Tugas Pemrograman Berorientasi Objek Praktik

Pertemuan ke-1

Nama : Muhammad Zaki As Shidiqi

NPM : 5240411230

|  |
| --- |
| Soal No.1  Jelaskan pengertian Object Oriented Programming (OOP). Sertakan tujuan utama penerapan OOP dalam pengembangan perangkat lunak. |
| Jawaban Soal No.1 |
| Pendekatan untuk membuat program dengan basis objek seperti jalannya dunia nyata. Daripada kita harus mengulang banyak sintaks seperti pendekatan prosedural, kita dapat menggunakan OOP untuk mempersingkat sintaks karena setiap object pada OOP sudah memiliki atribut dan method. |

|  |
| --- |
| Soal No.2  Jelaskan konsep yang menjadi pilar utama dalam OOP yaitu Encapsulation, Inheritance, Polymorphism. |
| Jawaban Soal No.2 |
| 1. Enscapsulation: menyembunyikan detail internal dan hanya mengekspos yang penting saja, seperti panel TV, kita tidak perlu mengetahui isi dari panel tersebut. 2. Inheritance: mewariskan sifat dari class induk(parent class) ke class anak(child class), seperti ikan mewariskan sifat bernafas dengan insang ke anak-anaknya. 3. Polymorphism: perbedaan perilaku dari setiap nama method |

|  |
| --- |
| Soal No.3  Apa perbedaan utama antara ***class*** dan ***object***? Jelaskan dengan contoh nyata dari kehidupan sehari-hari. |
| Jawaban Soal No.3 |
| Perbedaan utama dari class dan objek yaitu, class adalah sebuah cetakan(blueprint), sedangkan object adalah wujud yang dibuat dari class/cetakan tersebut.  Contoh pada kehidupan sehari-harinya seperti resep kue, class sebagai resep untuk membuat sebuah kue dan objek adalah hasil dari resep tersebut atau kue itu sendiri. |

|  |
| --- |
| Soal No.4  Bagaimana konsep ***constructor*** dan ***destructor*** bekerja dalam OOP? Sertakan contoh kode sederhana dalam bahasa python untuk memperjelas jawaban Anda! |
| Jawaban Soal No.4 |
| Constructor adalah method yang digunakan untuk menginisialisai sebuah objek Ketika petama kali dibuat.  Destructor adalah method yang dipanggil secara otomatis sesaat sebelum sebuah objek dihapus dari memori  Program: class KoneksiDatabase:      def \_\_init\_\_(self, nama\_db):          print("--- [INIT] Memulai koneksi... ---")          self.nama\_db = nama\_db          self.terhubung = True          print(f"Koneksi ke database '{self.nama\_db}' berhasil DIBUAT.")      def kirim\_data(self, data):          if self.terhubung:              print(f"Mengirim data '{data}' ke database '{self.nama\_db}'...")          else:              print("Koneksi sudah ditutup!")      def \_\_del\_\_(self):          print("--- [DEL] Membersihkan... ---")          self.terhubung = False          print(f"Koneksi ke database '{self.nama\_db}' telah DITUTUP.")  print("Membuat object koneksi...")  db = KoneksiDatabase("TokoOnlineDB")  print("\nMenggunakan object...")  db.kirim\_data("Sepatu Baru")  print("\nMenghapus object...")  del db |

|  |
| --- |
| Soal No.5  Buatkan tabel perbandingan yang menunjukkan perbedaan paradigma OOP dengan paradigma pemrograman prosedural. Sebutkan minimal tiga perbedaan dan berikan contoh kasus di mana OOP lebih unggul daripada pemrograman prosedural. |
| Jawaban Soal No.5 |
| |  |  |  | | --- | --- | --- | | **Paradigma** | **Prosedural** | **OOP** | | Fokus Utama | Fokus pada urutan instruksi (fungsi/prosedur). | Fokus pada objek, di mana data dan fungsi menjadi satu kesatuan. | | Struktur Data & Fungsi | Data dan fungsi terpisah. | Data dan fungsi digabungkan di dalam objek. | | Kemudahan Pemeliharaan | Cenderung lebih sulit dipelihara seiring bertambahnya skala program. | Lebih mudah dipelihara karena sifatnya yang modular. | | Skala Program Ideal | Cocok untuk program kecil dan sederhana. | Cocok untuk program besar dan kompleks. |  1. Mudah Diperluas (Scalable): Untuk membuat 100 Goblin, kita hanya perlu membuat 100 objek dari class Karakter dalam sebuah perulangan. Tidak perlu membuat ratusan variabel baru. 2. Terorganisir: Semua yang berhubungan dengan Karakter (data HP & attack, serta perilaku serang()) terbungkus rapi dalam satu Class. 3. Mudah Dipelihara (Maintainable): Jika kita ingin mengubah cara kerja serangan, kita cukup mengubahnya di satu tempat, yaitu di dalam method serang() pada class Karakter. Semua objek akan otomatis mengikuti perubahan tersebut. |

Ketentuan Pengumpulan:  
Kerjakan Tugas PBO Praktik M01 menggunakan file: M1\_PBOP\_TGS.docx

File yang dikumpulkan ke portal adalah file docx

Portal Pengumpulan: Portal Pengumpulan Tugas - PBO Praktik M01

Rename File menjadi:

NPM\_TGS\_PBOPrak\_I\_M01.docx

Batas Upload

Sesuai Pada E-Learning