

**TUGAS JURNAL  
KONSTRUKSI PERANGKAT LUNAK**

**MODUL VII  
GRAMMAR-BASED INPUT PROCESSING PARSING**



**Disusun Oleh :**

**Atika Aji Hadiyani**

**2211104003**

**SE-06-01**

**Dosen Pengampu :**

**Yudha Islami Sulistya, S.Kom., M.Cs.**

**PROGRAM STUDI S1 SOFTWARE ENGINEERING FAKULTAS**

**INFORMATIKA**

**TELKOM UNIVERSITY PURWOKERTO**

**2025**

## TUGAS JURNAL

### 1. MENAMBAHKAN JSON DESERIALIZATON 1

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

- A. 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.
- B. Ganti isi dari file json tersebut dengan detail yang benar dari praktikan.
- C. Buatlah sebuah file class baru dengan nama “DataMahasiswa<NIM\_PRAKTIKAN>”.
- D. Buat method “ReadJSON() yang melakukan parsing untuk file tersebut menjadi object sesuai.
- E. 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\_2211104003.json

```
{
  "firstName": "Atika Aji",
  "lastName": "Hadiyani",
  "gender": "female",
  "age": 20,
  "address": {
    "streetAddress": "Jl. Gerilya",
    "city": "Purwokerto",
    "state": "Central Java"
  },
  "courses": [
    {
      "code": "CRI3R5",
      "name": "Konstruksi Perangkat Lunak"
    },
    {
      "code": "CRI2B8",
      "name": "Manajemen Konfigurasi dan Evolusi Perangkat Lunak"
    }
  ]
}
```

## File DataMahasiswa2211104003.cs

```
1  using System;
2  using System.IO;
3  using System.Text.Json;
4
5  0 references
6  public class DataMahasiswa2211104003
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 }
```

```
31  0 references
32  public static void ReadJSON()
33  {
34      string filePath = "jurnal7_1_2211104003.json";
35
36      if (File.Exists(filePath))
37      {
38          string jsonString = File.ReadAllText(filePath);
39
40          Mahasiswa mahasiswa = JsonSerializer.Deserialize<Mahasiswa>(jsonString);
41
42          Console.WriteLine("===== Data Mahasiswa =====");
43          Console.WriteLine($"Nama : {mahasiswa.firstName} {mahasiswa.lastName}");
44          Console.WriteLine($"Gender: {mahasiswa.gender}");
45          Console.WriteLine($"Usia : {mahasiswa.age}");
46          Console.WriteLine($"Alamat: {mahasiswa.address.streetAddress}, " +
47              $"{mahasiswa.address.city}, {mahasiswa.address.state}");
48          Console.WriteLine("\n===== Mata Kuliah =====");
49
50          foreach (var course in mahasiswa.courses)
51          {
52              Console.WriteLine($"Kode: {course.code} - Nama: {course.name}");
53          }
54      }
55      else
56      {
57          Console.WriteLine("File JSON tidak ditemukan.");
58      }
59 }
```

File Program.cs

```
1 class Program
2 {
3     0 references
4     static void Main(string[] args)
5     {
6         DataMahasiswa2211104003.ReadJSON();
7         ///TeamMembers2211104003.ReadJSON();
8         //GlossaryItem2211104003.ReadJSON();
9     }
```

#### - Screenshot hasil run

```
Microsoft Visual Studio Debu
===== Data Mahasiswa =====
Nama : Atika Aji Hadiyani
Gender: female
Usia : 20
Alamat: Jl. Gerilya, Purwokerto, Central Java

===== Mata Kuliah =====
Kode: CRI3R5 - Nama: Konstruksi Perangkat Lunak
Kode: CRI2B8 - Nama: Manajemen Konfigurasi dan Evolusi Perangkat Lunak
```

#### - Penjelasan

Program ini dibuat untuk membaca dan menampilkan data mahasiswa dari file JSON dengan menggunakan teknik parsing. File **jurnal7\_1\_2211104003.json** menyimpan data mahasiswa seperti nama, usia, gender, alamat, dan daftar mata kuliah. File ini disimpan di dalam folder proyek, lalu program akan membacanya dan mengubah (parsing) data tersebut menjadi objek di C#. Parsing ini pada dasarnya adalah proses mengonversi teks JSON menjadi struktur objek yang bisa digunakan dalam kode.

Di dalam file **DataMahasiswa2211104003.cs**, ada kelas **Mahasiswa** beserta sub-kelas **Address** dan **Course** untuk menyesuaikan struktur JSON. Metode **ReadJSON()** akan membaca file menggunakan **File.ReadAllText()**, lalu memprosesnya dengan **JsonSerializer.Deserialize()**, supaya isi file bisa diubah menjadi objek **Mahasiswa**. Setelah itu, program akan menampilkan datanya ke console dalam format yang lebih gampang dibaca.

Sementara itu, file **Program.cs** bertugas untuk memanggil metode **ReadJSON()**, sehingga parsing berjalan otomatis saat program dijalankan. Dengan teknik ini, program bisa membaca dan mengolah data JSON dengan lebih 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.

DataMahasiswa2211104003.cs	Menambahkan Kode Program DataMahasiswa2211104003
jurnal7_1_2211104003.json	Menambahkan file jurnal7_1_2211104003

### 3. MENAMBAHKAN JSON DESERIALIZATON 2

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

- A. 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.
- B. Ubah isi dari file json tersebut dengan daftar anggota kelompok (untuk tubes).
- C. Buatlah sebuah file class baru dengan nama “TeamMembers<NIM\_PRAKTIKAN>”.
- D. Buat method “ReadJSON() yang melakukan parsing untuk file tersebut menjadi object sesuai.
- E. 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.

**Jawab:**

**- Source Code**

File jurnal7\_2\_2211104003.json

```
1  {
2    "members": [
3      {
4        "firstName": "Atika Aji",
5        "lastName": "Hadiyani",
6        "gender": "Female",
7        "age": 20,
8        "nim": "2211104003"
9      },
10     {
11       "firstName": "Kholifah",
12       "lastName": "Dina",
13       "gender": "Female",
14       "age": 21,
15       "nim": "2211104004"
16     },
17     {
18       "firstName": "Afad Fath",
19       "lastName": "Musyarof",
20       "gender": "Male",
21       "age": 21,
22       "nim": "2211104030"
23     },
24     {
25       "firstName": "Rachmat",
26       "lastName": "Taufik",
27       "gender": "Male",
28       "age": 21,
29       "nim": "2211104028"
30     },
31     {
32       "firstName": "Noor Alam",
33       "lastName": "Fauzan",
34       "gender": "Male",
35       "age": 21,
36       "nim": "2211104031"
37     }
38   ]
39 }
```

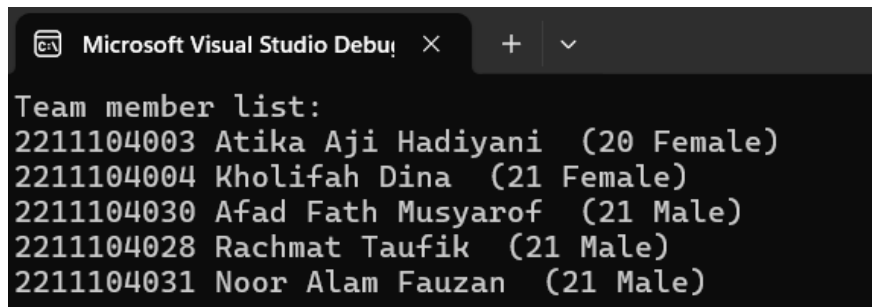
## File TeamMember2211104003.cs

```
1  using System;
2  using System.IO;
3  using System.Text.Json;
4
5  0 references
6  class TeamMembers2211104003
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_2211104003.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

```
1  0 references
2  class Program
3  {
4      0 references
5      static void Main(string[] args)
6      {
7          //DataMahasiswa2211104003.ReadJSON();
8          TeamMembers2211104003.ReadJSON();
9          //GlossaryItem2211104003.ReadJSON();
10     }
11 }
```

- **Screenshot hasil run**



```
Team member list:
2211104003 Atika Aji Hadiyani (20 Female)
2211104004 Kholifah Dina (21 Female)
2211104030 Afad Fath Musyarof (21 Male)
2211104028 Rachmat Taufik (21 Male)
2211104031 Noor Alam Fauzan (21 Male)
```

- **Penjelasan**

Program ini dibuat untuk membaca dan menampilkan daftar anggota kelompok dari file JSON dengan teknik parsing. File `jurnal7_2_2211104003.json` menyimpan data anggota tim dalam format JSON, termasuk NIM, nama, gender, dan usia setiap anggota. File ini disimpan di dalam folder proyek supaya program bisa membacanya dengan mudah. Parsing sendiri adalah proses mengubah data JSON (yang masih berbentuk teks) menjadi objek dalam bahasa C#, sehingga datanya bisa langsung digunakan dalam program.

Di dalam file `TeamMember2211104003.cs`, program membaca file JSON dengan `File.ReadAllText()`, lalu memprosesnya menggunakan `JsonSerializer.Deserialize()`, yang akan mengonversi data JSON menjadi objek `Team`. Objek ini berisi array `members`, yang menyimpan informasi setiap anggota kelompok. Setelah parsing selesai, program akan mencetak daftar anggota ke layar sesuai format yang sudah ditentukan. Sementara itu, file `Program.cs` bertugas memanggil metode `ReadJSON()`, supaya proses parsing dan tampilan data langsung berjalan saat program dijalankan.

#### 4. MELAKUKAN COMMIT

Pada branch yang dibuat sebelumnya:

- A. 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.

- A. 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.
- B. Buatlah sebuah file class baru dengan nama “GlossaryItem<NIM\_PRAKTIKAN>”.
- C. Buat method “ReadJSON() yang melakukan parsing untuk file tersebut menjadi object sesuai.
- D. Pada method tersebut, lakukan print hasil deserialisasi dari object yang dibuat dengan format bebas untuk bagian “GlossEntry” saja.

**Jawab:**

### - Source Code

File jurnal7\_3\_2211104003.json

```
1  {
2    "glossary": {
3      "title": "example glossary",
4      "GlossDiv": {
5        "title": "S",
6        "GlossList": {
7          "GlossEntry": {
8            "ID": "SGML",
9            "SortAs": "SGML",
10           "GlossTerm": "Standard Generalized Markup Language",
11           "Acronym": "SGML",
12           "Abbrev": "ISO 8879:1986",
13           "GlossDef": {
14             "para": "A meta-markup language, used to create markup languages such as DocBook.",
15             "GlossSeeAlso": [
16               "GML",
17               "XML"
18             ]
19           },
20           "GlossSee": "Markup"
21         }
22       }
23     }
24   }
25 }
```

File GlossaryItem2211104003.cs

```
1  using System;
2  using System.IO;
3  using System.Text.Json;
4
5  1 reference
6  class GlossaryItem2211104003
7  {
8      1 reference
9      class GlossDef
10     {
11         1 reference
12         public string para { get; set; }
13         1 reference
14         public string[] GlossSeeAlso { get; set; }
15     }
16
17     2 references
18     class GlossEntry
19     {
20         1 reference
21         public string ID { get; set; }
22         1 reference
23         public string SortAs { get; set; }
24         1 reference
25         public string GlossTerm { get; set; }
26         1 reference
27         public string Acronym { get; set; }
28         1 reference
29         public string Abbrev { get; set; }
30         2 references
31         public GlossDef GlossDef { get; set; }
32         1 reference
33         public string GlossSee { get; set; }
34     }
35
36     1 reference
37     class GlossList
38     {
39         1 reference
40         public GlossEntry GlossEntry { get; set; }
41     }
42
43     1 reference
44     class GlossDiv
45     {
46         0 references
47         public string title { get; set; }
48         1 reference
49         public GlossList GlossList { get; set; }
50     }
51 }
```

```

35 1 reference
36 class Glossary
37 {
38     0 references
39     public string title { get; set; }
40     1 reference
41     public GlossDiv GlossDiv { get; set; }
42 }
43
44 2 references
45 class Root
46 {
47     1 reference
48     public Glossary glossary { get; set; }
49 }
50
51 0 references
52 public static void ReadJSON()
53 {
54     string filePath = "jurnal7_3_2211104003.json";
55
56     if (!File.Exists(filePath))
57     {
58         Console.WriteLine("File JSON tidak ditemukan.");
59         return;
60     }
61
62     // Membaca dan parsing JSON
63     Root data = JsonSerializer.Deserialize<Root>(File.ReadAllText(filePath));
64
65     // Mengambil bagian GlossEntry
66     GlossEntry entry = data.glossary.GlossDiv.GlossList.GlossEntry;
67
68     Console.WriteLine("==== Glossary Entry =====");
69     Console.WriteLine($"ID : {entry.ID}");
70     Console.WriteLine($"SortAs : {entry.SortAs}");
71     Console.WriteLine($"Term : {entry.GlossTerm}");
72     Console.WriteLine($"Acronym : {entry.Acronym}");
73     Console.WriteLine($"Abbrev : {entry.Abbrev}");
74     Console.WriteLine($"Definition: {entry.GlossDef.para}");
75     Console.WriteLine($"See Also : ");
76     Console.WriteLine(string.Join(", ", entry.GlossDef.GlossSeeAlso));
77     Console.WriteLine($"GlossSee : {entry.GlossSee}");
78 }
79

```

#### File Program.cs

```

1 0 references
2 class Program
3 {
4     0 references
5     static void Main(string[] args)
6     {
7         //DataMahasiswa2211104003.ReadJSON();
8         //TeamMembers2211104003.ReadJSON();
9         GlossaryItem2211104003.ReadJSON();
10    }
11 }

```

#### - 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 dibuat untuk membaca dan menampilkan data dari file JSON "jurnal7\_3\_2211104003.json", yang berisi informasi glossary. Struktur datanya terdiri dari **glossary**, di dalamnya ada **GlossDiv**, lalu **GlossList**, dan **GlossEntry** yang menyimpan data utama seperti ID, GlossTerm, Acronym, Abbreviation, Definition, serta referensi terkait. Program membaca file JSON dan menggunakannya dalam C# dengan teknik parsing (deserialisasi JSON), mengubah data dari format teks JSON menjadi objek yang bisa diproses.

Di dalam file `GlossaryItem2211104003.cs`, program mendefinisikan kelas sesuai dengan struktur JSON. Method `ReadJSON()` membaca file dengan **`File.ReadAllText()`**, kemudian melakukan parsing menggunakan **`JsonSerializer.Deserialize()`**. Setelah data diproses, program mengambil **`GlossEntry`** dan mencetak informasinya ke konsol dengan format yang lebih rapi. Teknik parsing ini mempermudah akses ke elemen JSON seperti `ID`, `GlossTerm`, dan referensi di **`GlossSeeAlso`**, tanpa perlu membaca file secara manual.

## 6. MELAKUKAN COMMIT

Pada branch yang dibuat sebelumnya:

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

