

Nama : Salsabila

NIM : 242410103032

Kelas : PBO (C)

Prodi : Informatika

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

Code Program

```
using System;

namespace ManajemenKaryawan
{
    class Karyawan
    {
        private string nama;
        private string id;
        private string jenis;
        private double gajiPokok;

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

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

        public string Jenis
        {
            get { return jenis; }
            set { jenis = value; }
        }

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

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

    // Karyawan Tetap
    class KaryawanTetap : Karyawan
    {
        private const double BonusTetap = 500000;
        public override double HitungGaji()
        {
            return GajiPokok + BonusTetap;
        }
    }

    // Karyawan Kontrak
    class KaryawanKontrak : Karyawan
    {
        private const double PotonganKontrak = 200000;
        public override double HitungGaji()
        {
            return GajiPokok - PotonganKontrak;
        }
    }
}
```

```

// Karyawan Magang
class KaryawanMagang : Karyawan
{
    public override double HitungGaji()
    {
        return GajiPokok;
    }
}

class Program
{
    static void Main()
    {
        Console.WriteLine("=== Sistem Manajemen Karyawan ===");
        Console.WriteLine("Jenis Karyawan (tetap / kontrak / magang)");
        Console.Write("Masukkan jenis karyawan\t: ");
        string jenis = Console.ReadLine().ToLower();
        Console.Write("Masukkan nama karyawan\t: ");
        string nama = Console.ReadLine();
        Console.Write("Masukkan ID karyawan\t: ");
        string id = Console.ReadLine();
        Console.Write("Masukkan gaji pokok\t: ");
        double gajiPokok = Convert.ToDouble(Console.ReadLine());

        Karyawan karyawan;

        switch (jenis.ToLower())
        {
            case "tetap":
                karyawan = new KaryawanTetap();
                break;
            case "kontrak":
                karyawan = new KaryawanKontrak();
                break;
            case "magang":
                karyawan = new KaryawanMagang();
                break;
            default:
                Console.WriteLine("Jenis karyawan tidak valid.");
                return;
        }

        karyawan>Nama = nama;
        karyawan.ID = id;
        karyawan.Jenis = jenis;
        karyawan.GajiPokok = gajiPokok;

        Console.WriteLine("\n=== Data Karyawan ===");
        Console.WriteLine("Nama\t\t: " + karyawan>Nama);
        Console.WriteLine("ID\t\t: " + karyawan.ID);
        Console.WriteLine("Jenis Karyawan\t: " + karyawan.Jenis);
        Console.WriteLine("Gaji Pokok\t: " + karyawan.GajiPokok);
        Console.WriteLine("Gaji Akhir\t: " + karyawan.HitungGaji());
    }
}

```

Penjelasan Code Program

Kode program tersebut merupakan penerapan System Manajemen Karyawan Perusahaan yang mencakup tiga jenis karyawan yaitu:

- Karyawan Tetap
- Karyawan Kontrak
- Karyawan Magang

Setiap karyawan tersebut memiliki perbedaan dalam perhitungan gaji akhir.

Struktur Kode

- **Namespace**, merupakan tempat pengelompokan class-class dalam satu wadah/logika tertentu. Pada program ini terdapat Namespace **ManajemenKaryawan** yang digunakan untuk mengelompokkan semua kelas yang terkait dengan Sistem Manajemen Karyawan.

Di dalam namespace **ManajemenKaryawan** ini, terdapat macam-macam Class, yaitu:

```
class Karyawan;  
class KaryawanTetap;  
class KaryawanKontrak;  
class KaryawanMagang;  
class Program;
```

- **Class**
 - ❖ **Karyawan**, merupakan *Parent Class* yang menyimpan atribut (nama, ID, jenis, gajiPokok). Atribut-atribut ini di-enkapsulasi dengan menggunakan properti yaitu getter (get) dan setter (set) dengan tujuan untuk menjaga agar data tetap terlindungi dan tidak sembarangan diubah dari luar kelas. Properti tersebut berfungsi untuk mengakses dan mengubah nilai dari atribut yang tidak bisa diakses langsung dari luar atau bersifat private di dalam class (get digunakan untuk mengambil nilainya sedangkan set digunakan untuk mengubah nilainya dengan aman). Kelas ini memiliki method HitungGaji() yang dideklarasikan sebagai virtual sehingga dapat di-override oleh Child Class atau Kelas Turunannya.

```
class Karyawan  
{  
    private string nama;  
    private string id;  
    private string jenis;  
    private double gajiPokok;  
  
    public string Nama
```

```

{
    get { return nama; }
    set { nama = value; }
}

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

public string Jenis
{
    get { return jenis; }
    set { jenis = value; }
}

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

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

```

- ❖ **KaryawanTetap**, merupakan *Child Class* dari Parent Class Karyawan. Class ini menambahkan bonus tetap pada gaji pokok dan mengoverride metode **HitungGaji()** untuk menghitung gaji akhir dari karyawan.

```

class KaryawanTetap : Karyawan
{
    private const double BonusTetap = 500000;
    public override double HitungGaji()
    {
        return GajiPokok + BonusTetap;
    }
}

```

- ❖ **KaryawanKontrak**, merupakan *Child Class* dari Parent Class Karyawan. Class ini menghitung gaji akhir karyawan tetap yang mengurangi potongan dari gaji pokok dengan mengoverride method **HitungGaji()**

```

class KaryawanKontrak : Karyawan
{
    private const double PotonganKontrak = 200000;
    public override double HitungGaji()
    {
        return GajiPokok - PotonganKontrak;
    }
}

```

- ❖ **KaryawanMagang**, merupakan *Child Class* dari Parent Class Karyawan. Class ini tidak memotong atau memberi bonus gaji pokok, sehingga gaji akhir sama dengan gaji pokok.

```
class KaryawanMagang : Karyawan
{
    public override double HitungGaji()
    {
        return GajiPokok;
    }
}
```

Pada Child Class Karyawan ini, terdapat keyword **const** (Konstanta). Fungsi **const** dalam program digunakan untuk mendeklarasikan sebuah variabel yang nilainya tidak dapat diubah setelah ditetapkan.

- ❖ **Program**, merupakan Class Class ini berisi metode **Main()**, dimana metode ini meminta input dari pengguna untuk atribut dari Class Karyawan, yaitu (jenis karyawan, nama, ID, dan gaji pokok). Berdasarkan input pengguna tersebut, objek karyawan yang sesuai dibuat menggunakan **switch** statement dengan tujuan untuk memeriksa berbagai kemungkinan nilai dari sebuah variable dan menjalankan kode yang sesuai dengan nilainya. Setelah itu, data karyawan dan gaji akhir ditampilkan ke layar.

```
class Program
{
    static void Main()
    {
        Console.WriteLine("=== Sistem Manajemen Karyawan ===");
        Console.WriteLine("Jenis Karyawan (tetap / kontrak / magang)");
        Console.Write("Masukkan jenis karyawan\t: ");
        string jenis = Console.ReadLine().ToLower();
        Console.Write("Masukkan nama karyawan\t: ");
        string nama = Console.ReadLine();
        Console.Write("Masukkan ID karyawan\t: ");
        string id = Console.ReadLine();
        Console.Write("Masukkan gaji pokok\t: ");
        double gajiPokok = Convert.ToDouble(Console.ReadLine());

        Karyawan karyawan;

        switch (jenis.ToLower())
        {
            case "tetap":
                karyawan = new KaryawanTetap();
                break;
            case "kontrak":
                karyawan = new KaryawanKontrak();
                break;
            case "magang":
                karyawan = new KaryawanMagang();
                break;
            default:
                Console.WriteLine("Jenis karyawan tidak valid.");
                return;
        }
    }
}
```

```

    }

    karyawan.Nama = nama;
    karyawan.ID = id;
    karyawan.Jenis = jenis;
    karyawan.GajiPokok = gajiPokok;

    Console.WriteLine("\n=== Data Karyawan ===");
    Console.WriteLine("Nama\t\t: " + karyawan.Nama);
    Console.WriteLine("ID\t\t: " + karyawan.ID);
    Console.WriteLine("Jenis Karyawan\t: " + karyawan.Jenis);
    Console.WriteLine("Gaji Pokok\t: " + karyawan.GajiPokok);
    Console.WriteLine("Gaji Akhir\t: " + karyawan.HitungGaji());
}
}

```

Cara Kerja Class Program ini ialah:

- **Input Pengguna:** Program meminta pengguna untuk memasukkan jenis karyawan, nama, ID, dan gaji pokok karyawan.
- **Pembuatan Objek:** Berdasarkan jenis karyawan yang dimasukkan, program akan membuat objek dari kelas yang sesuai (**KaryawanTetap**, **KaryawanKontrak**, atau **KaryawanMagang**).
- **Pengaturan Atribut:** Atribut objek diatur menggunakan setter.
- **Menghitung Gaji:** Metode **HitungGaji()** dipanggil untuk menghitung gaji karyawan sehingga dapat menghasilkan output gaji akhir karyawan.

Output Program

```

=== Sistem Manajemen Karyawan ===
Jenis Karyawan (tetap / kontrak / magang)
Masukkan jenis karyawan :
Masukkan nama karyawan :
Masukkan ID karyawan :
Masukkan gaji pokok :

=== Data Karyawan ===
Nama :
ID :
Jenis Karyawan :
Gaji Pokok :
Gaji Akhir :

```

Class Diagram

