

Soal

Kelas A

## **Sistem Manajemen Perpustakaan Mini**

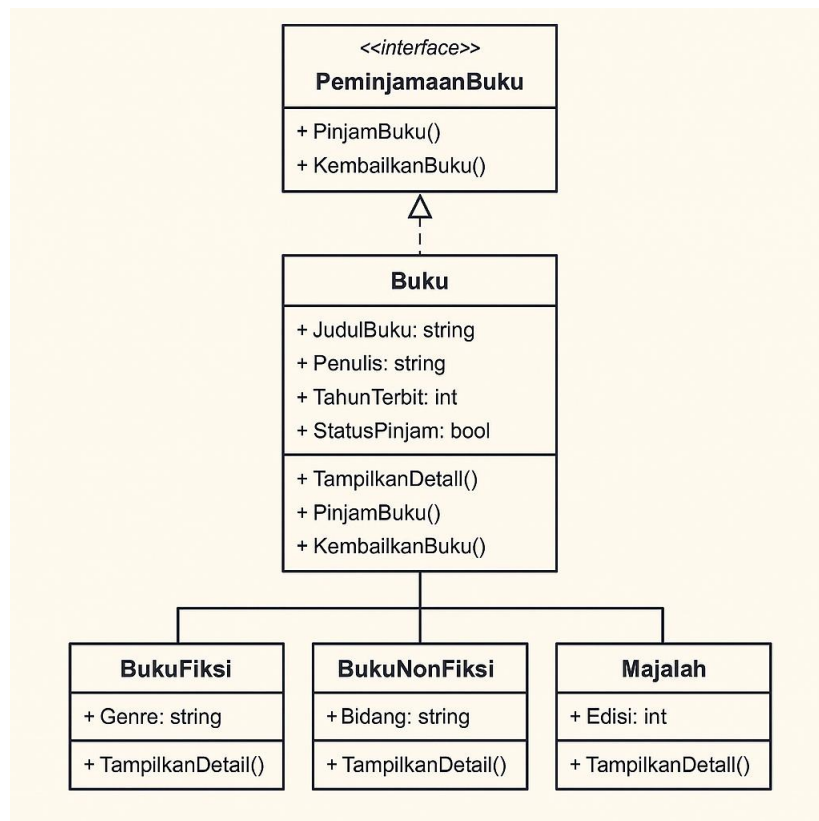
Sebuah perpustakaan kecil ingin membuat sistem manajemen sederhana menggunakan aplikasi console. Dalam sistem ini, terdapat berbagai jenis buku seperti Buku Fiksi, Buku Non-Fiksi, dan Majalah. Setiap buku memiliki informasi dasar seperti judul, penulis, dan tahun terbit.

Petugas ingin dapat melakukan hal-hal berikut:

- Menambahkan buku ke dalam daftar perpustakaan.
- Mengubah data buku
- Melihat informasi setiap buku secara umum.
- Menyimpan data buku dalam daftar menggunakan
- Meminjam buku (maksimal 3 buku).
- Mengembalikan buku.
- Melihat daftar buku yang sedang dipinjam.

Syarat : Program menerapkan 4 pilar OOP yaitu inheritance, enkapsulasi, polymorphism, abstraction dan interface.

## CLASS DIAGRAM



## CODE PROGRAM

```
using System;  
  
using System.Collections.Generic;  
  
public interface PeminjamanBuku  
{  
    void PinjamBuku();  
    void KembalikanBuku();  
}
```

## PENJELASAN CODE PROGRAM :

Dalam sistem manajemen perpustakaan ini, terdapat sebuah **antarmuka (interface)** bernama PeminjamanBuku yang dirancang untuk memastikan bahwa setiap objek buku yang dikelola oleh sistem memiliki kemampuan dasar untuk dipinjam dan dikembalikan. Antarmuka ini mendefinisikan dua metode penting, yaitu **PinjamBuku()** dan **KembalikanBuku()**,

**void PinjamBuku()** → Mengubah status buku menjadi dipinjam.

**void KembalikanBuku()** → Mengembalikan status buku ke tersedia.

Antarmuka ini sangat penting karena menyederhanakan manajemen koleksi buku. Sistem tidak perlu tahu secara detail jenis buku apa yang sedang diproses, selama objek tersebut mengimplementasikan PeminjamanBuku, maka sistem bisa memanggil metode **PinjamBuku()** dan **KembalikanBuku()** dengan cara yang konsisten.

```
public abstract class Buku : PeminjamanBuku  
{  
    public string JudulBuku { get; set; }  
    public string Penulis { get; set; }  
    public int TahunTerbit { get; set; }  
    public bool StatusPinjam { get; protected set; }  
}
```

```

protected Buku(string judul, string pencipta, int tahun)
{
    JudulBuku = judul;
    Penulis = pencipta;
    TahunTerbit = tahun;
    StatusPinjam = false;
}

public abstract void TampilkanDetail();

public void PinjamBuku()
{
    StatusPinjam = true;
}

public void KembalikanBuku()
{
    StatusPinjam = false;
}
}

```

#### PENJELASAN CODE PROGRAM :

kelas **Buku** berfungsi sebagai **kerangka dasar** atau **template umum** untuk semua jenis buku yang tersedia di perpustakaan. Karena tidak digunakan secara langsung untuk membuat objek, kelas ini dideklarasikan sebagai **abstrak**. Artinya, kelas ini tidak lengkap secara mandiri dan harus diturunkan ke kelas lain agar bisa digunakan.

Kelas ini memiliki beberapa **properti penting** yang mendeskripsikan karakteristik umum dari semua buku, yaitu:

- JudulBuku: berisi judul buku.
- Penulis: menyimpan nama penulis atau penerbit (khusus untuk majalah).
- TahunTerbit: menunjukkan tahun kapan buku diterbitkan.
- StatusPinjam: sebuah nilai boolean yang menunjukkan apakah buku sedang dipinjam (true) atau sedang tersedia (false).

Ketika sebuah objek dari kelas turunan dibuat (seperti BukuFiksi, BukuNonFiksi, atau Majalah), konstruktor dari kelas Buku akan menginisialisasi properti-properti tersebut, dan secara otomatis mengatur StatusPinjam menjadi false. Ini berarti, setiap buku yang baru dimasukkan ke sistem dianggap dalam keadaan **tersedia** dan belum dipinjam.

Selain properti, kelas ini juga menyertakan dua **metode konkret**, yaitu:

- PinjamBuku(): yang akan mengubah nilai StatusPinjam menjadi true, artinya buku telah dipinjam.
- KembalikanBuku(): yang akan mengubah nilai StatusPinjam kembali menjadi false, artinya buku telah dikembalikan dan siap dipinjam lagi.

```
public class BukuFiksi : Buku
```

```
{
```

```
    public string Genre { get; set; }
```

```
    public BukuFiksi(string judul, string penulis, int tahun, string genre)
```

```
        : base(judul, penulis, tahun)
```

```
{
```

```
    Genre = genre;
```

```
}
```

```
    public override void TampilkanDetail()
```

```
{
```

```

        Console.WriteLine($"Novel {Genre}: {JudulBuku}\n" +
            $"Penulis: {Penulis}, Tahun: {TahunTerbit}\n" +
            $"Status: {(StatusPinjam ? "Dipinjam" : "Tersedia")}\n");
    }
}

public class BukuNonFiksi : Buku
{
    public string Bidang { get; set; }

    public BukuNonFiksi(string judul, string penulis, int tahun, string bidang)
        : base(judul, penulis, tahun)
    {
        Bidang = bidang;
    }

    public override void TampilkanDetail()
    {
        Console.WriteLine($"Buku {Bidang}: {JudulBuku}\n" +
            $"Penulis: {Penulis}, Tahun: {TahunTerbit}\n" +
            $"Status: {(StatusPinjam ? "Dipinjam" : "Tersedia")}\n");
    }
}

public class Majalah : Buku
{
    public int Edisi { get; set; }
}

```

```

public Majalah (string judul, string penerbit, int tahun, int edisi)
    : base(judul, penerbit, tahun)
{
    Edisi = edisi;
}

public override void TampilkanDetail()
{
    Console.WriteLine($"Terbitan #{Edisi}: {JudulBuku}\n" +
        $"Penerbit: {Penulis}, Tahun: {TahunTerbit}\n" +
        $"Status: {(StatusPinjam ? "Dipinjam" : "Tersedia")}\n");
}
}

```

#### PENJELASAN CODE PROGRAM :

Pada tiga kelas turunan BukuFiksi, BukuNonFiksi, dan Majalah, mengambil kerangka dasar dari kelas Buku dan menambahkan ciri khas masing-masing sehingga sistem bisa membedakan satu jenis bacaan dengan yang lain tanpa mengorbankan konsistensi

1. **Di kelas BukuFiksi**, kita menambahkan properti Genre untuk mencatat spektrum cerita yang diusung—mulai dari fantasi yang memikat imajinasi hingga roman yang menyentuh hati.
2. **Di BukuNonFiksi** memfokuskan diri pada pengetahuan dan fakta, sehingga ia memperkenalkan properti Bidang bisa berupa “Sains” untuk penjelasan ilmiah, “Sejarah” untuk narasi masa lalu, atau topik lain sesuai konteks.
3. **Di kelas Majalah** memodelkan publikasi periodik dengan properti Edisi, sebuah nomor yang menandai keluaran ke-berapa dalam rangkaian terbitan. Dalam TampilkanDetail(), objek majalah menampilkan judul dengan awalan “Terbitan [Edisi]”, kemudian baris berikutnya menulis nama penerbit (menggunakan properti Penulis dari kelas dasar) dan tahun keluar, serta status pinjam yang bergeser sesuai interaksi peminjaman.

```
public class Perpustakaan
{
    private List<Buku> inventaris = new List<Buku>();
    private List<Buku> dalamPinjaman = new List<Buku>();

    public void TambahKeInventaris(Buku item)
    {
        inventaris.Add(item);

        Console.WriteLine($"{item.JudulBuku}' berhasil ditambahkan ke rak buku");
    }

    public void PerbaruiStatus(string pencarian, string JudulBaru, string PenulisBaru, int
    TahunTerbitTerbaru)
    {
        var target = inventaris.Find(i => i.JudulBuku.Equals(pencarian,
        StringComparison.OrdinalIgnoreCase));

        if (target != null)
        {
            target.JudulBuku = JudulBaru;

            target.Penulis = PenulisBaru;

            target.TahunTerbit = TahunTerbitTerbaru;

            Console.WriteLine($"{pencarian}' sudah diperbarui");
        }
        else
        {
            Console.WriteLine("maaf bang buku tidak ditemukan");
        }
    }
}
```



```
}
```

```
public void TampilkanInventaris()
```

```
{
```

```
    Console.WriteLine("\n=== DAFTAR KOLEKSI PERPUSTAKAAN ===");
```

```
    foreach (var item in inventaris)
```

```
    {
```

```
        item.TampilkanDetail();
```

```
    }
```

```
}
```

```
public void ProsesPinjam(string judul)
```

```
{
```

```
    if (dalamPinjaman.Count >= 3)
```

```
    {
```

```
        Console.WriteLine("\nmaksimal boleh meminjam 3 kak");
```

```
        return;
```

```
    }
```

```
    var item = inventaris.Find(i => i.JudulBuku.Equals(judul,  
StringComparison.OrdinalIgnoreCase));
```

```
    if (item == null)
```

```
    {
```

```
        Console.WriteLine("\nBuku Tidak Tersedia");
```

```
        return;
```

```
    }
```

```
        if (item.StatusPinjam)
        {
            Console.WriteLine("\n Buku Lagi dipinjam");

            return;
        }

        item.PinjamBuku();

        dalamPinjaman.Add(item);

        Console.WriteLine($" \n Buku '{judul}' Berhasil Dipinjam");
    }

    public void ProsesPengembalian(string judul)
    {
        var item = dalamPinjaman.Find(i => i.JudulBuku.Equals(judul,
StringComparison.OrdinalIgnoreCase));

        if (item != null)
        {
            item.KembalikanBuku();

            dalamPinjaman.Remove(item);

            Console.WriteLine($" \n Buku '{judul}' sudah dikembalikan");
        }
        else
        {
            Console.WriteLine("\nBuku Tidak Ada Di Daftar Pinjaman");
        }
    }
}
```

```

public void TampilkanPinjaman()
{
    Console.WriteLine("\n=== DAFTAR BUKU PINJAMAN ===");

    foreach (var item in dalamPinjaman)
    {
        item.TampilkanDetail();
    }
}
}

```

#### PENJELASAN CODE PROGRAM :

Dalam arsitektur sistem perpustakaan ini, **kelas Perpustakaan** berperan sebagai otak pengelola koleksi dan transaksi peminjaman. Kelas ini memelihara dua kumpulan data utama: satu daftar lengkap bernama inventaris, yang menyimpan semua objek buku yang pernah ditambahkan ke dalam sistem; dan sebuah daftar terpisah bernama `dalamPinjaman`, yang secara dinamis mencatat buku-buku mana saja yang sedang dipinjam oleh pengguna.

Saat seorang pustakawan atau pengguna menambahkan buku baru, mereka akan memanggil metode `TambahKeInventaris()`. Metode ini tidak hanya menempatkan objek buku ke dalam daftar inventaris, tetapi juga secara otomatis mencetak umpan balik ke layar—memberi tahu bahwa buku tersebut telah berhasil ditempatkan di rak digital.

Ada kalanya diperlukan pembaruan informasi buku—misalnya judul yang salah ketik atau edisi baru yang perlu dicatat. Di sinilah `PerbaruiStatus()` berfungsi. Metode ini mencari buku dalam inventaris berdasarkan judul yang diberikan, dan jika menemukan kecocokan, ia menggantikan judul, nama penulis, dan tahun terbit lama dengan data baru yang disediakan.

Untuk menampilkan kondisi terkini koleksi, kelas `Perpustakaan` menyediakan `TampilkanInventaris()`. Daripada menuliskan detail kode pemformatan berulang-ulang, metode ini cukup berjalan melalui setiap objek dalam inventaris dan memanggil `TampilkanDetail()`

```

class Program
{
    static void Main(string[] args)
    {
        Perpustakaan perpustakaan = new Perpustakaan();

        bool aktif = true;

        while (aktif)
        {
            Console.WriteLine("\n===== SISTEM MANAJEMEN PERPUSTAKAAN
=====");

            Console.WriteLine("1. Tampilkan Semua Buku");
            Console.WriteLine("2. Tambah Koleksi Buku Baru");
            Console.WriteLine("3. Pinjam Buku");
            Console.WriteLine("4. Kembalikan Buku");
            Console.WriteLine("5. Lihat Buku Yang Sedang Dipinjam");
            Console.WriteLine("6. Keluar");

            Console.Write("Pilih menu: ");

            switch (Console.ReadLine())
            {
                case "1":
                    perpustakaan.TampilkanInventaris();

                    break;

                case "2":
                    TambahKoleksiMenu(perpustakaan);

```

```
        break;

    case "3":
        ProsesPinjamMenu(perpustakaan);
        break;

    case "4":
        ProsesKembaliMenu(perpustakaan);
        break;

    case "5":
        perpustakaan.TampilkanPinjaman();
        break;

    case "6":
        aktif = false;
        Console.WriteLine("\nTerimakasih Ya Kembali Lagi Kapan Kapan");
        break;

    default:
        Console.WriteLine("\nSalah Input Min");
        break;
    }
}

}
```

```
static void TambahKoleksiMenu(Perpustakaan sys)
```

```
{  
    Console.WriteLine("\nJenis Buku: ");  
    Console.WriteLine("1. Buku Fiksi ");  
    Console.WriteLine("2. Buku Non Fiksi ");  
    Console.WriteLine("3. Majalah ");  
    Console.Write("Pilih jenis: ");  
  
    var jenis = Console.ReadLine();  
    Console.Write("Judul Buku : ");  
    var judul = Console.ReadLine();  
    Console.Write("Penulis: ");  
    var pencipta = Console.ReadLine();  
    Console.Write("Tahun Terbit : ");  
    int tahun = int.Parse(Console.ReadLine());  
  
    switch (jenis)  
    {  
        case "1":  
            Console.Write("Genre : ");  
            sys.TambahKeInventaris(new BukuFiksi(judul, pencipta, tahun,  
Console.ReadLine()));  
            break;  
  
        case "2":  
            Console.Write("Bidang : ");  
            sys.TambahKeInventaris(new BukuNonFiksi(judul, pencipta, tahun,  
Console.ReadLine()));  
        }  
    }
```

```

        break;

    case "3":

        Console.Write("Edisi: ");

        sys.TambahKeInventaris(new Majalah(judul, pencipta, tahun,
int.Parse(Console.ReadLine())));

        break;

    default:

        Console.WriteLine("\nJenis Buku Yang Dipilih Tidak Ada");

        break;

    }

}

static void ProsesPinjamMenu(Perpustakaan sys)
{
    Console.Write("\nMasukkan judul buku yang ingin dipinjam: ");

    sys.ProsesPinjam(Console.ReadLine());
}

static void ProsesKembaliMenu(Perpustakaan sys)
{
    Console.Write("\nMasukkan judul buku yang akan dikembalikan: ");

    sys.ProsesPengembalian(Console.ReadLine());
}

```

## PENJELASAN CODE PROGRAM :

Kelas **Program** bertindak sebagai pintu masuk aplikasi dan mengatur semua interaksi dengan pengguna melalui antarmuka konsol. Begitu dijalankan, Main() memulai sebuah loop yang terus-menerus menampilkan menu pilihan:

### 1. Tampilkan Semua Buku

Memanggil TampilkanInventaris(), lalu sistem menelusuri setiap buku di koleksi dan mencetak detailnya sesuai implementasi TampilkanDetail().

### 2. Tambah Koleksi Buku Baru

Memanggil TambahKoleksiMenu(), yang pertama-tama meminta pengguna memilih jenis item—buku fiksi, nonfiksi, atau majalah. Selanjutnya pengguna memasukkan judul, penulis, tahun terbit, dan atribut spesifik (genre, bidang, atau edisi). Setelah itu, objek buku baru dibuat dan ditambahkan ke inventaris.

### 3. Pinjam Buku

Memanggil ProsesPinjamMenu(), meminta judul buku kepada pengguna, kemudian ProsesPinjam() akan:

- Memeriksa apakah batas tiga buku sudah terlampaui.
- Mengecek ketersediaan buku di inventaris.
- Memperbarui status buku menjadi “dipinjam” dan menambahkannya ke daftar peminjaman jika semua syarat terpenuhi.

### 4. Kembalikan Buku

Memanggil ProsesKembaliMenu(), meminta judul buku yang dikembalikan, lalu ProsesPengembalian() menandai buku tersebut kembali tersedia dan mengeluarkannya dari daftar peminjaman.

### 5. Lihat Buku Yang Sedang Dipinjam

Memanggil TampilkanPinjaman(), yang mencetak semua buku dengan status “dipinjam” beserta detailnya.

### 6. Keluar

Mengakhiri loop menu dan menutup aplikasi dengan pesan perpisahan.