

LAPORAN PROGRAM SISTEM MANAJEMEN KARYAWAN PADA PERUSAHAAN

Laporan Ini Dibuat Sebagai Salah Satu Tugas pada Mata
Kuliah Pemrograman Berorientasi Objek



Disusun Oleh,

Nama : Aditya Bayu Pratama

NIM : 242410103043

Kelas : E

UNIVERSITAS JEMBER

2025

TUJUAN

Tujuan dari program ini adalah untuk menghitung total gaji karyawan berdasarkan tipe karyawan yang dipilih yaitu Karyawan Tetap, Karyawan Kontrak, dan Karyawan Magang. Dimana untuk ketiga jenis karyawan tersebut memiliki perhitungan gaji yang berbeda-beda. Pada program ini juga akan diminta informasi-informasi penting, sehingga dapat memberikan data yang baik. Pengguna dapat memasukkan data karyawan secara interaktif, seperti nama, ID, dan gaji pokok, lalu program akan menghitung total gaji dengan memperhitungkan bonus atau potongan yang sesuai dengan tipe karyawan.

Selain itu, tujuan pembuatan program ini untuk memperdalam sekaligus mengimplementasikan materi mata kuliah Pemrograman Berorientasi Objek, khususnya materi Encapsulasi dan Inheritance. Dimana pada program ini akan mengimplementasikan encapsulasi berupa private property. Sehingga program dapat mengolah data secara lebih berhati-hati dengan adanya private property.

DESKRIPSI MASALAH

Pada sistem penggajian konvensional, perhitungan gaji sering kali dilakukan secara manual, sehingga rentan terhadap kesalahan dan membutuhkan waktu lebih lama. Setiap tipe karyawan biasanya memiliki aturan yang berbeda dalam penghitungan gaji. Pada sebuah perusahaan terdapat aturan perhitungan gaji seperti berikut.

- Karyawan Tetap mendapat bonus tetap sebesar Rp500.000.
- Karyawan Kontrak mendapat potongan tetap sebesar Rp200.000.
- Karyawan Magang hanya menerima gaji pokok.

Selain itu, setiap tipe karyawan wajib memiliki private property sebagai berikut.

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

Property-property tersebut hanya dapat diakses dengan setter dan getter. Hal ini dilakukan untuk menghindari kesalahan dan mengamankan property. Sehingga private property tersebut tidak bisa langsung diakses.

Masalah yang ingin diselesaikan adalah bagaimana membuat sistem yang fleksibel dan mudah digunakan untuk menghitung total gaji berdasarkan tipe karyawan tersebut dengan property-property yang diperlukan. Selain itu program juga harus dapat menampilkan informasi total gaji secara jelas dengan informasi penting seperti nama dan ID.

ANALISIS KEBUTUHAN SYSTEM

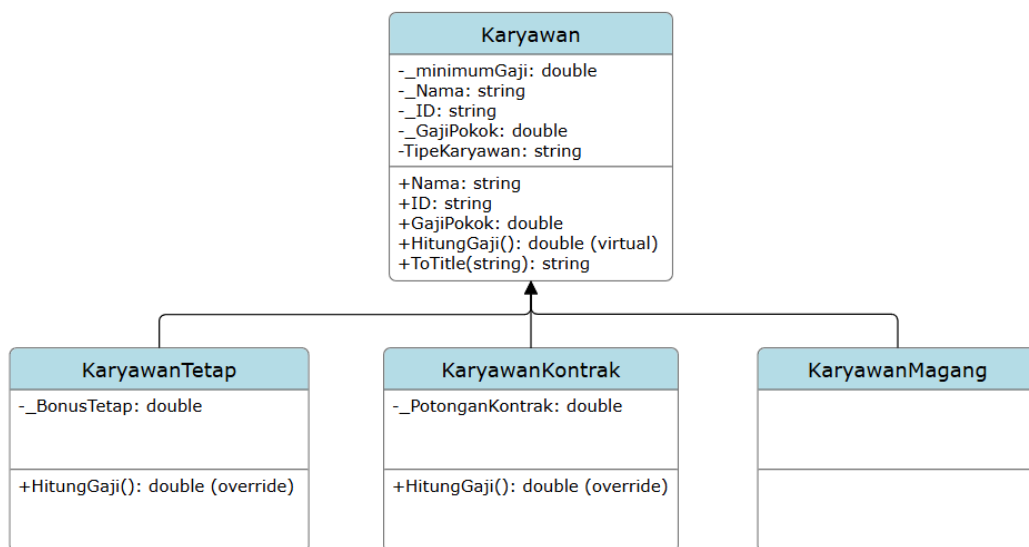
Fungsional:

- Memilih tipe karyawan (Tetap, Kontrak, Magang).
- Memasukkan data karyawan: nama, ID, dan gaji pokok.
- Validasi data input (hanya huruf untuk nama, angka untuk gaji, minimal gaji).
- Menampilkan hasil perhitungan gaji sesuai tipe karyawan.

Non-Fungsional:

- Antarmuka berbasis teks (CLI).
- Input yang robust dengan validasi yang jelas sehingga dapat menghindari kemungkinan error.
- Kode modular dengan prinsip OOP (inheritance dan polymorphism).

CLASS DIAGRAM



Private Class dan Setter Getter

`_Nama`: string memiliki setter getter Nama: string

`_ID`: string memiliki setter getter ID: string

`_GajiPokok`: double memiliki setter getter GajiPokok: double

IMPLEMENTASI

```
using System;
class Program
{
    static void Main()
    {
        Karyawan karyawan;
        while (true) {
            Console.Write("Program Hitung Gaji Karyawan\nPilih Tipe
Karyawan (Tulis Angka Saja)\n1. Karyawan Tetap\n2. Karyawan
Kontrak\n3. Karyawan Magang\n0. Keluar\nPilih : ");
            string choice = Console.ReadLine();
            switch (choice)
            {
                case "1":
                    karyawan = new KaryawanTetap();
                    break;
                case "2":
                    karyawan = new KaryawanKontrak();
                    break;
                case "3":
                    karyawan = new KaryawanMagang();
                    break;
                case "0":
                    Console.WriteLine("Keluar dari program.");
                    return; // Exit the program
                default:
                    Console.WriteLine("Pilihan tidak valid.
Silakan coba lagi.\n");
                    continue; // Go back to the start of the loop
            }
            while(karyawan>Nama == null){
                Console.Write("Masukkan nama karyawan: ");
                string nama = Console.ReadLine();
                karyawan>Nama = nama;
            }
            while(karyawan>ID == null){
                Console.Write("Masukkan Id karyawan: ");
                string id = Console.ReadLine();
                karyawan>ID = id;
            }
        }
    }
}
```

```

        while(karyawan.GajiPokok == 0){
            Console.Write("Masukkan Gaji Pokok karyawan: ");
            string gaji = Console.ReadLine();
            if(gaji.All(char.IsDigit) && gaji!=""){
                karyawan.GajiPokok = double.Parse(gaji);
            }else{
                Console.WriteLine("Inputan harus berupa angka
saja");
            }
        }

        Console.WriteLine($"
---\tRANGKUMAN\t---
Nama\t\t:
{karyawan.Nama}
Tipe Karyawan\t: {karyawan.TipeKaryawan}
Id\t\t:
{karyawan.ID}
Total Gaji\t:
Rp{karyawan.HitungGaji().ToString("N0").Replace(",",".")}
");
        Console.WriteLine("Tekan \"Enter\" untuk lanjut");
        Console.ReadLine();
    }
}

class Karyawan{
    private double _minimumGaji = 1_500_000; // to avoid
getting negative value
    public string TipeKaryawan {get; private set;} // in Karyawan
tipe Karyawan Kontrak
    private string _Nama;
    private string _ID;
    private double _GajiPokok = 0;
    public Karyawan(string TipeKaryawan){
        this.TipeKaryawan = TipeKaryawan;
    }
    public string Nama{
        get { return _Nama; }
        set {
            if (value.Replace(" ", "").All(Char.IsLetter) && value !=
"") {
                _Nama = ToTitle(value);}
            else {
                Console.WriteLine("Nama harus huruf saja");
            }
        }
    }
    public string ID{

```

```

        get { return _ID; }
        set {
            if (value != ""){
                _ID = value;
            }else{
                Console.WriteLine("ID tidak boleh kosong");
            }
        }
    }
}

public double GajiPokok{
    get { return _GajiPokok; }
    set {
        if (value >= _minimumGaji) {
            _GajiPokok = value;
        } else {
            Console.WriteLine($"Gaji Pokok harus lebih dari
{_minimumGaji}");
        }
    }
}

public virtual double HitungGaji(){
    return _GajiPokok;
}

public string ToTitle(string words){           //Merubah setiap huruf
didepan menjadi besar, ex. bayu pratama -> Bayu Pratama
    string[] wordsSplit = words.Split(" ");
    string placeHolder = "";
    foreach(string var in wordsSplit){
        placeHolder += " " + char.ToUpper(var[0]) +
var.Substring(1);
    }
    placeHolder = placeHolder.Substring(1);
    return placeHolder;
}
}

class KaryawanTetap:Karyawan{
    private int _BonusTetap = 500_000;
    public KaryawanTetap():base("Karyawan Tetap"){
    public override double HitungGaji()
    {
        return GajiPokok + _BonusTetap;
    }
}
}

```

```

}
class KaryawanKontrak:Karyawan{
    private double _PotonganKontrak = 200_000;
    public KaryawanKontrak():base("Karyawan Kontrak"){
    public override double HitungGaji()
    {
        Console.WriteLine($"Potongan Gaji: {_PotonganKontrak}");
        return GajiPokok - _PotonganKontrak;
    }
}
class KaryawanMagang:Karyawan{
    public KaryawanMagang():base("Karyawan Magang"){
}
}

```

Program ini diimplementasikan menggunakan bahasa pemrograman C# dengan pendekatan Object-Oriented Programming (OOP) khususnya Encapsulasi dan Inheritance. Berikut fitur-fitur utamanya:

- **Class Karyawan:** Merupakan kelas dasar yang menyimpan atribut umum seperti nama, ID, dan gaji pokok. Juga berisi metode `HitungGaji()` yang bisa dioverride.
- **Class Turunan (KaryawanTetap, KaryawanKontrak, KaryawanMagang):** Masing-masing mengimplementasikan aturan perhitungan gaji spesifik. Hal tersebut dapat dicapai dengan melakukan override method `HitungGaji()`.
- **Validasi Input:** Program hanya menerima input yang sesuai, seperti nama yang hanya mengandung huruf dan gaji yang berupa angka.
- **Loop Interaktif:** Program berjalan terus-menerus sampai pengguna memilih keluar (0), dan menampilkan rangkuman setiap perhitungan.

Selain itu, ada beberapa fitur pendukung yaitu:

- **Property _minimumGaji** untuk menghindari minus pada tipe karyawan kontrak. Hal ini akan membuat inputan `GajiPokok` harus lebih dari sama dengan `_minimumGaji`.
- **Method ToTitle()** untuk memastikan nama ditulis dengan format huruf besar disetiap katanya.
- **Property TipeKaryawan** untuk menyimpan data tipe karyawan dan dapat dibaca saja, tidak dapat dirubah.

SIMPULAN

Program ini berhasil mengimplementasikan sistem penghitungan gaji yang modular dan interaktif. Dengan menerapkan prinsip OOP, kode menjadi lebih terstruktur dan mudah diperluas untuk tipe karyawan lain di masa depan. Program ini cocok digunakan sebagai dasar dari sistem penggajian sederhana yang dapat ditingkatkan menjadi aplikasi berbasis GUI atau terintegrasi dengan database. Dengan pengecekan input yang cukup, program ini terbilang robust sehingga dapat menangani berbagai kemungkinan inputan user.