PEMROGRAMAN BERORIENTASI OBJEK

UTS

ADHE WIDYA GALIH KARTIKA

SIB 2C_03

244107060067

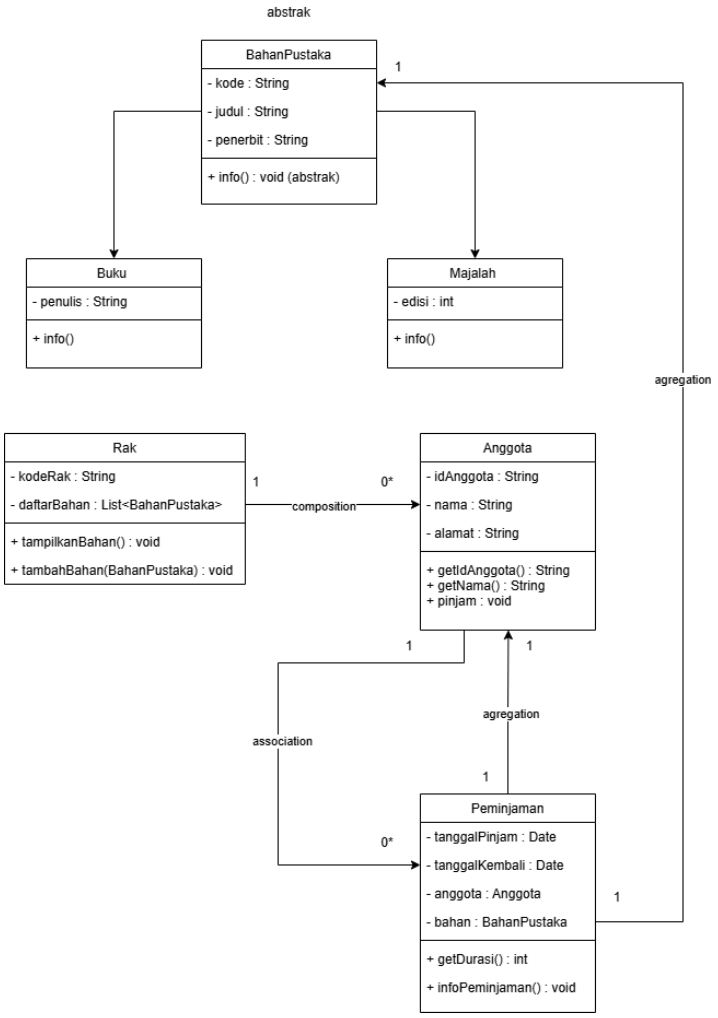
Jurusan Teknologi Informasi

POLITEKNIK NEGERI MALANG

2025

Bagian A – Perancangan

1. Gambar class diagram lengkap untuk skenario varian Anda. Tunjukkan multiplicity dan jenis relasi (association/aggregation/composition)



2. Tuliskan metode yang akan dioverride
   → info() di kelas Buku dan Majalah merupakan override dari info() (abstrak) di (BahanPustaka)
3. Tulis 3 keputusan desain terkait access modifier, enkapsulasi, dan pewarisan
   → Semua atribut dibuat private untuk menjaga enkapsulasi
   → Disediakan getter/setter hanya bila dibutuhkan
   → Menggunakan pewarisan agar dari class Buku dan Majalah dapat berbagi atribut umum dari class BahanPustaka

Bagian B – Implementasi

Implementasikan kelas inti berikut: Anggota, BahanPustaka (abstrak), Buku, Majalah, Peminjaman, Rak.

4. Syarat minimal implementasi:
   • Semua atribut private, sediakan getter/setter yang diperlukan saja.
   • 1 konstruktor berparameter di tiap kelas domain utama.
   • 1 inheritance + 1 override di tiap subclass.
   • 1 method overloading fungsional (beda signature).
   • Method main singkat yang mendemokan relasi & polimorfisme.

class Anggota

```java
public class Anggota {
    private String idAnggota;
    private String nama;

    public Anggota(String idAnggota, String nama) {
        this.idAnggota = idAnggota;
        this.nama = nama;
    }

    public String getIdAnggota() {
        return idAnggota;
    }

    public String getNama() {
        return nama;
    }

    public void pinjam(BahanPustaka bahan) {
        System.out.println(nama + " meminjam buku " + bahan.getJudul());
        bahan.info();
    }

    public void kembalikan(BahanPustaka bahan) {
        System.out.println(nama + " mengembalikan buku " + bahan.getJudul());
    }
}
```

class Peminjaman

```java
import java.time.LocalDate;
import java.time.temporal.ChronoUnit;

public class Peminjaman {
    private LocalDate tanggalPinjam;
    private LocalDate tanggalKembali;
    private Anggota anggota;
    private BahanPustaka bahan;

    public Peminjaman(Anggota anggota, BahanPustaka bahan, LocalDate tanggalPinjam, LocalDate tanggalKembali) {
        this.anggota = anggota;
        this.bahan = bahan;
        this.tanggalPinjam = tanggalPinjam;
        this.tanggalKembali = tanggalKembali;
    }

    public long getDurasi() {
        return ChronoUnit.DAYS.between(tanggalPinjam, tanggalKembali);
    }

    public void infoPeminjaman() {
        System.out.println("Anggota            : " + anggota.getNama());
        System.out.println("Judul Buku         : " + bahan.getJudul());
        System.out.println("Tanggal Pinjam     : " + tanggalPinjam);
        System.out.println("Tanggal Kembali    : " + tanggalKembali);
        System.out.println("Durasi Peminjaman  : " + getDurasi() + " hari");
        System.out.println(x: "----------------------------------");
    }
}
```

class BahanPustaka

```java
abstract class BahanPustaka {
    private String kode;
    private String judul;
    private String penerbit;

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

    public String getKode() {
        return kode;
    }

    public String getJudul() {
        return judul;
    }

    public abstract void info();
}
```

class Buku

```java
public class Buku extends BahanPustaka {
    private String penulis;

    public Buku(String kode, String judul, String penerbit, String penulis) {
        super(kode, judul, penerbit);
        this.penulis = penulis;
    }

    public String getPenulis() {
        return penulis;
    }

    @Override
    public void info() {
        System.out.println("Kode    : " + getKode());
        System.out.println("Judul   : " + getJudul());
        System.out.println("Penulis : " + penulis);
        System.out.println(x: "-----------------------------------");
    }
}
```

class Majalah

```java
public class Majalah extends BahanPustaka {
    private int edisi;

    public Majalah(String kode, String judul, String penerbit, int edisi) {
        super(kode, judul, penerbit);
        this.edisi = edisi;
    }

    public int getEdisi() {
        return edisi;
    }

    @Override
    public void info() {
        System.out.println("Kode    : " + getKode());
        System.out.println("Judul   : " + getJudul());
        System.out.println("Edisi   : " + edisi);
        System.out.println(x: "-----------------------------------");
    }
}
```

class Rak

```java
import java.util.ArrayList;
import java.util.List;

public class Rak {
    private String kodeRak;
    private List<BahanPustaka> daftarBahan = new ArrayList<>();

    public Rak(String kodeRak) {
        this.kodeRak = kodeRak;
    }

    public void tambahBahan(BahanPustaka bahan) {
        daftarBahan.add(bahan);
    }

    public void tampilkanBahan() {
        System.out.println("\n==============================");
        System.out.println("Daftar Isi Rak : " + kodeRak);
        System.out.println("==============================");
        for (BahanPustaka b : daftarBahan) {
            b.info();
        }
    }
}
```

class PerpustakaanMain

```java
import java.time.LocalDate;

public class PerpustakaanMain {
    Run main | Debug main | Run | Debug
    public static void main(String[] args) {

        Buku buku1 = new Buku(kode: "B001", judul: "Belajar Java", penerbit: "Penerbit A", penulis: "John Doe");
        Buku buku2 = new Buku(kode: "B002", judul: "Pemrograman Python", penerbit: "Penerbit B", penulis: "Jane Smith");
        Buku buku3 = new Buku(kode: "B003", judul: "Data Science", penerbit: "Penerbit E", penulis: "Alice Johnson");

        Majalah majalah1 = new Majalah(kode: "M001", judul: "Teknologi Terkini", penerbit: "Penerbit C", edisi: 2023);
        Majalah majalah2 = new Majalah(kode: "M002", judul: "Sains dan Alam", penerbit: "Penerbit D", edisi: 5);

        Rak rak1 = new Rak(kodeRak: "Rak A");
        rak1.tambahBahan(buku1);
        rak1.tambahBahan(majalah1);
        rak1.tambahBahan(buku2);

        Anggota anggota1 = new Anggota(idAnggota: "A001", nama: "Alice");
        Anggota anggota2 = new Anggota(idAnggota: "A002", nama: "Bob");
        Anggota anggota3 = new Anggota(idAnggota: "A003", nama: "Charlie");
        anggota1.pinjam(buku1);
        anggota2.pinjam(majalah1);
        anggota3.kembalikan(buku3);
```

```java
        System.out.println(x: "\n===== DATA PEMINJAMAN =====");
        anggota1.pinjam(buku1);
        anggota2.pinjam(majalah1);
        anggota3.kembalikan(buku3);

        System.out.println(x: "\n===== DATA PEMINJAMAN LENGKAP =====");
        Peminjaman peminjaman1 = new Peminjaman(anggota1, buku1, LocalDate.of(year: 2023,month: 10,dayOfMonth: 01), LocalDate.of(year: 2023,month: 10,dayOfMonth… 10));
        Peminjaman peminjaman2 = new Peminjaman(anggota2, majalah1, LocalDate.of(year: 2023,month: 10,dayOfMonth: 05), LocalDate.of(year: 2023,month: 10,dayOfMonth… 12));
        peminjaman1.infoPeminjaman();
        peminjaman2.infoPeminjaman();

        rak1.tampilkanBahan();
    }
}
```

Output

```
Alice meminjam buku Belajar Java
Kode    : B001
Judul   : Belajar Java
Penulis : John Doe
-----------------------------------
Bob meminjam buku Teknologi Terkini
Kode    : M001
Judul   : Teknologi Terkini
Edisi   : 2023
-----------------------------------
Charlie mengembalikan buku Data Science

===== DATA PEMINJAMAN =====
Alice meminjam buku Belajar Java
Kode    : B001
Judul   : Belajar Java
Penulis : John Doe
-----------------------------------
Bob meminjam buku Teknologi Terkini
Kode    : M001
Judul   : Teknologi Terkini
Edisi   : 2023
-----------------------------------
Charlie mengembalikan buku Data Science

===== DATA PEMINJAMAN LENGKAP =====
Anggota             : Alice
Judul Buku          : Belajar Java
Tanggal Pinjam      : 2023-10-01
Tanggal Kembali     : 2023-10-10
Durasi Peminjaman   : 9 hari
-----------------------------------
Anggota             : Bob
Judul Buku          : Teknologi Terkini
Tanggal Pinjam      : 2023-10-05
Tanggal Kembali     : 2023-10-12
Durasi Peminjaman   : 7 hari
-----------------------------------

============================
Daftar Isi Rak : Rak A
============================
Kode    : B001
Judul   : Belajar Java
Penulis : John Doe
-----------------------------------
Kode    : M001
Judul   : Teknologi Terkini
Edisi   : 2023
-----------------------------------
Kode    : B002
Judul   : Pemrograman Python
Penulis : Jane Smith
-----------------------------------
PS C:\UTS PBO>
```

Bagian C – Analisis

5. Jelaskan perbedaan overloading vs overriding dan berikan contoh dari kode Anda.
   → Overloading : Membuat beberapa metode dengan nama sama tetapi parameternya berbeda, contoh : public void pinjam(BahanPustaka bahan) { ... } dan public void pinjam(BahanPustaka bahan, int hari) { ... }
   → Overriding : Ketika subclass mengubah method dari superclass-nya dengan nama, parameter, dan return type yang sama, tapi isinya berbeda, contoh :

```java
@Override
    public void info() {
        System.out.println("Kode    : " + getKode());
        System.out.println("Judul   : " + getJudul());
        System.out.println("Edisi   : " + edisi);
```

```
System.out.println("--------------------------------------");
    }
```

Di class Majalah menampilkan Edisi

```
@Override
    public void info() {
        System.out.println("Kode    : " + getKode());
        System.out.println("Judul   : " + getJudul());
        System.out.println("Penulis : " + penulis);

System.out.println("--------------------------------------");
    }
```

Di class Buku menampilkan Penulis

6. Mengapa Anda memilih composition untuk X dan aggregation untuk Y?
   → Composition : Rak memiliki BahanPustaka. Ketika Rak dihapus, maka koleksi BahanPustaka di dalamnya juga ikut hilang. Tanpa Rak, daftar bahan pustaka tersebut tidak relevan.
   → Aggregation : Peminjaman memiliki Anggota dan BahanPustaka. Anggota dan BahanPustaka tetap bisa berjalan tanpa Peminjaman. Hubungan “meminjam” bersifat sementara, tidak saling memiliki.

7. Sebutkan aturan method signature dan contoh yang tidak valid.
   → Method signature terdiri dari :
   - Nama method
   - Daftar tipe parameter (beserta urutannya)

   Contoh tidak valid :
   public void pinjam(BahanPustaka bahan)
   public int pinjam(BahanPustaka bahan)

   Contoh valid :
   public void pinjam(BahanPustaka bahan)
   public void pinjam(BahanPustaka bahan, int hari)

8. Alasan pemilihan access modifier pada minimal 2 atribut & 1 method.
   → private String idAnggota : dipilih private agar hanya bisa diakses dari dalam class Anggota untuk menjaga keamanan data ID anggota supaya tidak diubah sembarangan dari luar.
   → private List<BahanPustaka> daftarBahan : dipilih private agar daftar isi rak hanya bisa dimanipulasi lewat method resmi (tambahBahan) bukan langsung dari luar.
   → public void tampilkanBahan() : bersifat public karena method ini harus bisa dipanggil dari luar class misalnya oleh PerpustakaanMain untuk menampilkan isi rak.