

Nama: Achmad Gibran

Nim: 202412003

Mata Kuliah: Pemrograman Berorientasi Objek

Program Karyawan

1. Sourcecode

```
1  #Achmad Gibran - 202412003
2  #Tugas 4 OOP
3
4  # Class Karyawan (Parent Class)
5  class Karyawan:
6
7      def __init__(self, nama, id_karyawan, gaji_pokok):
8          self.nama = nama
9          self.id_karyawan = id_karyawan
10         self.gaji_pokok = gaji_pokok
11
12     def hitung_gaji(self):
13         return self.gaji_pokok
14
15     def info(self):
16         # Gaji
17         gaji_terformat = float(self.hitung_gaji())
18
19         return f"Karyawan : {self.nama}, ID: {self.id_karyawan}, Gaji: {gaji_terformat}"
20
21 # Class Manager (Subclass)
22 class Manager(Karyawan):
23
24     def __init__(self, nama, id_karyawan, gaji_pokok, tunjangan):
25         super().__init__(nama, id_karyawan, gaji_pokok)
26         self.tunjangan = tunjangan
27
28     def hitung_gaji(self):
29         return self.gaji_pokok + self.tunjangan
30
31     def info(self):
32         parent_info = super().info().replace("Karyawan", "Manager")
33         return parent_info
34
35 # Class Programmer (Subclass)
36 class Programmer(Karyawan):
37
38     def __init__(self, nama, id_karyawan, gaji_pokok, bonus):
39         super().__init__(nama, id_karyawan, gaji_pokok)
40         self.bonus = bonus
41
42     def hitung_gaji(self):
43         return self.gaji_pokok + self.bonus
44
45     def info(self):
46         parent_info = super().info().replace("Karyawan", "Programmer")
47         return parent_info
48
49
50 # Manager: Achmad, A001, Total Gaji 8.000.000 (7jt + 1jt)
51 manager1 = Manager("Achmad", "A001", 7000000, 1000000)
52
53 # Programmer: Gibran, G001, Total Gaji 13.000.000 (11jt + 2jt)
54 programmer1 = Programmer("Gibran", "G001", 11000000, 2000000)
55
56 # Menampilkan
57 print(manager1.info())
58 print(programmer1.info())
```

2. Output

```
Manager : Achmad, ID: A001, Gaji: 8000000.0
Programmer : Gibran, ID: G001, Gaji: 13000000.0
PS C:\Users\ACER NITRO\Documents\Achmad_Gibran_Tugas>
```

3. Analisis

Program berhasil mengimplementasikan sistem manajemen karyawan sederhana menggunakan:

1) Struktur Program (Inheritance)

Program membangun hierarki pewarisan (Inheritance):

- a. Karyawan berfungsi sebagai *superclass* yang menetapkan atribut dasar (nama, id_karyawan, gaji_pokok) untuk semua karyawan.
- b. Manager dan Programmer adalah *subclass* yang mewarisi semua sifat dari Karyawan namun menambahkan atribut unik (tunjangan dan bonus) sesuai peran masing-masing.

2) Perilaku Khusus (Polymorphism)

Program mencapai Polymorphism melalui Method Overriding, di mana satu *method* (hitung_gaji()) memberikan hasil yang berbeda-beda:

- a. Karyawan.hitung_gaji() hanya mengembalikan gaji pokok.
- b. Manager.hitung_gaji() di-override untuk menambahkan tunjangan.
- c. Programmer.hitung_gaji() di-override untuk menambahkan bonus.

Dengan cara ini, program menunjukkan bahwa meskipun semua objek dipanggil dengan perintah yang sama (.info()), setiap objek merespons dengan cara berbeda yang sesuai berdasarkan jenisnya (Manajer atau Programmer).