

**JURNAL
KONSTRUKSI PERANGKAT LUNAK**

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



**Disusun Oleh :
Mohammad Dhimas Afrizal
2211104023
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**

[Link Guthub Repository](#)

1. Screenshot hasil run dan potongan code, penjelasan singkat (hasil console output untuk masing-masing hasil deserialisasi)

a. Class Data Mahasiswa

```
1 using System;
2 using tpmodul7_kelompok_5;
3
4 class Program
5 {
6     static void Main(string[] args)
7     {
8         DataMahasiswa2211104023.ReadJSON();
9         Console.WriteLine("\n=====\\n");
10        KuliahMahasiswa2211104023.ReadJSON();
11    }
12 }
```

tp7_1_2211104023.json

```
{
  "nama": {
    "depan": "Mohammad",
    "belakang": "Dhimas Afrizal"
  },
  "nim": "2211104023",
  "fakultas": "informatika"
}
```

tp7_2_2211104023.json

```
1 {
2   "courses": [
3     {
4       "code": "CCK3KAB3",
5       "name": "MANAJEMEN KONFIGURASI DAN EVOLUSI PERANGKAT LUNAK"
6     },
7     {
8       "code": "CCK2LAB3",
9       "name": "KECERDASAN BUATAN"
10    },
11    {
12      "code": "CCK2DAB3",
13      "name": "PROSES PERANGKAT LUNAK"
14    },
15    {
16      "code": "CCK1LAB3",
17      "name": "DESIGN THINKING"
18    },
19    {
20      "code": "CCK2JAC2",
21      "name": "PROYEK TINGKAT II"
22    },
23    {
24      "code": "CCK2KAB4",
25      "name": "KONSTRUKSI PERANGKAT LUNAK"
26    },
27    {
28      "code": "CCK2KAB4",
29      "name": "PRAKTIKUM KONSTRUKSI PERANGKAT LUNAK"
30    },
31    {
32      "code": "CCK3IAB2",
33      "name": "TATA TULIS ILMIAH"
34    }
35  ]
36 }
```

DataMahasiswa2211104023.cs

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Text.Json;
6 using System.Text.Json.Serialization;
7 using System.Threading.Tasks;
8
9 namespace tpmodul7_kelompok_5
10 {
11     class Nama
12     {
13         [JsonPropertyName("depan")]
14         public string Depan { get; set; }
15
16         [JsonPropertyName("belakang")]
17         public string Belakang { get; set; }
18     }
19     class DataMahasiswa2211104023
20     {
21         [JsonPropertyName("nama")]
22         public Nama Nama { get; set; }
23
24         [JsonPropertyName("nim")]
25         public string Nim { get; set; }
26
27         [JsonPropertyName("fakultas")]
28         public string Fakultas { get; set; }
29
30         public static void ReadJSON()
31         {
32             string pathFile = Path.Combine(Directory.GetCurrentDirectory(), "tp7_1_2211104023.json");
33             if (File.Exists(pathFile))
34             {
35                 string jsonData = File.ReadAllText(pathFile);
36                 DataMahasiswa2211104023 mahasiswa = JsonSerializer.Deserialize<DataMahasiswa2211104023>(jsonData);
37
38                 Console.WriteLine($"Nama {mahasiswa.Nama.Depan} {mahasiswa.Nama.Belakang} dengan nim {mahasiswa.Nim} dari fakultas {mahasiswa.Fakultas}");
39             }
40             else
41             {
42                 Console.WriteLine("File tidak ditemukan");
43             }
44         }
45     }
46 }
```

KuliahMahasiswa2211104023.json

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Text.Json;
6 using System.Text.Json.Serialization;
7 using System.Threading.Tasks;
8
9 namespace tpmodul7_kelompok_5
10 {
11     class Course
12     {
13         [JsonPropertyName("code")]
14         public string Code { get; set; }
15
16         [JsonPropertyName("name")]
17         public string Name { get; set; }
18     }
19
20     class KuliahMahasiswa2211104023
21     {
22         [JsonPropertyName("courses")]
23         public List<Course> courses { get; set; }
24
25         public static void ReadJSON()
26         {
27             string pathFile = Path.Combine(Directory.GetCurrentDirectory(), "tp7_2_2211104023.json");
28             if (File.Exists(pathFile))
29             {
30                 string jsonData = File.ReadAllText(pathFile);
31                 KuliahMahasiswa2211104023 courses = JsonSerializer.Deserialize<KuliahMahasiswa2211104023>(jsonData);
32                 Console.WriteLine("Daftar mata kuliah yang diambil:");
33                 int index = 1;
34                 foreach (var course in courses.courses)
35                 {
36                     Console.WriteLine($"MK {index} {course.Code} - {course.Name}");
37                     index++;
38                 }
39             }
40             else
41             {
42                 Console.WriteLine("File tidak ditemukan");
43             }
44         }
45     }
46 }
```

Output:

```
Terminal - tpmodul7_kelompok_5

Nama Mohammad Dhimas Afrizal dengan nim 2211104023 dari fakultas informatika

=====

Daftar mata kuliah yang diambil:
MK 1 CCK3KAB3 - MANAJEMEN KONFIGURASI DAN EVOLUSI PERANGKAT LUNAK
MK 2 CCK2LAB3 - KECERDASAN BUATAN
MK 3 CCK2DAB3 - PROSES PERANGKAT LUNAK
MK 4 CCK1LAB3 - DESIGN THINKING
MK 5 CCK2JAC2 - PROYEK TINGKAT II
MK 6 CCK2KAB4 - KONSTRUKSI PERANGKAT LUNAK
MK 7 CCK2KAB4 - PRAKTIKUM KONSTRUKSI PERANGKAT LUNAK
MK 8 CCK3IAB2 - TATA TULIS ILMIAH
```

Penjelasan:

Program ini bertujuan untuk membaca data mahasiswa dan mata kuliah dari file JSON, kemudian menampilkannya di konsol. **Program.cs** hanya bertanggung jawab untuk memanggil metode `ReadJSON()`, yang terdapat dalam dua kelas terpisah, yaitu **DataMahasiswa2211104023.cs** dan **KuliahMahasiswa2211104023.cs**.

Pada **DataMahasiswa2211104023.cs**, terdapat metode `ReadJSON()`, yang membaca file JSON berisi data mahasiswa dan mengonversinya menjadi objek yang sesuai. Begitu juga pada **KuliahMahasiswa2211104023.cs**, metode `ReadJSON()` membaca file JSON berisi data mata kuliah yang diambil mahasiswa. Jika file JSON ditemukan dan valid, program akan menampilkan data mahasiswa serta mata kuliah di konsol. Namun, jika file tidak ditemukan atau terdapat kesalahan dalam pembacaannya, program akan memberikan pesan kesalahan yang sesuai. Dengan pendekatan ini, program menjadi lebih modular, terstruktur, dan mudah dipahami.

b. Class Team Members

Program.cs

```
1 using System;
2 using tp7;
3
4 class Program
5 {
6     static void Main(string[] args)
7     {
8         Console.WriteLine("\n=====n");
9         TeamMembers2211104023.DisplayTeamData("jurnal7_2_2211104023.json");
10    }
11 }
12
```

TeamMembers2211104023.cs

```
1 using System;
2 using System.Collections.Generic;
3 using System.IO;
4 using Newtonsoft.Json;
5
6 namespace tp7
7 {
8     public class Member
9     {
10         public string FirstName { get; set; }
11         public string LastName { get; set; }
12         public string Gender { get; set; }
13         public int Age { get; set; }
14         public string NIM { get; set; }
15     }
16
17     public class Team
18     {
19         [JsonProperty("members")]
20         public List<Member> Members { get; set; }
21     }
22
23     public class TeamMembers2211104023
24     {
25         public static Team ReadJSON(string filePath)
26         {
27             try
28             {
29                 if (File.Exists(filePath))
30                 {
31                     string jsonData = File.ReadAllText(filePath);
32                     return JsonConvert.DeserializeObject<Team>(jsonData);
33                 }
34                 else
35                 {
36                     Console.WriteLine("File JSON tidak ditemukan!");
37                     return null;
38                 }
39             }
40             catch (Exception ex)
41             {
42                 Console.WriteLine($"Error membaca file JSON: {ex.Message}");
43                 return null;
44             }
45         }
46
47         public static void DisplayTeamData(string filePath)
48         {
49             Team team = ReadJSON(filePath);
50
51             if (team != null && team.Members != null)
52             {
53                 Console.WriteLine("Daftar Anggota Tim:");
54                 foreach (var member in team.Members)
55                 {
56                     Console.WriteLine($"{member.NIM} - {member.FirstName} {member.LastName} ({member.Age} tahun, {member.Gender})");
57                 }
58             }
59             else
60             {
61                 Console.WriteLine("Data anggota tim tidak ditemukan atau file JSON tidak valid.");
62             }
63         }
64     }
65 }
```

jurnal_2_2211104023.json

```
1  {
2    "members" : [
3      {
4        "firstName": "Mohammad",
5        "lastName": "Dhimas Afrizal",
6        "gender": "male",
7        "age": 2,
8        "nim": "2211104023"
9      },
10     {
11       "firstName": "Ahmad",
12       "lastName": "Junaidi",
13       "gender": "male",
14       "age": 21,
15       "nim": "2211104002"
16     },
17     {
18       "firstName": "Nadia",
19       "lastName": "Putri Rahmaniar",
20       "gender": "female",
21       "age": 21,
22       "nim": "2211104012"
23     },
24     {
25       "firstName": "Devrin",
26       "lastName": "Anggun Saputri",
27       "gender": "female",
28       "age": 21,
29       "nim": "2211104001"
30     },
31     {
32       "firstName": "Raflin",
33       "lastName": "Dhafin Kamil",
34       "gender": "male",
35       "age": 21,
36       "nim": "2211104018"
37     },
38     {
39       "firstName": "Nita",
40       "lastName": "Fitrotul Mar'ah",
41       "gender": "female",
42       "age": 21,
43       "nim": "2211104005"
44     },
45     {
46       "firstName": "Muhammad",
47       "lastName": "Edgar Nadhif",
48       "gender": "male",
49       "age": 21,
50       "nim": "2211104019"
51     }
52   ]
53 }
```

Output:

```
Terminal - tp7

=====

Daftar Anggota Tim:
2211104023 - Mohammad Dhimas Afrizal (2 tahun, male)
2211104002 - Ahmad Junaidi (21 tahun, male)
2211104012 - Nadia Putri Rahmaniar (21 tahun, female)
2211104001 - Devrin Anggun Saputri (21 tahun, female)
2211104018 - Rafli Dhafin Kamil (21 tahun, male)
2211104005 - Nita Fitrotul Mar'ah (21 tahun, female)
2211104019 - Muhammad Edgar Nadhif (21 tahun, male)
```

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 **TeamMembers2211104023**, sehingga kode menjadi lebih modular.

Pada **TeamMembers2211104023.cs**, terdapat dua metode utama:

1. **ReadJSON()** → Membaca file JSON dan mengubahnya menjadi objek `Team`. Jika file ditemukan, data akan di-deserialize menjadi daftar anggota tim. Jika file tidak ditemukan, program akan menampilkan pesan kesalahan.
2. **DisplayTeamData()** → Menampilkan data anggota tim jika file JSON berhasil dibaca. Jika file tidak valid atau tidak berisi data yang benar, program akan memberikan pesan kesalahan yang sesuai.

Dengan pendekatan ini, program menjadi lebih terstruktur, mudah dipahami, dan fleksibel dalam membaca serta menampilkan data dari file JSON.

c. Class GlossaryItem

Program.cs

```
1  using System;
2  using tp7_kelompok_5;
3
4  class Program
5  {
6      static void Main(string[] args)
7      {
8          Console.WriteLine("\n=====\\n");
9          GlossaryItem.ReadJSON("jurnal7_3_2211104023.json");
10     }
11 }
12
```

GlossaryItem2211104023.cs

```
1  using System;
2  using System.IO;
3  using System.Collections.Generic;
4  using Newtonsoft.Json;
5
6  namespace tp7_kelompok_5
7  {
8      public class GlossDef
9      {
10         public string Para { get; set; }
11         public List<string> GlossSeeAlso { get; set; }
12     }
13
14     public class GlossEntry
15     {
16         public string ID { get; set; }
17         public string SortAs { get; set; }
18         public string GlossTerm { get; set; }
19         public string Acronym { get; set; }
20         public string Abbrev { get; set; }
21         public GlossDef GlossDef { get; set; }
22         public string GlossSee { get; set; }
23     }
24
25     public class GlossList
26     {
27         public GlossEntry GlossEntry { get; set; }
28     }
29
30     public class GlossDiv
31     {
32         public string Title { get; set; }
33         public GlossList GlossList { get; set; }
34     }
35
36     public class Glossary
37     {
38         public string Title { get; set; }
39         public GlossDiv GlossDiv { get; set; }
40     }
41
42     public class Root
43     {
44         public Glossary Glossary { get; set; }
45     }
46
47     public class GlossaryItem
48     {
49         public static void ReadJSON(string filePath)
50         {
51             try
52             {
53                 if (File.Exists(filePath))
54                 {
55                     string jsonData = File.ReadAllText(filePath);
56                     Root data = JsonConvert.DeserializeObject<Root>(jsonData);
57                     if (data != null && data.Glossary?.GlossDiv?.GlossList?.GlossEntry != null)
58                     {
59                         var entry = data.Glossary.GlossDiv.GlossList.GlossEntry;
60                     }
61                 }
62             }
63         }
64     }
65 }
```



```

62         Console.WriteLine("\n--- Glossary Entry ---");
63         Console.WriteLine($"ID: {entry.ID}");
64         Console.WriteLine($"Sort As: {entry.SortAs}");
65         Console.WriteLine($"Gloss Term: {entry.GlossTerm}");
66         Console.WriteLine($"Acronym: {entry.Acronym}");
67         Console.WriteLine($"Abbreviation: {entry.Abbrev}");
68         Console.WriteLine($"Definition: {entry.GlossDef.Para}");
69         Console.WriteLine("GlossSeeAlso: " + string.Join(", ", entry.GlossDef.GlossSeeAlso));
70         Console.WriteLine($"GlossSee: {entry.GlossSee}");
71     }
72     else
73     {
74         Console.WriteLine("GlossEntry data not found.");
75     }
76 }
77 else
78 {
79     Console.WriteLine("File JSON tidak ditemukan!");
80 }
81 }
82 catch (Exception ex)
83 {
84     Console.WriteLine($"Error reading JSON file: {ex.Message}");
85 }
86 }
87 }
88 }

```

Output:

```

Terminal - tp7_kelompok_5

=====

--- 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** berfungsi sebagai titik masuk utama yang memanggil metode `ReadJSON()` dari kelas `GlossaryItem` untuk menjaga modularitas.

GlossaryItem.cs menangani parsing JSON menggunakan `Newtonsoft.Json`. Program mendeserialisasi data JSON ke dalam struktur objek yang telah didefinisikan, seperti `Glossary`, `GlossDiv`, dan `GlossEntry`. Informasi yang diekstrak meliputi ID, istilah, akronim, singkatan, definisi, serta referensi terkait.

Jika file JSON tidak ditemukan atau terjadi kesalahan selama proses pembacaan, program akan menangani error dan menampilkan pesan yang sesuai. Dengan pendekatan ini, program menjadi lebih terstruktur, mudah diperluas, dan fleksibel dalam membaca berbagai format glossary dari file JSON.