

**TUGAS JURNAL
KONSTRUKSI PERANGKAT LUNAK
MODUL VII
GRAMMAR-BASED INPUT PROCESSING (PARSING)**



**Disusun Oleh :
Ahmad Junaidi / 2211104002
SE-06-01**

**Asisten Praktikum:
Naufal El Kamil Aditya Pratama Rahman
Imelda**

**Dosen Pengampu :
Yudha Islami Sulistya, S.Kom., M.Cs.**

**PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO**

2025

TUGAS JURNAL

1. Link Github Repository:
https://github.com/Ahmadjunaidi101105/KPL_AHMAD-JUNAIIDI_2211104002_SE0601/tree/main/07_Grammar-Based_Input_Processing_Parsing
2. Screenshot hasil run dan potongan code, penjelasan singkat (hasil console output untuk masing-masing hasil deserialisasi)
 - a. Class Data Mahasiswa

Source Code

Program.cs :

```
0 references
1  class Program
2  {
3      0 references
4      static void Main()
5      {
6          DataMahasiswa_2211104002.ReadJSON();
7      }
8  }
```

Jurnal7_1_2211104002.json :

```
1  {
2      "firstName": "Ahmad ",
3      "lastName": "Junaidi",
4      "gender": "male",
5      "age": 21,
6      "address": {
7          "streetAddress": "Jatimulyo",
8          "city": "Kebumen",
9          "state": "Central Java"
10     },
11     "courses": [
12         {
13             "code": "CRI2C4",
14             "name": "Konstruksi Perangkat Lunak"
15         },
16         {
17             "code": "CRI2XX",
18             "name": "Nama Mata Kuliah"
19         }
20     ]
21 }
```

DataMahasiswa2211104002.cs :

```
1  using System;
2  using System.IO;
3  using System.Collections.Generic;
4  using Newtonsoft.Json;
5
6  namespace modul7_kelompok_3
7  {
8      public class Address
9      {
10         public string StreetAddress { get; set; }
11         public string City { get; set; }
12         public string State { get; set; }
13     }
14
15     public class Course
16     {
17         public string Code { get; set; }
18         public string Name { get; set; }
19     }
20
21     public class Person
22     {
23         public string FirstName { get; set; }
24         public string LastName { get; set; }
25         public string Gender { get; set; }
26         public int Age { get; set; }
27         public Address Address { get; set; }
28         public List<Course> Courses { get; set; }
29     }
30
31     public class DataMahasiswa2211104002
32     {
33         public static void ReadJSON()
34         {
35             string path = @"D:\modul7_kelompok_3\Jurnal7_1_2211104002.json";
36
37             if (File.Exists(path))
38             {
39                 string jsonData = File.ReadAllText(path);
40                 var person = JsonConvert.DeserializeObject<Person>(jsonData);
41
42                 Console.WriteLine($"Name: {person.FirstName} {person.LastName}");
43                 Console.WriteLine($"Gender: {person.Gender}");
44                 Console.WriteLine($"Age: {person.Age}");
45                 Console.WriteLine($"Address: {person.Address.StreetAddress}, {person.Address.City}, {person.Address.State}");
46                 Console.WriteLine("Courses:");
47
48                 foreach (var course in person.Courses)
49                 {
50                     Console.WriteLine($"- {course.Code}: {course.Name}");
51                 }
52             }
53             else
54             {
55                 Console.WriteLine("File JSON tidak ditemukan!");
56             }
57         }
58     }
59 }
```

Output :

```
Microsoft Visual Studio Debug Console
Mencari file JSON di: C:\SEMESTER 6\PRAKTIKUM KONSTRUKSI PERANGKAT LUNAK\modul7_1_2211104002.json
===== Data Mahasiswa =====
Nama      : Ahmad Junaidi
Gender    : male
Age       : 21
Alamat    : Jatimulyo, Kebumen, Central Java
===== Mata Kuliah =====
Kode: CRI2C4, Nama: Konstruksi Perangkat Lunak
Kode: CRI2XX, Nama: Nama Mata Kuliah
=====

C:\SEMESTER 6\PRAKTIKUM KONSTRUKSI PERANGKAT LUNAK\modul7_kelompok7\modul7_
cess 18548) exited with code 0 (0x0).
To automatically close the console when debugging stops, enable Tools->Opti
le when debugging stops.
Press any key to close this window . . .|
```

Penjelasan :

Program ini dibuat untuk membaca data mahasiswa dari sebuah file JSON dan menampilkannya ke layar konsol. Program terdiri dari dua bagian utama, yaitu class DataMahasiswa2211104002 yang bertanggung jawab membaca dan

memproses data dari file JSON, serta class Program yang menjalankan metode pembacaan tersebut.

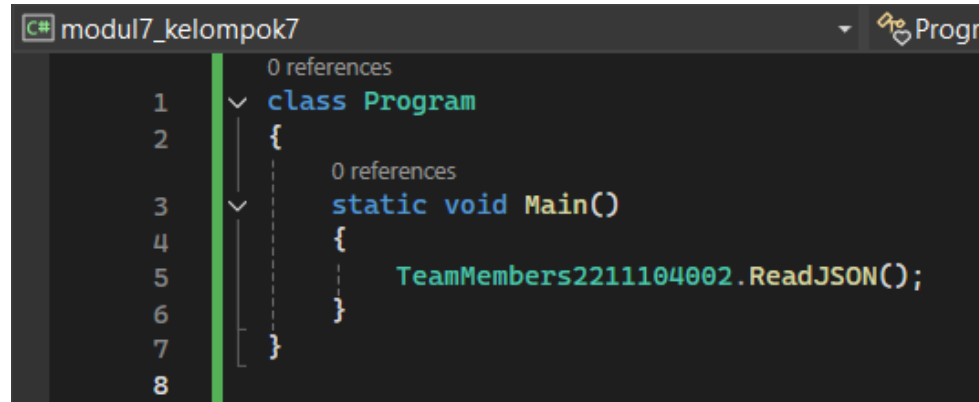
Pada class DataMahasiswa2211104002, metode ReadJSON() digunakan untuk membaca file JSON yang berisi informasi mahasiswa, seperti nama, jenis kelamin, usia, alamat, dan daftar mata kuliah yang diambil. Program menggunakan library Newtonsoft.Json untuk melakukan deserialisasi JSON menjadi objek dalam bahasa C#. Setelah data berhasil dibaca, informasi tersebut akan ditampilkan di layar konsol secara terstruktur.

Sementara itu, pada class Program, terdapat metode Main() yang menjadi titik awal eksekusi program. Di dalamnya, metode ReadJSON() dipanggil untuk memproses data JSON dan menampilkan outputnya. Untuk memastikan program dapat berjalan dengan baik, kedua kelas ini harus berada dalam namespace yang sama, serta file JSON harus tersedia di lokasi yang benar sesuai dengan path yang telah ditentukan. Jika file tidak ditemukan, program akan menampilkan pesan kesalahan sebagai pemberitahuan kepada pengguna. Dengan implementasi ini, program mampu membaca data mahasiswa dari JSON secara otomatis dan menyajikannya dengan format yang mudah dipahami, sehingga dapat digunakan sebagai referensi dalam pengelolaan data mahasiswa berbasis file JSON.

b. Class Team Members

Source Code

Program.cs :



```
C# modul7_kelompok7
0 references
1 class Program
2 {
3     0 references
4     static void Main()
5     {
6         TeamMembers2211104002.ReadJSON();
7     }
8 }
```

Jurnal7_2_2211104002.json :

```

1  {
2  "members": [
3  {
4      "firstName": "Ahmad junaidi",
5      "lastName": "Junaidi",
6      "gender": "male",
7      "age": 21,
8      "nim": "2211104002"
9  },
10 {
11     "firstName": " Alfian mutakim",
12     "lastName": "Mutakim",
13     "gender": "male",
14     "age": 21,
15     "nim": "2211104002"
16 },
17 {
18     "firstName": "Dhimas Afrizal",
19     "lastName": "Aftizal",
20     "gender": "male",
21     "age": 21,
22     "nim": "2211104002"
23 },
24 {
25     "firstName": "Nita",
26     "lastName": "Fitrotul mar ah",
27     "gender": "male",
28     "age": 21,
29     "nim": "2211104002"
30 },
31 {
32     "firstName": " nadia",
33     "lastName": "rahmaniar",
34     "gender": "male",
35     "age": 21,
36     "nim": "2211104002"
37 },
38 {
39     "firstName": "muhammad edgar",
40     "lastName": "Nadhif",
41     "gender": "male",
42     "age": 21,
43     "nim": "2211104002"
44 },
45 {
46     "firstName": " Rafli dhafin",
47     "lastName": "Kamil",
48     "gender": "male",
49     "age": 21,
50     "nim": "2211104002"
51 }
52 ]
53 }

```

TeamMembers2211104002.cs :

```

1  using System;
2  using System.IO;
3  using Newtonsoft.Json;
4  using System.Collections.Generic;
5
6  class TeamMember
7  {
8      public string firstName { get; set; }
9      1 reference
10     public string lastName { get; set; }
11     1 reference
12     public string gender { get; set; }
13     1 reference
14     public int age { get; set; }
15     1 reference
16     public string nim { get; set; }
17 }
18
19 3 references
20  class TeamMembers2211104002
21  {
22      1 reference
23      public List<TeamMember> members { get; set; }
24
25      1 reference
26      public static void ReadJSON()
27      {
28          try
29          {
30              string filePath = @"C:\SEMESTER 6\PRAKTIKUM KONSTRUKSI PERANGKAT LUNAK\modul7_kelompok7\bin\Debug\jurnal7_2_2211104002.json";
31              Console.WriteLine($"Mencari file JSON di: {filePath}");
32
33              if (!File.Exists(filePath))
34              {
35                  Console.WriteLine("File JSON tidak ditemukan.");
36                  return;
37              }
38
39              string jsonData = File.ReadAllText(filePath);
40              TeamMembers2211104002 team = JsonConvert.DeserializeObject<TeamMembers2211104002>(jsonData);
41
42              Console.WriteLine("Team member list:");
43              foreach (var member in team.members)
44              {
45                  Console.WriteLine($"{member.nim} {member.firstName} {member.lastName} ({member.age} {member.gender})");
46              }
47
48              catch (Exception ex)
49              {
50                  Console.WriteLine($"Terjadi kesalahan: {ex.Message}");
51              }
52          }
53      }
54  }

```

Output :

```

Microsoft Visual Studio Debug Console
Mencari file JSON di: C:\SEMESTER 6\PRAKTIKUM KONSTRUKSI PERANGKAT LUNAK\modul7_kelompok7\bin\Debug\jurnal7_2_2211104002.json
Team member list:
2211104002 Ahmad junaidi Junaidi (21 male)
2211104002 Alfian mutakim Mutakim (21 male)
2211104002 Dhimas Afrizal Aftizal (21 male)
2211104002 Nita Fitrotul mar ah (21 male)
2211104002 nadia rahmaniar (21 male)
2211104002 muhammad edgar Nadhif (21 male)
2211104002 Rafli dhafin kamil (21 male)

C:\SEMESTER 6\PRAKTIKUM KONSTRUKSI PERANGKAT LUNAK\modul7_kelompok7\bin\Debug\jurnal7_2_2211104002.exe (Process 11360) exited with code 0 (0x0).
To automatically close the console when debugging stops, enable Tools |> Options |> Debugging |> Automatically close console when debugging stops.
Press any key to close this window . . .|

```

Penjelasan :

Program ini bertujuan untuk membaca data anggota tim dari file JSON dan menampilkannya di konsol. Program.cs hanya bertanggung jawab untuk memanggil metode `DisplayTeamData()`, yang terdapat dalam kelas TeamMembers2211104002, sehingga kode menjadi lebih modular. Pada TeamMembers2211104002.cs, terdapat dua metode utama: `ReadJSON()`, yang membaca file JSON dan mengubahnya menjadi objek Team, serta `DisplayTeamData()`, yang menampilkan data anggota tim jika file berhasil dibaca. Jika file tidak ditemukan atau tidak valid, program akan memberikan pesan kesalahan yang sesuai. Dengan pendekatan ini, program menjadi lebih terstruktur, mudah dipahami, dan fleksibel.

c. Class GlossaryItem

Source Code

Program.cs :

```
1  using System;
2
3  0 references
4  class Program
5  {
6      0 references
7      static void Main()
8      {
9          GlossaryItem2211104002.ReadJSON()
10     }
```

Jurnal7_3_2211104002.json :

File: <No Schema Selected>

```
{
  "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"
        }
      }
    }
  }
}
```

GlossaryItem2211104002.cs :

```

public class GlossEntry
{
    1 reference
    public string ID { get; set; }
    0 references
    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; }
    1 reference
    public GlossDef GlossDef { get; set; }
    0 references
    public string GlossSee { get; set; }
}

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

1 reference
public static void ReadJSON()
{
    try
    {
        string filePath = "Jurnal7_3_22111040802.json"; // Pastikan file berada dalam direktori proyek
        if (!File.Exists(filePath))
        {
            Console.WriteLine("File JSON tidak ditemukan.");
            return;
        }

        string jsonData = File.ReadAllText(filePath);
        Glossary glossary = JsonConvert.DeserializeObject<Glossary>(jsonData);

        Console.WriteLine("==== Glossary Entry =====");
        Console.WriteLine($"Title       : {glossary.Title}");
        Console.WriteLine($"Glossary ID : {glossary.GlossDiv.GlossList.GlossEntry.ID}");
        Console.WriteLine($"Term       : {glossary.GlossDiv.GlossList.GlossEntry.GlossTerm}");
        Console.WriteLine($"Acronym    : {glossary.GlossDiv.GlossList.GlossEntry.Acronym}");
        Console.WriteLine($"Abbreviation: {glossary.GlossDiv.GlossList.GlossEntry.Abbrev}");
        Console.WriteLine($"Description: {glossary.GlossDiv.GlossList.GlossEntry.GlossDef.Paragraph}");
        Console.WriteLine("=====");
    }
    catch (Exception ex)
    {
        Console.WriteLine($"Terjadi kesalahan: {ex.Message}");
    }
}

```

Output :

```

=== Membaca Data Glossary dari JSON ===

--- Glossary Entry ---
ID: SGML
Sort As: SGML
Gloss Term: Standard Generalized Markup Language
Acronym: SGML
Abbreviation: ISO 8879:1986
Definition: A meta-markup language, used to create ma
GlossSeeAlso: GML, XML
GlossSee: markup

```

Penjelasan :

Program ini membaca dan menampilkan data glossary dari file JSON menggunakan C#. Program.cs hanya memanggil metode 'ReadJSON()' dari kelas GlossaryItem untuk menjaga modularitas. GlossaryItem.cs menangani parsing JSON menggunakan Newtonsoft.Json, kemudian menampilkan informasi glossary seperti ID, istilah, akronim, singkatan, dan definisi. Jika

file tidak ditemukan atau terjadi kesalahan, program akan menampilkan pesan error yang sesuai.