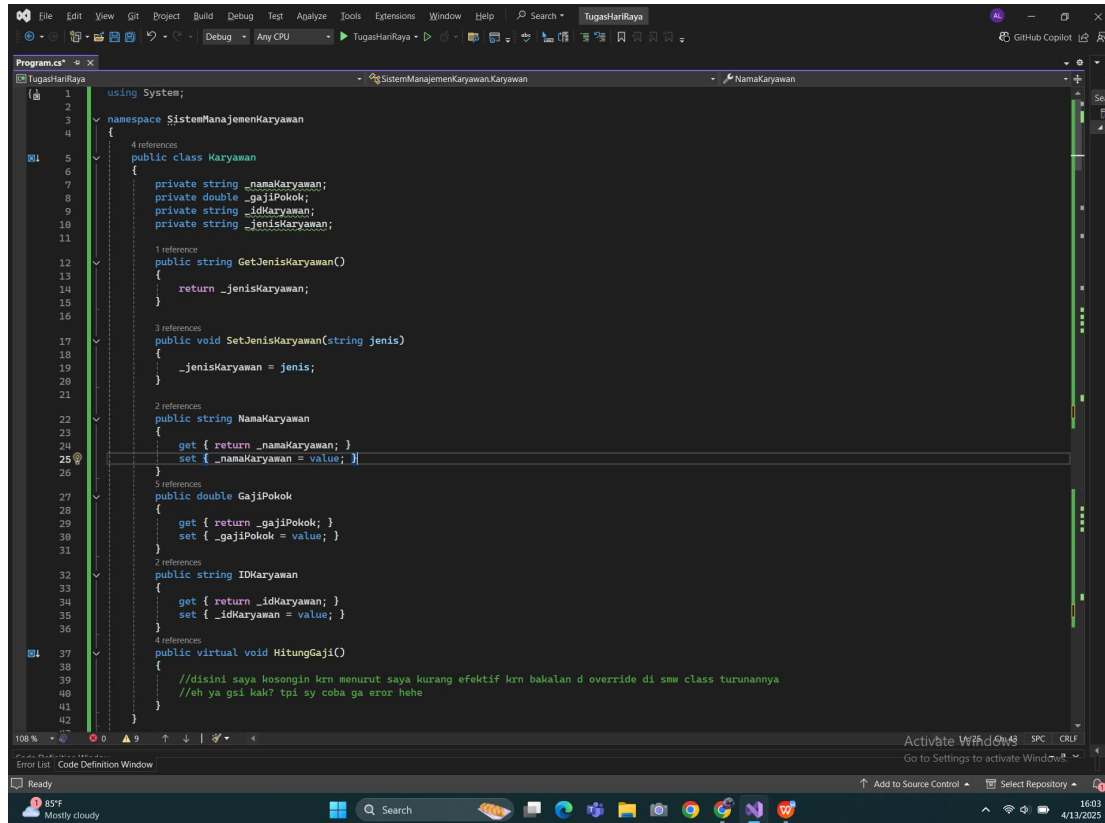


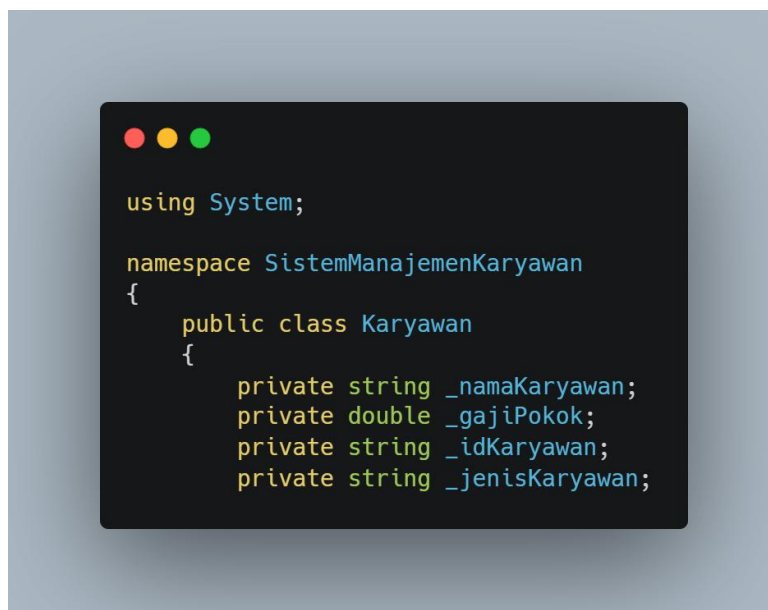
Nama : Aden Alexandria Syaiful Perdana
Kelas : PBO A
NIM : 242410102033

PBO PR Oleh Oleh Hari Raya

1. Class Karyawan



- Saya akan menjelaskan beberapa isinya menjadi beberapa bagian.



- Pertama saya membuat class Karyawan yang nantinya bakal menjadi class parent
- Di class Karyawan ini saya membuat beberapa atribut yaitu `_namaKaryawan`, `_gajiPokok`, `_idKaryawan`, `_jenisKaryawan` yang akan di override di class bawahannya.
- Saya private karena ketentuan dari tugasnya.

```

public string GetJenisKaryawan()
{
    return _jenisKaryawan;
}

public void SetJenisKaryawan(string jenis)
{
    _jenisKaryawan = jenis;
}

```

- private `_jenisKaryawan` tidak bisa diakses langsung dari luar class.
- Jadi kita buat method untuk:
 - Membaca data : `GetJenisKaryawan()`
 - Mengubah data : `SetJenisKaryawan(string jenis)`

```

public string NamaKaryawan
{
    get { return _namaKaryawan; }
    set { _namaKaryawan = value; }
}

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

public string IDKaryawan
{
    get { return _idKaryawan; }
    set { _idKaryawan = value; }
}

```

- Ini disebut property dengan accessor manual.
- get untuk mengambil nilai sedangkan set untuk mengisi nilai

```
public virtual void HitungGaji()
{
    //disini saya kosongin krn
    menurut saya kurang efektif krn bakalan
    d override di smw class turunannya
    //eh ya gsi kak? tpi sy coba ga
    eror hehe
}
```

- virtual artinya method ini akan di override pada class turunan.
- Di class Karyawan, method ini cuma sebagai class parent.
- Lalu setiap class turunan (Tetap, Kontrak, Magang) membuat versi masing-masing.
- Isinya saya kosongin karena nanti bakalan di override di class turunannya.

2. public class Tetap : Karyawan


```
public class Tetap : Karyawan
{
    private double BonusTetap = 500000;

    public override void HitungGaji()
    {
        double GajiAkhir = GajiPokok + BonusTetap;
        Console.WriteLine($"Gaji Akhir : Rp {GajiAkhir}");
    }
}
```

- Disini saya akan membuat class turunan Tetap dari class parent Karyawan.
- Di class Tetap ini bakalan mewarisi semua properti dan method yang ada di class Karyawan seperti :

- NamaKaryawan
- IDKaryawan
- GajiPokok
- HitungGaji() [khusus method ini bakalan di override]
- BonusTetap adalah atribut khusus untuk karyawan tetap. Nilainya sudah di tetapkan sebesar 500000.
- Keyword override digunakan karena method HitungGaji() di class Karyawan adalah virtual yang artinya sedang mengganti perilaku default HitungGaji() dengan logika yang ada pada karyawan tetap.
- Kemudian isi logika pada HitungGaji() : GajiPokok diambil dari property induk (Karyawan)
- BonusTetap adalah tambahan yang hanya dimiliki oleh karyawan tetap
- GajiAkhir adalah total dari keduanya

3. public class Kontrak : Karyawan



```
public class Kontrak : Karyawan
{
    private double PotonganKontrak = 200000;

    public override void HitungGaji()
    {
        double GajiAkhir = GajiPokok - PotonganKontrak;
        Console.WriteLine($"Gaji Akhir : Rp {GajiAkhir}");
    }
}
```

- Disini saya akan membuat class turunan Kontrak dari class parent Karyawan.
- Ini kurang lebih hampir sama dengan class Tetap, namun hanya ada perbedaan pada method HitungGaji().
- Pada class turunan Kontrak di method GajiAkhir logikanya adalah Gaji akhir = Gaji pokok dikurangi potongan kontrak yaitu 200000.

4. public class Magang : Karyawan

```
public class Magang : Karyawan
{
    public override void HitungGaji()
    {
        double GajiAkhir = GajiPokok;
        Console.WriteLine($"Gaji Akhir : Rp {GajiAkhir}");
    }
}
```

- Disini saya akan membuat class turunan Magang dari class parent Karyawan.
- Ini kurang lebih hampir sama dengan class Tetap dan Kontrak, namun hanya ada perbedaan pada method HitungGaji().
- Pada class turunan Magang di method GajiAkhir logikanya adalah Gaji akhir = Gaji pokok tanpa ada pengurangan atau penambahan.

5. Class Program

- Disini saya membuat class program yang dimana class program ini adalah class untuk menampilkan output pada layar.

```
class Program
{
    static void Main(string[] args)
    {
        Console.WriteLine("Sistem Manajemen Perusahaan SEKAREP");

        Console.WriteLine("Jenis Karyawan :");
        Console.WriteLine("[1]. Karyawan Tetap");
        Console.WriteLine("[2]. Karyawan Kontrak");
        Console.WriteLine("[3]. Karyawan Magang");
        Console.Write("\nPilih [1/2/3] : ");
        int jenisKaryawan = int.Parse(Console.ReadLine());
    }
}
```

- Awalnya kita akan disuruh melakukan input jenis karyawan apa yang berisi Karyawan Tetap, Karyawan Kontrak, atau Karyawan Magang. Kita isi dengan pilihan 1/2/3.

```

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

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

Console.Write("Gaji Pokok : ");
double gajiPokok = double.Parse(Console.ReadLine());

```

- Kemudian kita akan melakukan input kembali untuk mengisi nama, ID Karyawan, dan Gaji Pokok. Karena di Gaji Pokok akan ada fungsi matematika, maka menggunakan variabel tipe double. Kita harus mengisi sebuah angka untuk Gaji Pokok.

```

Karyawan karyawan;

if (jenisKaryawan == 1)
{
    karyawan = new Tetap();
    karyawan.SetJenisKaryawan("Karyawan tetap");
}
else if (jenisKaryawan == 2)
{
    karyawan = new Kontrak();
    karyawan.SetJenisKaryawan("Karyawan Kontrak");
}
else if (jenisKaryawan == 3)
{
    karyawan = new Magang();
    karyawan.SetJenisKaryawan("Karyawan Magang");
}
else
{
    Console.WriteLine("Pilihan jenis karyawan tidak ditemukan.");
    return;
}

```

- Kita akan mendeklarasi ulang Variabel Karyawan yang berada pada class induk yang artinya variabel ini nantinya bisa menampung objek dari kelas turunannya karena menggunakan konsep polimorfisme.
- Kemudian ada fungsi if-else yaitu :
- Jika input jenisKaryawan = 1 berarti pengguna memilih karyawan tetap. Maka dibuat objek dari class Tetap, lalu disimpan ke dalam variabel karyawan
- Kemudian kamu panggil SetJenisKaryawan(...) untuk menyimpan informasi jenisnya ke dalam objek.
- Jika input = 2, maka buat objek dari class Kontrak
- Jenis karyawan diset menjadi "Karyawan Kontrak"
- Jika input = 3, maka buat objek dari class Magang
- Jenis karyawan diset menjadi "Karyawan Magang"

- Jika input yang dimasukkan selain 1,2,3 maka akan menampilkan “Pilihan jenis karyawan tidak ditemukan.”
- Lalu menggunakan syntax return untuk menghentikan eksekusi program agar tidak error atau lanjut ke langkah berikutnya.

```
karyawan>NamaKaryawan = nama;
karyawan.IDKaryawan = id;
karyawan.GajiPokok = gajiPokok;
```

- Tiga baris ini adalah pengisian data utama Karyawan.
- karyawan>NamaKaryawan = nama;>NamaKaryawan adalah property publik milik class Karyawan.
- Field aslinya sebenarnya private (_namaKaryawan), tapi bisa diakses melalui property>NamaKaryawan.
- karyawan.IDKaryawan = id; Sama seperti di atas, IDKaryawan adalah property publik untuk field privat _idKaryawan.
- Mengisi ID yang dimasukkan user ke dalam objek.
- karyawan.GajiPokok = gajiPokok;
- GajiPokok adalah property publik untuk field _gajiPokok.

```
Console.WriteLine("\n Detail Karyawan Perusahaan SEKAREP");
Console.WriteLine($"Nama Karyawan : {karyawan>NamaKaryawan}");
Console.WriteLine($"Jenis Karyawan : {karyawan.GetJenisKaryawan()}");
Console.WriteLine($"ID Karyawan : {karyawan.IDKaryawan}");
Console.WriteLine($"Gaji Pokok : Rp {karyawan.GajiPokok}");
karyawan.HitungGaji();
    }
}
```

- Ini adalah hasil akhirnya dimana akan menampilkan Detail Karyawan yang sudah kita inputkan pada layar.
- Output yang keluar adalah Nama Karyawan, Jenis Karyawan, ID Karyawan, dan Gaji Pokok karyawan yang sudah kita hitung bersama sesuai dengan turunan class Tetap, Kontrak, dan Magang.

6. OUTPUT

- Jika input yang dimasukkan sesuai semua :

```
Microsoft Visual Studio Debu x + v
Sistem Manajemen Perusahaan SEKAREP
Jenis Karyawan :
[1]. Karyawan Tetap
[2]. Karyawan Kontrak
[3]. Karyawan Magang

Pilih [1/2/3] : 1
Nama Karyawan : Aden
ID Karyawan : 2033
Gaji Pokok : 5000000

Detail Karyawan Perusahaan SEKAREP
Nama Karyawan : Aden
Jenis Karyawan : Karyawan tetap
ID Karyawan : 2033
Gaji Pokok : Rp 5000000
Gaji Akhir : Rp 5500000

D:\TugasHariRaya\TugasHariRaya\bin\Debug\net8.0\TugasHariRaya.exe (process 4608) exited with code
0 (0x0).
Press any key to close this window . . .|

Activate Windows
Go to Settings to activate Windows.

83°F Mostly cloudy 18:59 4/13/2025
```

- Jika input pilih Jenis Karyawan salah :

```
Microsoft Visual Studio Debu x + v
Sistem Manajemen Perusahaan SEKAREP
Jenis Karyawan :
[1]. Karyawan Tetap
[2]. Karyawan Kontrak
[3]. Karyawan Magang

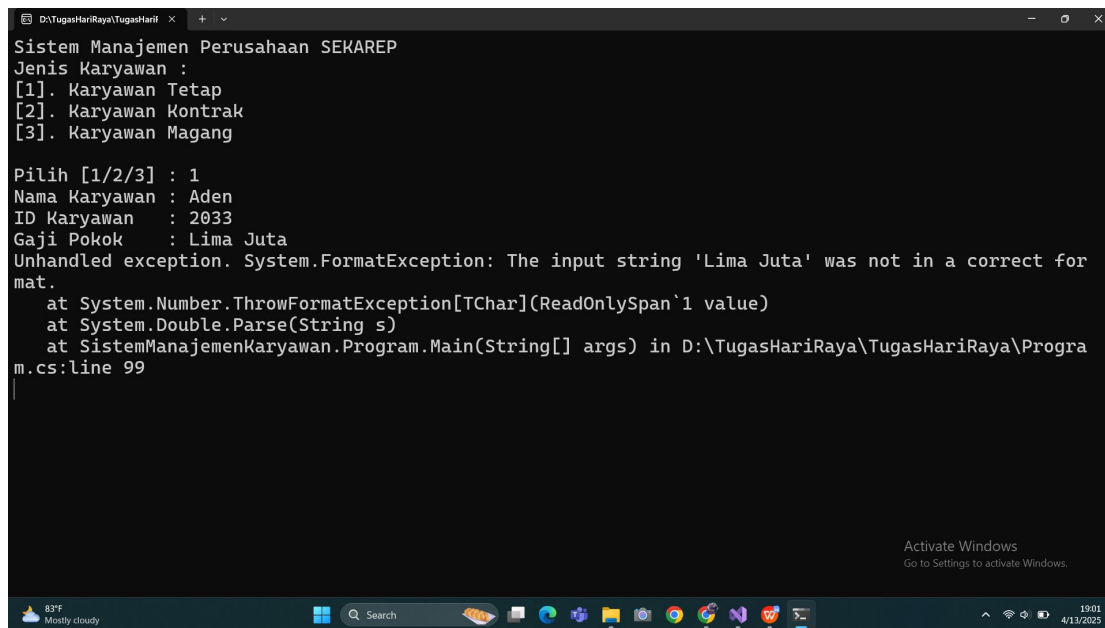
Pilih [1/2/3] : 4
Nama Karyawan : Aden
ID Karyawan : 2033
Gaji Pokok : 5000000
Pilihan jenis karyawan tidak ditemukan.

D:\TugasHariRaya\TugasHariRaya\bin\Debug\net8.0\TugasHariRaya.exe (process 14792) exited with code
0 (0x0).
Press any key to close this window . . .

Activate Windows
Go to Settings to activate Windows.

83°F Mostly cloudy 19:00 4/13/2025
```


- Jika input gaji pokok tidak berupa angka :



```
Sistem Manajemen Perusahaan SEKAREP
Jenis Karyawan :
[1]. Karyawan Tetap
[2]. Karyawan Kontrak
[3]. Karyawan Magang

Pilih [1/2/3] : 1
Nama Karyawan : Aden
ID Karyawan : 2033
Gaji Pokok : Lima Juta
Unhandled exception. System.FormatException: The input string 'Lima Juta' was not in a correct format.
at System.Number.ThrowFormatException[TChar](ReadOnlySpan`1 value)
at System.Double.Parse(String s)
at SistemManajemenKaryawan.Program.Main(String[] args) in D:\TugasHariRaya\TugasHariRaya\Program.cs:line 99
```

7. GPT

25% menggunakan GPT untuk membantu memberikan penjelasan pada code.