

## Soal UTS Praktikum PBO Berbasis Python

### Format Ujian:

UTS dilaksanakan dalam bentuk proyek praktikum terstruktur dan berkelanjutan, terdiri dari 3 tahapan (Project 1–3) yang saling berhubungan.

Dikerjakan secara berkelompok. Setiap kelompok bebas menentukan tema proyeknya, namun harus relevan dengan prinsip Pemrograman Berorientasi Objek (OOP).

### Project 1 — Struktur Dasar dan Enkapsulasi

(Bobot: 30%)

#### Tujuan:

Menguji kemampuan mahasiswa membangun class dan object, menerapkan enkapsulasi, serta mengimplementasikan logika dasar program berbasis objek.

#### Instruksi:

1. Buat minimal dua class utama.
2. Gunakan atribut privat (diawali `__`) dan metode getter/setter sesuai kebutuhan.
3. Implementasikan beberapa method publik yang memanipulasi data objek.
4. Buat minimal tiga objek dari class utama untuk mendemonstrasikan fungsionalitasnya.
5. Tampilkan hasil program yang menunjukkan interaksi antar objek (misal: pengguna meminjam barang, pelanggan membeli produk, dll.).

#### Penilaian Utama:

- Penggunaan class & object dengan benar.
- Penerapan enkapsulasi.
- Struktur dan keterbacaan kode.
- Output program sesuai skenario yang logis.

## **Project 2 — Inheritance dan Polymorphism**

**(Bobot: 35%)**

### **Tujuan:**

Mengukur kemampuan mahasiswa dalam membangun relasi antar class menggunakan konsep pewarisan dan polymorphism.

### **Instruksi:**

1. Kembangkan proyek sebelumnya dengan menambahkan class induk dan beberapa class turunan.
  - a. Gunakan konsep inheritance (pewarisan atribut & method).
  - b. Gunakan method overriding untuk menunjukkan polymorphism.
2. Implementasikan fungsi `super()` untuk memanggil konstruktor dari parent class.
3. Gunakan `datetime` untuk mencatat aktivitas atau transaksi yang terjadi.
4. Buat list berisi beberapa objek turunan dan lakukan iterasi yang memanggil method yang sama untuk menunjukkan polymorphism (hasil berbeda untuk tiap class turunan).

### **Penilaian Utama:**

- Penerapan inheritance dan polymorphism yang tepat.
- Penggunaan `super()` secara benar.
- Output yang konsisten dan menunjukkan perilaku berbeda antar class turunan.
- Integrasi logis dengan proyek tahap 1.

### Project 3 - Modularisasi dan Integrasi Pustaka Standar (Bobot: 35%)

#### Tujuan:

Menilai kemampuan mahasiswa dalam menata struktur proyek secara modular serta mengintegrasikan pustaka standar Python.

#### Instruksi:

1. Refactor seluruh proyek ke dalam struktur folder modular seperti berikut: (Contoh)

```
nama_proyek/  
|  
├─ main.py  
├─ models/  
|   ├─ __init__.py  
|   ├─ entitas.py  
|   ├─ pengguna.py  
|   └─ transaksi.py  
└─ utils/  
    ├─ __init__.py  
    └─ laporan.py
```

2. Pisahkan setiap class ke file modulnya masing-masing.
3. Tambahkan class tambahan (misal: Laporan, Analisis, LogAktivitas) yang:
  - a. Menggunakan pustaka datetime untuk mencatat waktu aktivitas.
  - b. Menggunakan pustaka os untuk membaca/menulis file laporan teks.
  - c. (Opsional) Menggunakan pustaka math atau lainnya sesuai kebutuhan proyek.
4. Program utama (main.py) harus:
  - a. Mengimpor seluruh class dari package terkait.
  - b. Menjalankan simulasi lengkap sesuai alur tema yang dipilih kelompok.
  - c. Menghasilkan output berupa file laporan (laporan.txt) berisi ringkasan aktivitas program.

**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).

**Output yang Diharapkan**

- Menunjukkan keterkaitan antar class (relasi objek nyata).
- Menunjukkan praktik OOP: enkapsulasi, inheritance, polymorphism, modularisasi, pustaka standar.
- Setiap anggota kelompok memiliki peran (misal: pengembang class, integrator modul, penguji, dokumentator, dsb.).

**Kriteria Penilaian**

Aspek	Keterangan	Bobot
<b>Struktur &amp; Desain OOP</b>	Implementasi class, atribut, method, dan relasi	25%
<b>Penerapan Inheritance &amp; Polymorphism</b>	Penggunaan pewarisan & overriding method	20%
<b>Modularisasi &amp; Import</b>	Struktur folder & file modular berjalan tanpa error	20%
<b>Integrasi Pustaka Standar</b>	Penggunaan datetime, os, math, dll.	15%
<b>Dokumentasi &amp; Refleksi</b>	Kode bersih, komentar, dan laporan ringkas	10%
<b>Presentasi Proyek</b>	Demonstrasi kelompok dan pembagian peran	10%