

MODUL 11 – Komposisi dan Agregasi

11.1. CAPAIAN PEMBELAJARAN

- 1 Mahasiswa mampu memahami konsep komposisi dan agregasi dalam PBO.
- 2 Mahasiswa Mampu menerapkan komposisi dan agregasi untuk membangun hubungan antar objek di program Java.
- 3 Mahasiswa mampu membuat dan mengelola objek yang saling bergantung dan tidak bergantung satu sama lain.

11.2. ALAT DAN BAHAN

- 1 Seperangkat komputer lengkap/Laptop dengan koneksi internet
- 2 Web Browser (Chrome/Firefox/Opera/Edge/Safari/dll)
- 3 Aplikasi Kantor (Microsoft Office/Libre Office/WPS Office/etc)
- 4 JDK (<https://www.oracle.com/java/technologies/downloads/>)
- 5 Netbeans (<https://netbeans.apache.org/front/main/download/>)

11.3. DASAR TEORI

11.3.1. Komposisi dan Agregasi

Dalam PBO, sering kali objek-objek yang dibuat tidak berdiri sendiri, tetapi memiliki hubungan dengan objek lain. Hubungan ini dapat dikelompokkan dalam berbagai jenis, salah satunya adalah hubungan has-a (memiliki), di mana objek "induk" memiliki objek "bagian". Hubungan has-a dibagi menjadi dua bentuk utama, yaitu Komposisi dan Agregasi.

11.3.2. Komposisi

Komposisi adalah hubungan yang sangat kuat di mana satu objek (induk) terdiri dari objek lain (bagian) sebagai komponen yang tidak terpisahkan. Pada hubungan ini jika objek induk dihancurkan, objek bagian juga akan dihancurkan, sehingga objek bagian tidak memiliki kehidupan terpisah di luar objek induk; objek ini tidak akan ada tanpa objek induknya.

Komposisi memiliki hubungan yang kuat atau tertutup (strong association), di mana objek bagian menjadi komponen integral dari objek induknya. Kehidupan objek bagian sepenuhnya bergantung pada keberadaan objek induk. Jika objek induk dihancurkan atau dihapus, maka objek bagian tersebut akan ikut terhapus secara otomatis. Contoh di dunia nyata adalah hubungan antara mobil dan mesin; mesin tidak dapat berdiri sendiri tanpa mobil yang menjadi wadahnya, sehingga kehidupan mesin bergantung sepenuhnya pada keberadaan mobil tersebut.

Komposisi digunakan ketika suatu objek bagian adalah bagian integral dari objek induknya dan tidak masuk akal untuk eksis tanpa objek induk tersebut. Dalam desain perangkat lunak, komposisi membantu menjaga integritas data dan relasi antara objek.

11.3.3. Agregasi

Agregasi adalah hubungan yang lebih lemah dibandingkan komposisi. Hubungan ini menunjukkan bahwa suatu objek "induk" memiliki objek "bagian" tanpa merangkul penuh kehidupannya. Dengan kata lain, objek bagian bisa eksis sendiri, terlepas dari keberadaan objek induk.

Agregasi memiliki hubungan yang longgar atau kurang erat (loose association), di mana objek induk dan objek bagian terhubung secara tidak terlalu erat. Kehidupan objek bagian dalam agregasi tidak tergantung pada keberadaan objek induk; objek bagian dapat tetap eksis secara mandiri meskipun objek induk dihancurkan. Contoh di dunia nyata adalah hubungan antara kelas dan mahasiswa; mahasiswa tetap dapat eksis sebagai entitas independen meskipun kelas tertentu yang diambilnya telah berakhir atau tidak lagi ada.

Agregasi digunakan ketika objek bagian adalah objek independen yang hanya terkait sementara atau longgar dengan objek induk. Ini memungkinkan objek yang sama digunakan dalam konteks atau hubungan yang berbeda.

11.3.4. Perbandingan komposisi dan agregasi

Tabel 1. Perbandingan Komposisi dengan Agregasi

Aspek	Komposisi	Agregasi
Hubungan	Hubungan kuat; bagian adalah bagian esensial	Hubungan longgar; bagian adalah objek independen
Kehidupan Objek	Objek bagian tidak bisa eksis tanpa objek induk	Objek bagian bisa eksis tanpa objek induk
Contoh	Mobil dan Mesin	Kelas dan Mahasiswa

11.4. PRAKTIKUM

10.4.1. Perisapan

- 1) Buka Netbeans yang sudah terinstall pada komputer
- 2) Buat proyek baru dengan nama PraktikumPBO_11.
- 3) Buat package baru dengan nama praktikum11.

10.4.2. Studi Kasus Komposisi (Sistem Perpustakaan)

- Deskripsi

Perpustakaan memiliki beberapa buku. Jika perpustakaan dihapus, maka semua buku di dalamnya juga ikut dihapus.

- Kelas

- Perpustakaan: Memiliki daftar buku (komposisi).
- Buku: Berisi detail buku seperti judul dan ISBN.

```
class Buku {
    private String judul;

    public Buku(String judul) {
        this.judul = judul;
    }

    public void infoBuku() {
        System.out.println("Judul Buku: " + judul);
    }
}

class Perpustakaan {
    private List<Buku> bukuList;

    public Perpustakaan() {
        this.bukuList = new ArrayList<>();
    }

    public void tambahBuku(Buku buku) {
        bukuList.add(buku);
    }

    public void infoPerpustakaan() {
        for (Buku buku : bukuList) {
            buku.infoBuku();
        }
    }
}
```

10.4.3. Studi Kasus Agregasi (Sistem Klub dan Anggota)

- Deskripsi

Klub memiliki anggota, tetapi anggota tetap ada meskipun mereka keluar dari klub.

- Kelas

- Klub: Memiliki daftar anggota (agregasi).

- Anggota: Menyimpan nama dan data lainnya.

```
class Anggota {
    private String nama;

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

    public void infoAnggota() {
        System.out.println("Nama Anggota: " + nama);
    }
}

class Klub {
    private String namaKlub;
    private List<Anggota> anggotaList;

    public Klub(String namaKlub) {
        this.namaKlub = namaKlub;
        this.anggotaList = new ArrayList<>();
    }

    public void tambahAnggota(Anggota anggota) {
        anggotaList.add(anggota);
    }

    public void infoKlub() {
        System.out.println("Nama Klub: " + namaKlub);
        for (Anggota anggota : anggotaList) {
            anggota.infoAnggota();
        }
    }
}
```

11.5. TUGAS MODUL 11

10.5.1. Soal

Anda diminta membuat program untuk sistem perpustakaan yang memodelkan hubungan antara Perpustakaan, Buku, dan Pengarang. Komposisi →
Perpustakaan memiliki Buku. Jika Perpustakaan dihapus, maka Buku juga akan

dihapus. Agregasi → Buku memiliki Pengarang. Pengarang bisa eksis tanpa Buku tertentu.

Instruksi :

- Buat kelas Buku dengan atribut judul dan metode infoBuku().
- Buat kelas Perpustakaan yang memiliki atribut daftar buku (gunakan array) dan metode untuk menambahkan buku. Tampilkan informasi semua buku di perpustakaan.
- Buat kelas Pengarang dengan atribut namaPengarang dan metode infoPengarang().
- Hubungkan Buku dan Pengarang menggunakan agregasi.

Petunjuk Pengerjaan

a) Laporan:

- Buatlah laporan akhir sesuai dengan praktikum yang dilakukan
- Tuliskan laporan ke dalam akun masing-masing web medium
- Publikasikan laporan tersebut
- Kirimkan tautan laporan ke elita.
- **Batas Pengumpulan:** Sebelum Pertemuan Praktik Ke 12.