

TUGAS JURNAL 7 KONTRUKSI PERANGKAT LUNAK

Nama: Rifqi M Ramdani

NIM : 2311104044

Kelas: SE-07-02

Link Github Kelompok: https://github.com/dimastianaji/modul7_kelompok_1.git

1.

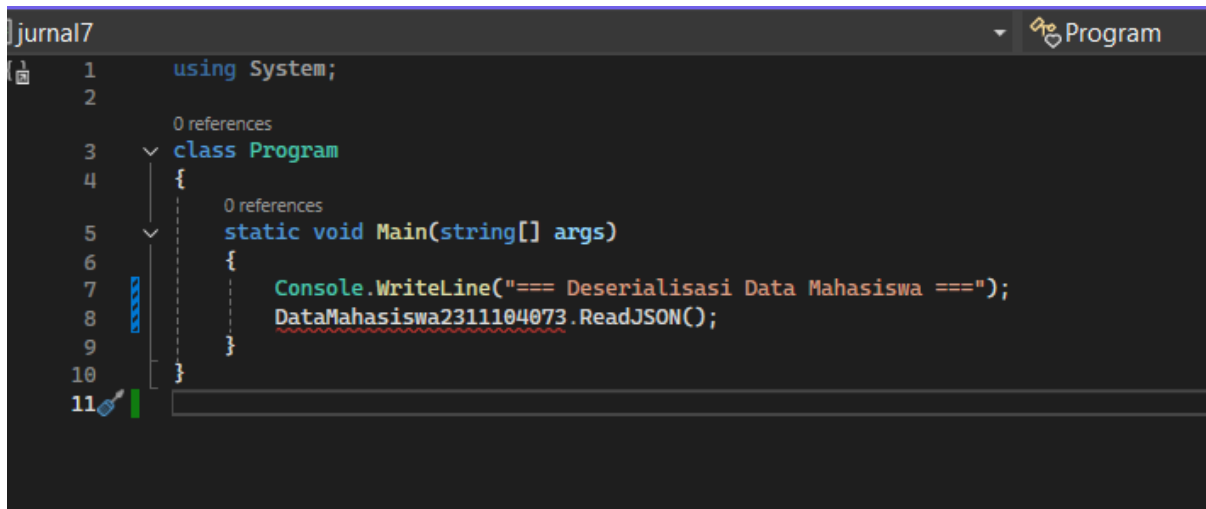
DataMahasiswa2311104044.cs



```
jurnal7
1  using System;
2  using System.IO;
3  using System.Text.Json;
4
5  0 references
6  public class DataMahasiswa2311104044
7  {
8      0 references
9      public static void ReadJSON()
10     {
11         string fileName = "jurnal7_1_2311104044.json";
12
13         if (!File.Exists(fileName))
14         {
15             Console.WriteLine("File tidak ditemukan.");
16             return;
17         }
18
19         string jsonString = File.ReadAllText(fileName);
20         var data = JsonSerializer.Deserialize<Mahasiswa>(jsonString);
21
22         Console.WriteLine("Data Mahasiswa:");
23         Console.WriteLine($"Nama : {data.Nama}");
24         Console.WriteLine($"NIM : {data.NIM}");
25         Console.WriteLine($"Umur : {data.Umur}");
26         Console.WriteLine($"Kelas : {data.Kelas}");
27         Console.WriteLine($"Asal : {data.Asal}");
28     }
29
30     1 reference
31     public class Mahasiswa
32     {
33         1 reference
34         public string Nama { get; set; }
35         1 reference
36         public string NIM { get; set; }
37         1 reference
38         public int Umur { get; set; }
39         1 reference
40         public string Kelas { get; set; }
41         1 reference
42         public string Asal { get; set; }
43     }
```

berisi kelas DataMahasiswa2311104044 yang memiliki satu metode statis ReadJSON() untuk membaca dan mendeserialisasi file JSON bernama jurnal7_1_2311104044.json. Metode ini pertama-tama memeriksa apakah file tersebut ada, lalu membaca seluruh isi file dan mengubah data JSON menjadi objek dari kelas Mahasiswa. Setelah berhasil dideserialisasi, program akan menampilkan data mahasiswa seperti nama, NIM, umur, kelas,

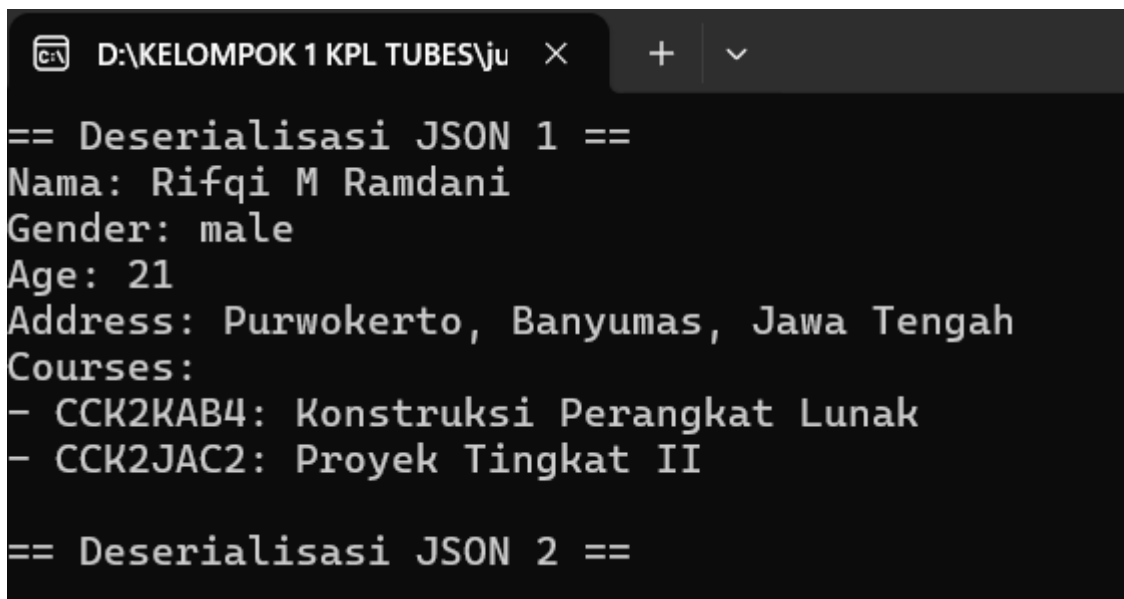
dan asal. Kelas Mahasiswa sendiri merupakan representasi struktur data yang memuat properti-properti tersebut, dan digunakan untuk menyesuaikan struktur JSON yang dibaca.



```
1 using System;
2
3 class Program
4 {
5     static void Main(string[] args)
6     {
7         Console.WriteLine("=== Deserialisasi Data Mahasiswa ===");
8         DataMahasiswa2311104073.ReadJSON();
9     }
10 }
11
```

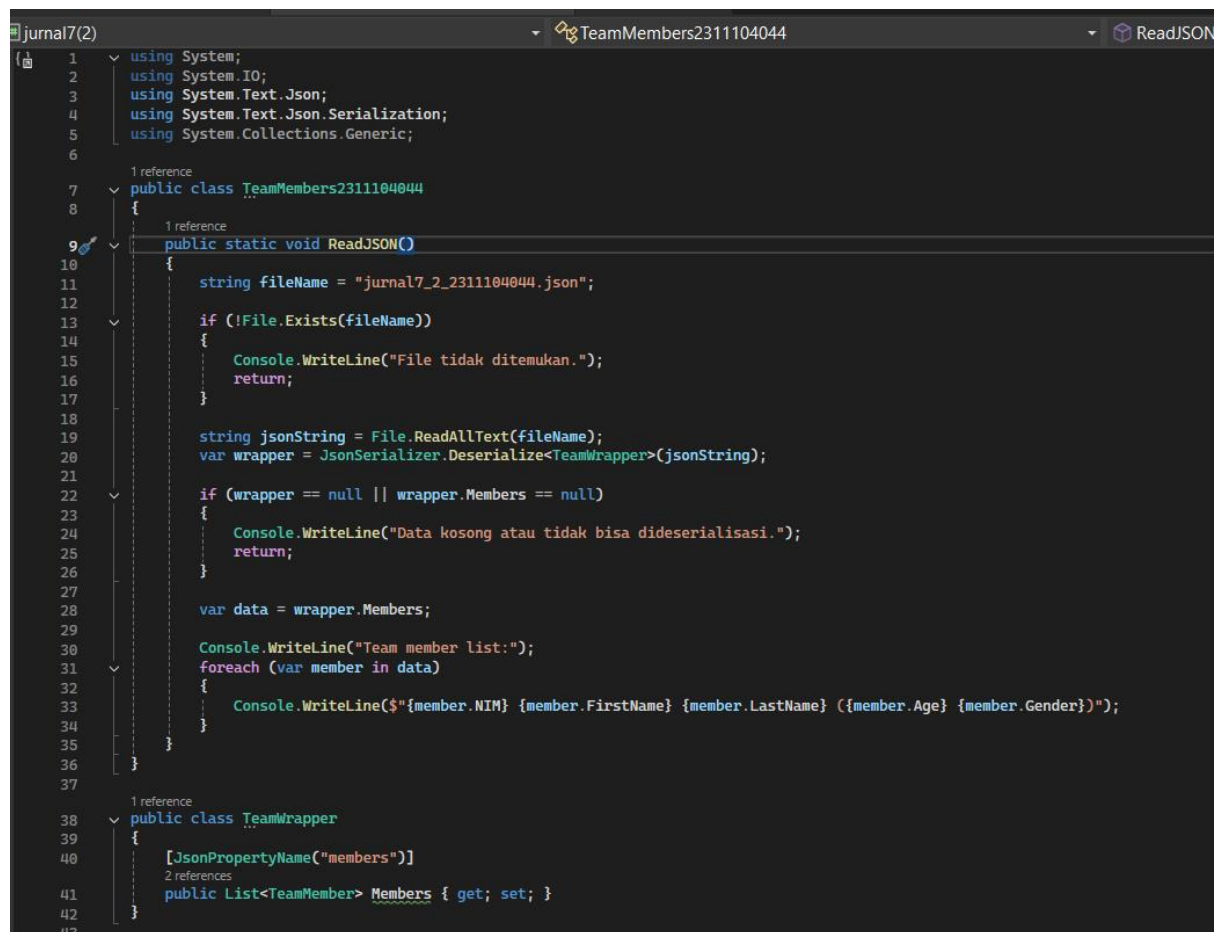
ini merupakan titik masuk (entry point) dari program C#. Di dalamnya terdapat kelas Program dengan metode Main, yang akan dieksekusi pertama kali saat program dijalankan. Di dalam metode Main, akan dicetak judul ke konsol, lalu dipanggil metode ReadJSON() dari kelas DataMahasiswa2311104073 (yang seharusnya disesuaikan dengan nama kelas yang benar, yaitu DataMahasiswa2311104044, agar program dapat berjalan tanpa error). File ini berfungsi sebagai penghubung utama antara pengguna dengan fungsi pembacaan dan penampilan data mahasiswa dari file JSON.

Output



```
D:\KELOMPOK 1 KPL TUBES\ju >
== Deserialisasi JSON 1 ==
Nama: Rifqi M Ramdani
Gender: male
Age: 21
Address: Purwokerto, Banyumas, Jawa Tengah
Courses:
- CCK2KAB4: Konstruksi Perangkat Lunak
- CCK2JAC2: Proyek Tingkat II
== Deserialisasi JSON 2 ==
```

2. TeamMembers2311104044.cs



```
1 using System;
2 using System.IO;
3 using System.Text.Json;
4 using System.Text.Json.Serialization;
5 using System.Collections.Generic;
6
7 public class TeamMembers2311104044
8 {
9     public static void ReadJSON()
10     {
11         string fileName = "jurnal7_2_2311104044.json";
12
13         if (!File.Exists(fileName))
14         {
15             Console.WriteLine("File tidak ditemukan.");
16             return;
17         }
18
19         string jsonString = File.ReadAllText(fileName);
20         var wrapper = JsonSerializer.Deserialize<TeamWrapper>(jsonString);
21
22         if (wrapper == null || wrapper.Members == null)
23         {
24             Console.WriteLine("Data kosong atau tidak bisa dideserialisasi.");
25             return;
26         }
27
28         var data = wrapper.Members;
29
30         Console.WriteLine("Team member list:");
31         foreach (var member in data)
32         {
33             Console.WriteLine($"{member.NIM} {member.FirstName} {member.LastName} ({member.Age} {member.Gender})");
34         }
35     }
36 }
37
38 public class TeamWrapper
39 {
40     [JsonPropertyName("members")]
41     public List<TeamMember> Members { get; set; }
42 }
43
```

File ini berisi logika utama untuk membaca dan menampilkan data dari file JSON yang berisi daftar anggota tim. Di dalamnya terdapat kelas TeamMembers2311104044 dengan metode ReadJSON(), yang akan memeriksa keberadaan file jurnal7_2_2311104044.json, membaca isinya, lalu melakukan deserialisasi menggunakan System.Text.Json. Karena struktur JSON memiliki properti induk "members", digunakan kelas pembungkus TeamWrapper yang berisi list Members bertipe List<TeamMember>. Deserialisasi dilakukan ke objek TeamWrapper, dan jika berhasil, program akan mencetak setiap anggota tim ke konsol menggunakan foreach. Selain itu, properti di kelas TeamMember dilengkapi dengan atribut [JsonPropertyName] agar mapping JSON tetap sesuai meskipun nama propertinya menggunakan gaya penulisan berbeda (camelCase di JSON dan PascalCase di C#).

Program.cs

```
1 using System;
2
3 class Program
4 {
5     static void Main(string[] args)
6     {
7         Console.WriteLine("\n=== Deserialisasi Team Members ===");
8         TeamMembers2311104044.ReadJSON();
9     }
10 }
11
```

entry point dari aplikasi C#, yaitu tempat eksekusi program dimulai. Di dalamnya terdapat kelas Program dan metode statis Main(string[] args), yang akan mencetak judul ke layar dan memanggil TeamMembers2311104044.ReadJSON() untuk menjalankan proses pembacaan dan penampilan data dari file JSON. File ini berperan sebagai penggerak awal program dan menjadi penghubung antara pengguna dan logika deserialisasi yang ada di file utama.

Output:

```
Microsoft Visual Studio Debug Console
=== Deserialisasi Team Members ===
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)

D:\KELOMPOK 1 KPL TUBES\jurnal7(2)\jurnal7(2)\bin\Debug\net8.0\jurnal7(2).exe (process 20340) 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 . . .|
```

3. GlossaryItem2311104044.cs

```
jurnal7_3_2311104044.json  GlossaryItem2311104044.cs  Program.cs
C# jurnal7(3)  GlossList

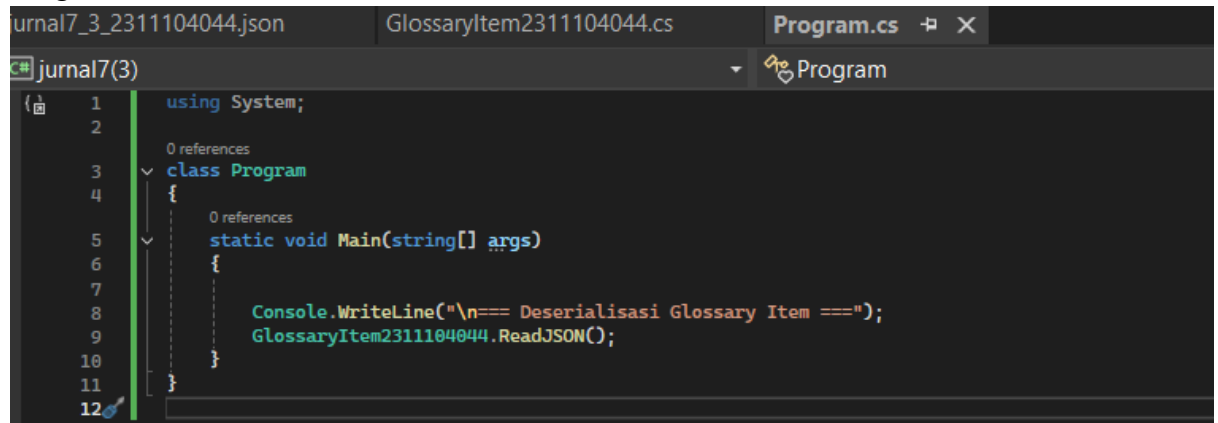
1  using System;
2  using System.IO;
3  using System.Text.Json;
4
5  1 reference
   public class GlossaryItem2311104044
6  {
7      1 reference
       public static void ReadJSON()
8      {
9          string fileName = "jurnal7_3_2311104044.json";
10
11         if (!File.Exists(fileName))
12         {
13             Console.WriteLine("File tidak ditemukan.");
14             return;
15         }
16
17         string jsonString = File.ReadAllText(fileName);
18         var wrapper = JsonSerializer.Deserialize<GlossaryWrapper>(jsonString);
19
20         if (wrapper == null || wrapper.glossary == null ||
21             wrapper.glossary.GlossDiv == null ||
22             wrapper.glossary.GlossDiv.GlossList == null ||
23             wrapper.glossary.GlossDiv.GlossList.GlossEntry == null)
24         {
25             Console.WriteLine("Data glossary tidak lengkap.");
26             return;
27         }
28
29         var entry = wrapper.glossary.GlossDiv.GlossList.GlossEntry;
30
31         Console.WriteLine("Glossary Entry:");
32         Console.WriteLine($"ID           : {entry.ID}");
33         Console.WriteLine($"SortAs      : {entry.SortAs}");
34         Console.WriteLine($"GlossTerm   : {entry.GlossTerm}");
35         Console.WriteLine($"Acronym     : {entry.Acronym}");
36         Console.WriteLine($"Abbrev      : {entry.Abbrev}");
37         Console.WriteLine($"Title       : {entry.Title}");
38         Console.WriteLine($"GlossSee    : {entry.GlossSee}");
39     }
40 }
41
42 // Wrapper untuk bagian "glossary"
43 1 reference
   public class GlossaryWrapper
44 {
45     5 references
       public Glossary glossary { get; set; }
46 }
47
```

```
jurnal7_3_2311104044.json  GlossaryItem2311104044.cs  Program.cs
C# jurnal7(3)  GlossList

45  public Glossary glossary { get; set; }
46  }
47
48  // Kelas-kelas turunan
49  1 reference
50  public class Glossary
51  {
52      4 references
53      public GlossDiv GlossDiv { get; set; }
54  }
55
56  1 reference
57  public class GlossDiv
58  {
59      3 references
60      public GlossList GlossList { get; set; }
61  }
62
63  1 reference
64  public class GlossList
65  {
66      2 references
67      public GlossEntry GlossEntry { get; set; }
68  }
69
70  1 reference
71  public class GlossEntry
72  {
73      1 reference
74      public string ID { get; set; }
75      1 reference
76      public string SortAs { get; set; }
77      1 reference
78      public string GlossTerm { get; set; }
79      1 reference
80      public string Acronym { get; set; }
81      1 reference
82      public string Abbrev { get; set; }
83      1 reference
84      public string Title { get; set; }
85      1 reference
86      public string GlossSee { get; set; }
87  }
```

logika utama untuk membaca dan menampilkan data dari file JSON yang berisi informasi glossary. Di dalamnya terdapat kelas GlossaryItem2311104044 dengan metode ReadJSON() yang bertugas memeriksa apakah file jurnal7_3_2311104044.json tersedia, membaca isinya, lalu mendeserialisasi isi file ke dalam objek bertipe GlossaryWrapper, karena struktur JSON memiliki properti root bernama "glossary". Proses deserialisasi dilakukan secara berlapis, dari GlossaryWrapper, ke Glossary, kemudian ke GlossDiv, GlossList, dan akhirnya GlossEntry yang berisi detail informasi glossary seperti ID, term, acronym, abbreviation, dan sebagainya. Program kemudian menampilkan informasi tersebut ke konsol. Kelas-kelas seperti Glossary, GlossDiv, GlossList, dan GlossEntry dibuat sesuai struktur bertingkat pada JSON agar proses deserialisasi berjalan lancar dan sesuai.

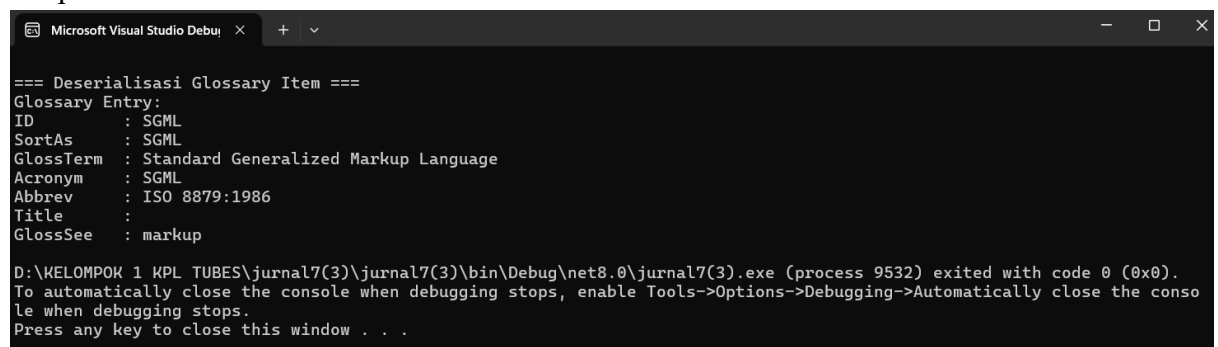
Program.cs

A screenshot of a Visual Studio code editor showing the Program.cs file. The code is in C# and defines a class Program with a static Main method. The Main method calls Console.WriteLine to print a separator and then calls GlossaryItem2311104044.ReadJSON(). The code is as follows:

```
1 using System;
2
3 class Program
4 {
5     static void Main(string[] args)
6     {
7
8         Console.WriteLine("\n=== Deserialisasi Glossary Item ===");
9         GlossaryItem2311104044.ReadJSON();
10    }
11 }
12
```

titik masuk (entry point) program yang berisi kelas Program dengan metode Main, yang akan dijalankan pertama kali ketika program dijalankan. Dalam metode Main, program mencetak judul ke layar, lalu memanggil metode ReadJSON() dari kelas GlossaryItem2311104044 untuk memulai proses pembacaan dan penampilan data glossary dari file JSON. File ini berfungsi sebagai penggerak awal program yang mengarahkan eksekusi ke logika utama deserialisasi.

Output

A screenshot of the Microsoft Visual Studio Debug Console window. It shows the output of the program, which is a formatted display of a glossary entry for SGML. The output is as follows:

```
=== Deserialisasi Glossary Item ===
Glossary Entry:
ID       : SGML
SortAs   : SGML
GlossTerm : Standard Generalized Markup Language
Acronym  : SGML
Abbrev   : ISO 8879:1986
Title    :
GlossSee  : markup

D:\KELOMPOK 1 KPL TUBES\jurnal7(3)\jurnal7(3)\bin\Debug\net8.0\jurnal7(3).exe (process 9532) 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 . . .
```