UTS

Nama : JULIANA GLORIA br SIPAYUNG

NIM : 4243250053

Kelas : PSIK 24 A

Matkul : Pemrograman Berorientasi Objek

1) Empat prinsip utama OOP, berperan besar dalam sistem perangkat lunak yang mudah dikelola.

- Encapsulation; memungkinkan data disimpan dalam satu kesatuan dan hanya bisa diakses melalui method tertentu, → seperti dompet yang bisa dibuka dengan resleting
- Inheritance; memungkinkan class baru mewarisi fitur dari class sebelumnya → seperti tumbuhan yang mewarisi sifat induk tanamannya
- Polymorphism; memungkinkan objek memiliki banyak bentuk, → seperti tombol save yang berfungsi menyimpan dokumen di word dan menyimpan gambar di paint
- Abstraction; menyederhanakan kompleksitas dengan hanya menampilkan hal penting → seperti remot TV yang hanya menunjukkan tombol-tombol meskipun mekanismenya kompleks di dalam.

2) Java 21 memberikan berbagai peningkatan yang membantu dalam pengembangan OOP, seperti fitur String templates, unamed patterns, dan sequenced collections. Fitur String String templates mempermudah untuk penggabungan string dan variabel tanpa banyak tanda tambah (+). Hal ini mempercepat penulisan kode dan mengurangi kemungkinan kesalahan dalam proyek yang membutuhkan kejelasan dalam membaca kode.

3) Class adalah blue print atau cetak biru, sedangkan Object adalah hasil nyata dari blueprint tersebut. contoh lainnya adalah class adalah resep masakan dan object adalah makanan yang dibuat dari resep tsb. Dalam sistem manajemen data mahasiswa, class mahasiswa mendefinisikan atribut seperti nama, NIM. Sedangkan Object Mahasiswa menyimpan data aktual dari masing-masing mahasiswa

4) Untuk mencegah data balance agar tidak sembarangan diubah, kita bisa menggunakan encapsulation. Caranya dengan menjadikan variabel balance bersifat privat dan menyediakan method public seperti deposit() dan withdraw() yang memverifikasi data sebelum melakukan perubahan, a

5) Constructor chaining memungkinkan pemanggilan constructor dari superclass secara eksplisit. menggunakan kata kunci super()

Contoh:

```
Class karyawan {
    karyawan (string nama) {
      system.out.print ("karyawan :" +nama);
    }
}

class Manager extends karyawan {
  Manager (string nama) {
    super(nama);
    sytem.out. Println ("Manager: " + nama);
    }
}
```

Jika super tidak memanggil saat dibutuhkan, maka akan terjadi error karena Java perlu memastikan bahwa bagian atas dari hierarki class telah di inisialisasi.

6). Polymorphism membuat kode lebih fleksibel dan mudah dimaintain karena objek dapat diakses melalui referensi superclass atau interface-nya. Dengan menggunakan Interface, program dapat mengubah implementasi tanpa mengubah kode yang memakainya. Misalnya, interface PesananOnline dapat diimplementasikan oleh class Gofood, Grabfood atau Shopeefood. Saat kita memanggil PesanMakan(), masing-masing class bisa memberikan implementasi yang sama

7) Abstraction menyembunyikan detail yang tidak penting dan hanya memperlihatkan hal yang relevan untuk pengguna. Misalnya, class kendaraan hanya memiliki method Jalan(), tanpa memperlihatkan mekanisme internal seperti sistem penggerak atau jenis mesin Abstraction bisa dicapai melalui abstract class atau interface. Abstract class cocok digunakan saat ada implementasi umum yang bisa diwariskan, sedangkan interface lebih baik untuk mendefenisikan kemampuan yang bisa dimiliki oleh class yang berbeda tanpa memaksakan struktur pewarisan.

KIKY