

TUGAS JURNAL
KONSTRUKSI PERANGKAT LUNAK
MODUL VII
GRAMMATR-BASED INPUT PROCESSING PARSING



Disusun Oleh:
Maria Nathasya Desfera Pangestu
2211104008
SE0601

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

PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK
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.

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

Jurnal7_1_2211104008.json

```
1  {
2    "firstName": "Maria",
3    "lastName": "Nathasya",
4    "gender": "female",
5    "age": 21,
6    "address": {
7      "streetAddress": "Jl. DI. Panjaitan",
8      "city": "Purwokerto",
9      "state": "Central Java"
10   },
11   "courses": [
12     {
13       "code": "CRI2C4",
14       "name": "Konstruksi Perangkat Lunak"
15     },
16     {
17       "code": "CRI2XX",
18       "name": "Kecerdasan Buatan"
19     }
20   ]
21 }
```

DataMahasiswa2211104008.cs

```

1  using System;
2  using System.IO;
3  using System.Text.Json;
4
5  // 0 references
6  public class DataMahasiswa2211104008
7  {
8      // 1 reference
9      // Class untuk merepresentasikan struktur JSON
10     public class Address
11     {
12         // 1 reference
13         public string streetAddress { get; set; }
14         // 1 reference
15         public string city { get; set; }
16         // 1 reference
17         public string state { get; set; }
18     }
19
20     // 1 reference
21     public class Course
22     {
23         // 1 reference
24         public string code { get; set; }
25         // 1 reference
26         public string name { get; set; }
27     }
28
29     // 2 references
30     public class Mahasiswa
31     {
32         // 1 reference
33         public string firstName { get; set; }
34         // 1 reference
35         public string lastName { get; set; }
36         // 1 reference
37         public string gender { get; set; }
38         // 1 reference
39         public int age { get; set; }
40         // 3 references
41         public Address address { get; set; }
42         // 1 reference
43         public Course[] courses { get; set; }
44     }
45 }
46
47 // 0 references
48 public static void ReadJSON()
49 {
50     string filePath = "jurnal7_1_2211104008.json";
51
52     if (File.Exists(filePath))
53     {
54         string jsonString = File.ReadAllText(filePath);
55
56         Mahasiswa mahasiswa = JsonSerializer.Deserialize<Mahasiswa>(jsonString);
57
58         Console.WriteLine("==== Data Mahasiswa =====");
59         Console.WriteLine($"Nama : {mahasiswa.firstName} {mahasiswa.lastName}");
60         Console.WriteLine($"Gender: {mahasiswa.gender}");
61         Console.WriteLine($"Usia : {mahasiswa.age}");
62         Console.WriteLine($"Alamat: {mahasiswa.address.streetAddress}, " +
63             $"{mahasiswa.address.city}, {mahasiswa.address.state}");
64         Console.WriteLine("\n==== Mata Kuliah =====");
65
66         foreach (var course in mahasiswa.courses)
67         {
68             Console.WriteLine($"Kode: {course.code} - Nama: {course.name}");
69         }
70     }
71     else
72     {
73         Console.WriteLine("File JSON tidak ditemukan.");
74     }
75 }

```

Program.cs

```

0 references
1  class Program
2  {
3      0 references
4      static void Main(string[] args)
5      {
6          //DataMahasiswa2211104008.ReadJSON();
7          //TeamMembers2211104008.ReadJSON();
8          GlossaryItem2211104008.ReadJSON();
9      }
}

```

- Output

```

Microsoft Visual Studio Debug Console

===== Data Mahasiswa =====
Nama : Maria Nathasya
Gender: Female
Usia : 21
Alamat: JL. DI Panjaitan, Purwokerto, Central Java

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

```

- Penjelasan

Program ini membaca dan menguraikan (parse) data mahasiswa dari file JSON ke dalam objek C#. Data mahasiswa meliputi nama, usia, gender, alamat, dan mata kuliah. Proses parsing dilakukan oleh metode ReadJSON() di kelas Mahasiswa (bersama sub-kelas Address dan Course), yang kemudian menampilkan data yang terstruktur ke konsol.

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.

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

Jurnal7_2_2211104008.json

```
1  {
2    "members": [
3      {
4        "firstName": "Maria",
5        "lastName": "Nathasya",
6        "gender": "Female",
7        "age": 21,
8        "nim": "2211104008"
9      },
10     {
11       "firstName": "Aubrey",
12       "lastName": "Raihan",
13       "gender": "Male",
14       "age": 21,
15       "nim": "2211104010"
16     },
17     {
18       "firstName": "Syahrul",
19       "lastName": "Zaki",
20       "gender": "Male",
21       "age": 21,
22       "nim": "2211104014"
23     },
24     {
25       "firstName": "Wahyu",
26       "lastName": "Isnantia",
27       "gender": "Male",
28       "age": 21,
29       "nim": "2211104021"
30     }
31   ]
32 }
33
```

TeamMember2211104008.cs

```

1  using System;
2  using System.IO;
3  using System.Text.Json;
4
5  0 references
6  class TeamMembers2211104008
7  {
8      1 reference
9      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     2 references
24     class Team
25     {
26         1 reference
27         public Member[] members { get; set; }
28     }
29
30     0 references
31     public static void ReadJSON()
32     {
33         string filePath = "jurnal7_2_2211104008.json";
34
35         if (!File.Exists(filePath))
36         {
37             Console.WriteLine("File JSON tidak ditemukan.");
38             return;
39         }
40
41         // Membaca dan melakukan parsing JSON
42         Team team = JsonSerializer.Deserialize<Team>(File.ReadAllText(filePath));
43
44         Console.WriteLine("Team member list:");
45         foreach (var member in team.members)
46         {
47             Console.WriteLine($"{member.nim} {member.firstName} {member.lastName} " +
48                               $"{member.age} {member.gender}");
49         }
50     }
51 }

```

Program.cs

```

1  0 references
2  class Program
3  {
4      0 references
5      static void Main(string[] args)
6      {
7          //DataMahasiswa2211104008.ReadJSON();
8          //TeamMembers2211104008.ReadJSON();
9          GlossaryItem2211104008.ReadJSON();
10     }
11 }

```

- Output

```

Microsoft Visual Studio Debug Console
Team member list:
2211104008 Maria Nathasya (21 Female)
2211104010 Aubrey Raihan (21 Male)
2211104014 Syahrul Zaki (21 Male)
2211104021 Wahyu Insnantia (21 Male)

```

- Penjelasan

Program ini bertujuan membaca dan menampilkan daftar anggota tim dari file JSON (jurnal7_2_2211104008.json). Ini dilakukan dengan teknik parsing JSON, mengubah data teks JSON (berisi NIM, nama, gender, usia) menjadi objek C#

(Team). File TeamMembers2211104008.cs membaca dan mem-parsing JSON menggunakan JsonSerializer.Deserialize<Team>(). Objek Team memiliki array members berisi informasi setiap anggota. Setelah parsing, program mencetak daftar anggota ke layar. File Program.cs memanggil metode ReadJSON() untuk menjalankan proses parsing dan tampilan ini 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

Jurnal7_3_221104008

```
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 }
```

GlossaryItem2211104008.cs

```

1  using System;
2  using System.IO;
3  using System.Text.Json;
4
5  0 references
6  class GlossaryItem2211104008
7  {
8      1 reference
9      class GlossDef
10     {
11         1 reference
12         public string para { get; set; }
13         1 reference
14         public string[] GlossSeeAlso { get; set; }
15     }
16
17     2 references
18     class GlossEntry
19     {
20         1 reference
21         public string ID { get; set; }
22         1 reference
23         public string SortAs { get; set; }
24         1 reference
25         public string GlossTerm { get; set; }
26         1 reference
27         public string Acronym { get; set; }
28         1 reference
29         public string Abbrev { get; set; }
30         2 references
31         public GlossDef GlossDef { get; set; }
32         1 reference
33         public string GlossSee { get; set; }
34     }
35
36     1 reference
37     class Glosslist
38     {
39         1 reference
40         public GlossEntry GlossEntry { get; set; }
41     }
42
43     1 reference
44     class GlossDiv
45     {
46         0 references
47         public string title { get; set; }
48         1 reference
49         public GlossList GlossList { get; set; }
50     }
51
52     1 reference
53     class Glossary
54     {

```



```

37 |         0 references
    |         public string title { get; set; }
    |         1 reference
38 |         public GlossDiv GlossDiv { get; set; }
39 |     }
40 |
    |     2 references
41 |     class Root
42 |     {
    |         1 reference
43 |         public Glossary glossary { get; set; }
44 |     }
45 |
    |     0 references
46 |     public static void ReadJSON()
47 |     {
48 |         string filePath = "jurnal7_3_2211104008.json";
49 |
50 |         if (!File.Exists(filePath))
51 |         {
52 |             Console.WriteLine("File JSON tidak ditemukan.");
53 |             return;
54 |         }
55 |
56 |         // Membaca dan parsing JSON
57 |         Root data = JsonSerializer.Deserialize<Root>(File.ReadAllText(filePath));
58 |
59 |         // Mengambil bagian GlossEntry
60 |         GlossEntry entry = data.glossary.GlossDiv.GlossList.GlossEntry;
61 |
62 |         Console.WriteLine("===== Glossary Entry =====");
63 |         Console.WriteLine($"ID : {entry.ID}");
64 |         Console.WriteLine($"SortAs : {entry.SortAs}");
65 |         Console.WriteLine($"Term : {entry.GlossTerm}");
66 |         Console.WriteLine($"Acronym : {entry.Acronym}");
67 |         Console.WriteLine($"Abbrev : {entry.Abbrev}");
68 |         Console.WriteLine($"Definition: {entry.GlossDef.para}");
69 |         Console.WriteLine("See Also : ");
70 |         Console.WriteLine(string.Join(", ", entry.GlossDef.GlossSeeAlso));
71 |         Console.WriteLine($"GlossSee : {entry.GlossSee}");
72 |     }
73 | }

```

Program.cs

```

    |     0 references
1 |     class Program
2 |     {
    |         0 references
3 |         static void Main(string[] args)
4 |         {
5 |             //DataMahasiswa2211104008.ReadJSON();
6 |             //TeamMembers2211104008.ReadJSON();
7 |             GlossaryItem2211104008.ReadJSON();
8 |         }
9 |     }

```

- Output

```

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

```

- Penjelasan

Program ini membaca dan menampilkan data glosarium dari file JSON (jurnal7_3_2211104008.json). File JSON ini berisi bagian utama "glossary" dengan "GlossDiv" dan "GlossList" yang menyimpan "GlossEntry" berisi ID, GlossTerm,

Acronym, Abbreviation, Definition, dan referensi terkait. Prosesnya melibatkan parsing (deserialisasi) data JSON menjadi objek C#. Di file `GlossaryItem2211104008.cs`, program mendefinisikan kelas-kelas yang sesuai dengan struktur JSON. Metode `ReadJSON()` membaca file JSON, lalu melakukan parsing menggunakan `JsonSerializer.Deserialize<Root>()`. Setelah data berhasil di-parse, program mengekstrak bagian "GlossEntry" dan mencetaknya ke konsol dalam format yang mudah dibaca. Teknik parsing ini memungkinkan akses mudah ke setiap elemen data tanpa perlu membaca file secara manual.