



Mata Kuliah : Teori Pemrograman Berorientasi Objek  
Program Studi : D4 – Teknik Informatika  
Semester : 3

---

Kelas : 2A  
NIM : 244107020030  
Nama : Athaulla Hafizh  
Pertemuan Ke- : 5

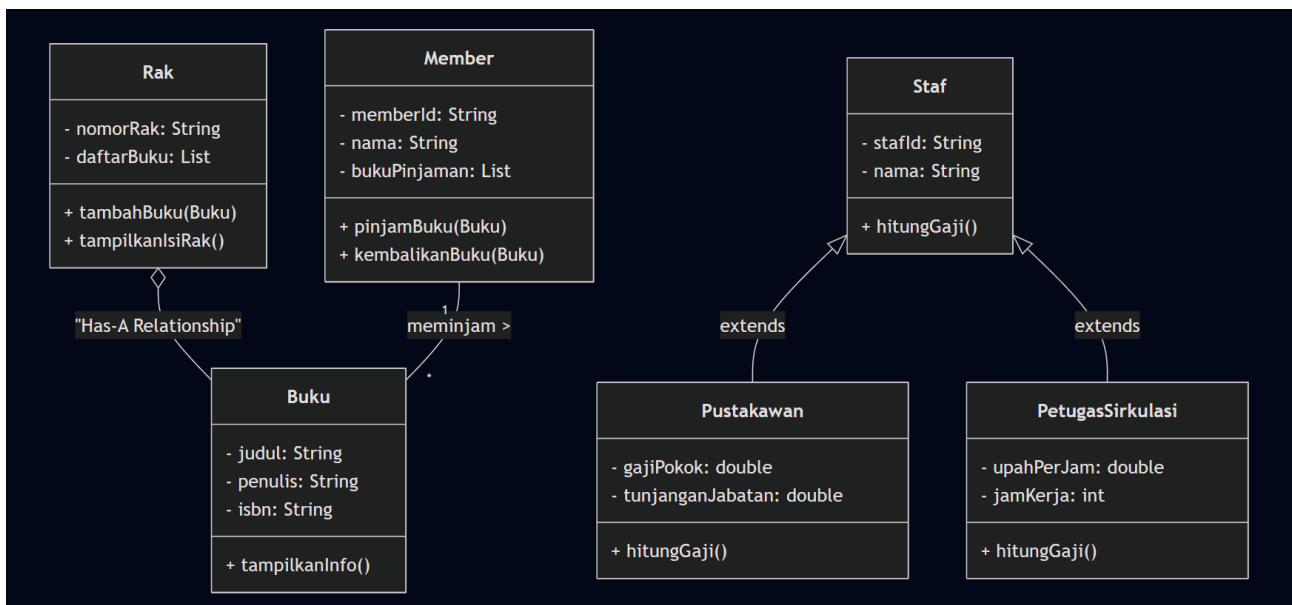
### Laporan Jobsheet

	Jawaban / Deskripsi
Studi Kasus	<p><b>Sistem Manajemen Perpustakaan</b></p> <p>Sebuah perpustakaan universitas ingin membuat sistem sederhana untuk mengelola koleksi buku dan aktivitas peminjaman. Sistem ini perlu mendata setiap buku yang dimiliki, anggota yang terdaftar, dan staf perpustakaan yang bertugas.</p> <p>Setiap buku memiliki judul, penulis, dan kode ISBN. Perpustakaan memiliki banyak rak buku, dan setiap rak digunakan untuk menyimpan kumpulan buku. Jika sebuah rak dibongkar atau dipindahkan, buku-buku yang ada di dalamnya tidak akan hilang dari data perpustakaan; buku-buku tersebut hanya perlu ditempatkan di rak lain.</p> <p>Anggota perpustakaan (Member) dapat meminjam buku. Setiap anggota memiliki ID unik, nama, dan dapat meminjam beberapa buku sekaligus. Di sisi lain, ada staf perpustakaan (Staf) yang bertugas mengelola sirkulasi buku. Staf juga memiliki ID dan nama. Ada dua jenis staf: Pustakawan (Librarian) yang bertanggung jawab atas pengelolaan koleksi dan Petugas Sirkulasi (Clerk) yang menangani peminjaman dan pengembalian. Keduanya memiliki cara perhitungan gaji yang berbeda.</p>



Agregasi	Hubungan ini terjadi antara kelas Rak dan Buku. Sebuah rak "memiliki" beberapa buku (has-a relationship). Namun, keberadaan buku tidak bergantung pada rak. Jika objek Rak dihapus, objek Buku yang terkait dengannya akan tetap ada dalam sistem.
Relasi/ Asosiasi	Relasi ini terjadi antara Member dan Buku. Seorang anggota dapat meminjam banyak buku, dan satu buku dapat dipinjam oleh satu anggota pada satu waktu. Ini adalah hubungan "uses-a" di mana objek Member berinteraksi dengan objek Buku tanpa kepemilikan yang kuat. Hubungan serupa juga terjadi antara Staf dan Member, di mana staf melayani anggota.
Inheritance	Pewarisan (inheritance) diterapkan pada kelas Staf sebagai <i>parent class</i> . Kelas Pustakawan dan PetugasSirkulasi adalah <i>subclass</i> atau <i>child class</i> yang mewarisi properti dan metode dari Staf (seperti nama dan ID) tetapi memiliki implementasi hitungGaji() yang berbeda sesuai dengan perannya masing-masing.

## Class Diagram





### Implementasi Kode

class	kode
Buku	<pre>public class Buku {     private String judul;     private String penulis;     private String isbn;      Buku(String judul, String penulis, String isbn) {         this.judul = judul;         this.penulis = penulis;         this.isbn = isbn;     }      String getJudul() {         return this.judul;     }      public void tampilkanInfo() {         System.out.println("Judul    : " + this.judul);         System.out.println("Penulis : " + this.penulis);         System.out.println("ISBN    : " + this.isbn);         System.out.println("-----");     } }</pre>
Rak	<pre>import java.util.ArrayList; import java.util.List;  public class Rak {     private String nomorRak;     private List&lt;Buku&gt; daftarBuku;      Rak(String nomorRak) {         this.nomorRak = nomorRak;         this.daftarBuku = new ArrayList&lt;&gt;();     } }</pre>



	<pre>}  public void tambahBuku(Buku buku) {     this.daftarBuku.add(buku);      System.out.println("Buku " + buku.getJudul() + " berhasil ditambahkan ke rak " + this.nomorRak); }  public void tampilkanIsiRak() {     System.out.println("\n===== Daftar Buku di Rak " + this.nomorRak + " =====");     for (Buku buku : this.daftarBuku) {         System.out.println("- " + buku.getJudul());     }     System.out.println("=====\n"); } }</pre>
Staf (Parent Class)	<pre>public class Staf {     private String stafId;     private String nama;      Staf(String stafId, String nama) {         this.stafId = stafId;         this.nama = nama;     }      String getNama() {         return this.nama;     }      public void melayani(Member member) {         System.out.println("Staf " + this.nama + " sedang melayani member " + member.getNama());     } }</pre>



	<pre>public void hitungGaji() {     System.out.println("Menghitung gaji untuk " + this.nama + "..."); } }</pre>
<b>Pustakawan</b> <b>(Subclass)</b>	<pre>public class Pustakawan extends Staf {     private double gajiPokok;     private double tunjanganJabatan;      Pustakawan(String stafId, String nama, double gajiPokok, double tunjanganJabatan) {         super(stafId, nama);         this.gajiPokok = gajiPokok;         this.tunjanganJabatan = tunjanganJabatan;     }      @Override     public void hitungGaji() {         super.hitungGaji();         double totalGaji = this.gajiPokok + this.tunjanganJabatan;         System.out.println("Total Gaji: Rp. " + totalGaji);     } }</pre>
<b>PetugasSirkulasi</b> <b>(Subclass)</b>	<pre>public class PetugasSirkulasi extends Staf {     private double upahPerJam;     private int jamKerja;      PetugasSirkulasi(String stafId, String nama, double upahPerJam, int jamKerja) {         super(stafId, nama);         this.upahPerJam = upahPerJam;         this.jamKerja = jamKerja;     }      @Override</pre>



	<pre>public void hitungGaji() {     super.hitungGaji();     double totalGaji = this.upahPerJam * this.jamKerja;     System.out.println("Total Gaji: Rp. " + totalGaji); } }</pre>
Member	<pre>import java.util.ArrayList; import java.util.List;  public class Member {     private String memberId;     private String nama;     private List&lt;Buku&gt; bukuPinjaman;      Member(String memberId, String nama) {         this.memberId = memberId;         this.nama = nama;         this.bukuPinjaman = new ArrayList&lt;&gt;();     }      String getNama(){         return this.nama;     }      public void pinjamBuku(Buku buku) {         this.bukuPinjaman.add(buku);         System.out.println("Member '" + this.nama + "' berhasil meminjam buku '" + buku.getJudul() + "'.");     }      public void tampilkanBukuPinjaman() {         System.out.println("\n===== Buku yang Dipinjam oleh " + this.nama + " =====");         for (Buku buku : this.bukuPinjaman) {</pre>



	<pre>        System.out.println("- " + buku.getJudul());          }          System.out.println("=====\n");          }      }</pre>
<b>Main Class:</b> <b>LibraryDemo</b> <b>o</b>	<pre>public class LibraryDemo {     public static void main(String[] args) {         // Membuat objek Buku         Buku buku1 = new Buku("Pemrograman Berorientasi Objek", "Dr. Sutrisno", "978-602-0-1");         Buku buku2 = new Buku("Dasar-Dasar Jaringan Komputer", "Ahmad Fauzi", "978-602-0-2");          // Menampilkan info buku         System.out.println("==== Informasi Buku Awal =====");         buku1.tampilkanInfo();          // Agregasi: Menambahkan buku ke rak         Rak rakA1 = new Rak("A1-Computer");         rakA1.tambahBuku(buku1);         rakA1.tambahBuku(buku2);         rakA1.tampilkanIsiRak();          // Relasi/Asosiasi: Member meminjam buku         Member member1 = new Member("M001", "Budi");         member1.pinjamBuku(buku1);         member1.tampilkanBukuPinjaman();          // Inheritance: Membuat objek dari subclass Staf         Pustakawan pustakawan1 = new Pustakawan("S001", "Ani", 5000000, 1500000);         PetugasSirkulasi petugas1 = new PetugasSirkulasi("S002", "Charlie", 50000, 40);          // Polimorfisme dalam aksi</pre>



	<pre>System.out.println("==== Perhitungan Gaji Staf ====");  pustakawan1.hitungGaji();  System.out.println();  petugas1.hitungGaji();  // Relasi Staf dan Member  System.out.println();  pustakawan1.melayani(member1);  }  }</pre>
Output	<pre>==== Informasi Buku Awal ==== Judul   : Pemrograman Berorientasi Objek Penulis : Dr. Sutrisno ISBN    : 978-602-0-1 ----- Buku 'Pemrograman Berorientasi Objek' berhasil ditambahkan ke rak A1-Computer Buku 'Dasar-Dasar Jaringan Komputer' berhasil ditambahkan ke rak A1-Computer  ==== Daftar Buku di Rak A1-Computer ==== - Pemrograman Berorientasi Objek - Dasar-Dasar Jaringan Komputer =====  Member 'Budi' berhasil meminjam buku 'Pemrograman Berorientasi Objek'.  ==== Buku yang Dipinjam oleh Budi ==== - Pemrograman Berorientasi Objek =====  ==== Perhitungan Gaji Staf ==== Menghitung gaji untuk Ani... Total Gaji: Rp. 6500000.0  Menghitung gaji untuk Charlie... Total Gaji: Rp. 2000000.0  Staf Ani sedang melayani member Budi</pre>