

**PRAKTIKUM KONSTRUKSI PERANGKAT LUNAK  
TUGAS JURNAL 06**

**Grammar-Based Input Processing Parsing**



**Telkom  
University**

disusun Oleh:  
Alfian Mutakim  
2211104024

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\_2211104024.json

```
{
  "firstName": "Alfian",
  "lastName": "Alfian Mutakim",
  "gender": "male",
  "age": 21,
  "address": {
    "streetAddress": "Jl. Gatot Subroto",
    "city": "Purwokerto",
    "state": "Central Java"
  },
  "courses": [
    {
      "code": "CRI2C4",
      "name": "Konstruksi Perangkat Lunak"
    },
    {
      "code": "CRI2XX",
      "name": "Kecerdasan Buatan"
    }
  ]
}
```

File DataMahasiswa2211104024.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        // 3 references
41        public Address address { get; set; }
42        // 1 reference
43        public Course[] courses { get; set; }
44    }
45 }
46
47 // 0 references
48 public static void ReadJSON()
49 {
50     string filePath = "jurnal7_1_2211104005.json";
51
52     if (File.Exists(filePath))
53     {
54         string jsonString = File.ReadAllText(filePath);
55
56         Mahasiswa mahasiswa = JsonSerializer.Deserialize<Mahasiswa>(jsonString);
57
58         Console.WriteLine("----- Data Mahasiswa -----");
59         Console.WriteLine($"Nama : {mahasiswa.firstName} {mahasiswa.lastName}");
60         Console.WriteLine($"Gender: {mahasiswa.gender}");
61         Console.WriteLine($"Usia : {mahasiswa.age}");
62         Console.WriteLine($"Alamat: {mahasiswa.address.streetAddress}, " +
63             $"{mahasiswa.address.city}, {mahasiswa.address.state}");
64         Console.WriteLine("\n----- Mata Kuliah -----");
65
66         foreach (var course in mahasiswa.courses)
67         {
68             Console.WriteLine($"Kode: {course.code} - Nama: {course.name}");
69         }
70     }
71     else
72     {
73         Console.WriteLine("File JSON tidak ditemukan.");
74     }
75 }

```

## File Program.cs

```

1 class Program
2 {
3     static void Main(string[] args)
4     {
5         // DataMahasiswa2211104005.ReadJSON();
6         // TeamMembers2211104017.ReadJSON();
7         GlossaryItem2211104024.ReadJSON();
8     }
9 }

```

- **Screenshot hasil run**

```
Alamat : Jl. Kalibener, Purwokerto, Central Java

===== Mata Kuliah =====
Kode: CRI2C4 - Nama: Konstruksi Perangkat Lunak
Kode: CRI2XX - Nama: Kecerdasan Buatan
```

- **Penjelasan**

Program ini bertujuan untuk membaca dan menampilkan data mahasiswa yang disimpan dalam file berformat JSON dengan memanfaatkan teknik parsing. File bernama jurnal7\_1\_2211104024.json berisi informasi mahasiswa, seperti nama, usia, jenis kelamin, alamat, serta daftar mata kuliah. File tersebut ditempatkan di dalam direktori proyek, dan program akan membaca serta mengonversinya menjadi objek C#.

Parsing di sini berarti mengubah data teks dalam format JSON menjadi objek yang dapat diolah di dalam kode program. Di file DataMahasiswa2211104024.cs, didefinisikan kelas Mahasiswa beserta subkelas Address dan Course yang merepresentasikan struktur data dalam JSON.

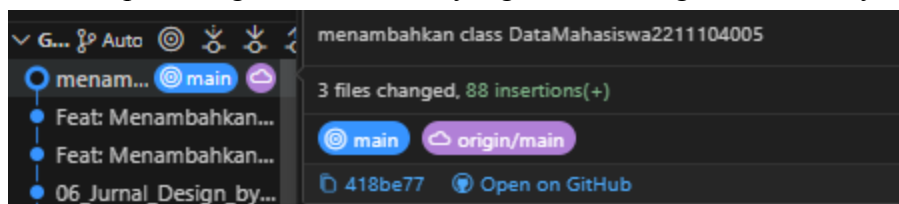
Metode ReadJSON() digunakan untuk membaca isi file menggunakan File.ReadAllText() dan kemudian melakukan parsing dengan JsonSerializer.Deserialize<Mahasiswa>() agar data dalam file dapat dipetakan ke dalam objek bertipe Mahasiswa. Setelah parsing selesai, informasi mahasiswa ditampilkan dalam format yang lebih rapi dan mudah dibaca.

File Program.cs memiliki peran untuk memanggil metode ReadJSON() saat program dijalankan, sehingga proses pembacaan dan konversi data dilakukan secara otomatis. Dengan pendekatan ini, program mampu mengakses dan mengelola data JSON dengan efisien.

## 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\_2211104024.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 "TeamMembers2211104024.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

```
class Program
{
    static void Main(string[] args)
    {
        // DataMahasiswa2211104005.ReadJSON();
        // TeamMembers2211104017.ReadJSON();
        GlossaryItem2211104024.ReadJSON();
    }
}
```

- **Screenshot hasil run**

```
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 dibuat untuk membaca dan menampilkan informasi anggota kelompok dari file JSON dengan menggunakan teknik parsing. File bernama jurnal7\_2\_2211104024.json berisi data anggota tim seperti NIM, nama lengkap, jenis kelamin, dan usia dalam format JSON. File ini diletakkan di dalam folder proyek agar dapat diakses dengan mudah oleh program.

Parsing dilakukan untuk mengonversi data teks JSON menjadi objek yang dapat dimanipulasi di dalam bahasa pemrograman C#. Pada file TeamMembers2211104024.cs, program membaca isi file menggunakan `File.ReadAllText()` dan memparsing-nya dengan `JsonSerializer.Deserialize<Team>()`, sehingga data dapat diubah menjadi objek bertipe `Team`. Objek ini memiliki properti `members`, yaitu sebuah array yang menyimpan detail tiap anggota kelompok.

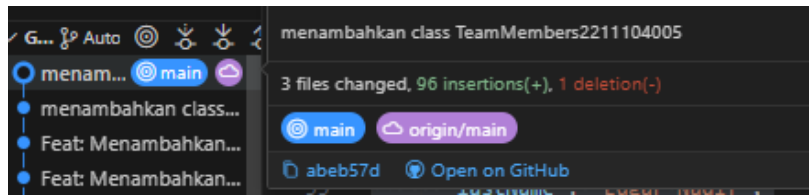
Setelah proses parsing selesai, data masing-masing anggota akan ditampilkan ke console sesuai format yang telah ditentukan. Proses ini dijalankan melalui file `Program.cs`, yang memanggil metode `ReadJSON()` saat program dieksekusi, sehingga pembacaan dan penampilan data berjalan secara otomatis dan sesuai dengan yang diharapkan.

#### 4. MELAKUKAN COMMIT

Pada branch main/master:

- Lakukan commit dengan pesan “menambahkan class <NAMA\_CLASS>”.

- b. 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\_2211104024.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 GlossaryItem2211104024.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

Program ini bertujuan untuk membaca dan menampilkan informasi dari file JSON berjudul "jurnal7\_3\_2211104024.json", yang berisi data mengenai sebuah entri glossary. Struktur utama file JSON tersebut terdiri dari elemen "glossary", yang di dalamnya terdapat "GlossDiv", lalu "GlossList", hingga ke "GlossEntry" yang menyimpan detail seperti ID, GlossTerm, Acronym, Abbreviation, Definition, serta referensi tambahan.

Program ini menggunakan teknik parsing (deserialisasi) untuk mengonversi data dalam format teks JSON menjadi objek C# yang dapat diolah lebih lanjut.

Pada file GlossaryItem2211104024.cs, didefinisikan sejumlah kelas yang merepresentasikan struktur hierarki dari JSON tersebut. Di dalam metode ReadJSON(), program membaca isi file menggunakan File.ReadAllText(), lalu memprosesnya menggunakan JsonSerializer.Deserialize<Root>(). Setelah proses parsing berhasil, data dari bagian GlossEntry akan diambil dan ditampilkan ke console dalam format yang rapi dan mudah dibaca.

Dengan teknik parsing ini, program dapat mengakses langsung elemen-elemen penting seperti ID, GlossTerm, Acronym, dan referensi yang terdapat di GlossSeeAlso, tanpa perlu membaca file secara manual baris demi baris, sehingga proses menjadi jauh lebih efisien.

## 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.

