

**TUGAS PENDAHULUAN  
KONSTRUKSI PERANGKAT LUNAK**

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



**Disusun Oleh:**

Nadia Putri Rahmaniar / 2211104012

S1 SE-06-01

**Dosen Pengampu:**

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

**PROGRAM STUDI S1 SOFTWARE ENGINEERING FAKULTAS  
INFORMATIKA  
TELKOM UNIVERSITY PURWOKERTO  
2025**

## TUGAS PENDAHULUAN

### 1. MENAMBAHKAN JSON DESERIALIZATON 1

Buatlah branch baru dengan nama branch “nama\_panggilan\_praktikan” dan checkout kesana.

- Download file “tp7\_1\_nim.json” dan rename file tersebut dengan mengganti “nim” dengan NIM praktikan kemudian pindahkan file json tersebut di folder solution projectnya.
- Ganti isian nama dan nim di dalam file tersebut dengan nama dan nim 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:  
*“Nama <nama depan + belakang> dengan nim <nim> dari fakultas <fakultas>”*

**Jawab :**

File tp7\_1\_2211104012.json

```
1 {
2   "nama": {
3     "depan": "Nadia Putri",
4     "belakang": "Rahmaniar"
5   },
6   "nim": 2211104012,
7   "fakultas": "informatika"
8 }
```

## File DataMahasiswa2211104012.cs

```
1  using System;
2  using System.IO;
3  using Newtonsoft.Json;
4
5  namespace TP07
6  {
7      1 reference
8      class Nama
9      {
10         1 reference
11         public string depan { get; set; }
12         1 reference
13         public string belakang { get; set; }
14     }
15
16     3 references
17     class DataMahasiswa2211104012
18     {
19         2 references
20         public Nama nama { get; set; }
21         1 reference
22         public string nim { get; set; }
23         1 reference
24         public string fakultas { get; set; }
25
26         1 reference
27         public static void ReadJSON()
28         {
29             try
30             {
31                 // Ambil path file JSON di dalam folder proyek
32                 string filePath = Path.Combine(Directory.GetCurrentDirectory(), "tp7_1_2211104012.json");
33
34                 // Cek apakah file ada
35                 if (!File.Exists(filePath))
```

## File Program.cs

```
1  using TP07;
2
3  0 references
4  class Program
5  {
6      0 references
7      static void Main()
8      {
9          DataMahasiswa2211104012.ReadJSON();
10     }
```

## Output

```
Nama: Nadia Putri Rahmaniar
NIM: 2211104012
Fakultas: informatika
```

## Penjelasan

File tp7\_1\_2211104012.json berisi data mahasiswa dalam format JSON, yang menyimpan informasi seperti nama depan, nama belakang, NIM, dan fakultas. File ini menjadi sumber data yang akan dibaca oleh program untuk kemudian diolah dan ditampilkan.

Untuk membaca data tersebut, terdapat kelas DataMahasiswa2211104012.cs yang memiliki metode ReadJSON(). Metode ini pertama-tama memeriksa keberadaan file JSON. Jika file tersedia, program akan membacanya dan

mengonversi isinya menjadi objek C# menggunakan pustaka Newtonsoft.Json. Setelah konversi berhasil, data dari file JSON seperti nama, NIM, dan fakultas akan diambil dan ditampilkan ke layar. Jika file tidak ditemukan, program akan memberikan pesan kesalahan agar pengguna mengetahui bahwa file tidak tersedia atau belum diletakkan di lokasi yang benar.

Sementara itu, file Program.cs berfungsi sebagai titik awal eksekusi program dengan metode Main(). Metode ini bertugas untuk memanggil ReadJSON() dari kelas DataMahasiswa2211104012, sehingga saat program dijalankan, data mahasiswa dari file JSON langsung diproses dan ditampilkan. Dengan mekanisme ini, program dapat secara otomatis mengambil data mahasiswa dan menampilkannya, serta memberikan notifikasi jika terjadi kesalahan saat membaca file JSON.

## 2. MELAKUKAN COMMIT

Pada branch yang dibuat sebelumnya:

- a. Lakukan commit dengan pesan “menambahkan class <NAMA\_CLASS>”.

```
ASUS@DESKTOP-I50G73S MINGW64 /d/School/KPL_NadiaPutriRahmaniar_2211104012 (nadia
putri)
$ git commit -m "menambahkan class DataMahasiswa2211104012"
[nadiaputri ab2bb41] menambahkan class DataMahasiswa2211104012
3 files changed, 68 insertions(+)
 create mode 100644 Praktik/07_Grammar-Based_Input_Processing_Parsing/DataMahasi
swa2211104012.cs
 create mode 100644 Praktik/07_Grammar-Based_Input_Processing_Parsing/Program.cs
 create mode 100644 Praktik/07_Grammar-Based_Input_Processing_Parsing/tp7_1_2211
104012.json
```

- b. Lakukan push ke github ke branch yang dibuat di bagian sebelumnya.

```
ASUS@DESKTOP-I50G73S MINGW64 /d/School/KPL_NadiaPutriRahmaniar_2211104012 (nadia
putri)
$ git push origin nadiaputri
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 8 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (7/7), 1.26 KiB | 322.00 KiB/s, done.
Total 7 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
remote:
remote: Create a pull request for 'nadiaputri' on GitHub by visiting:
remote:   https://github.com/nadiaputrirah/KPL_NadiaPutriRahmaniar_2211104012
/pull/new/nadiaputri
remote:
To https://github.com/nadiaputrirah/KPL_NadiaPutriRahmaniar_2211104012.git
 * [new branch]      nadiaputri -> nadiaputri
```

## 3. MENAMBAHKAN JSON DESERIALIZED 2

Buatlah branch baru dengan nama branch “nama\_panggilan\_praktikan” dan checkout kesana.

- a. Download file “tp7\_2\_nim.json” dan rename file tersebut dengan mengganti

“nim” dengan NIM praktikan kemudian pindahkan file json tersebut di folder solution projectnya.

- b. Ganti kode mata kuliah dan nama kuliah sesuai dengan daftar mata kuliah yang diambil di semester ini.
- c. Buatlah sebuah file class baru dengan nama “KuliahMahasiswa<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:

*“Daftar mata kuliah yang diambil:”*

*“MK 1 <kode\_matakuliah\_1> - <nama\_matakuliah\_1>”*

*“MK 2 <kode\_matakuliah\_2> - <nama\_matakuliah\_2>”*

*“MK 3 <kode\_matakuliah\_3> - <nama\_matakuliah\_3>”*

dst.

**Jawab :**

File tp7\_2\_2211104012.json

```
{
  "courses": [
    {
      "code": "CRI2C4",
      "name": "Konstruksi Perangkat Lunak"
    },
    {
      "code": "CRI2C5",
      "name": "Proyek Tingkat 2"
    },
    {
      "code": "CRI2C6",
      "name": "Design Thinking"
    },
    {
      "code": "CRI2C7",
      "name": "Kecerdasan Buatan"
    },
    {
      "code": "CRI2C8",
      "name": "Praktikum Konstruksi Perangkat Lunak"
    },
    {
      "code": "CRI2C9",
      "name": "Tata Tulis Ilmiah"
    },
    {
      "code": "CRI2C10",
      "name": "Manajemen Evolusi dan Konfigurasi Perangkat Lunak"
    }
  ]
}
```

File KuliahMahasiswa2211104017.cs

```
1  using System;
2  using System.IO;
3  using Newtonsoft.Json;
4  using System.Collections.Generic;
5
6  1 reference
7  class KuliahMahasiswa2211104012
8  {
9      1 reference
10     public class Course
11     {
12         1 reference
13         public string Code { get; set; }
14         1 reference
15         public string Name { get; set; }
16     }
17
18     2 references
19     public class CourseList
20     {
21         1 reference
22         public List<Course> Courses { get; set; }
23     }
24
25     1 reference
26     public static void ReadJSON()
27     {
28         string filePath = "tp7_2_2211104012.json";
29
30         // Cek apakah file JSON ada
31         if (!File.Exists(filePath))
32         {
33             Console.WriteLine("File JSON tidak ditemukan.");
34             return;
35         }
36
37         // Membaca isi file JSON
38         string jsonContent = File.ReadAllText(filePath);
39
40         // Melakukan deserialisasi JSON ke objek
41         CourseList data = JsonConvert.DeserializeObject<CourseList>(jsonContent);
42
43         // Menampilkan hasil
44         Console.WriteLine("Daftar mata kuliah yang diambil:");
45         int index = 1;
46         foreach (var course in data.Courses)
47         {
48             Console.WriteLine($"MK {index} {course.Code} - {course.Name}");
49             index++;
50         }
51     }
52 }
```

Output

```
Microsoft Visual Studio Debu: x + v
Nama: Nadia Putri Rahmania
NIM: 2211104012
Fakultas: informatika
Daftar mata kuliah yang diambil:
MK 1 CRI2C4 - Konstruksi Perangkat Lunak
MK 2 CRI2C5 - Proyek Tingkat 2
MK 3 CRI2C6 - Design Thinking
MK 4 CRI2C7 - Kecerdasan Buatan
MK 5 CRI2C8 - Praktikum Konstruksi Perangkat Lunak
MK 6 CRI2C9 - Tata Tulis Ilmiah
MK 7 CRI2C10 - Manajemen Evolusi dan Konfigurasi Perangkat Lunak
```

Penjelasan

Program ini berfungsi untuk membaca file JSON yang berisi daftar mata kuliah yang diambil mahasiswa dalam satu semester. File JSON ini berisi kode dan nama mata kuliah dalam format yang terstruktur. Agar dapat membaca file ini, dibuatlah kelas `KuliahMahasiswa2211104012`, yang bertanggung jawab untuk melakukan parsing file JSON dan menampilkannya dalam format yang lebih mudah dibaca oleh pengguna.

Kelas `KuliahMahasiswa2211104012` memiliki dua bagian utama, yaitu class `Course` yang merepresentasikan satu mata kuliah dengan properti `Code` dan `Name`, serta class `CourseList` yang menampung daftar semua mata kuliah dalam bentuk `List`. Kemudian, dalam metode `ReadJSON()`, program terlebih dahulu mengecek apakah file JSON tersedia dalam direktori proyek. Jika file tidak ditemukan, maka program akan menampilkan pesan kesalahan agar pengguna tahu bahwa file belum tersedia atau berada di lokasi yang salah.

Jika file ditemukan, program akan membaca isinya menggunakan `File.ReadAllText()` dan mengonversinya menjadi objek `CourseList` menggunakan pustaka `Newtonsoft.Json`. Setelah data berhasil diproses, program akan mencetak daftar mata kuliah dalam format yang telah ditentukan, di mana setiap mata kuliah akan ditampilkan dengan urutan yang sesuai.

Pada sisi lain, file `Program.cs` berfungsi sebagai titik masuk utama program. Di dalamnya terdapat metode `Main()` yang hanya memanggil metode `ReadJSON()` dari kelas `KuliahMahasiswa2211104012`. Dengan demikian, ketika program dijalankan, informasi mata kuliah yang tersimpan dalam file JSON akan langsung ditampilkan ke layar tanpa perlu interaksi tambahan dari pengguna.

Jika program berjalan dengan baik, pengguna akan melihat daftar mata kuliah beserta kode dan namanya. Namun, jika terjadi kesalahan seperti file JSON yang tidak ditemukan atau format yang tidak sesuai, program akan memberikan pesan kesalahan yang informatif agar pengguna dapat memperbaikinya.