|  |  |  |
| --- | --- | --- |
| **Nama** | : | Vincentius Leonanda Prabowo |
| **Nim** | : | 2341720149 |
| **Kelas** | : | TI-2A |

**LEMBAR JAWABAN PERCOBAAN 1**

|  |  |
| --- | --- |
| **Soal No** | **Jawaban** |
| 1 | **Class apa sajakah yang merupakan turunan dari class Employee?**   * PermanentEmployee dan InternshipEmployee. Kedua kelas ini mungkin memperluas (extend) class Employee |
| 2 | **Class apa sajakah yang implements ke interface Payable?**   * Class yang mengimplementasikan interface Payable adalah PermanentEmployee dan ElectricityBill. |
| 3 | **Mengapa e bisa diisi dengan objek pEmp (objek dari class PermanentEmployee) dan objek iEmp (objek dari class InternshipEmployee)?**   * Variabel e memiliki tipe data Employee, yang merupakan supertype dari PermanentEmployee dan InternshipEmployee. Dalam Java, tipe Employee dapat merujuk pada objek dari kelas turunannya, yaitu PermanentEmployee atau InternshipEmployee, karena aturan pewarisan yang mengizinkan referensi tipe parent untuk menunjuk ke objek child class. Ini adalah salah satu bentuk polimorfisme. |
| 4 | **Mengapa p bisa diisi dengan objek pEmp (objek dari class PermanentEmployee) dan objek eBill (objek dari class ElectricityBill)?**   * Variabel p bertipe Payable, yang memungkinkan p merujuk pada objek apa pun yang mengimplementasikan interface Payable, termasuk pEmp (objek PermanentEmployee) dan eBill (objek ElectricityBill). Karena keduanya mengimplementasikan Payable, variabel bertipe Payable dapat merujuk pada kedua objek ini. |
| 5 | **Mengapa terjadi error pada penambahan p = iEmp; dan e = eBill;?**   * Kesalahan ini terjadi karena iEmp (objek InternshipEmployee) tidak mengimplementasikan Payable, sehingga tidak kompatibel dengan tipe Payable yang diharapkan untuk p. Demikian juga, eBill (objek ElectricityBill) bukan subclass dari Employee, jadi tidak bisa ditugaskan ke variabel bertipe Employee. Ini adalah aturan tipe dalam Java yang mengharuskan jenis yang kompatibel. |
| 6 | **Kesimpulan tentang konsep/bentuk dasar polimorfisme:**   * Polimorfisme dalam Java memungkinkan variabel dari suatu tipe parent (superclass atau interface) untuk merujuk pada objek dari tipe child (subclass atau implementasi interface). Hal ini memungkinkan objek-objek yang berbeda untuk digunakan secara bergantian selama mereka memiliki kompatibilitas tipe, baik melalui pewarisan (inheritance) atau implementasi interface. Polimorfisme memberikan fleksibilitas dalam program untuk memproses objek yang berbeda dengan cara yang seragam, misalnya, dengan memanggil metode yang sama pada tipe variabel yang lebih umum. |

**LEMBAR JAWABAN EXE 2**

|  |  |
| --- | --- |
| **Soal No** | **Jawaban** |
| 1 |  |
| 2 |  |
| 3 |  |
| 4 |  |
| 5 |  |
| 6 |  |