

Nama : Yovitha Gracia Tavares

NIM : 202412044

Mata Kuliah : Pemrograman Berorientasi Objek

Konsep Inheritance & Polymorphism

a. Source Code

```
➊ Tugas Inheritance & Polymorphism ✎
C:\Users\Lenovo\Documents>New folder (3)\Tugas Inheritance & Polymorphism>...
1 # 
2 # Program: Sistem Manajemen Karyawan
3 # Konsep: Inheritance & Polymorphism
4 # dibuat oleh: yovitha
5 #
6
7 # ----- Parent Class -----
8 class Karyawan:
9     def __init__(self, nama, id_karyawan, gaji_pokok):
10        # Inisialisasi atribut dasar karyawan
11        self.nama = nama
12        self.id_karyawan = id_karyawan
13        self.gaji_pokok = gaji_pokok
14
15    def hitung_gaji(self):
16        # Mengembalikan gaji pokok karyawan
17        return self.gaji_pokok
18
19    def info(self):
20        # Mengembalikan informasi umum tentang karyawan
21        return f"Karyawan : {self.nama}, ID: {self.id_karyawan}, Gaji: {self.hitung_gaji()}"
22
23
24 # ----- Child Class: Manager -----
25 class Manager(Karyawan):
26     def __init__(self, nama, id_karyawan, gaji_pokok, tunjangan):
27        # Memanggil constructor dari parent class menggunakan super()
28        super().__init__(nama, id_karyawan, gaji_pokok)
29        # Menambahkan atribut baru khusus untuk manager
30        self.tunjangan = tunjangan
31
32    def hitung_gaji(self):
33        # override method: menambahkan tunjangan ke gaji pokok
34        return self.gaji_pokok + self.tunjangan
35
36    def info(self):
37        # override method: menampilkan info manager secara lengkap
```

```

38         return f"Manager : {self.nama}, ID: {self.id_karyawan}, Gaji: {self.hitung_gaji()}"
39
40
41 # ----- Child Class: Programmer -----
42 class Programmer(Karyawan):
43     def __init__(self, nama, id_karyawan, gaji_pokok, bonus):
44         # Memanggil constructor parent class
45         super().__init__(nama, id_karyawan, gaji_pokok)
46         # Menambahkan atribut baru khusus untuk programmer
47         self.bonus = bonus
48
49     def hitung_gaji(self):
50         # Override method: menambahkan bonus ke gaji pokok
51         return self.gaji_pokok + self.bonus
52
53     def info(self):
54         # Override method: menampilkan info programmer secara lengkap
55         return f"Programmer : {self.nama}, ID: {self.id_karyawan}, Gaji: {self.hitung_gaji()}"
56
57
58 # ----- Program Utama -----
59 if __name__ == "__main__":
60     # Membuat objek Manager dengan data contoh
61     m = Manager("Yovitha", "M002", 12000000, 3000000)
62
63     # Membuat objek Programmer dengan data contoh
64     p = Programmer("Gracia", "P002", 9000000, 2500000)
65
66     # Menampilkan informasi masing-masing karyawan
67     print(m.info())
68     print(p.info())

```

b. Output

```

Manager : Yovitha, ID: M002, Gaji: 15000000
Programmer : Gracia, ID: P002, Gaji: 11500000

```

c. Penjelasan :

Program dimulai dengan komentar pembuka yang berisi judul dan konsep program. Baris class Karyawan: membuat kelas utama bernama Karyawan sebagai dasar bagi semua karyawan. Di dalamnya terdapat fungsi `__init__` yang digunakan untuk menyimpan data awal yaitu nama, ID karyawan, dan gaji pokok ke dalam atribut `self.nama`, `self.id_karyawan`, dan `self.gaji_pokok`. Fungsi `hitung_gaji()` di kelas ini hanya mengembalikan gaji pokok, sedangkan fungsi `info()` menampilkan informasi lengkap karyawan seperti nama, ID, dan gaji menggunakan format string.

Selanjutnya, baris class Manager(Karyawan): membuat kelas turunan Manager yang mewarisi semua data dan fungsi dari Karyawan. Di dalamnya ada `__init__` yang menambahkan atribut baru yaitu tunjangan, dan menggunakan `super().__init__()` untuk memanggil constructor dari kelas induk. Fungsi `hitung_gaji()` pada Manager menimpa fungsi induknya dengan menghitung total gaji sebagai gaji pokok ditambah tunjangan, sedangkan `info()` juga diubah agar menampilkan tulisan “Manager” di depan data karyawan. Kemudian, dibuat kelas Programmer(Karyawan): yang juga menurunkan kelas Karyawan,

dengan tambahan atribut bonus. Constructor-nya memanggil super().__init__() untuk menyimpan data dasar dan menambahkan bonus.

Fungsi hitung_gaji() di sini menghitung total gaji sebagai gaji pokok ditambah bonus, dan fungsi info() menampilkan data Programmer secara lengkap. Pada bagian if __name__ == "__main__": program utama dijalankan, di mana dibuat objek m dari kelas Manager dengan data nama "Yovitha", ID "M002", gaji pokok 12 juta, dan tunjangan 3 juta, serta objek p dari kelas Programmer dengan data nama "Gracia", ID "P002", gaji pokok 9 juta, dan bonus 2,5 juta. Terakhir, baris print(m.info()) dan print(p.info()) digunakan untuk menampilkan informasi gaji masing-masing. Hasil akhirnya menunjukkan bahwa Manager memiliki total gaji 15 juta dan Programmer memiliki total gaji 11,5 juta, yang membuktikan bahwa konsep inheritance dan polymorphism bekerja dengan baik dalam program ini.