Pustaka : B001 Jumlah halaman : 450 2. Tempo Pengarang : Redaksi Tempo Kode Pustaka : M002 Edisi : 45</p> <p>ID : A002 Nama : Rafi Setiawan Email : rafi@mail.com Buku yang dipinjam : 1. Bumi Manusia Pengarang : Pramoedya Ananta Toer Kode Pustaka : B002 Jumlah halaman : 520</p>	<p>===== DETAIL PEMINJAMAN =====</p> <p>=== Detail Peminjaman ===</p> <p>Nama Anggota : Maria Savira Email Anggota : maria@mail.com Bahan Pustaka : Laskar Pelangi Kode Pustaka : B001 Tanggal Pinjam: 10-10-2025 Tanggal Kembali: 17-10-2025 Jenis : Buku Jumlah Halaman: 450</p> <p>Denda (tarif default): Rp3000 Denda (tarif custom) : Rp6000</p> <p>=== Detail Peminjaman ===</p> <p>Nama Anggota : Maria Savira Email Anggota : maria@mail.com Bahan Pustaka : Tempo Kode Pustaka : M002 Tanggal Pinjam: 11-10-2025 Tanggal Kembali: 18-10-2025 Jenis : Majalah Edisi : 45</p> <p>Denda (tarif default): Rp3000 Denda (tarif custom) : Rp6000</p>	<p>=== Detail Peminjaman ===</p> <p>Nama Anggota : Rafi Setiawan Email Anggota : rafi@mail.com Bahan Pustaka : Bumi Manusia Kode Pustaka : B002 Tanggal Pinjam: 09-10-2025 Tanggal Kembali: 16-10-2025 Jenis : Buku Jumlah Halaman: 520</p> <p>Denda (tarif default): Rp3000 Denda (tarif custom) : Rp6000</p>
<p>===== ISI RAK =====</p> <p>Rak Kategori: Rak Fiksi 1. Laskar Pelangi (Andrea Hirata) 2. Bumi Manusia (Pramoedya Ananta Toer)</p> <p>Rak Kategori: Rak Majalah 1. National Geographic (Redaksi NG) 2. Tempo (Redaksi Tempo)</p>		

C. Analisis

1. Perbedaan Overloading dan Overriding adalah sebagai berikut :

Overloading terjadi ketika dua atau lebih method memiliki nama yang sama tetapi parameter berbeda (jumlah atau tipe datanya berbeda).

Overriding terjadi ketika subclass (class turunan) membuat method dengan nama dan parameter yang sama seperti di superclass untuk mengubah perilakunya.

Seperti contoh dalam kode saya :

Overloading pada class Peminjaman :


```

32 public int hitungDenda(int hariTerlambat) {
33     return hariTerlambat * 1000;
34 }
35
36 public int hitungDenda(int hariTerlambat, int tarifPerHari) {
37     return hariTerlambat * tarifPerHari;
38 }

```

Sedangkan overriding terdapat pada BahanPustaka.Buku (pada konstruktor) :

```

6 public BahanPustaka(String judul, String pengarang, String kode) {
7     this.judul = judul;
8     this.pengarang = pengarang;
9     this.kode = kode;
10 }
11
12 > public void tampilkanInfo() { ...
17
18 > public String getJudul() { ...
21
22 > public String getPengarang() { ...
25
26 > public String getKode() { ...
29
30
31 public static class Buku extends BahanPustaka {
32     private int jumlahHalaman;
33
34     public Buku(String judul, String pengarang, String kode, int jumlahHalaman) {
35         super(judul, pengarang, kode);
36         this.jumlahHalaman = jumlahHalaman;
37     }

```

2. Alasan memilih composition untuk X dan Aggregation untuk Y :

Hubungan Anggota dan Peminjaman menggunakan composition, karena setiap objek Peminjaman tidak dapat ada tanpa Anggota. Jika Anggota dihapus dari sistem, maka seluruh data Peminjaman miliknya juga ikut hilang.

Sedangkan hubungan Rak dan BahanPustaka menggunakan aggregation, karena Rak hanya berfungsi sebagai wadah atau pengelompokan, tapi bahan pustaka tetap bisa ada tanpa rak.

3. Berikut adalah aturan signature dalam method :

- Harus memiliki nama method yang unik untuk kombinasi parameter tertentu.
- Tipe parameter dan urutannya termasuk dalam signature, tapi tipe return dan access modifier tidak dihitung.
- Tidak boleh ada dua method dengan nama dan parameter yang identik dalam satu class.

Misalnya :

```

32     public int hitungDenda(int hariTerlambat) {
33         return hariTerlambat * 1000;
34     }
35
36     public int hitungDenda(int hariTerlambat) {
37         return hariTerlambat * tarifPerHari;
38     }

```

Jika parameter kedua dihapus pada overloading hitungDenda, sehingga terdapat dua parameter yang sama dalam fungsi yang berbeda sehingga akan error.

4. Alasan pemilihan access modifier :

Atribut `private String nama;` dan `private ArrayList<Peminjaman> peminjaman;` di class Anggota diberi modifier `private` agar tidak bisa diakses langsung dari luar class, menjaga keamanan dan konsistensi data anggota. Akses dilakukan hanya lewat getter atau method seperti `tambahPeminjaman()`.

Method `public void tampilkanInfo()` diberi modifier `public` agar dapat dipanggil dari class lain (misalnya Main) untuk menampilkan data anggota.

Karena tujuannya adalah menampilkan informasi ke pengguna, method ini harus bersifat terbuka (`accessible`).