

## Contoh Penggeraan

### Tema Besar UTS: "Sistem Manajemen Perpustakaan Digital"

Membangun sistem berbasis OOP yang mensimulasikan operasi dasar sebuah perpustakaan digital, dengan penerapan prinsip class & object, enkapsulasi, inheritance, polymorphism, modularisasi, serta penggunaan pustaka standar Python.

Setiap proyek merupakan tahap pengembangan dari tahap sebelumnya.

### Project 1 — Pembuatan Struktur Dasar OOP dan Enkapsulasi (Modul 1–3)

Membuat modul utama dengan nama perpustakaan.py yang berisi definisi class dasar berikut:

1. Buku
  - a. Atribut privat: \_\_judul, \_\_penulis, \_\_tahun, \_\_tersedia (boolean).
  - b. Method:
    - tampilan\_info() → menampilkan informasi buku.
    - pinjam() → mengubah status buku menjadi tidak tersedia.
    - kembalikan() → mengubah status buku menjadi tersedia.
2. Anggota
  - a. Atribut privat: \_\_nama, \_\_nim, \_\_daftar\_pinjam (list of Buku).
  - b. Method:
    - pinjam\_buku(buku) → menambahkan buku ke daftar pinjam jika tersedia.
    - kembalikan\_buku(buku) → menghapus buku dari daftar pinjam.
    - lihat\_daftar\_pinjam() → menampilkan semua buku yang sedang dipinjam anggota tersebut.

#### Ketentuan Teknis

- Menggunakan konsep enkapsulasi (atribut privat dan getter/setter bila perlu).
- Menggunakan konsep instance dan object dengan minimal 3 objek Buku dan 2 objek Anggota.
- Output memperlihatkan proses peminjaman dan pengembalian buku oleh anggota.

## **Project 2 — Pewarisan dan Polymorphism dalam Sistem Transaksi (Modul 4–5)**

Menambahkan fitur transaksi ke sistem dengan menggunakan inheritance dan polymorphism.

1. Buat class induk Transaksi:
  - a. Atribut: id\_transaksi, tanggal, anggota.
  - b. Method proses\_transaksi() (akan dioverride di class turunan).
2. Buat dua class turunan:
  - a. TransaksiPeminjaman(Transaksi)
    - Override method proses\_transaksi() → menampilkan pesan bahwa anggota meminjam buku.
  - b. TransaksiPengembalian(Transaksi)
    - Override method proses\_transaksi() → menampilkan pesan bahwa anggota mengembalikan buku.
3. Implementasikan polymorphism
  - a. Buat list berisi beberapa objek TransaksiPeminjaman dan TransaksiPengembalian.
  - b. Lakukan loop dan panggil proses\_transaksi() untuk seluruh objek, tunjukkan bahwa method yang sama menghasilkan perilaku berbeda.

### **Ketentuan Teknis:**

- Menggunakan super() untuk mewarisi konstruktor class induk.
- Menggunakan datetime untuk mencatat waktu transaksi.
- Menggunakan Minimal ada 4 transaksi yang ditampilkan di output (2 peminjaman dan 2 pengembalian).

### **Project 3 — Modularisasi dan Integrasi Pustaka Standar (Modul 6–7)**

Refactor seluruh sistem ke dalam struktur proyek yang modular. Menambahkan fungsi analisis data buku menggunakan pustaka standar Python.

#### **Struktur Folder:**

```
perpustakaan_digital/
|
├── main.py
└── models/
    ├── __init__.py
    ├── buku.py
    ├── anggota.py
    └── transaksi.py
|
└── utils/
    ├── __init__.py
    └── laporan.py
```

#### **Tugas:**

1. Pisahkan setiap class (Buku, Anggota, Transaksi, dst.) ke dalam file modulnya masing-masing.
2. Di dalam laporan.py, buat class LaporanPerpustakaan dengan:
  - a. Method buat\_ringkasan()  
→ Menggunakan os dan datetime untuk mencatat waktu pembuatan laporan, dan menulis ringkasan ke file teks (laporan.txt) berisi jumlah buku tersedia, jumlah buku dipinjam, dan total transaksi.
3. Di main.py, tampilkan alur penuh sistem:
  - a. Menampilkan daftar buku & anggota.
  - b. Simulasi peminjaman dan pengembalian.
  - c. Membuat file laporan otomatis menggunakan class LaporanPerpustakaan.

**Ketentuan Teknis:**

- Gunakan blok if `__name__ == "__main__"`: sebagai entry point.
- Gunakan pustaka math, datetime, dan os sesuai konteks program.
- Struktur proyek harus berjalan tanpa error (bisa di-run langsung dari `main.py`).