

Nama : Tunggul Abdul Majid

NIM : 242410102058

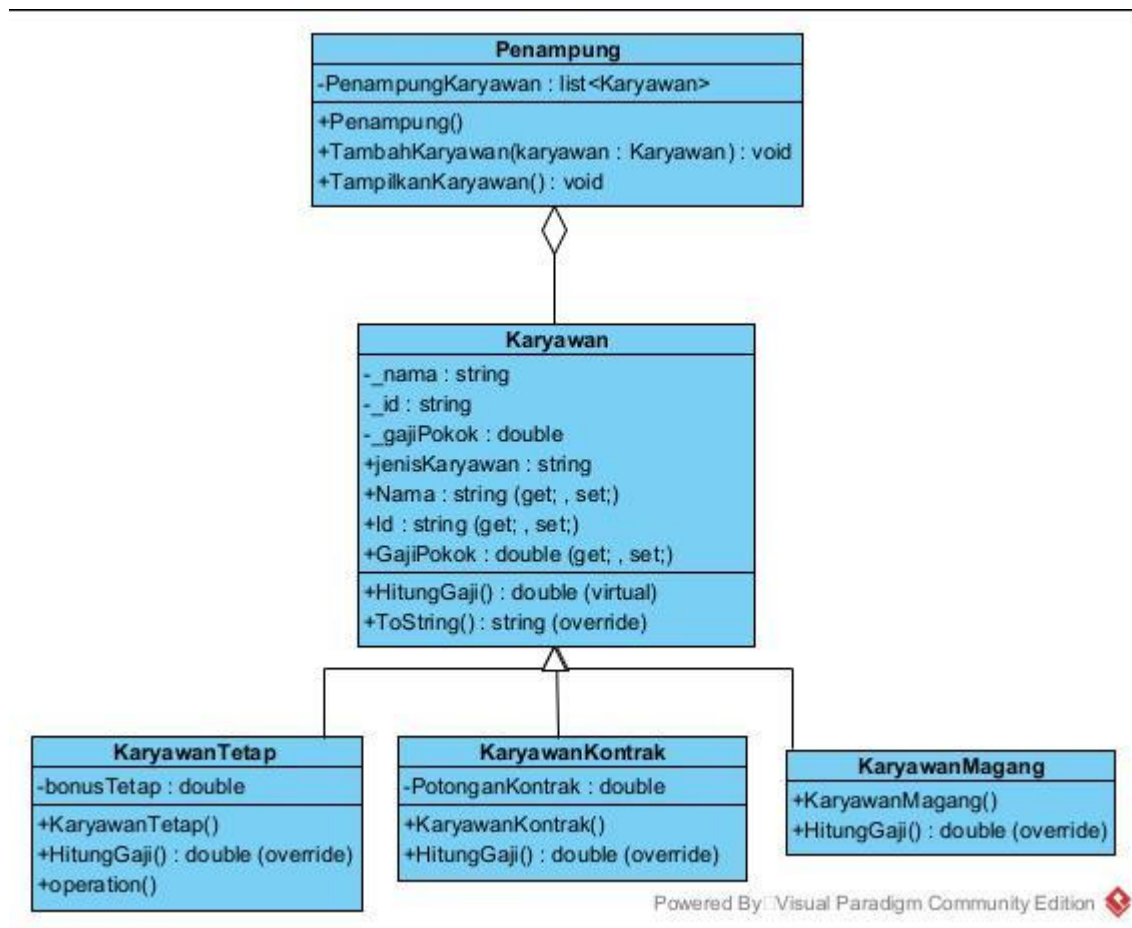
Prodi : Teknologi Informasi

Mata Kuliah : PBO (Praktikum, Kelas B)

Github : <https://github.com/tunggulmajid/Tugas-Praktikum-Encapsulasi/>

TUGAS ENCAPSULASI (Sistem Manajemen Karyawan di Perusahaan)

Class Diagram :



Penjelasan :

Dalam program ini terdapat beberapa kelas, yang meliputi :

- Penampung
- Karyawan
- KaryawanTetap
- KaryawanKontrak
- KaryawanMagang

Beberapa diantara kelas yang telah saya sebutkan tersebut memiliki beberapa hubungan atau relasi, yang meliputi :

➤ **Kelas Penampung dan Kelas Karyawan**

- Keduanya memiliki hubungan agregation yang mana kelas penampung disini bertugas untuk menampung objek dari kelas karyawan, hal tersebut dibuktikan dengan method yang ada dalam kelas penampung yaitu TambahkanKaryawan. Ketika kelas penampung dihapuskan tidak akan berdampak terhadap kelas karyawan, objek dari kelas karyawan akan tetap berdiri sendiri

➤ **Kelas Karyawan dengan ketiga kelas karyawan lainnya (KaryawanTetap, KaryawanKontrak, dan KaryawanMagang)**

- Memiliki hubungan Inheritance, kelas karyawan sebagai superclass atau parent class sedangkan ketiga kelas karyawan lainnya merupakan child class atau sub class nya. Attribut dari kelas Karyawan beserta methodnya akan diturunkan kepada sub class nya tersebut.

Code :

➤ **Karyawan**

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Tugas_Praktikum_Encapsulasi
{
    6 references
    internal class Karyawan
    {
        private string _nama, _id;
        private double _gajiPokok;
        public string jenisKaryawan = "Karyawan";
        5 references
        public string Nama
        {
            get { return _nama; }
            set
            {
                if (value != null && value.Length > 2)
                {
                    _nama = value;
                }
                else
                {
                    _nama = "";
                    Console.WriteLine("Nama Harus Melebihi 2 Karakter");
                    Program.Enter();
                }
            }
        }
    }
}
```

```

5 references
public string Id
{
    get { return _id; }
    set
    {
        if (value != null && value.Length > 2)
        {
            _id = value;
        }
        else
        {
            _id = "";
            Console.WriteLine("Id Tidak Boleh Kosong");
        }
    }
}

7 references
public double GajiPokok
{
    get { return _gajiPokok; }
    set
    {
        if (value > 0 && Convert.ToString(value) != null)
        {
            _gajiPokok = value;
        }
        else
        {
            Console.WriteLine("Gaji Pokok Tidak boleh Minus");
        }
    }
}

5 references
public virtual double HitungGaji()
{
    double gajiTotal = GajiPokok ;
    return gajiTotal;
}

0 references
public override string ToString()
{
    double jumlahGaji = HitungGaji();
    return $"Nama Karyawan : {Nama}\nJenis Karyawan : {jenisKaryawan}\nId Karyawan : {Id}\nTotal Gaji : {jumlahGaji}";
}

```

Penjelasan : Dalam Kelas Karyawan memiliki beberapa attribut dengan acces modifier private dan juga public, untuk private disini berarti attribut tersebut hanya dapat diakses dalam kelas karyawan saja dan tidak bisa diakses dalam kelas lain contoh : `_nama`, `_id`, `_gajiPokok`. Sedangkan yang memiliki acces modifier public bisa diakses di kelas manapun contoh : `jenisKaryawan`.

Untuk Attribut yang memiliki acces acces modifier private harus memiliki sebuah properti baru untuk mengakses nya (get dan set nya). Dalam kelas ini ada beberapa properti yang meliputi :

- Nama untuk mengakses `_nama`
- Id untuk mengakses `_id`
- GajiPokok untuk mengakses `_gajiPokok`

Masing masing properti tersebut memiliki "get" untuk mengakses value dalam variable private nya, dan juga "set" untuk memberikan value ke dalam variabel private, untuk set disini bisa di customize dengan menggunakan if else, jadi untuk value nya bisa kita sesuaikan dengan kehendak kita. Seperti contoh pada properti Nama, disana jika inputan berupa null atau inputan kurang dari 2 karakter maka akan masuk ke kondisi else yang mana kondisi tersebut

menyebabkan proses menambahkan karyawan error dalam kelas penampung nanti.

Dalam kelas ini juga memiliki beberapa method yang meliputi : HitungGaji yang merupakan method ber return type double untuk menghitung total gaji karyawan dan berbentuk virtual, jadi dapat di customize untuk body nya dalam sub class nya nanti. Kelas ini juga memiliki method bertipe override dengan nama ToString yang berfungsi untuk mengubah tampilan default atau format default ketika menampilkan objek dari kelas karyawan dalam list penampung nantinya. Jadi disini ToString itu merupakan method bawaan objek, dan dia bersifat virtual dan bisa di override sesuai dengan kebutuhan.

➤ KaryawanTetap

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Tugas_Praktikum_Encapsulasi
{
    internal class KaryawanTetap : Karyawan
    {
        private double bonusTetap = 500000;

        public KaryawanTetap()
        {
            jenisKaryawan = "Karyawan Tetap";
        }

        public override double HitungGaji()
        {
            double gajiTotal = GajiPokok + bonusTetap;
            return gajiTotal;
        }
    }
}
```

Penjelasan : Kelas KaryawanTetap ini merupakan kelas turunan atau sub class dari kelas Karyawan, jadi kelas ini memiliki attribut dan juga method dari kelas karyawan. Kelas ini memiliki attribut tambahan berupa bonusTetap yang bertipe double dengan acces modifier private. Kelas ini juga memiliki konstruktor yang mana ketika kelas ini dibuat akan langsung mengisi attribut jenisKaryawan dengan value berupa Karyawan Tetap.

Kelas ini memiliki method hasil override dari kelas karyawan yaitu method HitungGaji, dalam kelas ini method tersebut berisi value dari properti GajiPokok ditambahkan dengan value dari attribut bonusTetap.

➤ KaryawanKontrak

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Tugas_Praktikum_Encapsulasi
{
    internal class KaryawanKontrak : Karyawan
    {
        private double PotonganKontrak = 200000;

        public KaryawanKontrak()
        {
            jenisKaryawan = "Karyawan Kontrak";
        }

        public override double HitungGaji()
        {
            double gajiTotal = GajiPokok - PotonganKontrak;
            return gajiTotal;
        }
    }
}
```

Penjelasan : Kelas ini hampir sama dengan Kelas KaryawanTetap, hanya saja yang membedakan kelas KaryawanKontrak ini dengan kelas KaryawanTetap yaitu terletak dalam atribut tambahan dan juga isi dari method override HitungGajinya.

Dalam kelas ini memiliki atribut tambahan berupa potonganKontrak yang bertipe double dengan private sebagai akses modifiernya dan untuk isi dari method HitungGaji disini berupa value dari atribut GajiPokok dikurangi dengan value dari atribut PotonganKontrak.

➤ KaryawanMagang

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Tugas_Praktikum_Encapsulasi
{
    3 references
    internal class KaryawanMagang : Karyawan
    {
        1 reference
        public KaryawanMagang()
        {
            jenisKaryawan = "Karyawan Magang";
        }
        3 references
        public override double HitungGaji()
        {
            double gajiTotal = base.HitungGaji();
            return gajiTotal;
        }
    }
}
```

Penjelasan : Sama seperti kedua kelas sebelumnya, namun dalam kelas Karyawan Magang ini tidak terdapat atribut tambahan dan juga method HitungGaji yang di override tetap memiliki kesamaan hitungan dengan super class nya yaitu kelas Karyawan.

➤ Penampung

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Tugas_Praktikum_Encapsulasi
{
    3 references
    internal class Penampung
    {
        1 reference
        private List<Karyawan> PenampungKaryawan;

        1 reference
        public Penampung()
        {
            PenampungKaryawan = new List<Karyawan>();
        }

        3 references
        public void TambahKaryawan(Karyawan karyawan)
        {
            if (karyawan>Nama.Length > 0 && karyawan.Id.Length > 2 && karyawan.GajiPokok > 0)
            {
                PenampungKaryawan.Add(karyawan);
                Console.WriteLine($"{karyawan.jenisKaryawan} Berhasil Ditambahkan");
            }
            else
            {
                Console.WriteLine("Karyawan tidak Berhasil ditambahkan, silahkan ulangi proses");
            }
        }

        1 reference
        public void TampilkanKaryawan()
        {
            if (PenampungKaryawan.Count > 0)
            {
                for (int i = 0; i < PenampungKaryawan.Count; i++)
                {
                    Console.WriteLine($"{PenampungKaryawan[i]}\n");
                }
            }
            else
            {
                Console.WriteLine("Belum Ada Karyawan yang di Inputkan");
            }
        }
    }
}
```

Penjelasan : Kelas Penampung ini merupakan kelas yang berfungsi menampung objek dari kelas Karyawan beserta sub classnya. Didalam kelas penampung ini terdapat sebuah attribut berupa list yang memiliki tipe data yang ditampung yaitu objek dari kelas Karyawan dan juga memiliki acces modifier private.

Dalam kelas ini juga memiliki sebuah konstruktor, jadi ketika kelas ini dibuat maka akan langsung mengeksekusi program atau code yang berada dalam konstrukturnya, dalam kelas ini yaitu membuat list dari penampungKaryawan.

Selain itu dalam kelas ini memiliki 2 method ber return type void yaitu TambahKaryawan yang mana method ini berfungsi untuk menambahkan objek dari kelas karyawan yang berada dalam parameternya ke dalam list Penampung karyawan yang bersifat private. Terdapat syarat untuk

menambahkan objek kedalam list penampungKaryawan, ketika syarat atau salah satu kondisi tidak terpenuhi maka penambahan tidak akan berhasil.

Sedangkan method satu lagi adalah method TampilkanKaryawan yang memiliki fungsi untuk menampilkan value atau objek yang berada dalam list, untuk memanggil atau mengakses objek dalam list disini memakai pengulangan yaitu for, jadi nanti value dalam list akan dipanggil sesuai dengan index nya. Dan untuk outputnya / format output akan sesuai dengan yang telah di set dalam method override ToString yang berada dalam kelas karyawan.

➤ Program

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Runtime.Remoting.Messaging;
using System.Text;
using System.Threading.Tasks;

namespace Tugas_Praktikum_Encapsulasi
{
    1 reference
    internal class Program
    {
        public static Penampung penampung = new Penampung();
        0 references
        static void Main(string[] args)
        {
            awal();
        }
        14 references
        public static void garis()
        {
            Console.WriteLine(new string ('=', 60));
        }
        6 references
        public static void Header()
        {
            garis();
            Console.WriteLine("Tugas Hari Raya".PadLeft(35));
            garis();
        }
        9 references
        public static void Enter()
        {
            Console.Write("Tekan [Enter] Untuk Melanjutkan >> ");
            Console.ReadLine();
        }
    }
}
```


6 references

```
public static void awal()
{
    Console.Clear();
    Header();
    Console.WriteLine("1.Tambahkan Karyawan\n2.Lihat Karyawan\n3.Keluar");
    garis();
    Console.Write("Masukkan Pilihan >> ");
    string pilihan = Console.ReadLine();

    if (pilihan == "1")
    {
        TambahkanKaryawan();
    }
    else if (pilihan == "2")
    {
        Console.Clear();
        Header();
        Console.WriteLine("");

        penampung.TampilkanKaryawan();

        Console.WriteLine("");
        garis();
        Enter();
        awal();
    }
    else if (pilihan == "3")
    {
        absen();
    }
    else
    {
        Console.WriteLine("Pilihan Tidak Tersedia");
        Enter();
        awal();
    }
}
```

2 references

```
public static void TambahkanKaryawan()
{
    Console.Clear();
    Header();
    Console.WriteLine("1.Karyawan Tetap\n2.Karyawan Kontrak\n3.Karyawan Magang");
    garis();
    Console.Write("Masukkan Pilihan >> ");
    string pilihan = Console.ReadLine();
    Console.Clear();
    Header();

    if (pilihan == "1")
    {
        KaryawanTetap karyawanTetap = new KaryawanTetap();

        Console.Write("Masukkan Nama Karyawan Tetap >> ");
        karyawanTetap.Nama = Console.ReadLine();

        Console.Write("Masukkan Id Karyawan Tetap >> ");
        karyawanTetap.Id = Console.ReadLine();

        Console.Write("Masukkan Gaji Pokok Karyawan Tetap >> ");
        karyawanTetap.GajiPokok = Convert.ToDouble(Console.ReadLine());

        garis();
        penampung.TambahKaryawan(karyawanTetap);
        //Console.WriteLine("Karyawan Tetap Berhasil Ditambahkan");
        garis();

        Enter();
        awal();
    }
}
```

```

else if (pilihan == "2")
{
    KaryawanKontrak karyawanKontrak = new KaryawanKontrak();

    Console.Write("Masukkan Nama Karyawan Kontrak >> ");
    karyawanKontrak.Nama = Console.ReadLine();

    Console.Write("Masukkan Id Karyawan Kontrak >> ");
    karyawanKontrak.Id = Console.ReadLine();

    Console.Write("Masukkan Gaji Pokok Karyawan Kontrak >> ");
    karyawanKontrak.GajiPokok = Convert.ToDouble(Console.ReadLine());

    garis();
    penampung.TambahKaryawan(karyawanKontrak);
    //Console.WriteLine("Karyawan Kontrak Berhasil Ditambahkan");
    garis();

    Enter();
    awal();
}

else if (pilihan == "3")
{
    KaryawanMagang karyawanMagang = new KaryawanMagang();

    Console.Write("Masukkan Nama Karyawan Magang >> ");
    karyawanMagang.Nama = Console.ReadLine();

    Console.Write("Masukkan Id Karyawan Magang >> ");
    karyawanMagang.Id = Console.ReadLine();

    Console.Write("Masukkan Gaji Pokok Karyawan Magang >> ");
    karyawanMagang.GajiPokok = Convert.ToDouble(Console.ReadLine());

    garis();
    penampung.TambahKaryawan(karyawanMagang);
    //Console.WriteLine("Karyawan Magang Berhasil Ditambahkan");
    garis();
    Enter();
    awal();
}

```

```

else
{
    Console.WriteLine("Pilihan Tidak Valid");
    garis();
    Enter();
    TambahkanKaryawan();
}

1 reference
public static void absen()
{
    Console.Clear();
    Header();

    Console.WriteLine("\nAbsen Dulu Bang :");
    Console.WriteLine($"Nama : Tunggul Abdul Majid" +
        "\nNIM : 242410102058" +
        "\nProdi : Teknologi Informasi\n");
    garis();
    Enter();
    exit();
}

1 reference
public static void exit()
{
    Console.Clear();
    Header();
    Console.WriteLine("\nTerima Kasih\n");
    garis();
    Enter();
}

```

Penjelasan : Kelas Program ini merupakan kelas utama untuk mengeksekusi program, didalamnya terdapat objek dari kelas kelas lainnya, disini ada sebuah objek yang bersifat public statis, jadi dia dapat diakses dalam kelas manapun,

dengan menyebutkan lengkap alamatnya, **contoh : Program.penampung** , selain itu ada beberapa objek lain dari kelas kelas yang merupakan sub dari kelas Karyawan, yang mana nanti nya akan dimasukkan ke dalam list dalam objek penampung menggunakan method TambahKaryawan.

Dalam Kelas Program ini juga ada beberapa method yang memang di set sebagai statis agar memudahkan untuk memanggilnya dalam kelas lain. Untuk method nya meliputi :

- Main
 - Merupakan method yang secara default menjadi tempat dimana program dijalankan.
- garis
 - Method ini digunakan untuk mencetak garis sebuah garis dengan Panjang 60 karakter
- Header
 - Method ini digunakan untuk mencetak header, method ini memanggil method garis untuk mencetak garis dibagian atas dan bawah string judul nya.
- Enter
 - Method yang berisikan inputan kosong untuk membatasi program sebelum berpindah ke method atau proses lain.
- awal
 - Sebagai awal program, merupakan method pertama yang dipanggil dalam method Main. Menggunakan awal program agar memudahkan untuk membuat alur dari program. Dan juga membuat main tetap rapi karena hanya terdapat sebuah pemanggilan method saja.

Dalam method ini user diberikan pilihan atau menu yang berupa tambah karyawan, lihat karyawan atau keluar program. Apabila inputan dari user tidak sesuai dengan index dari ketiga pilihan tersebut maka akan masuk ke kondisi else yang memberikan peringatan ke user untuk memasukkan inputan lagi. Kemudian jika inputan user sesuai dengan salah satu indeks dari beberapa opsi tersebut maka akan diarahkan ke masing masing method sesuai dengan opsi yang dipilih.

Contoh : jika yang dipilih adalah opsi lihat karyawan maka akan memanggil method dari objek penampung berupa Tampilkan Karyawan.

- TambahkanKaryawan
 - Setelah dari method awal, jika user memilih opsi tambahkan karyawan maka akan langsung mengarah ke dalam method ini. Dalam method ini user diberikan

beberapa opsi karyawan yang akan ditambahkan (tetap, kontrak, magang). Apabila inputan selain dari ketiga indeks dari opsi tersebut maka akan masuk ke dalam kondisi else yang akan memunculkan peringatan dan user akan di arahkan untuk memasukkan inputan lagi.

Apabila user memilih salah satu dari ketiga opsi tersebut maka user akan diberi sebuah perintah untuk memasukkan beberapa inputan yang berupa nama, id, dan gaji pokok karyawan yang kemudian akan dibuatkan sebuah objek sesuai dengan pilihan karyawan yang telah dipilih dan inputan yang diberikan tadi akan menjadi value dari beberapa atribut yang ada dalam objek karyawan yang dipilih.

Setelah objek terbuat maka objek akan dimasukkan ke dalam list yang berada di objek penampung menggunakan method dari kelas Penampung yaitu TambahKaryawan.

- Absen
 - Merupakan method yang berisi nama nim dan prodi
- Exit
 - Method akhir dari program, jika dalam method awal user memilih untuk keluar maka akan diarahkan ke dalam method ini.

Output :

```
Tugas Hari Raya

1.Tambahkan Karyawan
2.Lihat Karyawan
3.Keluar

Masukkan Pilihan >> |
```

Penjelasan : User diberikan 3 pilihan opsi, tiap opsi tersebut dapat dipilih berdasarkan indeks yang ada didepan setiap opsi. Ini merupakan output dari method awal.

```
Tugas Hari Raya

1.Karyawan Tetap
2.Karyawan Kontrak
3.Karyawan Magang

Masukkan Pilihan >> |
```

Penjelasan : ketika user memilih menu tambahkan karyawan maka akan diarahkan selanjutnya ke menu ini, user diberi beberapa opsi jenis karyawan yang ingin ditambahkan, user dapat mengakses opsi menggunakan index yang ada di setiap depan opsi.

```
Tugas Hari Raya

Masukkan Nama Karyawan Tetap >> tunggul
Masukkan Id Karyawan Tetap >> 001
Masukkan Gaji Pokok Karyawan Tetap >> 7000000

Karyawan Tetap Berhasil Ditambahkan

Tekan [Enter] Untuk Melanjutkan >> |
```

Penjelasan : ketika user sudah memilih jenis karyawan maka user akan diminta untuk menginputkan data dari karyawan seperti nama, id dan gaji. Tentunya format tersebut memiliki syarat seperti nama dan id harus lebih dari 2 karakter dan juga gaji tidak boleh berupa minus, jika syarat terpenuhi maka akan muncul string karyawan sesuai jenis yang dipilih berhasil ditambahkan, jika syarat tidak terpenuhi maka user akan di kembalikan ke menu sebelumnya.

```
Tugas Hari Raya

Nama Karyawan : tunggul
Jenis Karyawan : Karyawan Tetap
Id Karyawan : 001
Total Gaji : 7500000

Nama Karyawan : roihan
Jenis Karyawan : Karyawan Kontrak
Id Karyawan : 002
Total Gaji : 6800000

Nama Karyawan : rafi
Jenis Karyawan : Karyawan Magang
Id Karyawan : 003
Total Gaji : 7000000

Tekan [Enter] Untuk Melanjutkan >> |
```

Penjelasan : Jika pada awal program user memilih opsi lihat karyawan maka outputnya akan menjadi seperti gambar diatas, output ini diambil dari list

penampungkaryawan dan format disesuaikan di method ToString dalam kelas Karyawan. User dapat menekan enter untuk kembali ke awal program.

```
=====
                          Tugas Hari Raya
=====

Absen Dulu Bang :

Nama : Tunggul Abdul Majid
NIM : 242410102058
Prodi : Teknologi Informasi

=====
Tekan [Enter] Untuk Melanjutkan >> |
```

Penjelasan : Ketika user memilih menu exit pada awal program, maka tampilan ini akan dimunculkan, berisi informasi terkait dengan data diri. User dapat menekan enter untuk melanjutkan.

```
=====
                          Tugas Hari Raya
=====

Terima Kasih

=====
Tekan [Enter] Untuk Melanjutkan >> |
```

Penjelasan : setelah user menekan enter dari menu data diri maka akan di arahkan ke akhir program (tampilan pada gambar diatas). User dapat menekan enter dan program akan berakhir.