

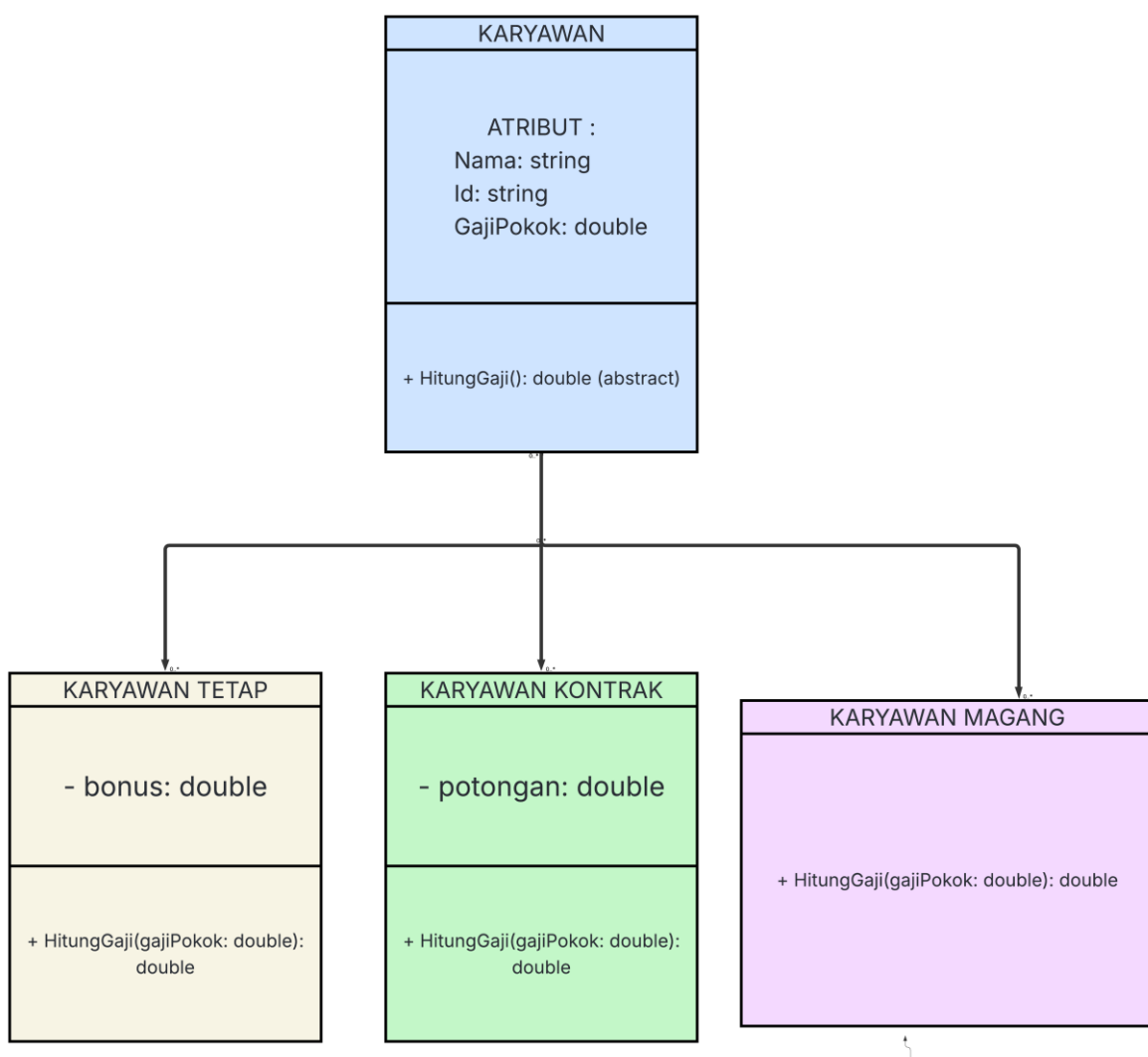
NAMA : Arul Pramana Bahari

NIM : 242410103035

KELAS : PBO_C

LAPORAN SISTEM MANAJEMEN KARYAWAN DI PERUSAHAAN

Class Diagram :



Keterangan Diagram :

- **Kelas Karyawan (Abstract):**
 - Atribut:
 1. Nama
 2. Id
 3. GajiPokok
 - Metode:
 - Property Nama, Id, GajiPokok (getter/setter).
 - HitungGaji() (abstract).
- **Subclass :**
 - KaryawanTetap:
 - Bonus tetap ditambahkan ke gaji.
 - Override HitungGaji().
 - KaryawanKontrak:
 - Potongan kontrak dikurangi dari gaji.
 - Override HitungGaji().
 - KaryawanMagang:
 - Gaji pokok tanpa perubahan.
 - Override HitungGaji().

CODE PROGRAM

1 . SuperClass Karyawan

```
1 using System;
2
3 //Superclass Karyawan
4 class Karyawan
5 {
6     private string nama;
7     private string id;
8     private double gajiPokok;
9
10    1 reference
11    public string GetNama() { return nama; }
12    1 reference
13    public void SetNama(string value) { nama = value; }
14
15    1 reference
16    public string GetID() { return id; }
17    1 reference
18    public void SetID(string value) { id = value; }
19
20    3 references
21    public double GetGajiPokok() { return gajiPokok; }
22    1 reference
23    public void SetGajiPokok(double value) { gajiPokok = value; }
24
25    // Metode untuk menghitung gaji
26    4 references
27    public virtual double HitungGaji()
28    {
29        return gajiPokok;
30    }
31 }
```

Peran :

Sebagai kelas dasar yang menentukan struktur umum untuk semua jenis karyawan.

- Menyediakan atribut dan metode yang wajib diwarisi oleh semua subclass.
- Memastikan konsistensi data dan logika antar jenis karyawan.

2. SubClass Karyawan Tetap

```
// Subclass Karyawan Tetap
1 reference
class KaryawanTetap : Karyawan
{
    2 references
    public override double HitungGaji()
    {
        return base.GetGajiPokok() + 500000;
    }
}
```

Peran :

Menghitung gaji karyawan tetap dengan bonus tetap.Override HitungGaji() :
Menambahkan bonus tetap ke GajiPokok yang diwarisi dari superclass.

Contoh : Gaji Pokok: 7.000.000 → Total Gaji: 7.500.000

3. SubClass Karyawan Kontrak

```
// Subclass Karyawan Kontrak
1 reference
class KaryawanKontrak : Karyawan
{
    2 references
    public override double HitungGaji()
    {
        return base.GetGajiPokok() - 200000;
    }
}
```

Peran :

Menghitung gaji karyawan kontrak dengan potongan kontrak .Override HitungGaji() :
Mengurangi potongan dari GajiPokok yang diwarisi dari superclass.

Contoh : Gaji Pokok: 5.000.000 → Total Gaji: 4.800.000

4. SubClass Karyawan Magang

```
// Subclass Karyawan Magang
1 reference
class KaryawanMagang : Karyawan
{
    2 references
    public override double HitungGaji()
    {
        return base.GetGajiPokok();
    }
}
```

Peran :

Menghitung gaji karyawan magang tanpa bonus/potongan.Override HitungGaji() :
Mengembalikan gajiPokok tanpa perubahan.

Contoh : Gaji Pokok: 2.500.000 → Total Gaji: 2.500.000

Subclass berperan **meng-override** metode `HitungGaji()` untuk menyediakan perilaku spesifik sesuai jenis karyawan.

Contoh:

- Karyawan Tetap → Bonus
- Karyawan Kontrak → Potongan
- Karyawan Magang → Gaji Pokok Murni

Class Main Program :

```
53 // Program utama
54 0 references
55 class Program
56 {
57     0 references
58     static void Main()
59     {
60         // Input jenis karyawan
61         Console.WriteLine("Pilih jenis karyawan:");
62         Console.WriteLine("1. Karyawan Tetap");
63         Console.WriteLine("2. Karyawan Kontrak");
64         Console.WriteLine("3. Karyawan Magang");
65         int pilihan = int.Parse(Console.ReadLine());
66
67         // Input data
68         Console.Write("Masukkan Nama: ");
69         string nama = Console.ReadLine();
70         Console.Write("Masukkan ID: ");
71         string id = Console.ReadLine();
72         Console.Write("Masukkan Gaji Pokok: ");
73         double gajiPokok = double.Parse(Console.ReadLine());
74
75         // Buat objek sesuai pilihan
76         Karyawan karyawan = null;
77         if (pilihan == 1)
78         {
79             karyawan = new KaryawanTetap();
80         }
81         else if (pilihan == 2)
82         {
83             karyawan = new KaryawanKontrak();
84         }
85         else if (pilihan == 3)
86         {
87             karyawan = new KaryawanMagang();
88         }
89         else
90         {
91             Console.WriteLine("Pilihan tidak valid!");
92             return;
93         }
94
95         // Set data ke objek
96         karyawan.SetNama(nama);
97         karyawan.SetID(id);
98         karyawan.SetGajiPokok(gajiPokok);
99
100         // Tampilkan hasil
101         Console.WriteLine("\n=== Hasil Perhitungan ===");
102         Console.WriteLine("Nama: " + karyawan.GetNama());
103         Console.WriteLine("ID: " + karyawan.GetID());
104         Console.WriteLine("Gaji Akhir: " + karyawan.HitungGaji().ToString("N0"));
```

Penjelasan :

Main Program berfungsi sebagai antarmuka interaktif yang menghubungkan logika perhitungan gaji dengan pengguna. Pertama, program menampilkan menu pilihan jenis karyawan (Tetap, Kontrak, atau Magang) dan meminta input berupa nama, ID, serta gaji pokok. Berdasarkan pilihan pengguna, program membuat objek karyawan menggunakan konsep polimorfisme, di mana objek yang dibuat (misalnya **KaryawanTetap**) direferensikan melalui variabel bertipe superclass **Karyawan**. Data yang diinputkan pengguna kemudian disimpan ke dalam objek melalui metode *setter* yang diwarisi dari superclass. Setelah data tersimpan, program memanggil metode **HitungGaji()** yang telah di-override di masing-masing subclass untuk menghitung total gaji sesuai aturan jenis karyawan (misalnya penambahan bonus untuk karyawan tetap atau pengurangan potongan untuk karyawan kontrak). Hasil perhitungan kemudian ditampilkan dengan format angka yang mudah dibaca (menggunakan pemisah ribuan). Program juga dilengkapi dengan loop **do-while** yang memungkinkan pengguna mengulang proses input dan perhitungan tanpa perlu merestart aplikasi.