

**PRAKTIKUM KONSTRUKSI PERANGKAT LUNAK  
TUGAS JURNAL 06**

**Grammar-Based Input Processing Parsing**



**Telkom  
University**

disusun Oleh:  
Nita Fitrotul Mar'ah  
2211104005

SE0601

**S1 REKAYASA PERANGKAT LUNAK  
FAKULTAS INFORMATIKA  
TELKOM UNIVERSITY  
2025**

## 1. MENAMBAHKAN JSON DESERIALIZATON 1

Buatlah branch baru dengan nama branch “nama\_panggilan\_praktikan” dan checkout kesana.

- Download file “jurnal7\_1\_nim.json” dan rename file tersebut dengan mengganti “nim” dengan NIM praktikan kemudian pindahkan file json tersebut di folder solution projectnya.
- Ganti isi dari file json tersebut dengan detail yang benar dari praktikan.
- Buatlah sebuah file class baru dengan nama “DataMahasiswa<NIM\_PRAKTIKAN>”.
- Buat method “ReadJSON() yang melakukan parsing untuk file tersebut menjadi object sesuai.
- Pada method tersebut, lakukan print hasil deserialisasi dari object yang dibuat dengan format bebas asalkan semua nilai ditampilkan di console/output.”

### Jawab

#### - Source Code

File jurnal7\_1\_2211104005.json

```
1  {
2    "firstName": "Nita",
3    "lastName": "Fitrotul Mar'ah",
4    "gender": "female",
5    "age": 21,
6    "address": {
7      "streetAddress": "Jl. Kalibener",
8      "city": "Purwokerto",
9      "state": "Central Java"
10   },
11   "courses": [
12     {
13       "code": "CRI2C4",
14       "name": "Konstruksi Perangkat Lunak"
15     },
16     {
17       "code": "CRI2XX",
18       "name": "Kecerdasan Buatan"
19     }
20   ]
21 }
22
```

File DataMahasiswa2211104005.cs

```

1 using System;
2 using System.IO;
3 using System.Text.Json;
4
5 // 0 references
6 public class DataMahasiswa2211104005
7 {
8     // Class untuk merepresentasikan struktur JSON
9     // 1 reference
10    public class Address
11    {
12        // 1 reference
13        public string streetAddress { get; set; }
14        // 1 reference
15        public string city { get; set; }
16        // 1 reference
17        public string state { get; set; }
18    }
19
20    // 1 reference
21    public class Course
22    {
23        // 1 reference
24        public string code { get; set; }
25        // 1 reference
26        public string name { get; set; }
27    }
28
29    // 2 references
30    public class Mahasiswa
31    {
32        // 1 reference
33        public string firstName { get; set; }
34        // 1 reference
35        public string lastName { get; set; }
36        // 1 reference
37        public string gender { get; set; }
38        // 1 reference
39        public int age { get; set; }
40        // 5 references
41        public Address address { get; set; }
42        // 1 reference
43        public Course[] courses { get; set; }
44    }
45
46    // 0 references
47    public static void ReadJSON()
48    {
49        string filePath = "jurnal7_1_2211104005.json";
50
51        if (File.Exists(filePath))
52        {
53            string jsonString = File.ReadAllText(filePath);
54
55            Mahasiswa mahasiswa = JsonSerializer.Deserialize<Mahasiswa>(jsonString);
56
57            Console.WriteLine("----- Data Mahasiswa -----");
58            Console.WriteLine($"Nama : {mahasiswa.firstName} {mahasiswa.lastName}");
59            Console.WriteLine($"Gender: {mahasiswa.gender}");
60            Console.WriteLine($"Usia : {mahasiswa.age}");
61            Console.WriteLine($"Alamat: {mahasiswa.address.streetAddress}, " +
62                $"{mahasiswa.address.city}, {mahasiswa.address.state}");
63            Console.WriteLine("\n----- Mata Kuliah -----");
64
65            foreach (var course in mahasiswa.courses)
66            {
67                Console.WriteLine($"Kode: {course.code} - Nama: {course.name}");
68            }
69        }
70        else
71        {
72            Console.WriteLine("File JSON tidak ditemukan.");
73        }
74    }
75 }

```

File Program.cs

```

1 class Program
2 {
3     // 0 references
4     static void Main(string[] args)
5     {
6         DataMahasiswa2211104005.ReadJSON();
7     }
8 }

```

- **Screenshot hasil run**

```
===== Data Mahasiswa =====  
Nama   : Nita Fitrotul Mar'ah  
Gender : female  
Usia   : 21  
Alamat : Jl. Kalibener, Purwokerto, Central Java  
  
===== Mata Kuliah =====  
Kode: CRI2C4 - Nama: Konstruksi Perangkat Lunak  
Kode: CRI2XX - Nama: Kecerdasan Buatan
```

- **Penjelasan**

Tujuan dari program ini adalah untuk membaca dan menampilkan informasi mahasiswa yang tersimpan dalam file berformat JSON dengan menggunakan teknik parsing JSON. File bernama jurnal7\_1\_2211104005.json menyimpan data mahasiswa yang mencakup nama, usia, jenis kelamin, alamat, serta daftar mata kuliah. File ini diletakkan di dalam direktori proyek, dan program akan membacanya, lalu mengubah isinya menjadi objek dalam bahasa C#.

Proses parsing merujuk pada konversi data dari format teks JSON menjadi objek yang bisa diproses dalam kode program. Pada file DataMahasiswa2211104005.cs, didefinisikan kelas Mahasiswa beserta sub-kelas Address dan Course untuk mencerminkan struktur JSON.

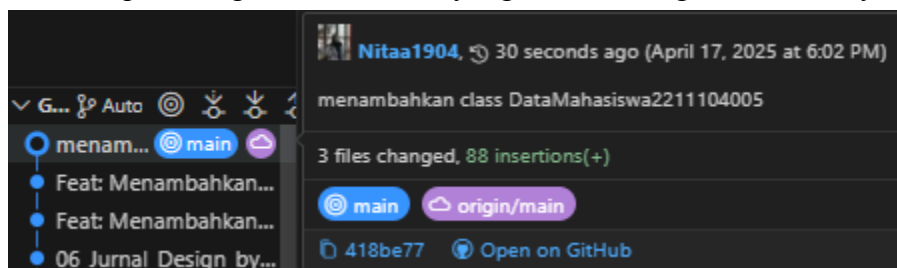
Metode ReadJSON() digunakan untuk membaca file JSON menggunakan File.ReadAllText(), lalu melakukan parsing dengan JsonSerializer.Deserialize<Mahasiswa>() agar data dari file dapat dipetakan ke objek bertipe Mahasiswa. Setelah proses parsing berhasil, informasi yang diperoleh ditampilkan ke layar dalam format yang lebih terstruktur dan mudah dibaca.

File Program.cs bertugas memanggil ReadJSON() saat program dijalankan, sehingga proses pembacaan dan parsing data berjalan otomatis. Dengan cara ini, program dapat memanfaatkan dan mengelola data JSON secara efektif.

## 2. MELAKUKAN COMMIT

Pada branch yang dibuat sebelumnya:

- Lakukan commit dengan pesan “menambahkan class <NAMA\_CLASS>”
- Lakukan push ke github ke branch yang dibuat di bagian sebelumnya.



### 3. MENAMBAHKAN JSON DESERIALIZATON 2

Buatlah branch baru dengan nama branch “nama\_panggilan\_praktikan” dan checkout kesana

- Download file “jurnal7\_2\_nim.json” dan rename file tersebut dengan mengganti “nim” dengan NIM praktikan kemudian pindahkan file json tersebut di folder solution projectnya.
- Ubah isi dari file json tersebut dengan daftar anggota kelompok (untuk tubes).
- Buatlah sebuah file class baru dengan nama “TeamMembers<NIM\_PRAKTIKAN>”.
- Buat method “ReadJSON() yang melakukan parsing untuk file tersebut menjadi object sesuai
- Pada method tersebut, lakukan print hasil deserialisasi dari object yang dibuat dengan format:  
“Team member list:”  
“<nim1> <firstname1 + lastname1> (<age1> <gender1>) ”  
“<nim2> <firstname2 + lastname2> (<age2> <gender2>) ”  
dst.

#### - Source code

File jurnal7\_2\_2211104005.json

```
1  {
2    "members": [
3      {
4        "firstName": "Alfian",
5        "lastName": "Mutakim",
6        "gender": "Male",
7        "age": 21,
8        "nim": "2211104014"
9      },
10     {
11       "firstName": "Nita",
12       "lastName": "Fitrotul Mar'ah",
13       "gender": "Female",
14       "age": 21,
15       "nim": "2211104005"
16     },
17     {
18       "firstName": "Ahmad",
19       "lastName": "Junaidi",
20       "gender": "Male",
21       "age": 21,
22       "nim": "2211104002"
23     },
24   ]
25 }
```

```

14 {
15     "firstName": "Rafli",
16     "lastName": "Dhafin Kamil",
17     "gender": "Male",
18     "age": 21,
19     "nim": "2211104018"
20 },
21 {
22     "firstName": "Muhammad",
23     "lastName": "Edgar Nadif",
24     "gender": "Male",
25     "age": 21,
26     "nim": "2211104019"
27 },
28 {
29     "firstName": "Nadia",
30     "lastName": "Putri Rahmaniar",
31     "gender": "Female",
32     "age": 21,
33     "nim": "2211104012"
34 },
35 {
36     "firstName": "Muhammad",
37     "lastName": "Dhimas Afrizal",
38     "gender": "Male",
39     "age": 21,
40     "nim": "2211104023"
41 }
42 ]
43 }
44

```

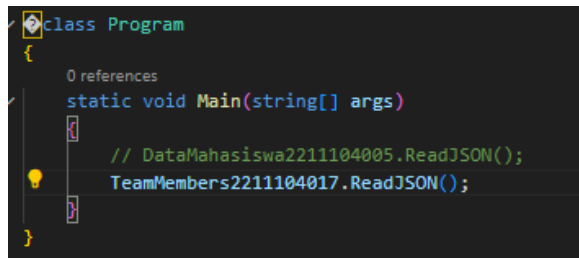
File "TeamMembers2211104005.cs

```

1 using System;
2 using System.IO;
3 using System.Text.Json;
4
5 0 references
6 class TeamMembers2211104005
7 {
8     1 reference
9     class Member
10     {
11         1 reference
12         public string firstName { get; set; }
13         1 reference
14         public string lastName { get; set; }
15         1 reference
16         public string gender { get; set; }
17         1 reference
18         public int age { get; set; }
19         1 reference
20         public string nim { get; set; }
21     }
22
23     2 references
24     class Team
25     {
26         1 reference
27         public Member[] members { get; set; }
28     }
29
30     0 references
31     public static void ReadJSON()
32     {
33         string filePath = "jurnal7_2_2211104005.json";
34
35         if (!File.Exists(filePath))
36         {
37             Console.WriteLine("File JSON tidak ditemukan.");
38             return;
39         }
40
41         // Membaca dan melakukan parsing JSON
42         Team team = JsonSerializer.Deserialize<Team>(File.ReadAllText(filePath));
43
44         Console.WriteLine("Team member list:");
45         foreach (var member in team.members)
46         {
47             Console.WriteLine($"{member.nim} {member.firstName} {member.lastName} " +
48                               $"{member.age} {member.gender}");
49         }
50     }
51 }

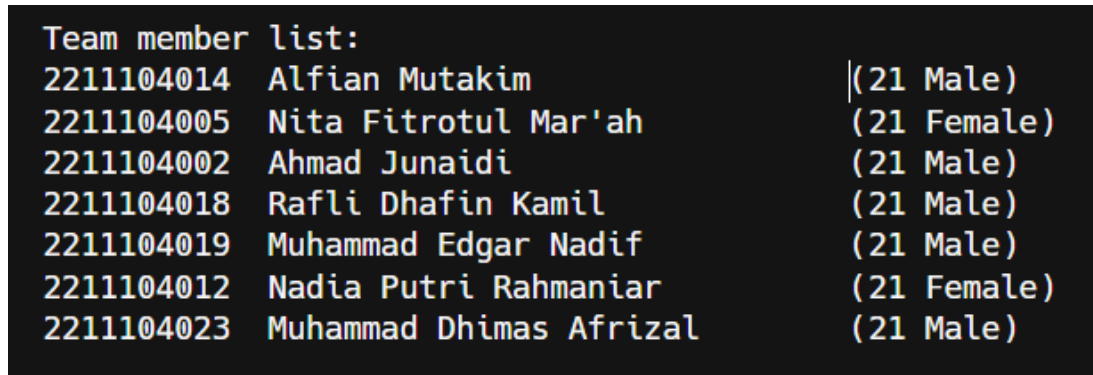
```

File Program.cs

A screenshot of a 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 ReadJSON() on two objects: DataMahasiswa2211104005 and TeamMembers2211104017. The code is as follows:

```
class Program
{
    0 references
    static void Main(string[] args)
    {
        // DataMahasiswa2211104005.ReadJSON();
        TeamMembers2211104017.ReadJSON();
    }
}
```

- **Screenshot hasil run**

A screenshot of a console window showing the output of the program. The output is a list of team members with their NIM, name, and age/gender in parentheses. The output is as follows:

```
Team member list:
2211104014 Alfian Mutakim (21 Male)
2211104005 Nita Fitrotul Mar'ah (21 Female)
2211104002 Ahmad Junaidi (21 Male)
2211104018 Rafli Dhafin Kamil (21 Male)
2211104019 Muhammad Edgar Nadif (21 Male)
2211104012 Nadia Putri Rahmaniar (21 Female)
2211104023 Muhammad Dhimas Afrizal (21 Male)
```

- **Pejelasan**

Program ini dirancang untuk membaca dan menampilkan informasi anggota kelompok dari file JSON menggunakan teknik parsing. File jurnal7\_2\_2211104005.json menyimpan data anggota tim berupa NIM, nama lengkap, jenis kelamin, dan usia dalam format JSON. File tersebut disimpan di dalam folder proyek agar dapat diakses oleh program.

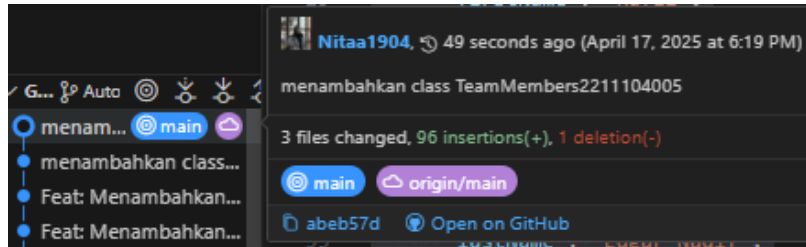
Proses parsing dilakukan untuk mengubah data dalam format teks JSON menjadi objek yang dapat digunakan dalam bahasa pemrograman C#. Di dalam file TeamMembers2211104005.cs, program membaca isi file JSON menggunakan File.ReadAllText() dan memprosesnya dengan JsonSerializer.Deserialize<Team>(), sehingga data diubah menjadi objek bertipe Team. Objek ini memiliki properti members berupa array yang menyimpan informasi masing-masing anggota.

Setelah parsing berhasil, program akan menampilkan data anggota tim ke console dengan format yang telah ditentukan. Eksekusi proses ini dikendalikan oleh file Program.cs, yang memanggil method ReadJSON() saat program dijalankan agar data dapat diproses dan ditampilkan dengan benar.

#### 4. MELAKUKAN COMMIT

Pada branch main/master:

- Lakukan commit dengan pesan “menambahkan class <NAMA\_CLASS>”.
- Lakukan push ke github ke branch yang dibuat di bagian sebelumnya.



## 5. MENAMBAHKAN JSON DESERIALIZATION 3

Buatlah branch baru dengan nama branch “nama\_panggilan\_praktikan” dan checkout kesana.

- Download file “jurnal7\_3\_nim.json” dan rename file tersebut dengan mengganti “nim” dengan NIM praktikan kemudian pindahkan file json tersebut di folder solution projectnya.
- Buatlah sebuah file class baru dengan nama “GlossaryItem<NIM\_PRAKTIKAN>”.
- Buat method “ReadJSON() yang melakukan parsing untuk file tersebut menjadi object sesuai.
- Pada method tersebut, lakukan print hasil deserialisasi dari object yang dibuat dengan format bebas untuk bagian “GlossEntry” saja.

**Jawab:**

### - Source Code

File jurnal7\_3\_2211104005.json

```
{
  "glossary": {
    "title": "example glossary",
    "GlossDiv": {
      "title": "S",
      "GlossList": {
        "GlossEntry": {
          "ID": "SGML",
          "SortAs": "SGML",
          "GlossTerm": "Standard Generalized Markup Language",
          "Acronym": "SGML",
          "Abbrev": "ISO 8879:1986",
          "GlossDef": {
            "para": "A meta-markup language, used to create markup languages such as DocBook.",
            "GlossSeeAlso": ["GML", "XML"]
          },
          "GlossSee": "Markup"
        }
      }
    }
  }
}
```



File GlossaryItem2211104005.cs

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

0 references
class GlossaryItem2211104005
{
    1 reference
    class GlossDef
    {
        1 reference
        public string para { get; set; }
        1 reference
        public string[] GlossSeeAlso { get; set; }
    }

    2 references
    class GlossEntry
    {
        1 reference
        public string ID { get; set; }
        1 reference
        public string SortAs { get; set; }
        1 reference
        public string GlossTerm { get; set; }
        1 reference
        public string Acronym { get; set; }
        1 reference
        public string Abbrev { get; set; }
        2 references
        public GlossDef GlossDef { get; set; }
        1 reference
        public string GlossSee { get; set; }
    }

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

    1 reference
    class GlossDiv
    {
        0 references
        public string title { get; set; }
        1 reference
        public GlossList GlossList { get; set; }
    }
}
```

```

1 reference
class Glossary
{
    0 references
    public string title { get; set; }
    1 reference
    public GlossDiv GlossDiv { get; set; }
}

2 references
class Root
{
    1 reference
    public Glossary glossary { get; set; }
}

0 references
public static void ReadJSON()
{
    string filePath = "jurnal7_3_2211104005.json";

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

    // Membaca dan parsing JSON
    Root data = JsonSerializer.Deserialize<Root>(File.ReadAllText(filePath));

    // Mengambil bagian GlossEntry
    GlossEntry entry = data.glossary.GlossDiv.GlossList.GlossEntry;

    Console.WriteLine("==== Glossary Entry =====");
    Console.WriteLine($"ID          : {entry.ID}");
    Console.WriteLine($"SortAs    : {entry.SortAs}");
    Console.WriteLine($"Term      : {entry.GlossTerm}");
    Console.WriteLine($"Acronym   : {entry.Acronym}");
    Console.WriteLine($"Abbrev    : {entry.Abbrev}");
    Console.WriteLine($"Definition: {entry.GlossDef.para}");
    Console.WriteLine("See Also  : ");
    Console.WriteLine(string.Join(", ", entry.GlossDef.GlossSeeAlso));
    Console.WriteLine($"GlossSee  : {entry.GlossSee}");
}

```

## File Program.cs

```

class Program
{
    0 references
    static void Main(string[] args)
    {
        // DataMahasiswa2211104005.ReadJSON();
        // TeamMembers2211104017.ReadJSON();
        GlossaryItem2211104017.ReadJSON();
    }
}

```

## - Screenshot hasil run

```

===== Glossary Entry =====
ID          : SGML
SortAs      : SGML
Term        : Standard Generalized Markup Language
Acronym     : SGML
Abbrev      : ISO 8879:1986
Definition: A meta-markup language, used to create markup languages such as DocBook.
See Also    : GML, XML
GlossSee    : Markup

```

## - Penjelasan

Tujuan dari program ini adalah untuk membaca dan menampilkan informasi dari file JSON bernama "jurnal7\_3\_2211104005.json", yang memuat data mengenai sebuah entri glossary. Struktur utama dalam file JSON ini terdiri dari bagian "glossary", yang di dalamnya terdapat "GlossDiv", kemudian "GlossList", dan akhirnya "GlossEntry" yang menyimpan detail seperti ID, GlossTerm, Acronym, Abbreviation, Definition, serta referensi tambahan.

Program ini membaca file JSON tersebut dan mengaplikasikan metode parsing (deserialisasi) untuk mengonversi data dari bentuk teks JSON menjadi objek dalam bahasa C# yang bisa diproses lebih lanjut.

Pada file GlossaryItem2211104005.cs, dibuat beberapa kelas yang mencerminkan struktur dari JSON tersebut. Dalam method ReadJSON(), program membaca isi file menggunakan File.ReadAllText(), lalu melakukan parsing dengan JsonSerializer.Deserialize<Root>(). Setelah berhasil diparsing, program mengambil data dari bagian GlossEntry dan menampilkannya ke console dengan format yang lebih terstruktur dan mudah dipahami.

Penggunaan teknik parsing ini sangat membantu untuk mengakses bagian-bagian tertentu dari JSON, seperti mengambil ID, GlossTerm, Acronym, dan daftar referensi yang terdapat dalam GlossSeeAlso, tanpa perlu membaca file secara manual baris demi baris.

## 6. MELAKUKAN COMMIT

Pada branch yang dibuat sebelumnya:

- Lakukan commit dengan pesan "menambahkan class <NAMA\_CLASS>"
- Lakukan push ke github ke branch yang dibuat di bagian sebelumnya.

