

Nama : Dimastian Aji Wibowo

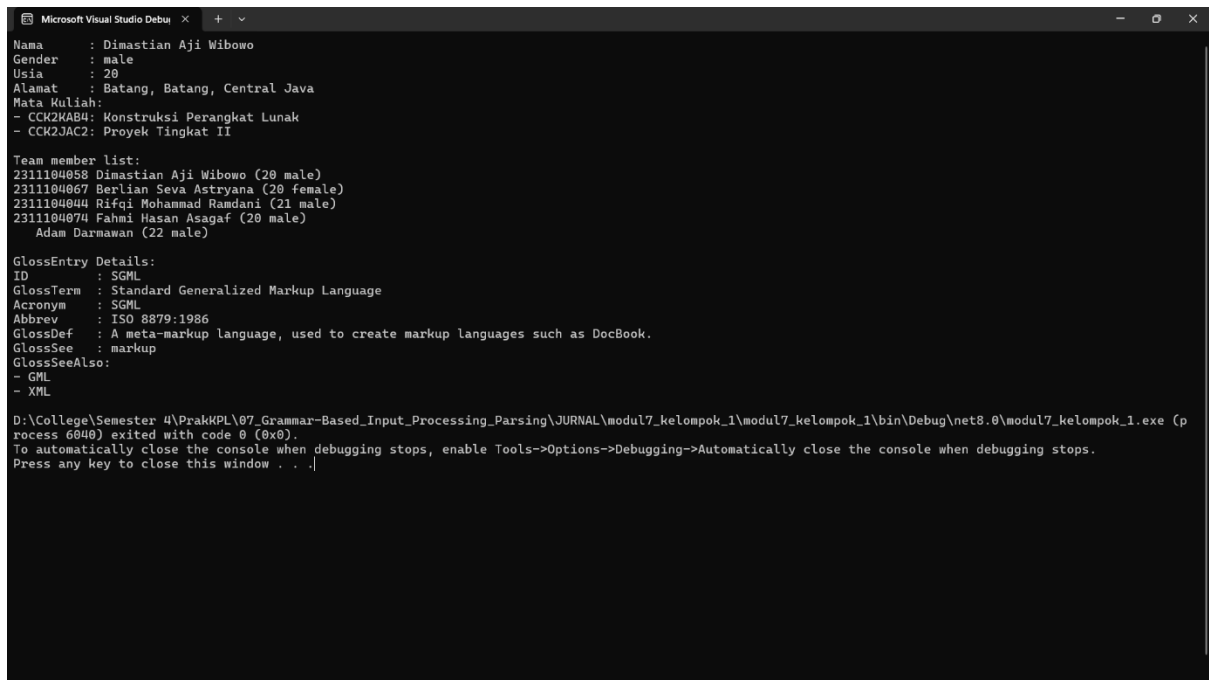
NIM : 2311104058

TP MODUL 7

Link Repository Github

https://github.com/dimastianaji/modul7_kelompok_1.git

Hasil Run



```
Microsoft Visual Studio Debug Console
Nama : Dimastian Aji Wibowo
Gender : male
Usia : 20
Alamat : Batang, Batang, Central Java
Mata Kuliah:
- CCK2KAB4: Konstruksi Perangkat Lunak
- CCK2JAC2: Proyek Tingkat II

Team member list:
2311104058 Dimastian Aji Wibowo (20 male)
2311104067 Berlian Seva Astryana (20 female)
2311104044 Rifqi Mohammad Ramdani (21 male)
2311104074 Fahmi Hasan Asagaf (20 male)
Adam Darmawan (22 male)

GlossEntry Details:
ID : SGML
GlossTerm : Standard Generalized Markup Language
Acronym : SGML
Abbrev : ISO 8879:1986
GlossDef : A meta-markup language, used to create markup languages such as DocBook.
GlossSee : markup
GlossSeeAlso:
- GML
- XML

D:\College\Semester 4\PrakKPL\07_Grammar-Based_Input_Processing_Parsing\JURNAL\modul7_kelompok_1\modul7_kelompok_1\bin\Debug\net8.0\modul7_kelompok_1.exe (process 6040) exited with code 0 (0x0).
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .
```

Source Code

DataMahasiswa2311104058

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.IO;
using System.Text.Json;
using System.Text.Json.Serialization;

namespace modul7_kelompok_1
{
    public class DataMahasiswa2311104058
    {
        public class Address
        {
            [JsonPropertyName("streetAddress")]
            public string StreetAddress { get; set; }

            [JsonPropertyName("city")]
            public string City { get; set; }
        }
    }
}
```

```

        [JsonPropertyName("state")]
        public string State { get; set; }
    }

    public class Course
    {
        [JsonPropertyName("code")]
        public string Code { get; set; }

        [JsonPropertyName("name")]
        public string Name { get; set; }
    }

    public class Mahasiswa
    {
        [JsonPropertyName("firstName")]
        public string FirstName { get; set; }

        [JsonPropertyName("lastName")]
        public string LastName { get; set; }

        [JsonPropertyName("gender")]
        public string Gender { get; set; }

        [JsonPropertyName("age")]
        public int Age { get; set; }

        [JsonPropertyName("address")]
        public Address Address { get; set; }

        [JsonPropertyName("courses")]
        public List<Course> Courses { get; set; }
    }

    public void ReadJSON()
    {
        string filePath = @"D:\College\Semester 4\PrakKPL\07_Grammar-
Based_Input_Processing_Parsing\JURNAL\modul7_kelompok_1\modul7_kelompok_1\jurnal7_1_2311104
058.json";

        string jsonString = File.ReadAllText(filePath);
        Mahasiswa mhs = JsonSerializer.Deserialize<Mahasiswa>(jsonString);

        Console.WriteLine($"Nama      : {mhs.FirstName} {mhs.LastName}");
        Console.WriteLine($"Gender    : {mhs.Gender}");
        Console.WriteLine($"Usia     : {mhs.Age}");

        if (mhs.Address != null)
        {
            Console.WriteLine($"Alamat      : {mhs.Address.StreetAddress},
{mhs.Address.City}, {mhs.Address.State}");
        }
        else
        {
            Console.WriteLine("Alamat      : (data tidak tersedia)");
        }

        Console.WriteLine("Mata Kuliah:");
        if (mhs.Courses != null)
        {
            foreach (var course in mhs.Courses)
            {
                Console.WriteLine($"- {course.Code}: {course.Name}");
            }
        }
    }
}

```

Kode di atas digunakan untuk membaca dan menampilkan data mahasiswa dari sebuah file JSON. Program ini terdiri dari beberapa kelas: Address untuk menyimpan data alamat, Course untuk menyimpan data mata kuliah, dan Mahasiswa sebagai representasi data mahasiswa secara keseluruhan. Properti-propri pada setiap kelas dihubungkan dengan nama-nama field dalam file JSON menggunakan atribut [JsonPropertyName]. Metode ReadJSON() berfungsi untuk membaca file JSON bernama *jurnal7_1_2311104058.json*, mendeserialisasi isi file tersebut menjadi objek Mahasiswa, dan kemudian menampilkan informasi mahasiswa tersebut ke konsol, termasuk nama, gender, usia, alamat, serta daftar mata kuliah yang diambil.

TeamMembers2311104058

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.IO;
using Newtonsoft.Json;

namespace modul7_kelompok_1
{
    public class TeamMembers2311104058
    {
        public class Member
        {
            public string firstName { get; set; }
            public string lastName { get; set; }
            public string gender { get; set; }
            public int age { get; set; }
            public string nim { get; set; }
        }

        public class MemberList
        {
            public List<Member> members { get; set; }
        }

        public static void ReadJSON()
        {
            string filePath = "jurnal7_2_2311104058.json";
            if (!File.Exists(filePath))
            {
                Console.WriteLine("File tidak ditemukan.");
                return;
            }

            string jsonContent = File.ReadAllText(filePath);
            MemberList team = JsonConvert.DeserializeObject<MemberList>(jsonContent);

            Console.WriteLine("Team member list:");
            foreach (var member in team.members)
            {
                Console.WriteLine($"{member.nim} {member.firstName} {member.lastName}
({member.age} {member.gender})");
            }
        }
    }
}
```

Kode di atas menggunakan library Newtonsoft.Json untuk membaca dan menampilkan data anggota tim dari sebuah file JSON. Terdapat dua kelas: Member untuk merepresentasikan satu anggota tim, dan MemberList yang berisi daftar anggota dalam bentuk list. Metode ReadJSON() akan membaca isi file JSON bernama *jurnal7_2_2311104058.json*, kemudian mendeserialisasi isi file tersebut menjadi objek MemberList. Setelah itu, data setiap anggota tim ditampilkan ke konsol, termasuk NIM, nama lengkap, usia, dan jenis kelamin. Jika file tidak ditemukan, program akan menampilkan pesan kesalahan.

GlossaryItem2311104058

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

namespace modul7_kelompok_1
{
    public class GlossaryItem2311104058
    {
        public class Glossary
        {
            public string title { get; set; }
            public GlossDiv GlossDiv { get; set; }
        }

        public class GlossDiv
        {
            public string title { get; set; }
        }
    }
}
```

```

    public GlossList GlossList { get; set; }
}

public class GlossList
{
    public GlossEntry GlossEntry { get; set; }
}

public class GlossEntry
{
    public string ID { get; set; }
    public string SortAS { get; set; }
    public string GlossTerm { get; set; }
    public string Acronym { get; set; }
    public string Abbrev { get; set; }
    public GlossDef GlossDef { get; set; }
    public string GlossSee { get; set; }
}

public class GlossDef
{
    public string para { get; set; }
    public List<string> GlossSeeAlso { get; set; }
}

public class Root
{
    public Glossary glossary { get; set; }
}

public static void ReadJSON()
{
    string filePath = @"D:\College\Semester 4\PrakKPL\07_Grammar-
Based_Input_Processing_Parsing\JURNAL\modul7_kelompok_1\modul7_kelompok_1\jurnal7_3_2311104
058.json";

    if (!File.Exists(filePath))
    {
        Console.WriteLine("File tidak ditemukan.");
        return;
    }

    string jsonContent = File.ReadAllText(filePath);
    Root data = JsonConvert.DeserializeObject<Root>(jsonContent);

    var entry = data.glossary.GlossDiv.GlossList.GlossEntry;

    Console.WriteLine("GlossEntry Details:");
    Console.WriteLine($"ID : {entry.ID}");
    Console.WriteLine($"GlossTerm : {entry.GlossTerm}");
    Console.WriteLine($"Acronym : {entry.Acronym}");
    Console.WriteLine($"Abbrev : {entry.Abbrev}");
    Console.WriteLine($"GlossDef : {entry.GlossDef.para}");
    Console.WriteLine($"GlossSee : {entry.GlossSee}");
    Console.WriteLine("GlossSeeAlso:");
    foreach (var item in entry.GlossDef.GlossSeeAlso)
    {
        Console.WriteLine($"- {item}");
    }
}
}

```

Kode ini digunakan untuk membaca dan menampilkan data glossary (daftar istilah) dari sebuah file JSON menggunakan library Newtonsoft.Json. Struktur data JSON yang kompleks dipecah menjadi beberapa kelas: Glossary, GlossDiv, GlossList, GlossEntry, dan GlossDef, semuanya dibungkus oleh kelas utama Root. Fungsi ReadJSON() membaca file JSON bernama *jurnal7_3_2311104058.json*, mendeserialisasi isinya menjadi objek Root, lalu mengekstrak dan menampilkan informasi glossary seperti ID istilah, istilah utama (GlossTerm), akronim, singkatan, definisi, dan istilah-istilah terkait (GlossSeeAlso) ke konsol. Jika file tidak ditemukan, maka program akan memberi pesan bahwa file tidak tersedia.