

**TUGAS JURNAL
KONSTRUKSI PERANGKAT LUNAK**

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



Disusun Oleh :

Rizky Hanifa Afania

2211104017

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_2211104017.json

```
{
  "firstName": "Rizky",
  "lastName": "Hanifa Afania",
  "gender": "female",
  "age": 22,
  "address": {
    "streetAddress": "Jl. Samparagin",
    "city": "Purwokerto",
    "state": "Central Java"
  },
  "courses": [
    {
      "code": "CRI2C4",
      "name": "Konstruksi Perangkat Lunak"
    },
    {
      "code": "CRI2XX",
      "name": "Kecerdasan Buatan"
    }
  ]
}
```

File DataMahasiswa2211104017.cs

```
using System;
using System.IO;
using System.Text.Json;

1 reference
public class DataMahasiswa2211104017
{
    // Class untuk merepresentasikan struktur JSON
    1 reference
    public class Address
    {
        1 reference
        public string streetAddress { get; set; }
        1 reference
        public string city { get; set; }
        1 reference
        public string state { get; set; }
    }

    1 reference
    public class Course
    {
        1 reference
        public string code { get; set; }
        1 reference
        public string name { get; set; }
    }

    2 references
    public class Mahasiswa
    {
        1 reference
        public string firstName { get; set; }
        1 reference
        public string lastName { get; set; }
        1 reference
        public string gender { get; set; }
        1 reference
        public int age { get; set; }
        3 references
        public Address address { get; set; }
        1 reference
        public Course[] courses { get; set; }
    }

    1 reference
    public static void ReadJSON()
    {
        string filePath = "jurnal7_1_2211104017.json";

        if (File.Exists(filePath))
        {
            string jsonString = File.ReadAllText(filePath);

            Mahasiswa mahasiswa = JsonSerializer.Deserialize<Mahasiswa>(jsonString);

            Console.WriteLine("===== Data Mahasiswa =====");
            Console.WriteLine($"Nama : {mahasiswa.firstName} {mahasiswa.lastName}");
            Console.WriteLine($"Gender: {mahasiswa.gender}");
            Console.WriteLine($"Usia : {mahasiswa.age}");
            Console.WriteLine($"Alamat: {mahasiswa.address.streetAddress}, " +
                $"{mahasiswa.address.city}, {mahasiswa.address.state}");
            Console.WriteLine("\n===== Mata Kuliah =====");

            foreach (var course in mahasiswa.courses)
            {
                Console.WriteLine($"Kode: {course.code} - Nama: {course.name}");
            }
        }
        else
        {
            Console.WriteLine("File JSON tidak ditemukan.");
        }
    }
}
```

File Program.cs

```
0 references
class Program
{
    0 references
    static void Main(string[] args)
    {
        DataMahasiswa2211104017.ReadJSON();
    }
}
```

- **Screenshot hasil run**

```
Microsoft Visual Studio Debug Console

===== Data Mahasiswa =====
Nama : Rizky Hanifa Afania
Gender: female
Usia : 22
Alamat: Jl. Samparagin, Purwokerto, Central Java

===== Mata Kuliah =====
Kode: CRI2C4 - Nama: Konstruksi Perangkat Lunak
Kode: CRI2XX - Nama: Kecerdasan Buatan

F:\Coding\Praktikum KPL\Praktikum Mandiri\Modul 7\JurnalModul7
```

- **Penjelasan**

Program ini bertujuan untuk membaca dan menampilkan data mahasiswa dari file JSON dan menerapkan teknik parsing JSON. File jurnal7_1_2211104017.json berisi data yang berisi informasi mahasiswa yaitu nama, usia, gender, alamat, dan daftar mata kuliah. File JSON ini disimpan dalam folder proyek, dan program akan membaca isinya lalu mengubah (parsing) data tersebut menjadi objek dalam bahasa C#. Parsing adalah proses mengonversi data dari format teks JSON menjadi struktur objek yang dapat digunakan dalam kode.

Di dalam file DataMahasiswa2211104017.cs, terdapat kelas Mahasiswa beserta sub-kelas Address dan Course untuk merepresentasikan struktur JSON. Method ReadJSON() membaca file JSON menggunakan File.ReadAllText(), kemudian melakukan parsing dengan JsonSerializer.Deserialize<Mahasiswa>() agar isi file JSON bisa digunakan sebagai objek Mahasiswa. Setelah parsing selesai, program menampilkan data tersebut ke console dalam format yang lebih mudah dibaca. File Program.cs memanggil method ReadJSON() untuk mengeksekusi proses parsing ini ketika

program dijalankan. Dengan teknik ini, program dapat mengakses dan mengolah data JSON dengan efisien.

2. 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.

 DataMahasiswa2211104017.cs	Menambahkan class DataMahasiswa2211104017
 Program.cs	Menambahkan class DataMahasiswa2211104017

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_2211104017.json

```
{
  "members": [
    {
      "firstName": "Andera",
      "lastName": "Singgih Pratama",
      "gender": "Male",
      "age": 21,
      "nim": "2211104007"
    },
    {
      "firstName": "Althafia Defiyandrea",
      "lastName": "Laskanadya Wibowo",
      "gender": "Female",
      "age": 21,
      "nim": "2211104011"
    },
    {
      "firstName": "Muhammad Idham",
      "lastName": "Cholid",
      "gender": "Male",
      "age": 21,
      "nim": "2211104016"
    },
    {
      "firstName": "Rizky Hanifa",
      "lastName": "Afania",
      "gender": "Female",
      "age": 22,
      "nim": "2211104017"
    },
    {
      "firstName": "Fauzan Wahyu",
      "lastName": "Mubarak",
      "gender": "Male",
      "age": 21,
      "nim": "2211104027"
    }
  ]
}
```

File "TeamMembers2211104017.cs

```
using System;
using System.IO;
using System.Text.Json;

1 reference
class TeamMembers2211104017
{
    1 reference
    class Member
    {
        1 reference
        public string firstName { get; set; }
        1 reference
        public string lastName { get; set; }
        1 reference
        public string gender { get; set; }
        1 reference
        public int age { get; set; }
        1 reference
        public string nim { get; set; }
    }

    2 references
    class Team
    {
        1 reference
        public Member[] members { get; set; }
    }
}
```

```
public static void ReadJSON()
{
    string filePath = "jurnal7_2_2211104017.json";

    if (!File.Exists(filePath))
    {
        Console.WriteLine("File JSON tidak ditemukan.");
        return;
    }

    // Membaca dan melakukan parsing JSON
    Team team = JsonSerializer.Deserialize<Team>(File.ReadAllText(filePath));

    Console.WriteLine("Team member list:");
    foreach (var member in team.members)
    {
        Console.WriteLine($"{member.nim} {member.firstName} {member.lastName} " +
            $"{member.age} {member.gender}");
    }
}
```

File Program.cs

```
0 references
class Program
{
    0 references
    static void Main(string[] args)
    {
        //DataMahasiswa2211104017.ReadJSON();
        TeamMembers2211104017.ReadJSON();
    }
}
```

- **Screenshot hasil run**

```
Microsoft Visual Studio Debug Console
Team member list:
2211104007 Andera Singgih Pratama (21 Male)
2211104011 Althafia Defiyandrea Laskanadya Wibowo (21 Female)
2211104016 Muhammad Idham Cholid (21 Male)
2211104017 Rizky Hanifa Afania (22 Female)
2211104027 Fauzan Wahyu Mubarak (21 Male)
F:\Coding\Praktikum KPL\Praktikum Mandiri\Modul 7\JurnalModul7\
```

- **Penjelasan**

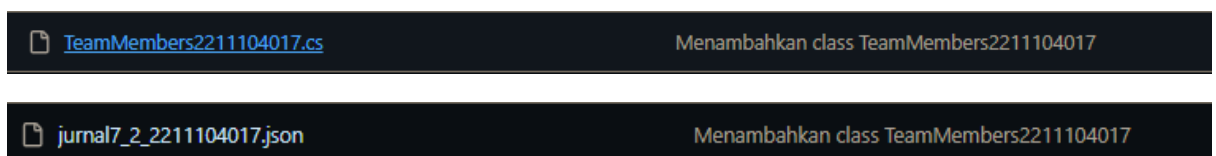
Program ini bertujuan untuk membaca dan menampilkan daftar anggota kelompok dari file JSON dengan teknik parsing JSON. File jurnal7_2_2211104017.json berisi data anggota tim dalam format JSON, yang berisi NIM, nama, gender, dan usia masing-masing anggota. File ditempatkan dalam folder proyek agar program dapat membacanya. Parsing adalah proses mengonversi data JSON (yang berbentuk teks) menjadi objek dalam bahasa C#, sehingga datanya bisa digunakan dalam kode program.

Dalam file TeamMembers2211104017.cs, program membaca file JSON dengan `File.ReadAllText()`, lalu program akan melakukan parsing menggunakan `JsonSerializer.Deserialize<Team>()`, yang mengubah data JSON menjadi objek `Team`. Objek ini memiliki array `members`, yang berisi informasi setiap anggota. Setelah parsing selesai, program mencetak daftar anggota ke layar dengan format yang telah ditentukan. File `Program.cs` bertugas memanggil method `ReadJSON()` agar proses parsing dan tampilan data dijalankan saat program dieksekusi.

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_2211104017.json

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

File GlossaryItem2211104017.cs

```
using System;
using System.IO;
using System.Text.Json;

1 reference
class GlossaryItem2211104017
{
    1 reference
    class GlossDef
    {
        1 reference
        public string para { get; set; }
        1 reference
        public string[] GlossSeeAlso { get; set; }
    }

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

    1 reference
    class GlossList
    {
        1 reference
        public GlossEntry GlossEntry { get; set; }
    }
}
```

```
1 reference
class GlossDiv
{
    0 references
    public string title { get; set; }
    1 reference
    public GlossList GlossList { get; set; }
}

1 reference
class Glossary
{
    0 references
    public string title { get; set; }
    1 reference
    public GlossDiv GlossDiv { get; set; }
}

2 references
class Root
{
    1 reference
    public Glossary glossary { get; set; }
}
```

```

1 reference
public static void ReadJSON()
{
    string filePath = "jurnal7_3_2211104017.json";

    if (!File.Exists(filePath))
    {
        Console.WriteLine("File JSON tidak ditemukan.");
        return;
    }

    // Membaca dan parsing JSON
    Root data = JsonSerializer.Deserialize<Root>(File.ReadAllText(filePath));

    // Mengambil bagian GlossEntry
    GlossEntry entry = data.glossary.GlossDiv.GlossList.GlossEntry;

    Console.WriteLine("==== Glossary Entry =====");
    Console.WriteLine($"*ID          : {entry.ID}*");
    Console.WriteLine($"*SortAs       : {entry.SortAs}*");
    Console.WriteLine($"*Term         : {entry.GlossTerm}*");
    Console.WriteLine($"*Acronym      : {entry.Acronym}*");
    Console.WriteLine($"*Abbrev       : {entry.Abbrev}*");
    Console.WriteLine($"*Definition   : {entry.GlossDef.para}*");
    Console.WriteLine("See Also : ");
    Console.WriteLine(string.Join(", ", entry.GlossDef.GlossSeeAlso));
    Console.WriteLine($"*GlossSee    : {entry.GlossSee}*");
}

```

File Program.cs

```

0 references
class Program
{
    0 references
    static void Main(string[] args)
    {
        //DataMahasiswa2211104017.ReadJSON();
        //TeamMembers2211104017.ReadJSON();
        GlossaryItem2211104017.ReadJSON();
    }
}

```

- Screenshot hasil run

```

Microsoft Visual Studio Debug Console
===== 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

F:\Coding\Praktikum KPL\Praktikum Mandiri\Modul 7\JurnalModul7\JurnalModul7\bin\Debug\

```

- Penjelasan

Program ini bertujuan untuk membaca dan menampilkan data dari file JSON "jurnal7_3_2211104017.json" yang berisi informasi tentang glossary. File JSON ini memiliki bagian utama yaitu "glossary", yang berisi "GlossDiv", lalu di dalamnya terdapat "GlossList", dan "GlossEntry" dan menyimpan data utama seperti ID, GlossTerm, Acronym, Abbreviation, Definition, serta


referensi terkait. Program membaca file JSON, lalu menggunakan teknik parsing (deserialisasi JSON) untuk mengubah data dari format teks JSON menjadi objek yang dapat digunakan dalam program C#.

Di dalam file `GlossaryItem2211104017.cs`, program mendefinisikan beberapa kelas yang sesuai dengan struktur JSON. Kemudian, pada method `ReadJSON()`, program membaca file JSON menggunakan `File.ReadAllText()`, lalu melakukan parsing dengan `JsonSerializer.Deserialize<Root>()`. Setelah data berhasil diparsing, program mengambil bagian `GlossEntry` dan mencetak informasinya ke konsol dalam format yang lebih mudah dibaca. Teknik parsing berguna untuk mengakses setiap elemen dalam JSON seperti mengambil ID, `GlossTerm`, `Acronym`, serta daftar referensi di `GlossSeeAlso` tanpa harus 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.

 `Glossary/Item2211104017.cs`

Menambahkan class `Glossary/Item2211104017`

 `jurnal7_3_2211104017.json`

Menambahkan class `Glossary/Item2211104017`