

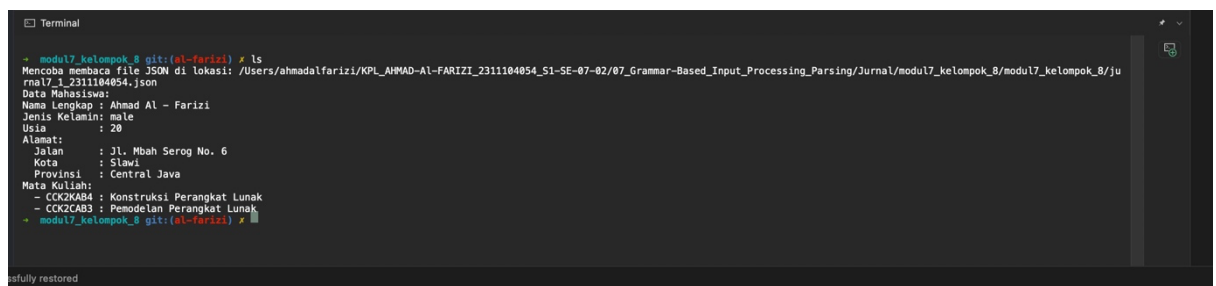
Nama: Ahmad Al – Farizi

NIM: 2311104054

I. Link github: https://github.com/alxfarizi/KPL_AHMAD-AL-FARIZI_2311104054_S1SE-07-02/tree/main/07_Grammar-Based_Input_Processing_Parsing/Jurnal

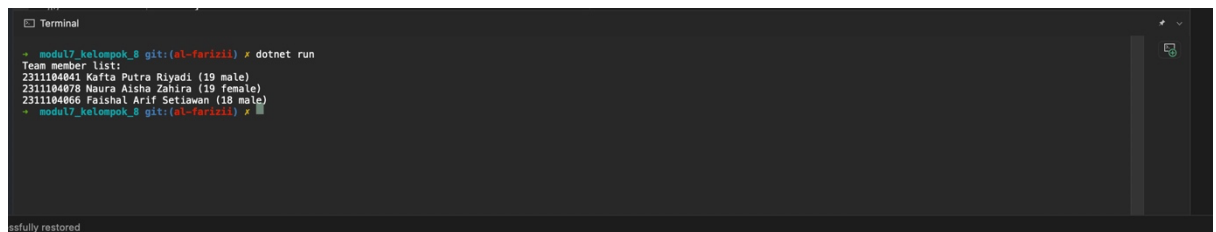
II. Hasil running:

Class DataMahasiswa2311104054:



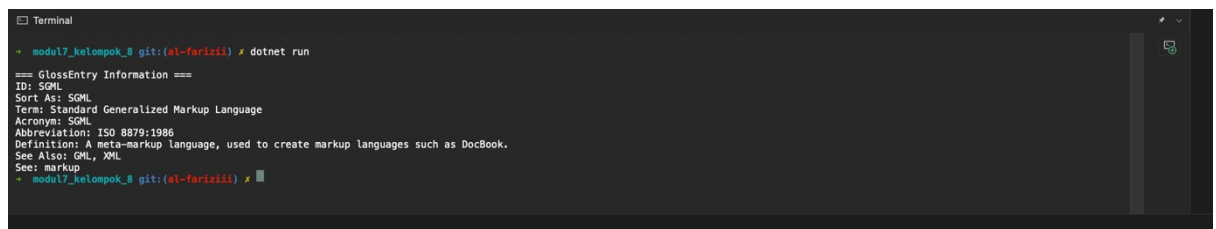
```
modul7_kelompok_8 git:(al-farizi) x ls
Mencoba membaca file JSON di lokasi: /Users/ahmadalfarizi/KPL_AHMAD-AL-FARIZI_2311104054_S1-SE-07-02/07_Grammar-Based_Input_Processing_Parsing/Jurnal/modul7_kelompok_8/modul7_kelompok_8/jurnal7_1_2311104054.json
Data Mahasiswa:
Nama Lengkap : Ahmad Al - Farizi
Jenis Kelamin : male
Usia          : 20
Alamat:
Jalan        : Jl. Mbah Serog No. 6
Kota         : Slawi
Provinsi     : Central Java
Mata Kuliah:
- CCK2AB4 : Konstruksi Perangkat Lunak
- CCK2CAB3 : Pemodelan Perangkat Lunak
modul7_kelompok_8 git:(al-farizi) x
```

Class TeamMembers2311104054:



```
modul7_kelompok_8 git:(al-farizi) x dotnet run
Team member list:
2311104041 Kafia Putra Riyadi (19 male)
2311104078 Naura Aisha Zahira (19 female)
2311104066 Faishal Arif Setiawan (18 male)
modul7_kelompok_8 git:(al-farizi) x
```

Class GlossaryItem2311104054:



```
modul7_kelompok_8 git:(al-farizi) x dotnet run
=== GlossEntry Information ===
ID: SGML
Sort As: SGML
Term: Standard Generalized Markup Language
Acronym: SGML
Abbreviation: ISO 8879:1986
Definition: A meta-markup language, used to create markup languages such as DocBook.
See Also: GML, XML
See: markup
modul7_kelompok_8 git:(al-farizi) x
```

III. Penjelasan singkat:

Class DataMahasiswa2311104054:

Program C# yang menggunakan namespace "System.Text.Json" untuk membaca dan menampilkan data siswa dari file JSON. Terdapat tiga kelas utama: "Mahasiswa", "Alamat", dan "Mata kuliah". Properti kelas "Mahasiswa" termasuk

nama depan, nama belakang, jenis kelamin, usia, alamat (dalam bentuk objek "Alamat") dan daftar mata kuliah (dalam bentuk daftar "Mata kuliah"). Metode "ReadJson()" membaca file JSON dari lokasi tertentu, memastikan Konsol akan melihat data siswa, termasuk alamat IP dan mata kuliah, jika berhasil. Program akan menampilkan pesan error yang sesuai jika terjadi kesalahan, seperti file tidak ditemukan atau format JSON tidak valid. Program ini melihatkan cara membaca dan memproses data JSON.

```
Program.cs | DataMahasiswa231104054.cs | Jurnal7_1_231104054.json |
DataMahasiswa231104054.cs | Course > No selection
1  using System;
2  using System.Collections.Generic;
3  using System.IO;
4  using System.Text.Json;
5  using System.Text.Json.Serialization;
6
7  namespace modul7_2311184854
8  {
9      internal class DataMahasiswa231104054
10     {
11         public class Mahasiswa
12         {
13             [JsonPropertyName("firstName")]
14             public string? FirstName { get; set; }
15
16             [JsonPropertyName("lastName")]
17             public string? LastName { get; set; }
18
19             [JsonPropertyName("gender")]
20             public string? Gender { get; set; }
21
22             [JsonPropertyName("age")]
23             public int Age { get; set; }
24
25             [JsonPropertyName("address")]
26             public Address? Address { get; set; }
27
28             [JsonPropertyName("courses")]
29             public List<Course>? Courses { get; set; }
30         }
31
32         public class Address
33         {
34             [JsonPropertyName("streetAddress")]
35             public string? StreetAddress { get; set; }
36
37             [JsonPropertyName("city")]
38             public string? City { get; set; }
39
40             [JsonPropertyName("state")]
41             public string? State { get; set; }
42         }
43
44         public class Course
45         {
46             [JsonPropertyName("code")]
47             public string? Code { get; set; }
48
49             [JsonPropertyName("name")]
50             public string? Name { get; set; }
51         }
52
53         public static void ReadJson()
54         {
55             string pathFile = Path.Combine(Directory.GetCurrentDirectory(), "Jurnal7_1_231104054.json");
56             Console.WriteLine($"Mencoba membaca file JSON di lokasi: {pathFile}");
57
58             if (File.Exists(pathFile))
59             {
60                 try
61                 {
62                     string jsonData = File.ReadAllText(pathFile);
```

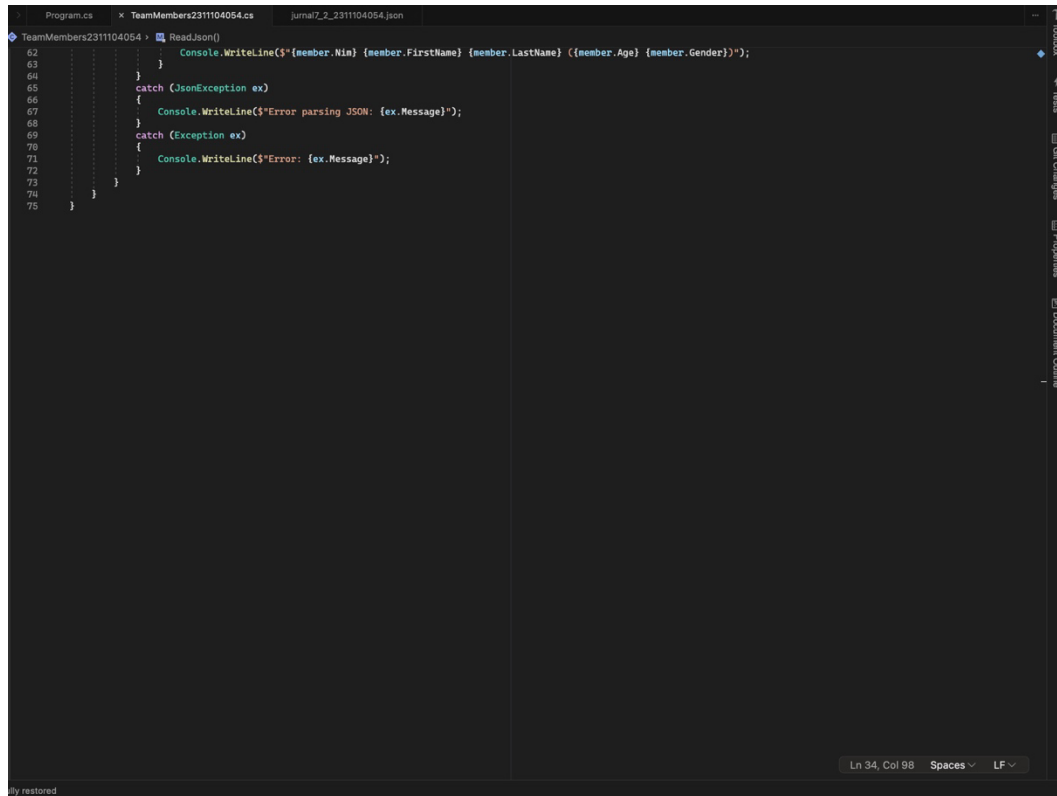
```
Program.cs | DataMahasiswa231104054.cs | Jurnal7_1_231104054.json |
DataMahasiswa231104054.cs | Course > No selection
62         string jsonData = File.ReadAllText(pathFile);
63         Mahasiswa? mahasiswa = JsonSerializer.Deserialize<Mahasiswa>(jsonData);
64
65         if (mahasiswa != null)
66         {
67             Console.WriteLine($"Data Mahasiswa:");
68             Console.WriteLine($"Nama Lengkap : {mahasiswa.FirstName} {mahasiswa.LastName}");
69             Console.WriteLine($"Jenis Kelamin : {mahasiswa.Gender}");
70             Console.WriteLine($"Usia : {mahasiswa.Age}");
71             Console.WriteLine($"Alamat:");
72             Console.WriteLine($"    Jalan : {mahasiswa.Address?.StreetAddress}");
73             Console.WriteLine($"    Kota : {mahasiswa.Address?.City}");
74             Console.WriteLine($"    Provinsi : {mahasiswa.Address?.State}");
75             Console.WriteLine($"Mata Kuliah:");
76             if (mahasiswa.Courses != null)
77             {
78                 foreach (var course in mahasiswa.Courses)
79                 {
80                     Console.WriteLine($" - {course.Code} : {course.Name}");
81                 }
82             }
83         }
84         catch (JsonException ex)
85         {
86             Console.WriteLine($"Error parsing JSON: {ex.Message}");
87         }
88     }
89     else
90     {
91         Console.WriteLine("File tidak ditemukan.");
92         Console.WriteLine($"Jalur file: {pathFile}");
93     }
94 }
95
96
97
98
```

```
Ln 51, Col 10 Spaces LF
by restored
```

Class TeamMembers2311104054:

Source code implementasikan dalam C# untuk membaca dan memproses data JSON dengan daftar anggota tim. Dua kelas utama dalam program ini adalah "TeamMember" dan "TeamMembers2311104054". Kelas pertama mengandung struktur data anggota tim dengan properti seperti "FirstName", "LastName", "Gender", "Age", dan "Nim", yang dimapping ke properti JSON menggunakan atribut "JsonPropertyName". Kelas kedua berisi metode "ReadJson()" Program akan menampilkan pesan error jika file tidak ditemukan atau data JSON tidak valid. Untuk memastikan kompatibilitas, proses parsing menggunakan "JsonSerializer" dengan opsi case-insensitive. Hasil yang mudah dibaca mencakup NIM, nama lengkap, usia, dan gender dari setiap anggota tim.

```
TeamMembers2311104054 > ReadJson()
1  using System;
2  using System.Collections.Generic;
3  using System.IO;
4  using System.Text.Json;
5  using System.Text.Json.Serialization;
6
7  namespace modul7_2311104054
8  {
9      public class TeamMember
10     {
11         [JsonPropertyName("firstName")]
12         public string FirstName { get; set; } = string.Empty;
13
14         [JsonPropertyName("lastName")]
15         public string LastName { get; set; } = string.Empty;
16
17         [JsonPropertyName("gender")]
18         public string Gender { get; set; } = string.Empty;
19
20         [JsonPropertyName("age")]
21         public int Age { get; set; }
22
23         [JsonPropertyName("nim")]
24         public string Nim { get; set; } = string.Empty;
25     }
26
27     public class TeamMembers2311104054
28     {
29         [JsonPropertyName("members")]
30         public List<TeamMember> Members { get; set; } = new List<TeamMember>();
31
32         public static void ReadJson()
33         {
34             string pathFile = Path.Combine(Directory.GetCurrentDirectory(), "jurnal7_2_2311104054.json");
35
36             if (!File.Exists(pathFile))
37             {
38                 Console.WriteLine("File tidak ditemukan");
39                 Console.WriteLine(pathFile);
40                 return;
41             }
42
43             try
44             {
45                 string jsonData = File.ReadAllText(pathFile);
46                 var options = new JsonSerializerOptions
47                 {
48                     PropertyNameCaseInsensitive = true
49                 };
50
51                 TeamMembers2311104054 team = JsonSerializer.Deserialize<TeamMembers2311104054>(jsonData, options);
52
53                 if (team?.Members == null)
54                 {
55                     Console.WriteLine("Data JSON tidak valid atau kosong");
56                     return;
57                 }
58
59                 Console.WriteLine("Team member list:");
60                 foreach (var member in team.Members)
61                 {
62                     Console.WriteLine($"{member.Nim} {member.FirstName} {member.LastName} {member.Age} {member.Gender}");
63                 }
64             }
65             catch { }
66         }
67     }
68 }
```



```
62         Console.WriteLine($"{member.Nin} {member.FirstName} {member.LastName} ({member.Age} {member.Gender})");
63     }
64 }
65 catch (JsonException ex)
66 {
67     Console.WriteLine($"Error parsing JSON: {ex.Message}");
68 }
69 catch (Exception ex)
70 {
71     Console.WriteLine($"Error: {ex.Message}");
72 }
73 }
74 }
75 }
```

Class GlossaryItem2311104054:

Source code menggunakan namespace "System.Text.Json" untuk membaca dan memproses data JSON yang berisi informasi glossary (daftar istilah). Ini juga memungkinkan deserialisasi data JSON ke dalam objek C#. Struktur data JSON diwakili oleh beberapa kelas, termasuk "GlossDef" yang berisi definisi dan referensi terkait, "GlossEntry" yang berisi detail istilah seperti ID, istilah, akronim, dll., "GlossDiv" yang berisi bagian glossary dengan judul, dan "Glossary" yang merupakan struktur utama glossary. Kelas utama "GlossaryItem2311104054" memiliki metode "ReadJSON". Jika berhasil, aplikasi akan menampilkan data dari "GlossEntry", termasuk ID, istilah, definisi, dan referensi. Selain itu, program memiliki penanganan error untuk kasus file tidak ditemukan atau format JSON yang tidak valid.

```

modul7_kelompok_8
GlossaryItem2311104054 > ReadJSON()
1 using System;
2 using System.IO;
3 using System.Text.Json;
4 using System.Text.Json.Serialization;
5
6 namespace GlossaryProject
7 {
8     public class GlossDef
9     {
10         [JsonPropertyName("para")]
11         public string Para { get; set; } = string.Empty;
12
13         [JsonPropertyName("GlossSeeAlso")]
14         public string[] GlossSeeAlso { get; set; } = Array.Empty<string>();
15     }
16
17     public class GlossEntry
18     {
19         [JsonPropertyName("ID")]
20         public string ID { get; set; } = string.Empty;
21
22         [JsonPropertyName("SortAs")]
23         public string SortAs { get; set; } = string.Empty;
24
25         [JsonPropertyName("GlossTerm")]
26         public string GlossTerm { get; set; } = string.Empty;
27
28         [JsonPropertyName("Acronym")]
29         public string Acronym { get; set; } = string.Empty;
30
31         [JsonPropertyName("Abbrev")]
32         public string Abbrev { get; set; } = string.Empty;
33
34         [JsonPropertyName("GlossDef")]
35         public GlossDef GlossDef { get; set; } = new GlossDef();
36
37         [JsonPropertyName("GlossSee")]
38         public string GlossSee { get; set; } = string.Empty;
39     }
40
41     public class GlossList
42     {
43         [JsonPropertyName("GlossEntry")]
44         public GlossEntry GlossEntry { get; set; } = new GlossEntry();
45     }
46
47     public class GlossDiv
48     {
49         [JsonPropertyName("title")]
50         public string Title { get; set; } = string.Empty;
51
52         [JsonPropertyName("GlossList")]
53         public GlossList GlossList { get; set; } = new GlossList();
54     }
55
56     public class Glossary
57     {
58         [JsonPropertyName("title")]
59         public string Title { get; set; } = string.Empty;
60
61         [JsonPropertyName("GlossDiv")]
62         public GlossDiv GlossDiv { get; set; } = new GlossDiv();
63     }
64 }
Ln 73, Col 98 Spaces LF

```

```

Program.cs x GlossaryItem2311104054.cs Jurnal7_3_2311104054.json
GlossaryItem2311104054 > ReadJSON()
62 }
63 public GlossDiv GlossDiv { get; set; } = new GlossDiv();
64 }
65 // Kelas utama GlossaryItem
66 public class GlossaryItem2311104054
67 {
68     [JsonPropertyName("glossary")]
69     public Glossary Glossary { get; set; } = new Glossary();
70
71     public static void ReadJSON()
72     {
73         string pathFile = Path.Combine(Directory.GetCurrentDirectory(), "Jurnal7_3_2311104054.json");
74
75         if (File.Exists(pathFile))
76         {
77             Console.WriteLine("File glossary.json tidak ditemukan");
78             Console.WriteLine("Path yang dicari: " + pathFile);
79             return;
80         }
81
82         try
83         {
84             string jsonData = File.ReadAllText(pathFile);
85             var options = new JsonSerializerOptions
86             {
87                 PropertyNameCaseInsensitive = true,
88                 WriteIndented = true
89             };
90
91             GlossaryItem2311104054? glossaryItem = JsonSerializer.Deserialize<GlossaryItem2311104054>(jsonData, options);
92
93             if (glossaryItem?.Glossary == null)
94             {
95                 Console.WriteLine("Data JSON tidak valid atau kosong");
96                 return;
97             }
98
99             // Menampilkan informasi GlossEntry
100             var entry = glossaryItem.Glossary.GlossDiv.GlossList.GlossEntry;
101             Console.WriteLine($"ID: {entry.ID}");
102             Console.WriteLine($"Sort As: {entry.SortAs}");
103             Console.WriteLine($"Term: {entry.GlossTerm}");
104             Console.WriteLine($"Acronym: {entry.Acronym}");
105             Console.WriteLine($"Abbreviation: {entry.Abbrev}");
106             Console.WriteLine($"Definition: {entry.GlossDef.Para}");
107             Console.WriteLine($"See Also: {string.Join(", ", entry.GlossDef.GlossSeeAlso)}");
108             Console.WriteLine($"See: {entry.GlossSee}");
109
110         }
111         catch (JsonException ex)
112         {
113             Console.WriteLine($"Error parsing JSON: {ex.Message}");
114         }
115         catch (Exception ex)
116         {
117             Console.WriteLine($"Error: {ex.Message}");
118         }
119     }
120 }
121 }
Ln 73, Col 98 Spaces LF

```