

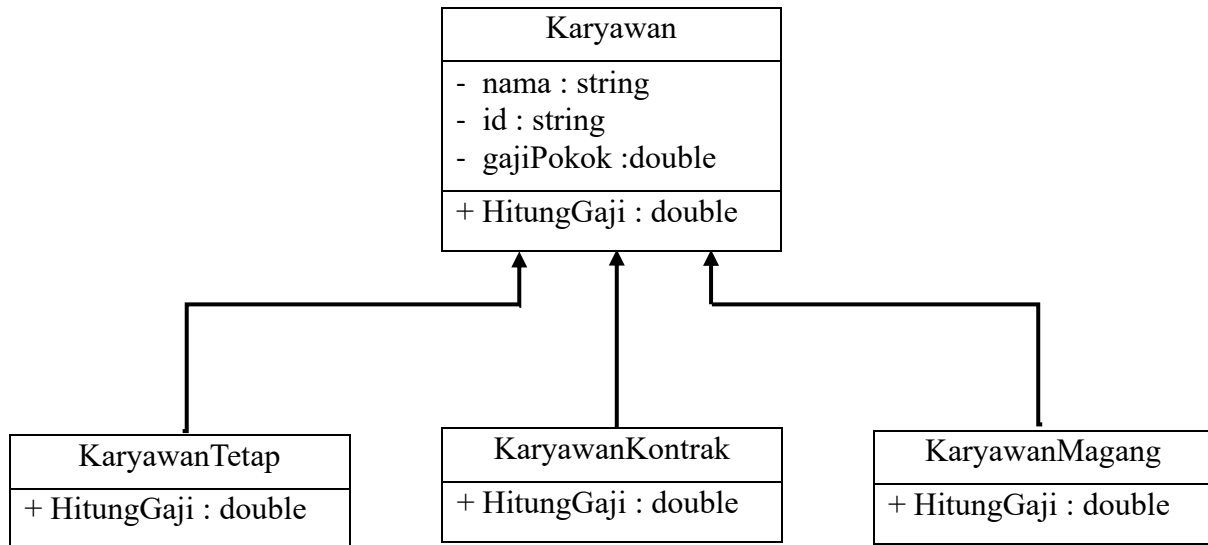
LAPORAN PEMROGRAMAN BERORIENTASI OBYEK

Sistem Manajemen Karyawan di Perusahaan

Yurissandy Al Aham Pratama

242410103083

Class Diagram :



Keterangan :

Kelas Induk (Karyawan)

- Atribut
 - `nama : string`, yang menyimpan nama karyawan (private)
 - `id : string`, yang menyimpan id dari karyawan (private)
 - `gajiPokok : double`, yang menyimpan gaji pokok setiap karyawan(private)
- Metode
 - `+ HitungGaji : double`, sebuah metode yang dapat dioverride oleh kelas turunannya untuk menghitung gaji dari setiap karyawan

Kelas Turunan (KaryawanTetap)

- Metode
 - `+ HitungGaji : double`, sebuah override dari metode `HitungGaji()` dari kelas induknya (Karyawan), serta menambahkan bonus tetap ke gaji pokok setiap karyawan

Kelas Turunan (KaryawanTetap)

- Metode
 - + HitungGaji : double, sebuah override dari metode HitungGaji() dari kelas induknya (Karyawan), serta mengurangi dengan potongan kontrak ke gaji pokok setiap karyawan

Kelas Turunan (KaryawanTetap)

- Metode
 - + HitungGaji : double, sebuah override dari metode HitungGaji() dari kelas induknya (Karyawan)
 -

Penjelasan Kode :

```
1 class Program
2 {
3     static void Main(string[] args)
4     {
5         Console.WriteLine("Masukkan jenis karyawan (tetap/kontrak/magang): ");
6         string jenis = Console.ReadLine().ToLower();
7
8         Console.Write("Nama: ");
9         string nama = Console.ReadLine();
10
11        Console.Write("ID: ");
12        string id = Console.ReadLine();
13
14        Console.Write("Gaji Pokok: ");
15        double gajiPokok = Convert.ToDouble(Console.ReadLine());
16
17        Karyawan karyawan;
18
19        if (jenis == "tetap")
20        {
21            karyawan = new KaryawanTetap(nama, id, gajiPokok);
22        }
23        else if (jenis == "kontrak")
24        {
25            karyawan = new KaryawanKontrak(nama, id, gajiPokok);
26        }
27        else if (jenis == "magang")
28        {
29            karyawan = new KaryawanMagang(nama, id, gajiPokok);
30        }
31        else
32        {
33            Console.WriteLine("Jenis karyawan tidak valid.");
34            return;
35        }
36
37        double gajiAkhir = karyawan.HitungGaji();
38        Console.WriteLine($"Gaji akhir untuk {karyawan>Nama} (ID: {karyawan.ID}) adalah Rp {gajiAkhir:N0}");
39    }
40 }
41
```

```
class Program
```

```
{
```

```
    static void Main(string[] args)
```

Metode `Main` adalah titik awal program, program mulai dieksekusi saat dijalankan.

```
Console.WriteLine("Masukkan jenis karyawan (tetap/kontrak/magang): ");
```

```
string jenis = Console.ReadLine().ToLower();
```

Menampilkan prompt untuk memilih jenis karyawan.

- `ReadLine()` membaca input dari user.

- `ToLower()` digunakan agar input seperti "Tetap" atau "TETAP" tetap dikenali sebagai `"tetap"`.

```
Console.Write("Nama: ");
```

```
string nama = Console.ReadLine();
```

```
Console.Write("ID: ");
```

```
string id = Console.ReadLine();
```

```
Console.Write("Gaji Pokok: ");
```

```
double gajiPokok = Convert.ToDouble(Console.ReadLine());
```

Mengambil input data karyawan dari pengguna: nama, ID, dan gaji pokok. Input dari konsol berupa string diubah ke tipe `double` menggunakan `Convert.ToDouble()`. Gaji pokok digunakan sebagai dasar perhitungan gaji akhir.

```
Karyawan karyawan;
```

Deklarasi variabel `karyawan` dengan tipe `Karyawan`.

```

if (jenis == "tetap")
    karyawan = new KaryawanTetap(nama, id, gajiPokok);
else if (jenis == "kontrak")
    karyawan = new KaryawanKontrak(nama, id, gajiPokok);
else if (jenis == "magang")
    karyawan = new KaryawanMagang(nama, id, gajiPokok);
else
{
    Console.WriteLine("Jenis karyawan tidak valid.");
    return;
}

```

Berdasarkan input user, program membuat objek dari kelas turunan (`KaryawanTetap`, `KaryawanKontrak`, atau `KaryawanMagang`) menggunakan konstruktor. Jika jenis tidak valid, program berhenti dengan `return`.

```

double gajiAkhir = karyawan.HitungGaji();
Console.WriteLine($"
Gaji akhir untuk {karyawan>Nama} (ID: {karyawan.ID}) adalah: Rp {gajiAkhir:N0}");

```

Menghitung dan menampilkan gaji akhir dengan memanggil metode `HitungGaji()` yang telah dioverride oleh masing-masing kelas turunan.

```

1 class Karyawan
2 {
3     private string nama;
4     private string id;
5     private double gajiPokok;
6
7     public Karyawan (string nama, string id, double gajiPokok)
8     {
9         this.nama = nama;
10        this.id = id;
11        this.gajiPokok = gajiPokok;
12    }
13
14    public string Nama
15    {
16        get { return nama; }
17        set { nama = value; }
18    }
19
20    public string ID
21    {
22        get { return id; }
23        set { id = value; }
24    }
25
26    public double GajiPokok
27    {
28        get { return gajiPokok; }
29        set { gajiPokok = value; }
30    }
31
32    public virtual double HitungGaji()
33    {
34        return gajiPokok;
35    }
36 }
37

```

- **Class Karyawan**

class Karyawan

{

private string nama;

private string id;

private double gajiPokok;

Atribut bersifat private agar hanya bisa diakses melalui getter/setter, agar tidak bisa diakses langsung dari luar.

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

Konstruktor untuk mengisi nilai properti saat objek dibuat.

```
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; }  
}
```

Properti (getter/setter) untuk mengakses data pribadi secara aman.

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

Fungsi `HitungGaji` diset sebagai `virtual` agar dapat dioverride oleh kelas anak untuk menerapkan logika perhitungan masing-masing.

```

1 class KaryawanTetap : Karyawan
2 {
3     public KaryawanTetap (string nama, string id, double gajiPokok) : base ( nama, id, gajiPokok) { }
4
5
6     public override double HitungGaji()
7     {
8         return GajiPokok + 500000;
9     }
10 }
11
12 class KaryawanKontrak : Karyawan
13 {
14     public KaryawanKontrak (string nama, string id, double gajiPokok) : base( nama, id, gajiPokok) { }
15
16     public override double HitungGaji()
17     {
18         return GajiPokok - 200000;
19     }
20 }
21
22 class KaryawanMagang : Karyawan
23 {
24     public KaryawanMagang(string nama, string id, double gajiPokok) : base( nama, id, gajiPokok) { }
25
26     public override double HitungGaji()
27     {
28         return GajiPokok;
29     }
30 }

```

- **Class KaryawanTetap**

```

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

```

Menambahkan bonus tetap Rp500.000.

- **Class KaryawanKontrak**

```

public override double HitungGaji()
{
    return GajiPokok - 200000;
}

```

Mengurangi potongan kontrak Rp200.000.

- **Class KaryawanMagang**

```
public override double HitungGaji()
```

```
{
```

```
    return GajiPokok;
```

```
}
```

Tidak ada bonus atau potongan / gaji pokok tetap.