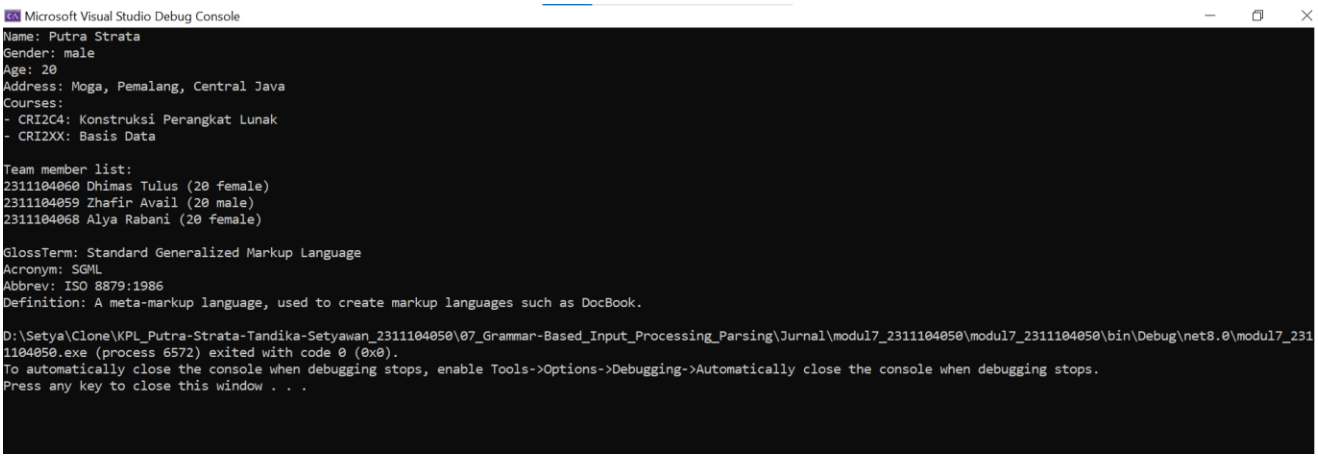
Putra Strata Tandika Setyawan

2311104050

S1SE0702

Jurnal Modul 7



1. Membuat Class DataMahasiswa

```
using System;
using System.Text.Json;
using System.Collections.Generic;
using System.IO;

public class DataMahasiswa2311104050
{
    public string firstName { get; set; }
    public string lastName { get; set; }
    public string gender { get; set; }
    public int age { get; set; }
    public Address address { get; set; }
    public List<Course> courses { get; set; }

    public static void ReadJSON()
    {
        try
        {
            // Pastikan file ada sebelum membaca
            if (!File.Exists("jurnal7_1_2311104050.json"))
            {
                Console.WriteLine("File JSON tidak ditemukan!");
                return;
            }

            var json = File.ReadAllText("jurnal7_1_2311104050.json");

            // Tambahkan options untuk handle case-insensitive dan lainnya
            var options = new JsonSerializerOptions
            {
                PropertyNameCaseInsensitive = true,
                AllowTrailingCommas = true,
```

```
                ReadCommentHandling = JsonCommentHandling.Skip
            };

            var data =
JsonSerializer.Deserialize<DataMahasiswa2311104050>(json, options);

            // Null check untuk data dan propertinya
            if (data == null)
            {
                Console.WriteLine("Data tidak valid atau struktur JSON
tidak sesuai");
                return;
            }

            Console.WriteLine($"Name: {data.firstName ?? "N/A"}
{data.lastName ?? "N/A"}");
            Console.WriteLine($"Gender: {data.gender ?? "N/A"}");
            Console.WriteLine($"Age: {data.age}");

            // Null check untuk address
            if (data.address != null)
            {
                Console.WriteLine($"Address: {data.address.streetAddress ??
"N/A"}, " +
                                  $"{data.address.city ?? "N/A"}, " +
                                  $"{data.address.state ?? "N/A"}");
            }
            else
            {
                Console.WriteLine("Address: N/A");
            }

            // Null check untuk courses
            Console.WriteLine("Courses:");
            if (data.courses != null)
            {
                foreach (var c in data.courses)
                {
                    if (c != null)
                    {
                        Console.WriteLine($"- {c.code ?? "N/A"}: {c.name ??
"N/A"}");
                    }
                }
            }
            else
            {
                Console.WriteLine("- No courses available");
            }
        }
        catch (JsonException jsonEx)
        {
            Console.WriteLine($"Error parsing JSON: {jsonEx.Message}");
        }
        catch (Exception ex)
        {
            Console.WriteLine($"Error: {ex.Message}");
        }
```

```
    }
}

public class Address
{
    public string streetAddress { get; set; }
    public string city { get; set; }
    public string state { get; set; }
}

public class Course
{
    public string code { get; set; }
    public string name { get; set; }
}
```

2. Membuat Class TeamMahasiswa

```
using System;
using System.Text.Json;
using System.Collections.Generic;
using System.IO;

public class TeamMembers2311104050
{
    public List<Member> members { get; set; }

    public static void ReadJSON()
    {
        try
        {
            // Pastikan file ada sebelum membaca
            if (!File.Exists("jurnal7_2_2311104050.json"))
            {
                Console.WriteLine("File JSON tidak ditemukan!");
                return;
            }

            var json = File.ReadAllText("jurnal7_2_2311104050.json");

            // Tambahkan options untuk handle case-insensitive dan lainnya
            var options = new JsonSerializerOptions
            {
                PropertyNameCaseInsensitive = true,
                AllowTrailingCommas = true,
                ReadCommentHandling = JsonCommentHandling.Skip
            };

            var data =
JsonSerializer.Deserialize<TeamMembers2311104050>(json, options);

            // Null check untuk data dan members
            if (data?.members == null)
            {
                Console.WriteLine("Data tidak valid atau struktur JSON
tidak sesuai");
                return;
            }
```

```
            Console.WriteLine("Team member list:");
            foreach (var m in data.members)
            {
                // Null check untuk setiap member dan propertinya
                Console.WriteLine($"{m.nim ?? "N/A"} {m.firstName ?? "N/A"}
{m.lastName ?? "N/A"} ({m.age} {m.gender ?? "N/A"})");
            }
        }
        catch (JsonException jsonEx)
        {
            Console.WriteLine($"Error parsing JSON: {jsonEx.Message}");
        }
        catch (Exception ex)
        {
            Console.WriteLine($"Error: {ex.Message}");
        }
    }
}

public class Member
{
    public string firstName { get; set; }
    public string lastName { get; set; }
    public string gender { get; set; }
    public string nim { get; set; }
    public int age { get; set; }
}
```

3. Membuat Class GlossaryItem

```
S
using System.Text.Json;

public class GlossaryItem2311104050
{
    public Glossary glossary { get; set; }

    public static void ReadJSON()
    {
        try
        {
            var json = File.ReadAllText("jurnal7_3_2311104050.json");
            var data =
JsonSerializer.Deserialize<GlossaryItem2311104050>(json);

            // Akses properti dengan cara yang benar
            if (data?.glossary?.GlossDiv?.GlossList?.GlossEntry != null)
            {
                GlossEntry entry =
data.glossary.GlossDiv.GlossList.GlossEntry;
                Console.WriteLine($"GlossTerm: {entry.GlossTerm}");
                Console.WriteLine($"Acronym: {entry.Acronym}");
                Console.WriteLine($"Abbrev: {entry.Abbrev}");
                Console.WriteLine($"Definition: {entry.GlossDef?.para}");
            }
            else
```

```
            {
                Console.WriteLine("Data tidak lengkap atau tidak sesuai
struktur");
            }
        }
        catch (Exception ex)
        {
            Console.WriteLine($"Error: {ex.Message}");
        }
    }
}

public class Glossary
{
    public GlossDiv GlossDiv { get; set; }
    public string title { 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 string GlossSee { get; set; }
    public GlossDef GlossDef { get; set; }
}

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

4. Menggabungkan semua class di program.cs

```
class Program
{
    static void Main(string[] args)
    {
        DataMahasiswa2311104050.ReadJSON();

        Console.WriteLine();

        TeamMembers2311104050.ReadJSON();
```

```
        Console.WriteLine();

        GlossaryItem2311104050.ReadJSON();
    }
}
```