

Nama : Farhan Kurniawan

NIM : 2311104073

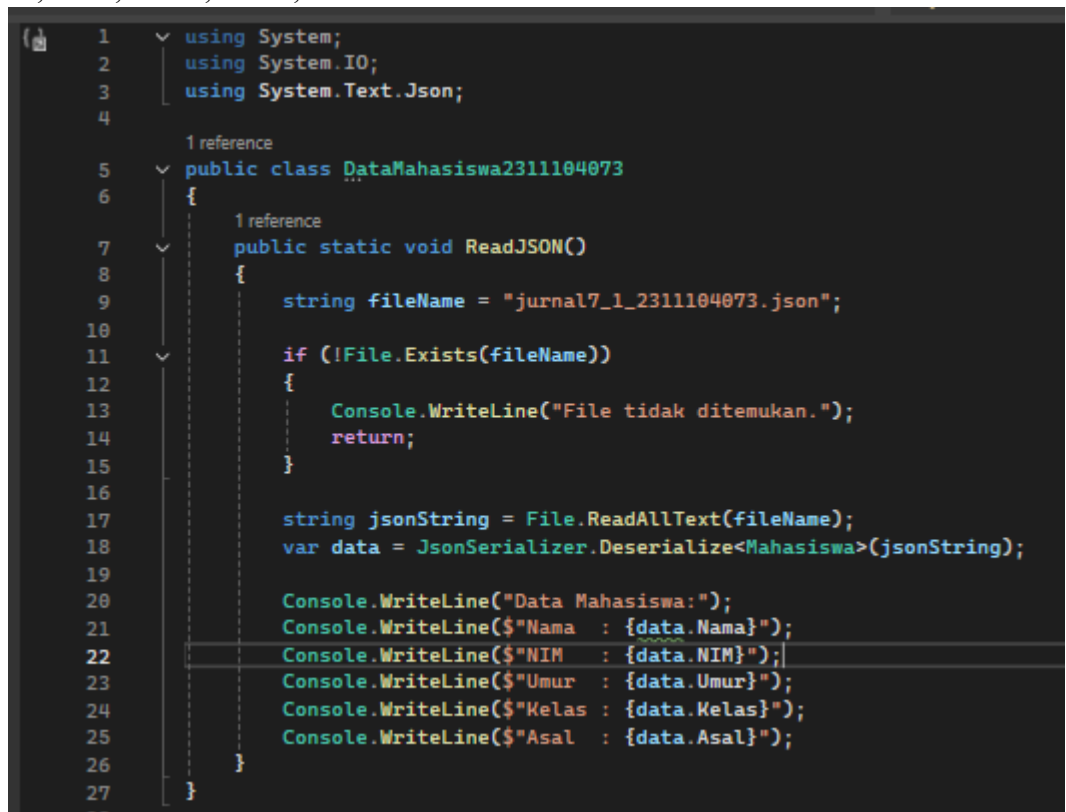
Kelas : SE-07-02

Link Github Kelompok: https://github.com/tiurmagrace/modul7_kelompok_4.git

1. Branch farhan1

a. DataMahasiswa2311104073.cs

File ini berisi dua kelas, yaitu DataMahasiswa2311104073 dan Mahasiswa. Kelas DataMahasiswa2311104073 memiliki method statis ReadJSON() yang berfungsi untuk membaca data dari file JSON bernama jurnal7_1_2311104073.json. Method ini terlebih dahulu memeriksa apakah file tersebut ada, kemudian membaca seluruh isi file, dan mendeserialisasi data JSON tersebut ke dalam objek Mahasiswa. Setelah berhasil, data mahasiswa seperti nama, NIM, umur, kelas, dan asal akan ditampilkan ke konsol. Kelas Mahasiswa sendiri merupakan representasi dari struktur data mahasiswa, yang memiliki properti Nama, NIM, Umur, Kelas, dan Asal.



```
1  using System;
2  using System.IO;
3  using System.Text.Json;
4
5  1 reference
6  public class DataMahasiswa2311104073
7  {
8      1 reference
9      public static void ReadJSON()
10     {
11         string fileName = "jurnal7_1_2311104073.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 }
```

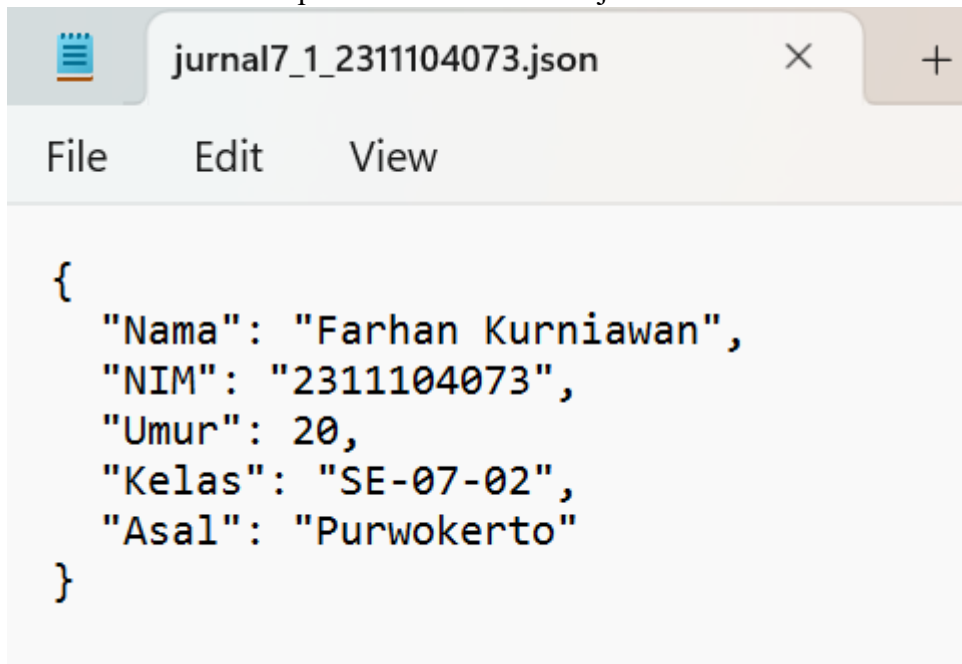
```

1 reference
29  public class Mahasiswa
30  {
31      1 reference
      public string Nama { get; set; }
32      1 reference
      public string NIM { get; set; }
33      1 reference
      public int Umur { get; set; }
34      1 reference
      public string Kelas { get; set; }
35      1 reference
      public string Asal { get; set; }
36  }

```

b. jurnal7_1_2311104073.json

File ini merupakan file eksternal berformat JSON yang berisi data statis mengenai seorang mahasiswa. Struktur data di dalamnya terdiri dari lima pasangan key-value yaitu Nama, NIM, Umur, Kelas, dan Asal, dengan masing-masing berisi nilai yang sesuai, seperti "Farhan Kurniawan" dan "2311104073". File ini digunakan sebagai sumber data yang akan dibaca oleh program melalui proses deserialisasi dan ditampilkan dalam bentuk objek C#.



```

{
  "Nama": "Farhan Kurniawan",
  "NIM": "2311104073",
  "Umur": 20,
  "Kelas": "SE-07-02",
  "Asal": "Purwokerto"
}

```

c. Program.cs

File ini berperan sebagai titik awal eksekusi program. Di dalamnya terdapat method Main() yang menampilkan judul "=== Deserialisasi Data Mahasiswa ===" ke konsol, lalu memanggil method ReadJSON() dari kelas DataMahasiswa2311104073. Tujuannya adalah untuk memulai proses pembacaan dan penampilan data mahasiswa dari file JSON yang sudah disiapkan. File ini bertugas menghubungkan alur eksekusi program dengan proses deserialisasi yang telah didefinisikan dalam file lain.

```

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 }

```

Output:

```

Microsoft Visual Studio Debug Console
=== Deserialisasi Data Mahasiswa ===
Data Mahasiswa:
Nama : Farhan Kurniawan
NIM : 2311104073
Umur : 20
Kelas : SE-07-02
Asal : Purwokerto

D:\Semester4\KPL\JM7\JM7\bin\Debug\net8.0\JM7.exe (process 24248) exited with code 0 (0x0).
To automatically close the console when debugging stops, enable Tools->Options->Debugging->
Close console when debugging stops.
Press any key to close this window . . .

```

2. Branch farhan2

a. TeamMembers2311104073.cs

File ini berisi dua class, yaitu TeamMembers2311104073 dan TeamMember. Class TeamMembers2311104073 memiliki method statis ReadJSON() yang bertugas membaca file JSON bernama jurnal7_2_2311104073.json, kemudian mendeserialisasi isi file tersebut menjadi list dari objek TeamMember menggunakan System.Text.Json. Setelah proses deserialisasi berhasil, program akan menampilkan informasi setiap anggota tim, seperti NIM, nama lengkap, usia, dan jenis kelamin ke console. Sementara itu, class TeamMember digunakan sebagai blueprint objek untuk menyimpan data dari masing-masing anggota tim dengan properti seperti NIM, FirstName, LastName, Age, dan Gender.

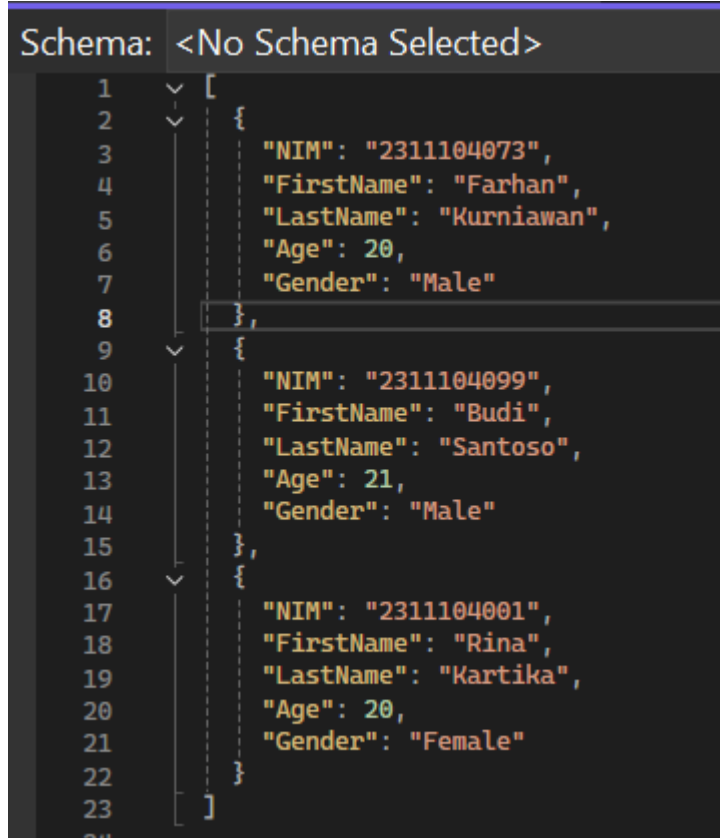
```

C# Miscellaneous Files TeamMembers2311104073
1  using System;
2  using System.IO;
3  using System.Text.Json;
4  using System.Collections.Generic;
5
6  public class TeamMembers2311104073
7  {
8      public static void ReadJSON()
9      {
10         string fileName = "jurnal7_2_2311104073.json";
11
12         if (!File.Exists(fileName))
13         {
14             Console.WriteLine("File tidak ditemukan.");
15             return;
16         }
17
18         string jsonString = File.ReadAllText(fileName);
19         var data = JsonSerializer.Deserialize<List<TeamMember>>(jsonString);
20
21         Console.WriteLine("Team member list:");
22         foreach (var member in data)
23         {
24             Console.WriteLine($"{member.NIM} {member.FirstName} {member.LastName} ({member.Age} {member.Gender})");
25         }
26     }
27 }

```

b. jurnal7_2_2311104073.json

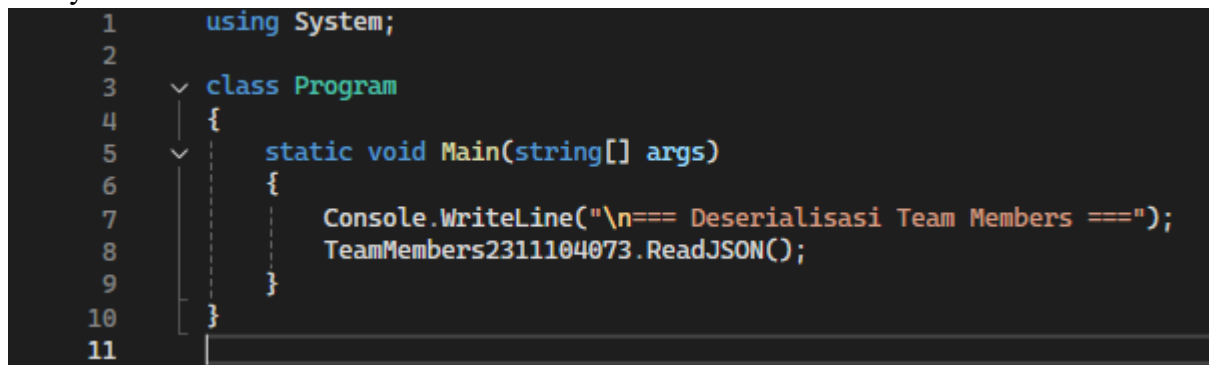
File ini merupakan file eksternal berformat JSON yang berisi data tiga orang anggota tim dalam bentuk array objek. Setiap objek merepresentasikan satu anggota tim dengan lima atribut: NIM, FirstName, LastName, Age, dan Gender. Data ini nantinya akan dibaca oleh program C# dan dikonversi menjadi objek-objek TeamMember yang ditampilkan dalam bentuk teks di console. File ini menjadi sumber utama data yang dibutuhkan oleh program dalam proses deserialisasi.



```
Schema: <No Schema Selected>
1  [
2  {
3      "NIM": "2311104073",
4      "FirstName": "Farhan",
5      "LastName": "Kurniawan",
6      "Age": 20,
7      "Gender": "Male"
8  },
9  {
10     "NIM": "2311104099",
11     "FirstName": "Budi",
12     "LastName": "Santoso",
13     "Age": 21,
14     "Gender": "Male"
15  },
16  {
17     "NIM": "2311104001",
18     "FirstName": "Rina",
19     "LastName": "Kartika",
20     "Age": 20,
21     "Gender": "Female"
22  }
23 ]
```

c. Program.cs

File ini berisi class utama bernama Program yang memiliki method Main() sebagai titik masuk utama aplikasi. Di dalam method ini, program pertama-tama mencetak judul ke layar yaitu “=== Deserialisasi Team Members ===” untuk memberi tahu pengguna bahwa proses deserialisasi akan dimulai. Setelah itu, method ReadJSON() dari class TeamMembers2311104073 dipanggil untuk menjalankan seluruh proses pembacaan file JSON dan penampilan data anggota tim ke layar.



```
1  using System;
2
3  class Program
4  {
5      static void Main(string[] args)
6      {
7          Console.WriteLine("\n=== Deserialisasi Team Members ===");
8          TeamMembers2311104073.ReadJSON();
9      }
10 }
11
```

Output:

```
Microsoft Visual Studio Debug Console

=== Deserialisasi Team Members ===
Team member list:
2311104073 Farhan Kurniawan (20 Male)
2311104099 Budi Santoso (21 Male)
2311104001 Rina Kartika (20 Female)

D:\jurnal7_2_2311104073\jurnal7_2_2311104073\bin\Debug\net8.0\jurnal7_2_2311104073.exe
0 (0x0).
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Close console when debugging stops.
Press any key to close this window . . .|
```

3. Branch farhan3

a. GlossaryItem2311104073.cs

Berisi kelas utama GlossaryItem2311104073 yang memiliki method statis ReadJSON() untuk melakukan proses deserialisasi file JSON bernama jurnal7_3_2311104073.json. Di dalam method tersebut, program akan memeriksa apakah file JSON tersedia, membaca seluruh isinya menjadi string, lalu mengubahnya menjadi objek C# menggunakan JsonSerializer.Deserialize. Data yang sudah dideserialisasi akan ditampilkan ke konsol, khususnya bagian GlossEntry yang berisi informasi seperti ID, istilah, akronim, singkatan, judul, dan referensi istilah lain. Untuk mendukung struktur JSON bersarang (nested), terdapat juga beberapa kelas pendukung seperti Glossary, GlossDiv, GlossList, dan GlossEntry yang masing-masing merepresentasikan hierarki dalam file JSON tersebut.

```
1  using System;
2  using System.IO;
3  using System.Text.Json;
4
5  1 reference
6  public class GlossaryItem2311104073
7  {
8      1 reference
9      public static void ReadJSON()
10     {
11         string fileName = "jurnal7_3_2311104073.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<Glossary>(jsonString);
21
22         var entry = data.GlossDiv.GlossList.GlossEntry;
23         Console.WriteLine("Glossary Entry:");
24         Console.WriteLine($"ID : {entry.ID}");
25         Console.WriteLine($"SortAs : {entry.SortAs}");
26         Console.WriteLine($"GlossTerm : {entry.GlossTerm}");
27         Console.WriteLine($"Acronym : {entry.Acronym}");
28         Console.WriteLine($"Abbrev : {entry.Abbrev}");
29         Console.WriteLine($"Title : {entry.Title}");
30         Console.WriteLine($"GlossSee : {entry.GlossSee}");
31     }
32 }
```

b. Program.cs

Merupakan file utama program yang mengeksekusi method Main() sebagai titik awal aplikasi. Di dalamnya, terdapat perintah untuk mencetak judul ke layar sebagai penanda bahwa proses deserialisasi akan dilakukan. Kemudian, program memanggil method ReadJSON() dari kelas GlossaryItem2311104073 untuk membaca dan menampilkan isi file JSON ke konsol. Dengan struktur yang sederhana, file ini berfungsi untuk menginisialisasi dan menjalankan proses utama yang sudah didefinisikan di file lainnya

```
1 using System;
2
3 0 references
4 class Program
5 {
6     0 references
7     static void Main(string[] args)
8     {
9         Console.WriteLine("\n=== Deserialisasi Glossary Item ===");
10        GlossaryItem2311104073.ReadJSON();
11    }
```

c. jurnal7_3_2311104073.json

file data yang berisi informasi dalam format JSON tentang sebuah entri glosarium. File ini memiliki struktur bersarang yang terdiri dari objek GlossDiv, di dalamnya terdapat GlossList, dan akhirnya GlossEntry yang memuat berbagai atribut seperti ID, SortAs, GlossTerm, Acronym, Abbrev, Title, dan GlossSee. Data ini akan dibaca dan ditampilkan oleh program melalui proses deserialisasi untuk diubah menjadi objek C# yang sesuai dengan struktur class yang telah dibuat.

```
jurnal7_3_2311104073.json
File Edit View

{
  "GlossDiv": {
    "GlossList": {
      "GlossEntry": {
        "ID": "SGML",
        "SortAs": "SGML",
        "GlossTerm": "Standard Generalized Markup Language",
        "Acronym": "SGML",
        "Abbrev": "ISO 8879:1986",
        "Title": "Glossary Entry",
        "GlossSee": "markup"
      }
    }
  }
}
```

Output:

```
Microsoft Visual Studio Debug Console
+ v

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

D:\Semester4\KPL\jurnal7_3_2311104073\jurnal7_3_2311104073\bin\Debug\net8.0\jurnal7_3_2311104073.exe
ted with code 0 (0x0).
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automaticall
le when debugging stops.
Press any key to close this window . . .|
```