
LAPORAN PRAKTIKUM TP MODUL 7

NIM / Nama: 2311104072 – Jauhar Fajar Zuhair

Bagian I: Pendahuluan

Tugas Pendahuluan Modul 7 ini berfokus pada dua aspek utama: penggunaan Git untuk manajemen versi dan kolaborasi, serta deserialisasi data JSON menggunakan C#. Praktikum ini melibatkan langkah-langkah mulai dari pengaturan repositori Git, pembuatan branch, implementasi pembacaan file JSON untuk data mahasiswa dan mata kuliah, hingga proses commit, push, dan pull request di GitHub. Laporan ini akan menjelaskan implementasi kode C# yang dibuat untuk membaca dan menampilkan data dari file JSON sesuai instruksi.

Bagian II: Konsep Utama

1. **JSON (JavaScript Object Notation):** Format pertukaran data yang ringan, mudah dibaca manusia, dan mudah diproses mesin. Sering digunakan untuk mengirim data antar aplikasi web atau menyimpan konfigurasi. Dalam TP ini, JSON digunakan untuk menyimpan data profil mahasiswa dan daftar mata kuliah.
2. **Deserialisasi JSON:** Proses mengubah teks berformat JSON menjadi objek dalam bahasa pemrograman (dalam kasus ini, objek C#). Library `System.Text.Json` bawaan .NET digunakan untuk melakukan tugas ini, memetakan properti dalam JSON ke properti dalam kelas C#.

Bagian III: Implementasi Kode

Kode C# yang dibuat terdiri dari beberapa bagian utama:

1. **DataMahasiswa_2311104072.cs:**
 - o Bertujuan untuk membaca file JSON berisi data profil mahasiswa (`tp7_1_2311104072.json`).
 - o Mendefinisikan kelas `JSONFormat` dan `NameFormat` untuk menampung struktur data dari JSON.
 - o Memiliki method `ReadJSON` yang melakukan deserialisasi.

```
// ... (Definisi kelas NameFormat)
// public class NameFormat { public string depan { get; set; } public
string belakang { get; set; } }

// ... (Definisi kelas JSONFormat)
// public class JSONFormat { public NameFormat nama { get; set; }
public double nim { get; set; } public string fakultas { get; set; } }

public class DataMahasiswa_2311104072
{
    public JSONFormat ReadJSON(string filename)
```

```

    {
        // Membaca teks dari file JSON
        string jsonString = File.ReadAllText(filename);
        // Melakukan deserialisasi JSON ke objek JSONFormat
        JSONFormat data =
        JsonSerializer.Deserialize<JSONFormat>(jsonString);
        return data;
    }
}

```

Penjelasan Snippet: Method `ReadJSON` menerima nama file, membaca seluruh isinya sebagai string, lalu menggunakan `JsonSerializer.Deserialize` untuk mengubah string JSON tersebut menjadi objek `JSONFormat`.

2. **KuliahMahasiswa_2311104072.cs:**

- Bertujuan untuk membaca file JSON berisi daftar mata kuliah (tp7_2_2311104072.json).
- Mendefinisikan kelas `JSONFormat2` (yang berisi `List<Course>`) dan `Course` untuk struktur data mata kuliah.
- Memiliki method `ReadJSON` untuk deserialisasi data mata kuliah.

```

3. // ... (Definisi kelas Course)
// public class Course { public string code { get; set; } public string
name { get; set; } }

// ... (Definisi kelas JSONFormat2)
// public class JSONFormat2 { public List<Course> courses { get; set; }
}

public class KuliahMahasiswa_2311104072
{
    public JSONFormat2 ReadJSON(string filename) // Nama method dan
return type disesuaikan dari Program.cs
    {
        string jsonString = File.ReadAllText(filename);
        // Melakukan deserialisasi JSON ke objek JSONFormat2
        JSONFormat2 data =
        JsonSerializer.Deserialize<JSONFormat2>(jsonString);
        return data;
    }
}

```

Penjelasan Snippet: Mirip dengan `DataMahasiswa`, method `ReadJSON` di sini membaca file JSON mata kuliah dan mendeserialisasinya menjadi objek `JSONFormat2`, yang di dalamnya terdapat list (`List<Course>`) dari objek `Course`.

4. **Program.cs:**

- Titik masuk utama aplikasi (Main method).
- Membuat instance dari `DataMahasiswa_2311104072` dan `KuliahMahasiswa_2311104072`.

- o Memanggil method `ReadJSON` dari kedua instance untuk mendapatkan data dari file JSON masing-masing.
- o Menampilkan data yang sudah dideserialisasi ke konsol sesuai format yang diminta dalam TP.

```
5. class Program
{
    static void Main(string[] args)
    {
        // Membaca data mahasiswa
        DataMahasiswa_2311104072 dataMahasiswa2311104072 = new
        DataMahasiswa_2311104072();
        var data =
        dataMahasiswa2311104072.ReadJSON("tp7_1_2311104072.json"); // Nama file
        sesuai TP

        // Menampilkan data mahasiswa
        Console.WriteLine($"Nama {data.nama.depan} {data.nama.belakang}
        " +
                                $"dengan nim {data.nim} " +
                                $"dari fakultas {data.fakultas}");

        Console.WriteLine(); // Memberi jarak antar output

        // Membaca data kuliah
        KuliahMahasiswa_2311104072 kuliahMahasiswa2311104072 = new
        KuliahMahasiswa_2311104072();
        var data1 =
        kuliahMahasiswa2311104072.ReadJSON("tp7_2_2311104072.json"); // Nama
        file sesuai TP

        // Menampilkan data kuliah
        Console.WriteLine("Daftar mata kuliah yang diambil:");
        int index = 1;
        foreach (var course in data1.courses) // Menggunakan nama
        properti 'courses' dari JSONFormat2
        {
            Console.WriteLine($"MK{index} {course.code}
            {course.name}");
            index++;
        }
    }
}
```

Penjelasan Snippet: Kode ini menunjukkan alur utama: membuat objek, memanggil method pembaca JSON, dan kemudian menggunakan `Console.WriteLine` dengan *string interpolation* (`$"{variable}"`) untuk mencetak output yang terformat. Looping `foreach` digunakan untuk menampilkan setiap mata kuliah dari list.

Bagian IV: Hasil Eksekusi (Contoh Tampilan Konsol)

Output program akan terlihat seperti berikut di konsol (nama, NIM, fakultas, dan mata kuliah akan sesuai dengan isi file JSON Anda):

```
Run  tpmodul7_2311104072  x
Nama jauhhar fajar dengan nim 2311104072 dari fakultas informatika
MK1 CCK2KAB4 Konstruksi Perangkat Lunak
MK2 CCK2HAB4 Basis Data
MK3 CCK2GAB3 Interaksi Manusia Komputer
MK4 CCK2JAC2 Proyek Tingkat II
MK5 CCK1KAB3 Rekayasa Kebutuhan Perangkat Lunak
MK6 CCK2CAB3 Pemodelan Perangkat Lunak
MK7 CCK2FAB4 Arsitektur Dan Desain Perangkat Lunak

Process finished with exit code 0.
```

Penjelasan Contoh Output: Baris pertama menampilkan informasi mahasiswa yang didapat dari deserialisasi `tp7_1_2311104072.json`. Baris berikutnya menampilkan daftar mata kuliah yang diambil, hasil deserialisasi dari `tp7_2_2311104072.json`, dengan format nomor urut (MK1, MK2, dst.), kode mata kuliah, dan nama mata kuliah.
