

TUGAS JURNAL
KONSTRUKSI PERANGKAT LUNAK

MODUL VII
GRAMMAR-BASED INPUT PROESSING (PARSING)



Disusun Oleh:

Aorinka Anendya Chazanah / 2211104013

S1 SE-06-01

Dosen Pengampu:

Yudha Islami Sulistya, S.Kom., M.Cs

PROGRAM STUDI S1 SOFTWARE ENGINEERING

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

TUGAS JURNAL

1. MENAMBAHKAN JSON DESERIALIZATION 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.

➤ Source code

File jurnal7_1_2211104013.json

```
1  {
2      "firstName": "Aorinka",
3      "lastName": "Anendya",
4      "gender": "Female",
5      "age": 22,
6      "address": {
7          "streetAddress": "Tayem Timur",
8          "city": "Cilacap",
9          "state": "Central Java"
10     },
11     "courses": [
12         { "code": "CRI2C4", "name": "Konstruksi Perangkat Lunak" },
13         { "code": "CRI2XX", "name": "Nama Mata Kuliah" }
14     ]
15 }
```

File DataMahasiswa2211104013.cs

```
1  using System;
2  using System.IO;
3  using Newtonsoft.Json;
4
5  namespace jurnal_modul7
6  {
7      1 reference
8      class DataMahasiswa2211104013
9      {
10         1 reference
11         public class Address
12         {
13             1 reference
14             public string StreetAddress { get; set; }
15             1 reference
16             public string City { get; set; }
17             1 reference
18             public string State { get; set; }
19         }
20
21         1 reference
22         public class Course
23         {
24             1 reference
25             public string Code { get; set; }
26             1 reference
27             public string Name { get; set; }
28         }
29
30         2 references
31         public class Mahasiswa
32         {
33             1 reference
34             public string FirstName { get; set; }
35             1 reference
36             public string LastName { get; set; }
37             1 reference
38             public string Gender { get; set; }
39             1 reference
40             public int Age { get; set; }
41             3 references
42             public Address Address { get; set; }
43             1 reference
44             public Course[] Courses { get; set; }
45         }
46
47         1 reference
48         public static void ReadJSON()
49         {
50             string filePath = "jurnal7_1_2211104013.json";
51             if (File.Exists(filePath))
52             {
53                 string jsonContent = File.ReadAllText(filePath);
54                 Mahasiswa mahasiswa = JsonConvert.DeserializeObject<Mahasiswa>(jsonContent);
55
56                 Console.WriteLine("Mahasiswa Details:");
57                 Console.WriteLine($"Name: {mahasiswa.FirstName} {mahasiswa.LastName}");
58                 Console.WriteLine($"Gender: {mahasiswa.Gender}");
59                 Console.WriteLine($"Age: {mahasiswa.Age}");
60                 Console.WriteLine($"Address: {mahasiswa.Address.StreetAddress}, {mahasiswa.Address.City}, {mahasiswa.Address.State}");
61                 Console.WriteLine("Courses:");
62                 foreach (var course in mahasiswa.Courses)
63                 {
64                     Console.WriteLine($"- {course.Code}: {course.Name}");
65                 }
66             }
67             else
68             {
69                 Console.WriteLine("File JSON tidak ditemukan.");
70             }
71         }
72     }
73 }
```

File Program.cs

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace jurnal_modul7
8  {
9      0 references
10     class Program
11     {
12         0 references
13         static void Main(string[] args)
14         {
15             DataMahasiswa2211104013.ReadJSON();
16         }
17     }
18 }
19
```

➤ Hasil running program

```
Microsoft Visual Studio Debug Console
Mahasiswa Details:
Name: Aorinka Anendya
Gender: Female
Age: 22
Address: Tayem Timur, Cilacap, Central Java
Courses:
- CRI2C4: Konstruksi Perangkat Lunak
- CRI2XX: Nama Mata Kuliah
```

➤ Penjelasan program

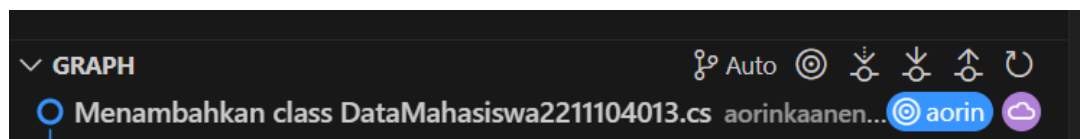
Program di atas membaca data mahasiswa dari file JSON dan menampilkannya di console. File JSON berisi informasi seperti nama, gender, usia, alamat, dan daftar mata kuliah yang diambil. Program ini menggunakan **Newtonsoft.Json** untuk melakukan deserialisasi JSON ke dalam objek C# yang sesuai dengan struktur data mahasiswa. Data kemudian ditampilkan dengan format yang jelas agar mudah dibaca.

Dalam implementasinya, terdapat dua file: `DataMahasiswa2211104013.cs` yang berisi kelas untuk merepresentasikan data mahasiswa serta method `ReadJSON()` untuk membaca dan menampilkan data, serta `Program.cs` yang berfungsi sebagai entry point aplikasi.

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.

➤ **Source code**

File jurnal7_2_2211104013.json

```
{
  "members" : [
    {
      "firstName": "Fadhila",
      "lastName": "Agil",
      "gender": "Male",
      "age": 21,
      "nim": "2211104006"
    },
    {
      "firstName": "Aorinka",
      "lastName": "Anendya",
      "gender": "Female",
      "age": 22,
      "nim": "2211104013"
    },
    {
      "firstName": "Muhammad Abdul",
      "lastName": "Aziz",
      "gender": "Male",
      "age": 21,
      "nim": "2211104026"
    },
    {
      "firstName": "Muhammad Luthfi",
      "lastName": "Hamdani",
      "gender": "Male",
      "age": 21,
      "nim": "2211104020"
    }
  ]
}
```

File TeamMembers2211104013.cs

```

1  using System;
2  using System.Collections.Generic;
3  using System.IO;
4  using Newtonsoft.Json;
5
6  namespace jurnal_modul7
7  {
8      1 reference
9      class TeamMembers2211104013
10     {
11         1 reference
12         public class Member
13         {
14             1 reference
15             public string FirstName { get; set; }
16             1 reference
17             public string LastName { get; set; }
18             1 reference
19             public string Gender { get; set; }
20             1 reference
21             public int Age { get; set; }
22             1 reference
23             public string NIM { get; set; }
24         }
25
26         2 references
27         public class Team
28         {
29             1 reference
30             public List<Member> Members { get; set; }
31         }
32
33         1 reference
34         public static void ReadJSON()
35         {
36             string filePath = "jurnal7_2_2211104013.json";
37             if (File.Exists(filePath))
38             {
39                 string jsonContent = File.ReadAllText(filePath);
40                 Team team = JsonConvert.DeserializeObject<Team>(jsonContent);
41
42                 Console.WriteLine("Team member list:");
43             }
44         }
45     }

```

```

19     public class Team
20     {
21         1 reference
22         public List<Member> Members { get; set; }
23     }
24
25     1 reference
26     public static void ReadJSON()
27     {
28         string filePath = "jurnal7_2_2211104013.json";
29         if (File.Exists(filePath))
30         {
31             string jsonContent = File.ReadAllText(filePath);
32             Team team = JsonConvert.DeserializeObject<Team>(jsonContent);
33
34             Console.WriteLine("Team member list:");
35             foreach (var member in team.Members)
36             {
37                 Console.WriteLine($"{member.NIM} {member.FirstName} {member.LastName} ({member.Age} {member.Gender})");
38             }
39         }
40         else
41         {
42             Console.WriteLine("File JSON tidak ditemukan.");
43         }
44     }
45 }

```

File Program.cs

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace jurnal_modul7
8  {
9      class Program
10     {
11         static void Main(string[] args)
12         {
13             //DataMahasiswa2211104013.ReadJSON();
14             TeamMembers2211104013.ReadJSON();
15         }
16     }
17 }
18
19
```

➤ Hasil running program

```
Microsoft Visual Studio Debug Console
Team member list:
2211104006 Fadhila Agil (21 Male)
2211104013 Aorinka Anendya (22 Female)
2211104026 Muhammad Abdul Aziz (21 Male)
2211104020 Muhammad Luthfi Hamdani (21 Male)
```

➤ Penjelasan program

Program di atas bertujuan untuk membaca dan menampilkan daftar anggota tim dari file JSON menggunakan bahasa C#. Kelas `TeamMembers2211104013` mendefinisikan struktur data yang sesuai dengan isi JSON, yaitu daftar anggota tim yang terdiri dari atribut `FirstName`, `LastName`, `Gender`, `Age`, dan `NIM`. Metode `ReadJSON()` membaca file JSON "`jurnal7_2_2211104013.json`", mendeserialisasinya menjadi objek `Team`, lalu mencetak daftar anggota tim dalam format yang telah ditentukan. Jika file JSON tidak ditemukan, program akan menampilkan pesan kesalahan.

File `Program.cs` berfungsi sebagai entry point aplikasi, di mana metode `Main()` memanggil `ReadJSON()` dari kelas `TeamMembers2211104013`. Program ini menggunakan pustaka `Newtonsoft.Json` untuk parsing JSON. Saat dijalankan, program akan membaca file JSON di folder proyek dan mencetak daftar anggota tim ke konsol sesuai dengan format yang telah ditentukan.

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.

➤ **Source code**

File jurnal7_3_2211104013.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 }
```


File GlossaryItem2211104013.cs

```

1  using System;
2  using System.IO;
3  using Newtonsoft.Json;
4
5  namespace jurnal_modul7
6  {
7      1 reference
8      class GlossaryItem2211104013
9      {
10         1 reference
11         public class GlossDef
12         {
13             public string Para { get; set; }
14             1 reference
15             public string[] GlossSeeAlso { get; set; }
16         }
17
18         public class GlossEntry
19         {
20             public string ID { get; set; }
21             public string SortAs { get; set; }
22             public string GlossTerm { get; set; }
23             public string Acronym { get; set; }
24             public string Abbrev { get; set; }
25             2 references
26             public GlossDef GlossDef { get; set; }
27             1 reference
28             public string GlossSee { get; set; }
29         }
30
31         1 reference
32         public class GlossList
33         {
34             1 reference
35             public GlossEntry GlossEntry { get; set; }
36         }
37
38         1 reference
39         public class GlossDiv
40         {
41             0 references
42             public string Title { get; set; }
43             1 reference
44             public GlossList GlossList { get; set; }
45         }
46
47         1 reference
48         public class Glossary
49         {
50             0 references
51             public string Title { get; set; }
52             1 reference
53             public GlossDiv GlossDiv { get; set; }
54         }
55
56         2 references
57         public class Root
58         {
59             1 reference
60             public Glossary Glossary { get; set; }
61         }
62
63         1 reference
64         public static void ReadJSON()
65         {
66             string filePath = "jurnal7_3_2211104013.json";
67             if (File.Exists(filePath))
68             {
69                 string jsonContent = File.ReadAllText(filePath);
70                 Root glossaryData = JsonConvert.DeserializeObject<Root>(jsonContent);
71
72                 GlossEntry entry = glossaryData.Glossary.GlossDiv.GlossList.GlossEntry;
73
74                 Console.WriteLine("Glossary Entry Details:");
75                 Console.WriteLine($"ID: {entry.ID}");
76                 Console.WriteLine($"GlossTerm: {entry.GlossTerm}");
77
78                 Console.WriteLine($"GlossTerm: {entry.GlossTerm}");
79                 Console.WriteLine($"Acronym: {entry.Acronym}");
80                 Console.WriteLine($"Abbreviation: {entry.Abbrev}");
81                 Console.WriteLine($"Definition: {entry.GlossDef.Para}");
82                 Console.WriteLine("GlossSeeAlso: " + string.Join(", ", entry.GlossDef.GlossSeeAlso));
83                 Console.WriteLine($"GlossSee: {entry.GlossSee}");
84             }
85             else
86             {
87                 Console.WriteLine("File JSON tidak ditemukan.");
88             }
89         }
90     }
91 }

```

File Program.cs

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace jurnal_modul7
8  {
9      class Program
10     {
11         static void Main(string[] args)
12         {
13             //DataMahasiswa2211104013.ReadJSON();
14             //TeamMembers2211104013.ReadJSON();
15             GlossaryItem2211104013.ReadJSON();
16         }
17     }
18 }
19
```

➤ Hasil running program

```
Microsoft Visual Studio Debug Console
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
```

➤ Penjelasan program

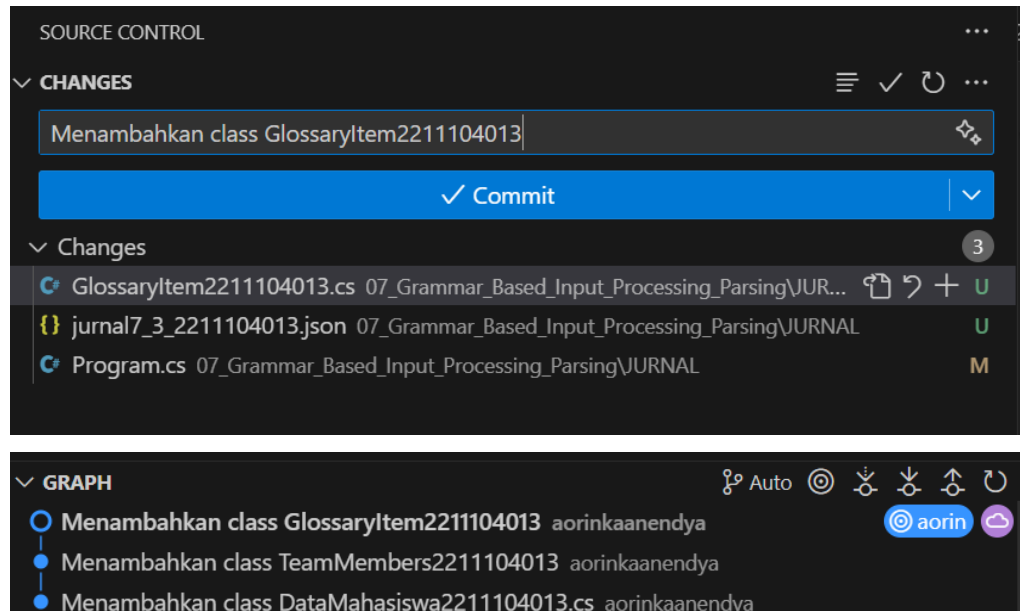
Program ini membaca dan memproses file JSON yang berisi data glossary menggunakan C# dan pustaka Newtonsoft.Json. Data dalam JSON terdiri dari berbagai elemen seperti GlossEntry, GlossTerm, Acronym, serta definisi istilah. Program pertama-tama membaca file jurnal7_3_2211104013.json, lalu melakukan deserialisasi ke dalam objek yang sesuai dengan struktur JSON. Setelah itu, data glossary ditampilkan di konsol dalam format yang lebih mudah dibaca.

Pemisahan kode dilakukan dengan menaruh logika utama dalam kelas GlossaryItem2211104013, sementara Main() di Program.cs hanya bertugas memanggil ReadJSON(). Dengan cara ini, struktur kode menjadi lebih rapi dan modular. Jika file JSON tidak ditemukan, program akan menampilkan pesan kesalahan untuk memberi tahu pengguna bahwa file tersebut belum tersedia.

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.



7. MELAKUKAN PULL REQUEST PADA GITHUB

Melakukan pull request dan merge branch aorin ke main

