

**JURNAL
KONSTRUKSI PERANGKAT LUNAK**

**PERTEMUAN 7
GRAMMAR-BASED INPUT PROCESSING (PARSING)**



**Disusun Oleh :
Muhammad Abdul Aziz
2211104026
SE0601**

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

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

**PROGRAM STUDI S1 SOFTWARE ENGINEERING
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2025**

1. Link Github Repository

https://github.com/akuazizz/KPL_MUHAMMAD-ABDUL-AZIZ_2211104026_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 :

```
kelompok_3 modul7_kelompok_3.Program Main
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace modul7_kelompok_3
8 {
9     class Program
10     {
11         static void Main(string[] args)
12         {
13             Console.WriteLine("=== Membaca Data Mahasiswa dari JSON ===");
14             DataMahasiswa2211104026.ReadJSON();
15         }
16     }
17 }
18
```

Jurnal7_1_2211104026.json :

```
na: <No Schema Selected>
{
  "firstName": "Abdul",
  "lastName": "Aziz",
  "gender": "male",
  "age": 21,
  "address": {
    "streetAddress": "Merden",
    "city": "Banjarnegara",
    "state": "Central Java"
  },
  "courses": [
    {
      "code": "CCK3IAB2",
      "name": "TATA TULIS ILMIAH"
    },
    {
      "code": "CCK3KAB3",
      "name": "MANAJEMEN KONFIGURASI DAN EVOLUSI PERANGKAT LUNAK"
    },
    {
      "code": "CCK2KAB4",
      "name": "KONSTRUKSI PERANGKAT LUNAK"
    },
    {
      "code": "CCK2JAC2",
      "name": "PROYEK TINGKAT II"
    },
    {
      "code": "CCK2KAB4",
      "name": "PRAKTIKUM KONSTRUKSI PERANGKAT LUNAK"
    },
    {
      "code": "CCK2DAB3",
      "name": "PROSES PERANGKAT LUNAK"
    },
    {
      "code": "CCK1LAB3",
      "name": "DESIGN THINKING"
    },
    {
      "code": "CCK2LAB3",
      "name": "KECERDASAN BUATAN"
    }
  ]
}
```

DataMahasiswa2211104026.cs :

modul7_kelompok_3 DataMahasiswa2211104026

```
1 using System;
2 using System.IO;
3 using System.Collections.Generic;
4 using Newtonsoft.Json;
5
6 1 reference
7 public class Address
8 {
9     1 reference
10     public string StreetAddress { get; set; }
11     1 reference
12     public string City { get; set; }
13     1 reference
14     public string State { get; set; }
15 }
16
17 1 reference
18 public class Course
19 {
20     1 reference
21     public string Code { get; set; }
22     1 reference
23     public string Name { get; set; }
24 }
25
26 1 reference
27 public class Person
28 {
29     1 reference
30     public string FirstName { get; set; }
31     1 reference
32     public string LastName { get; set; }
33     1 reference
34     public string Gender { get; set; }
35     1 reference
36     public int Age { get; set; }
37     3 references
38     public Address Address { get; set; }
39     1 reference
40     public List<Course> Courses { get; set; }
41 }
42
43 1 reference
44 public class DataMahasiswa2211104026
45 {
46     1 reference
47     public static void ReadJSON()
48     {
49         string path = @"E:\KPL\modul7_kelompok_3\modul7_kelompok_3\jurnal7_1_2211104026.json";
50         if (File.Exists(path))
51         {
52             string jsonData = File.ReadAllText(path);
53             var person = JsonConvert.DeserializeObject<Person>(jsonData);
54             Console.WriteLine($"Name: {person.FirstName} {person.LastName}");
55             Console.WriteLine($"Gender: {person.Gender}");
56             Console.WriteLine($"Age: {person.Age}");
57             Console.WriteLine($"Address: {person.Address.StreetAddress}, {person.Address.City}, {person.Address.State}");
58             Console.WriteLine("Courses:");
59             foreach (var course in person.Courses)
60             {
61                 Console.WriteLine($" - {course.Code}: {course.Name}");
62             }
63         }
64         else
65         {
66             Console.WriteLine("File JSON tidak ditemukan!");
67         }
68     }
69 }
```

Output :

Microsoft Visual Studio Debug Console

```
=== Membaca Data Mahasiswa dari JSON ===
Name: Abdul Aziz
Gender: male
Age: 21
Address: Merden, Banjarnegara, Central Java
Courses:
- CCK3IAB2: TATA TULIS ILMIAH
- CCK3KAB3: MANAJEMEN KONFIGURASI DAN EVOLUSI PERANGKAT LUNAK
- CCK2KAB4: KONSTRUKSI PERANGKAT LUNAK
- CCK2JAC2: PROYEK TINGKAT II
- CCK2KAB4: PRAKTIKUM KONSTRUKSI PERANGKAT LUNAK
- CCK2DAB3: PROSES PERANGKAT LUNAK
- CCK1LAB3: DESIGN THINKING
- CCK2LAB3: KECERDASAN BUATAN
```

Penjelasan :

Kode di atas merupakan implementasi deserialisasi JSON ke dalam objek C# menggunakan Newtonsoft.Json. Program ini terdiri dari beberapa kelas utama, yaitu `Address`, `Course`, dan `Person`, yang merepresentasikan data alamat, mata kuliah, dan identitas seseorang. Data dalam format JSON dibaca dari file eksternal yang berisi informasi mahasiswa, termasuk nama, gender, umur, alamat, serta daftar mata kuliah yang diambil.

Metode `ReadJSON()` dalam kelas `DataMahasiswa2211104026` bertanggung jawab untuk membaca file JSON, melakukan deserialisasi menggunakan `JsonConvert.DeserializeObject<Person>()`, lalu menampilkan data mahasiswa ke konsol dalam format yang mudah dibaca. Program ini juga memiliki mekanisme pengecekan apakah file JSON tersedia sebelum diproses. Kelas `Program` hanya berisi method `Main()`, yang memanggil `ReadJSON()` untuk menjalankan proses deserialisasi. Dengan pemisahan tugas ini, kode menjadi lebih modular, mudah dibaca, dan dapat diperluas sesuai kebutuhan.

b. Class Team Members

Source Code

Program.cs :

```
7_kelompok3
modul7_kelompok3.Program
Main(string[] args)

1 using System;
2
3 namespace modul7_kelompok3
4 {
5     0 references
6     class Program
7     {
8         0 references
9         static void Main(string[] args)
10        {
11            Console.WriteLine("=== Membaca Data Anggota Tim dari JSON ===");
12            TeamMembers2211104026.DisplayTeamData(@"E:\KPL\modul7_kelompok3\modul7_kelompok3\jurnal7_2_2211104026.json");
13        }
14    }
15 }
```

Jurnal7_2_2211104026.json :

```
ma: <No Schema Selected>
1  {
2    "members" : [
3      {
4        "firstName": "Abdul",
5        "lastName": "Aziz",
6        "gender": "male",
7        "age": 21,
8        "nim": "2211104026"
9      },
10     {
11       "firstName": "Fadhila Agil",
12       "lastName": "Permana",
13       "gender": "male",
14       "age": 20,
15       "nim": "2211104006"
16     },
17     {
18       "firstName": "Luthfi",
19       "lastName": "Hamdani",
20       "gender": "male",
21       "age": 20,
22       "nim": "2211104023"
23     },
24     {
25       "firstName": "Aorinka",
26       "lastName": "Anendya",
27       "gender": "female",
28       "age": 21,
29       "nim": "2211104011"
30     }
31   ]
32 }
```

TeamMembers2211104026.cs :

```
modul7_kelompok_3 modul7_kelompok_3.TeamMembers2211104026
1  using System;
2  using System.IO;
3  using System.Collections.Generic;
4  using Newtonsoft.Json;
5
6  namespace modul7_kelompok_3
7  {
8      1 reference
9      public 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     3 references
24     public class Team
25     {
26         2 references
27         public List<Member> Members { get; set; }
28     }
29
30     1 reference
31     public class TeamMembers2211104026
32     {
33         1 reference
34         public static Team ReadJSON(string filePath)
35         {
36             try
37             {
38                 if (File.Exists(filePath))
39                 {
40                     string jsonData = File.ReadAllText(filePath);
41                     return JsonConvert.DeserializeObject<Team>(jsonData);
42                 }
43                 else
44                 {
45                     Console.WriteLine("File JSON tidak ditemukan!");
46                     return null;
47                 }
48             }
49             catch (Exception ex)
50             {
51                 Console.WriteLine($"Error membaca file JSON: {ex.Message}");
52                 return null;
53             }
54         }
55
56         1 reference
57         public static void DisplayTeamData(string filePath)
58         {
59             Team team = ReadJSON(filePath);
60
61             if (team != null && team.Members != null)
62             {
63                 Console.WriteLine("Daftar Anggota Tim:");
64                 foreach (var member in team.Members)
65                 {
66                     Console.WriteLine($"{member.NIM} - {member.FirstName} {member.LastName} ({member.Age} tahun, {member.Gender})");
67                 }
68             }
69             else
70             {
71                 Console.WriteLine("Data anggota tim tidak ditemukan atau file JSON tidak valid.");
72             }
73         }
74     }
75 }
```

Output :

```
Microsoft Visual Studio Debug Console
=== Membaca Data Anggota Tim dari JSON ===
Daftar Anggota Tim:
2211104026 - Abdul Aziz (21 tahun, male)
2211104006 - Fadhila Agil Permana (20 tahun, male)
2211104023 - Luthfi Hamdani (20 tahun, male)
2211104011 - Aorinka Anendya (21 tahun, female)
```

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 TeamMembers2211104026, sehingga kode menjadi lebih modular. Pada TeamMembers2211104026.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 :

```
modul7_kelompok_3 modul7_kelompok_3.Program Main
1 using System;
2
3 namespace modul7_kelompok_3
4 {
5     0 references
6     class Program
7     {
8         0 references
9         static void Main(string[] args)
10        {
11            Console.WriteLine("=== Membaca Data Glossary dari JSON ===");
12            GlossaryItem.ReadJSON(@"E:\KPL\modul7_kelompok_3\modul7_kelompok_3\jurnal7_3_2211104026.json");
13        }
14    }
15 }
```

Jurnal7_3_2211104026.json :

```
ema: <No Schema Selected>
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": ["GML", "XML"]
16          },
17          "GlossSee": "markup"
18        }
19      }
20    }
21  }
22 }
```

GlossaryItem2211104026.cs :

```
Miscellaneous Files Program
1 using System;
2 using System.IO;
3 using System.Collections.Generic;
4 using Newtonsoft.Json;
5
6 public class GlossDef
7 {
8     public string Para { get; set; }
9     public List<string> GlossSeeAlso { get; set; }
10 }
11
12 public class GlossEntry
13 {
14     public string ID { get; set; }
15     public string SortAs { get; set; }
16     public string GlossTerm { get; set; }
17     public string Acronym { get; set; }
18     public string Abbrev { get; set; }
19     public GlossDef GlossDef { get; set; }
20     public string GlossSee { get; set; }
21 }
22
23 public class GlossList
24 {
25     public GlossEntry GlossEntry { get; set; }
26 }
27
28 public class GlossDiv
29 {
30     public string Title { get; set; }
31     public GlossList GlossList { get; set; }
32 }
33
34 public class Glossary
35 {
36     public string Title { get; set; }
37     public GlossDiv GlossDiv { get; set; }
38 }
39
40 public class Root
41 {
42     public Glossary Glossary { get; set; }
43 }
```

```
Miscellaneous Files Program
43 }
44
45 public class Program
46 {
47     public static void ReadJSON(string filePath)
48     {
49         try
50         {
51             string jsonData = File.ReadAllText(filePath);
52             Root data = JsonConvert.DeserializeObject<Root>(jsonData);
53
54             if (data != null && data.Glossary?.GlossDiv?.GlossList?.GlossEntry != null)
55             {
56                 var entry = data.Glossary.GlossDiv.GlossList.GlossEntry;
57
58                 Console.WriteLine("\n--- Glossary Entry ---");
59                 Console.WriteLine($"ID: {entry.ID}");
60                 Console.WriteLine($"Sort As: {entry.SortAs}");
61                 Console.WriteLine($"Gloss Term: {entry.GlossTerm}");
62                 Console.WriteLine($"Acronym: {entry.Acronym}");
63                 Console.WriteLine($"Abbreviation: {entry.Abbrev}");
64                 Console.WriteLine($"Definition: {entry.GlossDef.Para}");
65                 Console.WriteLine($"GlossSeeAlso: " + string.Join(", ", entry.GlossDef.GlossSeeAlso));
66                 Console.WriteLine($"GlossSee: {entry.GlossSee}");
67             }
68             else
69             {
70                 Console.WriteLine("GlossEntry data not found.");
71             }
72         }
73         catch (Exception ex)
74         {
75             Console.WriteLine($"Error reading JSON file: {ex.Message}");
76         }
77     }
78
79     public static void Main()
80     {
81         string filePath = @"E:\Data IITP\Perkuliahan\Semester 6\KPL\Praktikum\Praktikum5\tpmodul7_kelompok_4\jurnal7_3_2211104105.json";
82         ReadJSON(filePath);
83     }
84 }
```

Output :

```
Microsoft Visual Studio Debug Console
=== 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 markup languages such as DocBook.
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. Dengan struktur ini, program menjadi lebih terorganisir, mudah dipelihara, dan fleksibel.