

LAPORAN JOBSHEET 4 RELASI KELAS

Mata Kuliah : Praktikum Pemrograman Berbasis Objek

Dosen : Irsyad Arif Mashudi, S.Kom., M.Kom



Ilham Dharma Atmaja

24410702020

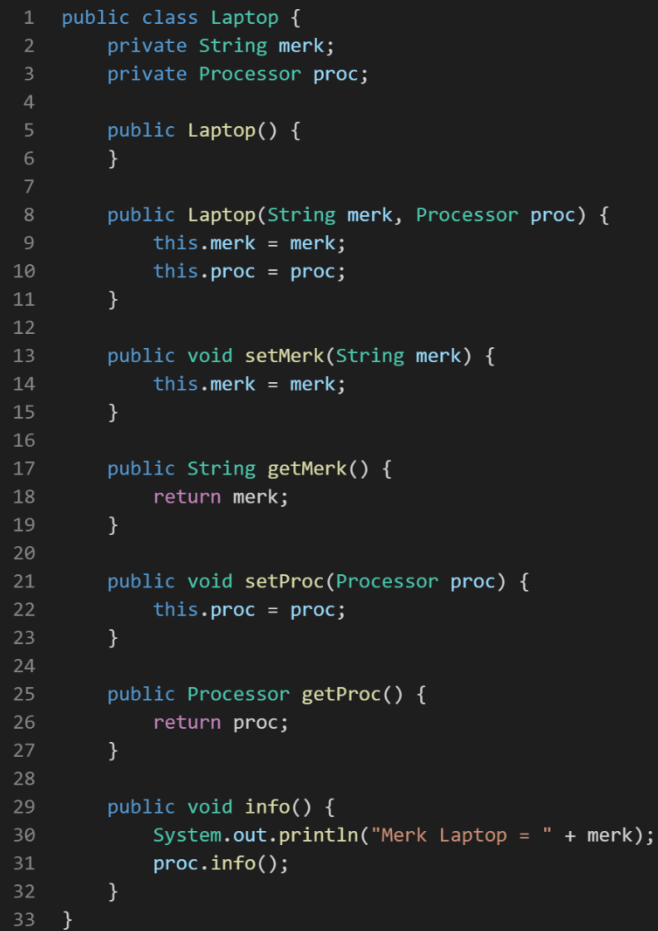
Kelas :TI 2D

**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG TAHUN 2025**

I. Percobaan 1: Relasi Agregasi


1. Verifikasi hasil Percobaan

Class Laptop

A screenshot of a Java code editor with a dark background and light-colored text. The code defines a public class named 'Laptop'. It has two private attributes: 'merk' of type 'String' and 'proc' of type 'Processor'. The class includes a default constructor, a parameterized constructor that initializes 'merk' and 'proc', and four methods: 'setMerk' to update the brand, 'getMerk' to retrieve the brand, 'setProc' to update the processor, and 'getProc' to retrieve the processor. Additionally, there is an 'info' method that prints the brand and calls the 'info' method of the processor object.

```
1 public class Laptop {
2     private String merk;
3     private Processor proc;
4
5     public Laptop() {
6     }
7
8     public Laptop(String merk, Processor proc) {
9         this.merk = merk;
10        this.proc = proc;
11    }
12
13    public void setMerk(String merk) {
14        this.merk = merk;
15    }
16
17    public String getMerk() {
18        return merk;
19    }
20
21    public void setProc(Processor proc) {
22        this.proc = proc;
23    }
24
25    public Processor getProc() {
26        return proc;
27    }
28
29    public void info() {
30        System.out.println("Merk Laptop = " + merk);
31        proc.info();
32    }
33 }
```

Class Processor




```

1  public class Processor {
2      private String merk;
3      private double cache;
4
5      public Processor() {
6      }
7
8      public Processor(String merk, double cache) {
9          this.merk = merk;
10         this.cache = cache;
11     }
12
13     public void setMerk(String merk) {
14         this.merk = merk;
15     }
16
17     public String getMerk() {
18         return merk;
19     }
20
21     public void setCache(double cache) {
22         this.cache = cache;
23     }
24
25     public double getCache() {
26         return cache;
27     }
28
29     public void info() {
30         System.out.printf("Merk Processor = %s\n", merk);
31         System.out.printf("Cache Memory = %.2f\n", cache);
32     }
33 }

```

Main



```

1  public class MainPercobaan1 {
2      public static void main(String[] args) {
3          Processor p = new Processor("Intel i5", 3);
4          Laptop l = new Laptop("Thinkpad", p);
5          l.info();
6
7          Processor p1 = new Processor();
8          p1.setMerk("Intel i5");
9          p1.setCache(4);
10         Laptop l1 = new Laptop();
11         l1.setMerk("Thinkpad");
12         l1.setProc(p1);
13         l1.info();
14     }
15 }

```

Output

```
Merk Laptop = Thinkpad
Merk Processor = Intel i5
Cache Memory = 3,00
Merk Laptop = Thinkpad
Merk Processor = Intel i5
Cache Memory = 4,00
PS C:\Users\ILHAM DHARMA A\OneD
```

2. Jawaban Pertanyaan Percobaan 1

1. Perbedaan Setter dan Getter

Method

setter digunakan untuk mengatur atau mengubah nilai dari atribut private dalam sebuah kelas. Ini memungkinkan Anda untuk memanipulasi data objek dengan cara yang terkontrol.

Sementara itu, method

getter digunakan untuk mengambil atau mendapatkan nilai dari atribut private. Ini memberikan akses baca ke data internal objek tanpa mengeksposnya secara langsung.

2. Perbedaan Penggunaan Konstruktor

- **Konstruktor default:** Konstruktor ini tidak memiliki parameter. Penggunaannya adalah untuk membuat objek tanpa harus menginisialisasi atributnya secara langsung pada saat pembuatan. Nilai atribut akan diisi menggunakan method setter setelah objek dibuat.
- **Konstruktor berparameter:** Konstruktor ini menerima parameter yang sesuai dengan atribut yang akan diinisialisasi. Penggunaannya adalah untuk membuat objek dan langsung menginisialisasi atributnya dengan nilai yang diberikan, seperti pada contoh `Laptop l = new Laptop("Thinkpad", p)`.

3. Atribut Bertipe Objek di Class Laptop

Dari atribut yang ada di class

Laptop, atribut yang bertipe objek adalah **proc**. Atribut

proc memiliki tipe data Processor, yang merupakan kelas lain.

4. Baris yang Menunjukkan Relasi

Baris kode yang menunjukkan bahwa class

Laptop memiliki relasi dengan class Processor adalah deklarasi atribut proc pada class Laptop.

```
private Processor proc;
```

5. Fungsi proc.info()

Sintaks

proc.info() pada class Laptop berfungsi untuk memanggil method **info()** yang ada di dalam objek **proc**. Karena

proc adalah objek dari class Processor, method ini akan menampilkan informasi dari objek Processor yang terkait, yaitu merk dan cache memory.

6. Perubahan Kode dan Hasil

Pada baris kode

Laptop l = new Laptop("Thinkpad", p);, p adalah sebuah objek Processor yang telah dibuat sebelumnya. Objek

p ini digunakan sebagai argumen untuk konstruktor class Laptop.

Jika kode diubah menjadi

Laptop l = new Laptop("Thinkpad", new Processor("Intel i5", 3));, hasilnya akan tetap sama. Perbedaannya hanya pada cara objek

Processor dibuat. Pada kode pertama, objek p dibuat terlebih dahulu dan disimpan dalam variabel, kemudian disalurkan ke konstruktor Laptop. Pada kode yang diubah, objek

Processor langsung dibuat sebagai argumen anonim di dalam konstruktor Laptop.

II. Percobaan 2: Relasi Agregasi

Verifikasi Hasil Percobaan

Class Mobil

```
1  public class Mobil {
2      private String merk;
3      private int biaya;
4
5      public Mobil() {
6      }
7
8      public String getMerk() {
9          return merk;
10     }
11
12     public void setMerk(String merk) {
13         this.merk = merk;
14     }
15
16     public int getBiaya() {
17         return biaya;
18     }
19
20     public void setBiaya(int biaya) {
21         this.biaya = biaya;
22     }
23
24     public int hitungBiayaMobil(int hari) {
25         return biaya * hari;
26     }
27 }
```

Class Sopir

```
1  public class Sopir {
2      private String nama;
3      private int biaya;
4
5      public Sopir() {
6      }
7
8      public String getNama() {
9          return nama;
10     }
11
12     public void setNama(String nama) {
13         this.nama = nama;
14     }
15
16     public int getBiaya() {
17         return biaya;
18     }
19
20     public void setBiaya(int biaya) {
21         this.biaya = biaya;
22     }
23
24     public int hitungBiayaSopir(int hari) {
25         return biaya * hari;
26     }
27 }
```

Class Pelanggan

```
1  public class Pelanggan {
2      private String nama;
3      private Mobil mobil;
4      private Sopir sopir;
5      private int hari;
6
7      public Pelanggan() {
8      }
9
10     public String getNama() {
11         return nama;
12     }
13
14     public void setNama(String nama) {
15         this.nama = nama;
16     }
17
18     public Mobil getMobil() {
19         return mobil;
20     }
21
22     public void setMobil(Mobil mobil) {
23         this.mobil = mobil;
24     }
25
26     public Sopir getSopir() {
27         return sopir;
28     }
29
30     public void setSopir(Sopir sopir) {
31         this.sopir = sopir;
32     }
33
34     public int getHari() {
35         return hari;
36     }
37
38     public void setHari(int hari) {
39         this.hari = hari;
40     }
41
42     public int hitungBiayaTotal() {
43         return mobil.hitungBiayaMobil(hari) + sopir.hitungBiayaSopir(hari);
44     }
45 }
```

Main

```
1 public class MainPercobaan2 {
2     public static void main(String[] args) {
3         Mobil m = new Mobil();
4         m.setMerk("Avanza");
5         m.setBiaya(350000);
6         Sopir s = new Sopir();
7         s.setNama("John Doe");
8         s.setBiaya(200000);
9         Pelanggan p = new Pelanggan();
10        p.setNama("Jane Doe");
11        p.setMobil(m);
12        p.setSopir(s);
13        p.setHari(2);
14        System.out.println("Biaya Total = " + p.hitungBiayaTotal());
15    }
16 }
```

Output

```
Biaya Total = 1100000
PS C:\Users\ILHAM DHARMA
```

2. Pertanyaan

1. Baris yang Menunjukkan Relasi

Dalam class

Pelanggan, baris program yang menunjukkan relasi dengan class Mobil dan class Sopir adalah deklarasi atribut mobil dan sopir.

```
private Mobil mobil;
```

```
private Sopir sopir;
```

2. Argumen hari

Method

hitungBiayaMobil pada class Mobil dan hitungBiayaSopir pada class Sopir memerlukan argumen hari karena biaya sewa dihitung per hari. Tanpa nilai hari, perhitungan biaya total tidak dapat dilakukan.

3. Fungsi mobil.hitungBiayaMobil(hari) dan sopir.hitungBiayaSopir(hari)

Perintah

mobil.hitungBiayaMobil(hari) dan sopir.hitungBiayaSopir(hari) digunakan untuk memanggil method perhitungan biaya dari objek Mobil dan Sopir yang terkait. Hasil dari kedua perhitungan ini kemudian dijumlahkan untuk mendapatkan total biaya sewa.

4. Fungsi p.setMobil(m) dan p.setSopir(s)

Sintaks

p.setMobil(m) dan p.setSopir(s) digunakan untuk menghubungkan objek Mobil (dengan nama m) dan objek Sopir (dengan nama s) ke dalam objek Pelanggan (dengan nama p). Ini merupakan implementasi dari relasi

has-a karena objek Pelanggan "memiliki" objek Mobil dan Sopir.

5. Fungsi p.hitungBiayaTotal()

Perintah

p.hitungBiayaTotal() berfungsi untuk memanggil method hitungBiayaTotal() yang ada di dalam objek Pelanggan. Method ini akan menghitung biaya sewa total dengan menjumlahkan biaya mobil dan biaya sopir, yang masing-masing dikalikan dengan jumlah hari sewa.

6. Fungsi p.getMobil().getMerk()

Ketika kode

System.out.println(p.getMobil().getMerk()); ditambahkan, program akan mencetak merk dari objek Mobil yang terhubung dengan objek Pelanggan.

- p.getMobil(): Mengambil objek Mobil yang ada di dalam objek Pelanggan p.
- .getMerk(): Memanggil method getMerk() pada objek Mobil yang baru saja didapatkan, untuk mengambil nilai atribut merk-nya.

III. Percobaan 3: Relasi Many-to-one

1. Verifikasi Hasil Percobaan

Class KeretaApi

```
1 public class KeretaApi {
2     private String nama;
3     private String kelas;
4     private Pegawai masinis;
5     private Pegawai asisten;
6
7     public KeretaApi(String nama, String kelas, Pegawai masinis) {
8         this.nama = nama;
9         this.kelas = kelas;
10        this.masinis = masinis;
11    }
12
13    public KeretaApi(String nama, String kelas, Pegawai masinis, Pegawai asisten) {
14        this.nama = nama;
15        this.kelas = kelas;
16        this.masinis = masinis;
17        this.asisten = asisten;
18    }
19
20    public void setNama(String nama) {
21        this.nama = nama;
22    }
23
24    public String getNama() {
25        return nama;
26    }
27
28    public void setKelas(String kelas) {
29        this.kelas = kelas;
30    }
31
32    public String getKelas() {
33        return kelas;
34    }
35
36    public void setMasinis(Pegawai masinis) {
37        this.masinis = masinis;
38    }
39
40    public Pegawai getMasinis() {
41        return masinis;
42    }
43
44    public void setAsisten(Pegawai asisten) {
45        this.asisten = asisten;
46    }
47
48    public Pegawai getAsisten() {
49        return asisten;
50    }
51
52    public String info() {
53        String info = "";
54        info += "Nama: " + this.nama + "\n";
55        info += "Kelas: " + this.kelas + "\n";
56        info += "Masinis: " + this.masinis.info() + "\n";
57        if (this.asisten != null) {
58            info += "Asisten: " + this.asisten.info() + "\n";
59        }
60        return info;
61    }
62 }
```


Class Pegawai

```
1 public class Pegawai {
2     private String nip;
3     private String nama;
4
5     public Pegawai(String nip, String nama) {
6         this.nip = nip;
7         this.nama = nama;
8     }
9
10    public void setNip(String nip) {
11        this.nip = nip;
12    }
13
14    public String getNip() {
15        return nip;
16    }
17
18    public void setNama(String nama) {
19        this.nama = nama;
20    }
21
22    public String getNama() {
23        return nama;
24    }
25
26    public String info() {
27        String info = "";
28        info += "Nip: " + this.nip + "\n";
29        info += "Nama: " + this.nama + "\n";
30        return info;
31    }
32 }
```

Main

```
1 public class MainPercobaan3 {
2     public static void main(String[] args) {
3         Pegawai masinis = new Pegawai("1234", "Spongebob Squarepants");
4         Pegawai asisten = new Pegawai("4567", "Patrick Star");
5         KeretaApi keretaApi = new KeretaApi("Gaya Baru", "Bisnis", masinis, asisten);
6         System.out.println(keretaApi.info());
7     }
8 }
```

Output

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

Asisten: Nip: 4567
Nama: Patrick Star
```

```
PS C:\Users\ILHAM DHARMA A\O
```

2. Pertanyaan

1. Fungsi `this.masinis.info()` dan `this.asisten.info()`

Di dalam method

`info()` pada class `Keretapi`, baris `this.masinis.info()` dan `this.asisten.info()` digunakan untuk memanggil method `info()` dari objek `Pegawai` yang merepresentasikan masinis dan asisten. Ini memungkinkan

`Keretapi` untuk menampilkan informasi lengkap mengenai masinis dan asistennya (NIP dan nama) tanpa perlu mengakses atributnya secara langsung.

2. Hasil Output Program Baru

Output dari program tersebut akan menghasilkan `NullPointerException`. Ini terjadi karena konstruktor yang dipanggil hanya memiliki tiga parameter, yaitu

nama, kelas, dan masinis. Atribut

asisten pada objek `keretaApi` tidak diinisialisasi dan nilainya tetap `null`. Ketika

`keretaApi.info()` dipanggil, program mencoba memanggil `info()` dari objek asisten yang `null`, sehingga terjadilah `NullPointerException`.

3. Perbaikan Class `KeretaApi`

Untuk memperbaiki class `KeretaApi`, tambahkan pengecekan `null` pada method `info()`

sebelum memanggil method `info()` dari objek asisten. Ini akan mencegah

`NullPointerException` jika asisten belum diinisialisasi.

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

IV. Percobaan 4: Relasi One-to-many

1. Verifikasi Hasil Percobaan

Class Kursi

```
1 public class Kursi {
2     private String nomor;
3     private Penumpang penumpang;
4
5     public Kursi(String nomor) {
6         this.nomor = nomor;
7     }
8
9     public void setNomor(String nomor) {
10         this.nomor = nomor;
11     }
12
13     public String getNomor() {
14         return nomor;
15     }
16
17     public void setPenumpang(Penumpang penumpang) {
18         this.penumpang = penumpang;
19     }
20
21     public Penumpang getPenumpang() {
22         return penumpang;
23     }
24
25     public String info() {
26         String info = "";
27         info += "Nomor: " + nomor + "\n";
28         if (this.penumpang != null) {
29             info += "Penumpang: " + penumpang.info() + "\n";
30         }
31         return info;
32     }
33 }
```

Class Gerbang

```
1 public class Gerbang {
2     private String kode;
3     private Kursi[] arrayKursi;
4
5     public Gerbang(String kode, int jumlah) {
6         this.kode = kode;
7         this.arrayKursi = new Kursi[jumlah];
8         this.initKursi();
9     }
10
11     private void initKursi() {
12         for (int i = 0; i < arrayKursi.length; i++) {
13             this.arrayKursi[i] = new Kursi(String.valueOf(i + 1));
14         }
15     }
16
17     public String getKode() {
18         return kode;
19     }
20
21     public void setKode(String kode) {
22         this.kode = kode;
23     }
24
25     public Kursi[] getArrayKursi() {
26         return arrayKursi;
27     }
28
29     public void setPenumpang(Penumpang penumpang, int nomor) {
30         // Modifikasi untuk mencegah penempatan penumpang yang sudah ada
31         if (this.arrayKursi[nomor - 1].getPenumpang() == null) {
32             this.arrayKursi[nomor - 1].setPenumpang(penumpang);
33         } else {
34             System.out.println("Maaf, kursi sudah terisi.");
35         }
36     }
37
38     public String info() {
39         String info = "";
40         info += "Kode: " + kode + "\n";
41         for (Kursi kursi : arrayKursi) {
42             info += kursi.info();
43         }
44         return info;
45     }
46 }
```

Class Penumpang

```
1 public class Penumpang {
2     private String ktp;
3     private String nama;
4
5     public Penumpang(String ktp, String nama) {
6         this.ktp = ktp;
7         this.nama = nama;
8     }
9
10    public void setKtp(String ktp) {
11        this.ktp = ktp;
12    }
13
14    public String getKtp() {
15        return ktp;
16    }
17
18    public void setName(String nama) {
19        this.nama = nama;
20    }
21
22    public String getName() {
23        return nama;
24    }
25
26    public String info() {
27        String info = "";
28        info += "Ktp: " + ktp + "\n";
29        info += "Nama: " + nama + "\n";
30        return info;
31    }
32 }
```

Main

```
1 public class MainPercobaan4 {
2     public static void main(String[] args) {
3         Penumpang p = new Penumpang("12345", "Mr. Krab");
4         Gerbong gerbong = new Gerbong("A", 10);
5         gerbong.setPenumpang(p, 1);
6         System.out.println(gerbong.info());
7     }
8 }
```

Output

```
Kode: A
Nomor: 1
Penumpang: Ktp: 12345
Nama: Mr. Krab
```

```
Nomor: 2
Nomor: 3
Nomor: 4
Nomor: 5
Nomor: 6
Nomor: 7
Nomor: 8
Nomor: 9
Nomor: 10
```

```
PS C:\Users\ILHAM DHARMA>
```

2. Pertanyaan

1. Jumlah Kursi

Pada main program dalam class

MainPercobaan4, jumlah kursi dalam Gerbong A adalah **10**. Ini ditentukan oleh argumen jumlah yang diberikan pada saat instansiasi objek Gerbong (new Gerbong("A", 10)).

2. Maksud Kode if (this.penumpang != null)

Potongan kode

if (this.penumpang != null) pada method info() di class Kursi berfungsi untuk memeriksa apakah kursi tersebut sudah memiliki penumpang. Jika atribut

penumpang tidak null, yang berarti ada objek Penumpang yang sudah dihubungkan, maka program akan memanggil method info() dari objek Penumpang tersebut untuk menampilkan detailnya. Jika

penumpang adalah null, maka tidak ada informasi penumpang yang ditampilkan untuk kursi tersebut.

3. Nilai nomor Dikurangi Satu

Nilai

nomor pada method setPenumpang() di class Gerbong dikurangi dengan angka 1 (nomor - 1) karena indeks array di Java dimulai dari **0**. Pengguna umumnya menganggap nomor kursi dimulai dari 1, sehingga perlu ada penyesuaian untuk mencocokkan dengan indeks array yang benar.

4. Apa yang Terjadi jika Kursi Sudah Terisi

Jika Anda menginstansiasi objek budi dan mencoba menetapkan ke kursi yang sama (gerbong.setPenumpang(budi, 1)), objek budi akan **menggantikan** objek Mr. Krab yang sudah ada sebelumnya. Ini karena method

setPenumpang() akan menimpa objek Penumpang yang ada pada indeks array yang ditunjuk (this.arrayKursi[nomor-1]) dengan objek Penumpang yang baru.

5. Modifikasi Program

Untuk mencegah kursi yang sudah terisi diduduki oleh penumpang lain, modifikasi method setPenumpang() pada class Gerbong dengan menambahkan logika pengecekan.

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

V. Tugas

Penjelasan Atribut dan Metode:

- **Anggota**
 - + idAnggota: String
 - + nama: String
 - + getNama(): String
 - + info(): String
- **Buku**
 - + idBuku: String
 - + judul: String
 - + getJudul(): String
 - + info(): String
- **Peminjaman**
 - + tanggalPinjam: String
 - + anggota: Anggota
 - + buku: Buku
 - + getAnggota(): Anggota
 - + getBuku(): Buku
 - + info(): String
- **Perpustakaan**
 - + nama: String
 - + daftarAnggota: Anggota[]
 - + daftarBuku: Buku[]
 - + daftarPeminjaman: Peminjaman[]
 - + tambahAnggota(Anggota a): void
 - + tambahBuku(Buku b): void
 - + pinjamBuku(Anggota a, Buku b): void
 - + info(): String

➤ Verifikasi Program

Class Anggota

```
1 public class Anggota {
2     private String idAnggota;
3     private String nama;
4
5     public Anggota(String idAnggota, String nama) {
6         this.idAnggota = idAnggota;
7         this.nama = nama;
8     }
9
10    public String getIdAnggota() {
11        return idAnggota;
12    }
13
14    public String getNama() {
15        return nama;
16    }
17
18    public String info() {
19        return "ID Anggota: " + idAnggota + ", Nama: " + nama;
20    }
21 }
```

Class Buku

```
1 public class Buku {
2     private String idBuku;
3     private String judul;
4
5     public Buku(String idBuku, String judul) {
6         this.idBuku = idBuku;
7         this.judul = judul;
8     }
9
10    public String getIdBuku() {
11        return idBuku;
12    }
13
14    public String getJudul() {
15        return judul;
16    }
17
18    public String info() {
19        return "ID Buku: " + idBuku + ", Judul: " + judul;
20    }
21 }
```

Class Peminjaman

```
1 import java.time.LocalDate;
2
3 public class Peminjaman {
4     private Anggota anggota;
5     private Buku buku;
6     private LocalDate tanggalPinjam;
7
8     public Peminjaman(Anggota anggota, Buku buku) {
9         this.anggota = anggota;
10        this.buku = buku;
11        this.tanggalPinjam = LocalDate.now();
12    }
13
14    public Anggota getAnggota() {
15        return anggota;
16    }
17
18    public Buku getBuku() {
19        return buku;
20    }
21
22    public LocalDate getTanggalPinjam() {
23        return tanggalPinjam;
24    }
25
26    public String info() {
27        return "Peminjaman oleh " + anggota.getNama() +
28            " dengan buku " + buku.getJudul() +
29            " pada tanggal " + tanggalPinjam.toString();
30    }
31 }
```

Class Perpustakaan

```
1 import java.util.ArrayList;
2
3 public class Perpustakaan {
4     private String nama;
5     private ArrayList<Anggota> daftarAnggota;
6     private ArrayList<Buku> daftarBuku;
7     private ArrayList<Peminjaman> daftarPeminjaman;
8
9     public Perpustakaan(String nama) {
10         this.nama = nama;
11         this.daftarAnggota = new ArrayList<>();
12         this.daftarBuku = new ArrayList<>();
13         this.daftarPeminjaman = new ArrayList<>();
14     }
15
16     public void tambahAnggota(Anggota a) {
17         daftarAnggota.add(a);
18         System.out.println("Anggota " + a.getNama() + " berhasil ditambahkan.");
19     }
20
21     public void tambahBuku(Buku b) {
22         daftarBuku.add(b);
23         System.out.println("Buku " + b.getJudul() + " berhasil ditambahkan.");
24     }
25
26     public void pinjamBuku(Anggota a, Buku b) {
27         Peminjaman p = new Peminjaman(a, b);
28         daftarPeminjaman.add(p);
29         System.out.println("Buku berhasil dipinjam: " + p.info());
30     }
31
32     public String info() {
33         String info = "Perpustakaan: " + nama + "\n";
34         info += "Total Anggota: " + daftarAnggota.size() + "\n";
35         info += "Total Buku: " + daftarBuku.size() + "\n";
36         info += "Total Peminjaman: " + daftarPeminjaman.size() + "\n";
37         return info;
38     }
39 }
```

Main

```
1 public class Main {
2     public static void main(String[] args) {
3         // Buat objek perpustakaan
4         Perpustakaan perpustakaanKu = new Perpustakaan("Perpustakaan Kota");
5
6         // Tambahkan anggota
7         Anggota a1 = new Anggota("001", "Budi Santoso");
8         Anggota a2 = new Anggota("002", "Siti Aminah");
9         perpustakaanKu.tambahAnggota(a1);
10        perpustakaanKu.tambahAnggota(a2);
11
12        // Tambahkan buku
13        Buku b1 = new Buku("B001", "Dasar-Dasar Pemrograman");
14        Buku b2 = new Buku("B002", "Struktur Data");
15        perpustakaanKu.tambahBuku(b1);
16        perpustakaanKu.tambahBuku(b2);
17
18        // Lakukan peminjaman
19        perpustakaanKu.pinjamBuku(a1, b1);
20        perpustakaanKu.pinjamBuku(a2, b2);
21
22        // Tampilkan informasi perpustakaan
23        System.out.println("\n--- Informasi Perpustakaan ---");
24        System.out.println(perpustakaanKu.info());
25    }
26 }
```

Output

```
Anggota Siti Aminah berhasil ditambahkan.
Buku Dasar-Dasar Pemrograman berhasil ditambahkan.
Buku Struktur Data berhasil ditambahkan.
Buku berhasil dipinjam: Peminjaman oleh Budi Santoso dengan buku Dasar-Dasar Pemrograman pada tanggal 2025-09-17
Buku berhasil dipinjam: Peminjaman oleh Siti Aminah dengan buku Struktur Data pada tanggal 2025-09-17

--- Informasi Perpustakaan ---
Perpustakaan: Perpustakaan Kota
Total Anggota: 2
Total Buku: 2
Total Peminjaman: 2
```

PS C:\Users\ILHAM DHARMA A\OneDrive\Documents\SEMESTER 3\PBO Pak Irsyad\PRAKTIKUM\minggu 4>

