

Nama : Dimastian Aji Wibowo

NIM : 2311104058

TP MODUL 14

Source Code

Before

```
using System;

class HaloGeneric
{
    // Method generic untuk menyapa user
    public void SapaUser<T>(T user)
    {
        Console.WriteLine($"Halo user {user}");
    }
}

class DataGeneric<T>
{
    private T data;
    // Konstruktor untuk menginisialisasi data
    public DataGeneric(T data)
    {
        this.data = data;
    }
    // Method untuk mencetak data
    public void PrintData()
    {
        Console.WriteLine($"Data yang tersimpan adalah: {data}");
    }
}

class Program
{
    static void Main()
    {
        HaloGeneric halo = new HaloGeneric();

        // Meminta input nama dari user
        Console.Write("Masukkan nama Anda: ");
        string nama = Console.ReadLine();

        // Memanggil method dengan input dari user
        halo.SapaUser(nama);

        // Meminta input NIM dari user
        Console.Write("Masukkan NIM Anda: ");
        string nim = Console.ReadLine();

        // Membuat objek DataGeneric dengan tipe string
        DataGeneric<string> dataGeneric = new DataGeneric<string>(nim);

        // Memanggil method PrintData
        dataGeneric.PrintData();
    }
}
```

After

```
using System;

namespace TModul14
{
    /// <summary>
    /// Class untuk menyapa user menggunakan method generic
    /// </summary>
    class GreetingService
    {
        /// <summary>
        /// Method generic untuk menyapa user
        /// </summary>
        /// <typeparam name="T">Tipe data user (contoh: string)</typeparam>
        /// <param name="user">Data user yang akan disapa</param>
        public void GreetUser<T>(T user)
        {
            Console.WriteLine($"Halo user {user}");
        }
    }

    /// <summary>
    /// Class untuk menyimpan dan mencetak data generic
    /// </summary>
    /// <typeparam name="T">Tipe data yang akan disimpan</typeparam>
    class DataContainer<T>
    {
        private T data;

        /// <summary>
        /// Konstruktor untuk menginisialisasi data
        /// </summary>
        /// <param name="data">Data yang akan disimpan</param>
        public DataContainer(T data)
        {
            this.data = data;
        }

        /// <summary>
        /// Method untuk mencetak data
        /// </summary>
        public void PrintData()
        {
            Console.WriteLine($"Data yang tersimpan adalah: {data}");
        }
    }

    /// <summary>
    /// Entry point program
    /// </summary>
    class Program
    {
        static void Main()
        {
            GreetingService greetingService = new GreetingService();

            // Meminta input nama dari user
            Console.Write("Masukkan nama Anda: ");
            string nama = Console.ReadLine();

            // Menyapa user
            greetingService.GreetUser(nama);

            // Meminta input NIM dari user
            Console.Write("Masukkan NIM Anda: ");
            string nim = Console.ReadLine();

            // Membuat objek DataContainer dengan tipe string
            DataContainer<string> dataContainer = new DataContainer<string>(nim);

            // Menampilkan data
            dataContainer.PrintData();
        }
    }
}
```

A. Naming Convention

1. Class HaloGeneric menjadi GreetingService

Nama class diubah agar lebih deskriptif dan sesuai dengan konvensi PascalCase untuk class di C# dan GreetingService lebih jelas menggambarkan fungsi utamanya, yaitu memberikan sapaan (greeting).

2. Method SapaUser<T> menjadi GreetUser<T>

Nama method diubah agar lebih konsisten menggunakan bahasa Inggris dan mengikuti PascalCase dan GreetUser<T> lebih mudah dipahami dan mengikuti standar C#.

3. Class DataGeneric<T> menjadi DataContainer<T>

Nama class diubah untuk lebih spesifik menggambarkan fungsinya sebagai container untuk data.

B. White Space dan Indentation

Menambahkan spasi kosong di antara blok kode utama untuk meningkatkan keterbacaan dengan menjaga konsistensi indentasi 4 spasi atau 1 tab di seluruh kode dan menambahkan spasi sebelum dan sesudah operator = untuk meningkatkan keterbacaan.

C. Variable Declarations

Deklarasi variabel data dalam class DataContainer<T> dipindahkan ke bagian atas class untuk memisahkan deklarasi dan logika. Ini juga mengikuti prinsip deklarasi sebelum penggunaan (declaration before use).

D. Comments

Menambahkan XML Documentation Comments di atas setiap class dan method untuk memberikan informasi lengkap mengenai fungsi dan parameter yang digunakan. Komentar ini tidak hanya meningkatkan keterbacaan kode, tetapi juga memungkinkan penggunaan fitur IntelliSense di Visual Studio.