

**TUGAS JURNAL
KONSTRUKSI PERANGKAT LUNAK**

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



Disusun Oleh :

Devrin Anggun Saputri
2211104001
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 DESERIALIZATION 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:

a. Source Code

- DataMahasiswa2211104001.cs

```
1  using System;
2  using System.IO;
3  using System.Collections.Generic;
4  using Newtonsoft.Json;
5
6  public class Address
7  {
8      1 reference
9      public string StreetAddress { get; set; }
10     1 reference
11     public string City { get; set; }
12     1 reference
13     public string State { get; set; }
14 }
15
16 public class Course
17 {
18     1 reference
19     public string Code { get; set; }
20     public string Name { get; set; }
21 }
22
23 1 reference
24 public class Person
25 {
26     1 reference
27     public string FirstName { get; set; }
28     1 reference
29     public string LastName { get; set; }
30     1 reference
31     public string Gender { get; set; }
32     1 reference
33     public int Age { get; set; }
34     3 references
35     public Address Address { get; set; }
36     1 reference
37     public List<Course> Courses { get; set; }
38 }
```

```

28
29 1 reference
30 public class DataMahasiswa2211104001
31 {
32     1 reference
33     public static void ReadJSON()
34     {
35         string path = @"C:\Users\LENOVO\source\repos\jurnal_modul7\jurnal_modul7\jurnal7_1_2211104001.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             foreach (var course in person.Courses)
48             {
49                 Console.WriteLine($" - {course.Code}: {course.Name}");
50             }
51         }
52         else
53         {
54             Console.WriteLine("File JSON tidak ditemukan!");
55         }
56     }
57 }

```

- jurnal7_1_2211104001.json

```

{
  "firstName": "Devrin",
  "lastName": "Anggun Saputri",
  "gender": "Perempuan",
  "age": 20,
  "address": {
    "streetAddress": "Karangtengah",
    "city": "Banjarnegara",
    "state": "Central Java"
  },
  "courses": [
    {
      "code": "CCK2KAB4",
      "name": "Konstruksi Perangkat Lunak"
    },
    {
      "code": "CCK2DAB3",
      "name": "Proses Perangkat Lunak"
    }
  ]
}

```

- Program.cs

```
1 using System;
2
3 namespace jurnal_modul7
4 {
5     0 references
6     class Program
7     {
8         0 references
9         static void Main(string[] args)
10        {
11            Console.WriteLine("=== Membaca Data Mahasiswa dari JSON ===\n");
12
13            // Memanggil method ReadJSON() dari class DataMahasiswa2211104001
14            DataMahasiswa2211104001.ReadJSON();
15
16            Console.WriteLine("\n☑ Program selesai. Tekan tombol apa saja untuk keluar...");
17            Console.ReadKey(); // Agar konsol tidak langsung tertutup
18        }
19    }
```

b. Output

```
C:\Users\LENOVO\source\rep. X + v
=== Membaca Data Mahasiswa dari JSON ===

Name: Devrin Anggun Saputri
Gender: Perempuan
Age: 20
Address: Karangtengah, Banjarnegara, Central Java
Courses:
- CCK2KAB4: Konstruksi Perangkat Lunak
- CCK2DAB3: Proses Perangkat Lunak
```

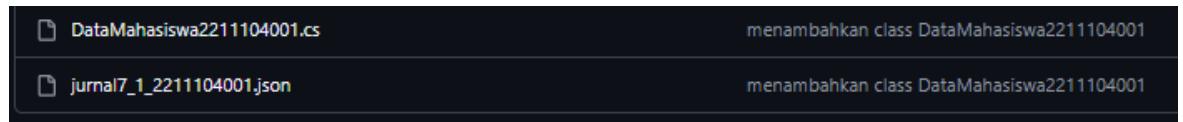
c. Penjelasan

Program ini membaca data mahasiswa dari file JSON menggunakan **Newtonsoft.Json** dan menampilkan informasi seperti nama, gender, usia, alamat, serta daftar mata kuliah ke konsol. Jika file tidak ditemukan, program akan menampilkan pesan error. Pastikan file JSON berada di lokasi yang benar agar program berjalan tanpa masalah. Program ini menggunakan **class** untuk merepresentasikan data JSON dan **metode deserialisasi** untuk mengonversi data JSON menjadi objek dalam C#.

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 DESERIALIZATION 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.

Jawab:

a. Source Code

- TeamMembers2211104001.cs

```
1 using System;
2 using System.IO;
3 using System.Collections.Generic;
4 using Newtonsoft.Json;
5
6 1 reference
7 public class Member
8 {
9     1 reference
10     public string FirstName { get; set; }
11     1 reference
12     public string LastName { get; set; }
13     1 reference
14     public string Gender { get; set; }
15     1 reference
16     public int Age { get; set; }
17     1 reference
18     public string NIM { get; set; }
19 }
20
21 2 references
22 public class Team
23 {
24     1 reference
25     public List<Member> Members { get; set; }
26 }
27
28 1 reference
29 public class TeamMembers2211104001
30 {
31     1 reference
32     public static void ReadJSON()
33     {
34         string path = @"C:\Users\LENOVO\source\repos\jurnal_modul7\jurnal_modul7\jurnal7_2_2211104001.json";
35
36         if (File.Exists(path))
37         {
38             string jsonData = File.ReadAllText(path);
39             Team team = JsonConvert.DeserializeObject<Team>(jsonData);
40         }
41     }
42 }
```

```
31 Console.WriteLine("=== Daftar Anggota Tim ===");
32 foreach (var member in team.Members)
33 {
34     Console.WriteLine($"NIM: {member.NIM}");
35     Console.WriteLine($"Nama: {member.FirstName} {member.LastName}");
36     Console.WriteLine($"Gender: {member.Gender}");
37     Console.WriteLine($"Umur: {member.Age} tahun");
38     Console.WriteLine("-----");
39 }
40
41 else
42 {
43     Console.WriteLine("File JSON tidak ditemukan!");
44 }
45 }
46
47
```

- Jurnal7_2_2211104001.json

```

1  {
2  }
3  "members" : [
4  {
5      "firstName": "Devrin",
6      "lastName": "Anggun Saputri",
7      "gender": "female",
8      "age": 20,
9      "nim": "2211104001"
10 }
11 {
12     "firstName": "Gideon",
13     "lastName": "Toranawa Ladiyo",
14     "gender": "male",
15     "age": 19,
16     "nim": "2211104022"
17 }
18 {
19     "firstName": "Atika",
20     "lastName": "Aji Hadiyani",
21     "gender": "female",
22     "age": 20,
23     "nim": "2211104001"
24 }
25 ]

```

- Program.cs

```

1  using System;
2
3  namespace jurnal_modul7
4  {
5      0 references
6      class Program
7      {
8          0 references
9          static void Main(string[] args)
10         {
11             Console.WriteLine("\n=== Membaca Data Anggota Tim ===\n");
12             // Memanggil method ReadJSON() dari class TeamMembers2211104001
13             TeamMembers2211104001.ReadJSON();
14         }
15     }
16 }

```

b. Output

```
Microsoft Visual Studio Debug Console
+ v

=== Member List ===

Team member list:
2211104001 Devrin Anggun Saputri (20 female)
2211104022 Gideon Toranawa Ladiyo (19 male)
2211104001 Atika Aji Hadiyani (20 female)
```

c. Penjelasan

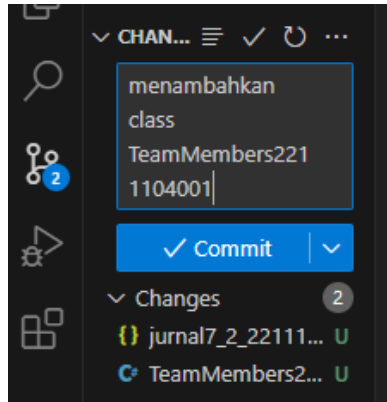
Kode di atas merupakan program C# yang membaca data anggota tim dari file JSON dan menampilkannya ke layar. Program ini menggunakan **Newtonsoft.Json** untuk melakukan deserialisasi JSON ke dalam objek C#. Struktur data terdiri dari **class Member**, yang merepresentasikan setiap anggota tim dengan atribut seperti **nama, gender, umur, dan NIM**. Kemudian, class Team menyimpan daftar anggota dalam bentuk **list Members**. Fungsi utama **ReadJSON()** berada dalam **class TeamMembers2211104001**, yang membaca file JSON dari lokasi tertentu di komputer. Jika file ditemukan, program akan mengekstrak datanya, lalu mencetak informasi setiap anggota tim secara berurutan.

Dalam file JSON, data disusun dalam bentuk array members, yang berisi objek anggota tim dengan atribut **firstName, lastName, gender, age, dan nim**. Setelah JSON dibaca dan dikonversi menjadi objek Team, program akan menampilkan setiap anggota dengan format yang mudah dibaca. Jika file tidak ditemukan, program akan menampilkan pesan **"File JSON tidak ditemukan!"**. Struktur ini memungkinkan program untuk dengan mudah membaca dan menampilkan daftar anggota tim dari file JSON tanpa perlu perubahan besar pada kode.

4. MELAKUKAN COMMIT KEDUA

Pada branch yang dibuat sebelumnya:

- 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:

a. Source Code

- GlossaryItem2211104001.cs

```
1  using System;
2  using System.IO;
3  using Newtonsoft.Json;
4
5  namespace jurnal_modul7
6  {
7      1 reference
8      class GlossaryItem2211104001
9      {
10         1 reference
11         public class GlossDef
12         {
13             1 reference
14             public string Para { get; set; }
15             1 reference
16             public string[] GlossSeeAlso { get; set; }
17         }
18
19         2 references
20         public class GlossEntry
21         {
22             1 reference
23             public string ID { get; set; }
24             0 references
25             public string SortAs { get; set; }
26             1 reference
27             public string GlossTerm { get; set; }
28             1 reference
29             public string Acronym { get; set; }
30             1 reference
31             public string Abbrev { get; set; }
32             2 references
33             public GlossDef GlossDef { get; set; }
34             1 reference
35             public string GlossSee { get; set; }
36         }
37
38         1 reference
39         public class GlossList
40         {
41             1 reference
```

```

26     1 reference
    public class GlossList
27     {
28         1 reference
    public GlossEntry GlossEntry { get; set; }
29     }
30
31     1 reference
    public class GlossDiv
32     {
33         0 references
    public string Title { get; set; }
34         1 reference
    public GlossList GlossList { get; set; }
35     }
36
37     1 reference
    public class Glossary
38     {
39         0 references
    public string Title { get; set; }
40         1 reference
    public GlossDiv GlossDiv { get; set; }
41     }
42
43     2 references
    public class Root
44     {
45         1 reference
    public Glossary Glossary { get; set; }
46     }
47
48     1 reference
    public static void ReadJSON()
49     {
50         string filePath = @"C:\Users\LENOVO\source\repos\jurnal_modul7\jurnal_modul7\jurnal7_3_2211104001.json";
51         if (File.Exists(filePath))
52         {
53             string jsonContent = File.ReadAllText(filePath);

```

```

53             string jsonContent = File.ReadAllText(filePath);
54             Root glossaryData = JsonConvert.DeserializeObject<Root>(jsonContent);
55
56             GlossEntry entry = glossaryData.Glossary.GlossDiv.GlossList.GlossEntry;
57
58             Console.WriteLine("Glossary Entry Details:");
59             Console.WriteLine($"ID: {entry.ID}");
60             Console.WriteLine($"GlossTerm: {entry.GlossTerm}");
61             Console.WriteLine($"Acronym: {entry.Acronym}");
62             Console.WriteLine($"Abbreviation: {entry.Abbrev}");
63             Console.WriteLine($"Definition: {entry.GlossDef.Para}");
64             Console.WriteLine("GlossSeeAlso: " + string.Join(", ", entry.GlossDef.GlossSeeAlso));
65             Console.WriteLine($"GlossSee: {entry.GlossSee}");
66
67         }
68         else
69         {
70             Console.WriteLine("File JSON tidak ditemukan.");
71         }
72     }
73 }

```

- Jurnal7_3_2211104001.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": [ "GML", "XML" ]
16                 },
17                 "GlossSee": "markup"
18             }
19         }
20     }
21 }
22
23

```

- Program.cs

```
1 using System;
2
3 namespace jurnal_modul7
4 {
5     0 references
6     class Program
7     {
8         0 references
9         static void Main(string[] args)
10         {
11             Console.WriteLine("=== Glosarry Item ===\n");
12             // Memanggil method ReadJSON() dari class TeamMahasiswa2211104001
13             GlossaryItem2211104001.ReadJSON();
14         }
15     }
16 }
```

b. Output

```
Microsoft Visual Studio Debug Console
=== Glosarry Item ===

Glossary Entry Details:
ID: SGML
GlossTerm: 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
```

c. Penjelasan

Program di atas merupakan implementasi dalam bahasa C# yang digunakan untuk membaca dan menampilkan data dari file JSON yang berisi informasi tentang sebuah glossary (daftar istilah). Struktur JSON yang digunakan memiliki beberapa tingkatan, di mana bagian utama adalah objek glossary, yang di dalamnya terdapat GlossDiv, lalu GlossList, dan akhirnya GlossEntry, yang berisi detail dari suatu istilah. Program ini membaca file JSON dari lokasi yang telah ditentukan (jurnal7_3_2211104001.json), kemudian mendeserialisasi isinya menjadi objek Root. Setelah data berhasil diproses, informasi mengenai istilah seperti ID, GlossTerm, Acronym, Abbreviation, serta deskripsinya akan ditampilkan di konsol. Selain itu, jika ada referensi lain dalam GlossSeeAlso, program akan mencetaknya sebagai daftar istilah terkait.

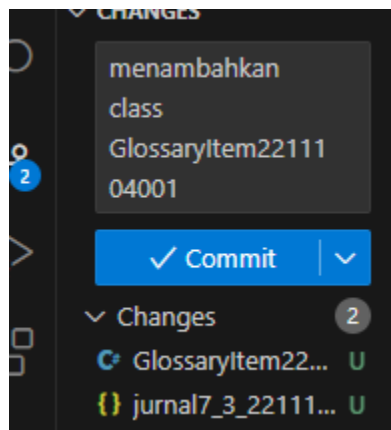
Dalam kode ini, setiap bagian dari JSON diwakili oleh kelas-kelas yang sesuai, seperti GlossDef, GlossEntry,

GlossList, GlossDiv, dan Glossary, sehingga mempermudah pemetaan data dari JSON ke dalam objek C#. Jika file JSON ditemukan, program akan membaca dan menampilkan data, tetapi jika tidak ditemukan, program akan memberikan pesan bahwa file JSON tidak tersedia. Dengan pendekatan ini, struktur JSON yang kompleks dapat dengan mudah dipahami dan diakses dalam kode C#. Hal ini sangat berguna dalam pengolahan data berbasis JSON, seperti untuk keperluan dokumentasi istilah atau kamus elektronik.

6. MELAKUKAN COMMIT TERAKHIR

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.

