

Sistem Manajemen Karyawan di Perusahaan

Sebuah perusahaan ingin membuat sistem manajemen karyawan yang mencakup karyawan tetap, karyawan kontrak, dan karyawan magang.

Persyaratan Implementasi

1. Buat kelas Karyawan yang memiliki atribut private:

- Nama (string)
- ID (string)
- Gaji Pokok (double)

Gunakan getter dan setter untuk mengakses atribut ini secara aman.

2. Karyawan adalah kelas parent.

Karyawan Tetap, Karyawan Kontrak, dan Karyawan Magang adalah kelas turunan. Setiap jenis karyawan memiliki metode Hitung Gaji(), tetapi cara perhitungannya berbeda:

- Karyawan Tetap: $\text{Gaji akhir} = \text{Gaji Pokok} + \text{Bonus Tetap (500000)}$
- Karyawan Kontrak: $\text{Gaji akhir} = \text{Gaji Pokok} - \text{Potongan Kontrak (200000)}$
- Karyawan Magang: $\text{Gaji akhir} = \text{Gaji Pokok}$ (tidak ada bonus atau potongan)

3. Gunakan override untuk mengganti metode Hitung Gaji() sesuai jenis karyawan.

4. Minta input dari pengguna untuk memasukkan jenis karyawan, nama, ID, dan gaji pokok. Buat objek sesuai jenis karyawan yang dipilih.

5. Hitung dan tampilkan gaji akhir karyawan tersebut

Jawab :

```
using System;
```

```
class Karyawan
```

```
{
```

```
    private string nama;
```

```
    private string id;
```

```
    private double gajiPokok;
```

```
    public Karyawan(string nama, string id, double gajiPokok)
```

```
    {
```

```
        this.nama = nama;
```

```
        this.id = id;
```

```
        this.gajiPokok = gajiPokok;
```

```
}
```

```
public string Nama
{
    get { return nama; }
    set { nama = value; }
}
```

```
public string ID
{
    get { return id; }
    set { id = value; }
}
```

```
public double GajiPokok
{
    get { return gajiPokok; }
    set { gajiPokok = value; }
}
```

```
public virtual double HitungGaji()
{
    return gajiPokok;
}
```

//Menggunakan metode virtual dan override untuk memberikan implementasi yang berbeda pada setiap kelas turunan.

```
}
```

//Pertama buat parent class Karyawan dengan atribut privat nama, id, gajiPokok dengan method HitungGaji untuk soal nomer 3,
//dan buat constructor untuk menginisialisasi atribut-atribut tersebut.

```
class KaryawanTetap : Karyawan
{
    public KaryawanTetap(string nama, string id, double gajiPokok)
        : base(nama, id, gajiPokok) { }

    public override double HitungGaji()
    {
        return GajiPokok + 500000;
    }
}
```

//Kedua buat subclass KaryawanTetap yang mewarisi dari Karyawan dengan menambahkan gaji tetap 500000 pada method HitungGaji.

```
class KaryawanKontrak : Karyawan
{
    public KaryawanKontrak(string nama, string id, double gajiPokok)
        : base(nama, id, gajiPokok) { }

    public override double HitungGaji()
    {
```

```

        return GajiPokok - 200000;
    }
}

```

//Ketiga buat subclass KaryawanKontrak yang mewarisi dari Karyawan dengan mengurangi gaji pokok 200000 pada method HitungGaji.

```

class KaryawanMagang : Karyawan
{
    public KaryawanMagang(string nama, string id, double gajiPokok)
        : base(nama, id, gajiPokok) { }

    public override double HitungGaji()
    {
        return GajiPokok;
    }
}

```

//Keempat buat subclass KaryawanMagang yang mewarisi dari Karyawan tanpa mengubah gaji pokok pada method HitungGaji.

```

class Program
{
    static void Main()
    {
        Console.WriteLine("+++++ Sistem Manajemen Karyawan +++++");
        Console.WriteLine("1. Karyawan Tetap");
        Console.WriteLine("2. Karyawan Kontrak");
        Console.WriteLine("3. Karyawan Magang");

        Console.Write("Pilih jenis karyawan (1/2/3): ");
        string pilihan = Console.ReadLine();

        Console.Write("Masukkan Nama: ");
        string nama = Console.ReadLine();

        Console.Write("Masukkan ID: ");
        string id = Console.ReadLine();

        Console.Write("Masukkan Gaji Pokok: ");
        double gajiPokok = Convert.ToDouble(Console.ReadLine());
        //Input gaji pokok menggunakan Convert.ToDouble untuk mengkonversi string ke
        double.

```

```

        Karyawan karyawan;

```

```

        switch (pilihan)
        {
            case "1":
                karyawan = new KaryawanTetap(nama, id, gajiPokok);
                break;
            case "2":
                karyawan = new KaryawanKontrak(nama, id, gajiPokok);
                break;

```

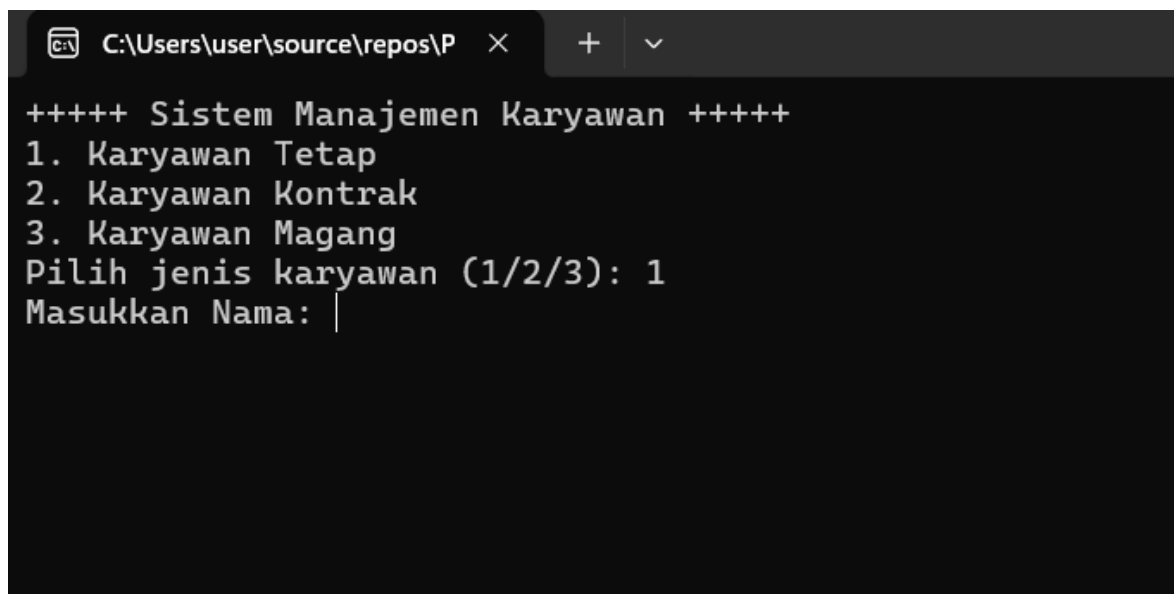
```

        case "3":
            karyawan = new KaryawanMagang(nama, id, gajiPokok);
            break;
        default:
            Console.WriteLine("Pilihan tidak valid.");
            return;
    }

    Console.WriteLine("\n+++++ Informasi Gaji +++++");
    Console.WriteLine($"Nama: {karyawan.Nama}");
    Console.WriteLine($"ID: {karyawan.ID}");
    Console.WriteLine($"Gaji Akhir: Rp {karyawan.HitungGaji():N0}");
}
}

```

//Program ini adalah sistem manajemen karyawan sederhana.
 //Program ini menggunakan konsep pewarisan dan polimorfisme untuk menghitung gaji karyawan berdasarkan jenisnya.
 //Program ini meminta input dari pengguna untuk jenis karyawan, nama, id, dan gaji pokok, lalu menampilkan informasi gaji akhir berdasarkan jenis karyawan yang dipilih.
 //KaryawanTetap menambahkan gaji tetap 500000, KaryawanKontrak mengurangi gaji pokok 200000, dan KaryawanMagang tidak mengubah gaji pokok.
 //Program ini juga menangani beberapa kesalahan input dari pengguna.



```

C:\Users\user\source\repos\P
+++++ Sistem Manajemen Karyawan +++++
1. Karyawan Tetap
2. Karyawan Kontrak
3. Karyawan Magang
Pilih jenis karyawan (1/2/3): 1
Masukkan Nama: |

```

Input jenis karyawan, nama, ID, dan gaji pokok.



```

Microsoft Visual Studio Debug
+++++ Sistem Manajemen Karyawan +++++
1. Karyawan Tetap
2. Karyawan Kontrak
3. Karyawan Magang
Pilih jenis karyawan (1/2/3): 1
Masukkan Nama: Me
Masukkan ID: 1234
Masukkan Gaji Pokok: 350000

+++++ Informasi Gaji +++++
Nama: Me
ID: 1234
Gaji Akhir: Rp 850.000

C:\Users\user\source\repos\PBO THR\PBO THR\bin\Debug\net8.0\PBO THR.exe (process 3444) exited with code 0 (0x0).
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .

```

Tampilkan gaji akhir karyawan.