

**LAPORAN JURNAL
MODUL 7**



Disusun Oleh :

Zaenarif Putra 'Ainurdin – 2311104049

Kelas :

SE-07-02

Dosen :

Yudha Islami Sulistya

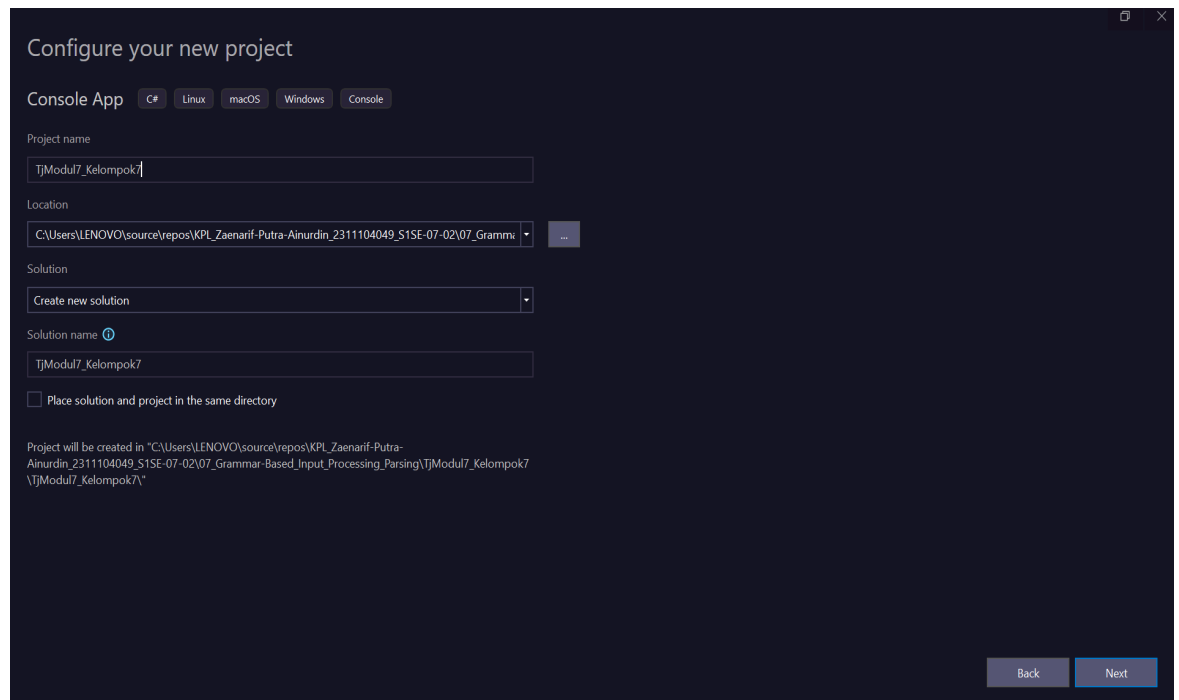
**PROGRAM STUDI SOFTWARE ENGINEERING
DIREKTORAT KAMPUS PURWOKERTO
TELKOM UNIVERSITY
PURWOKERTO
2025**

I. Link Github

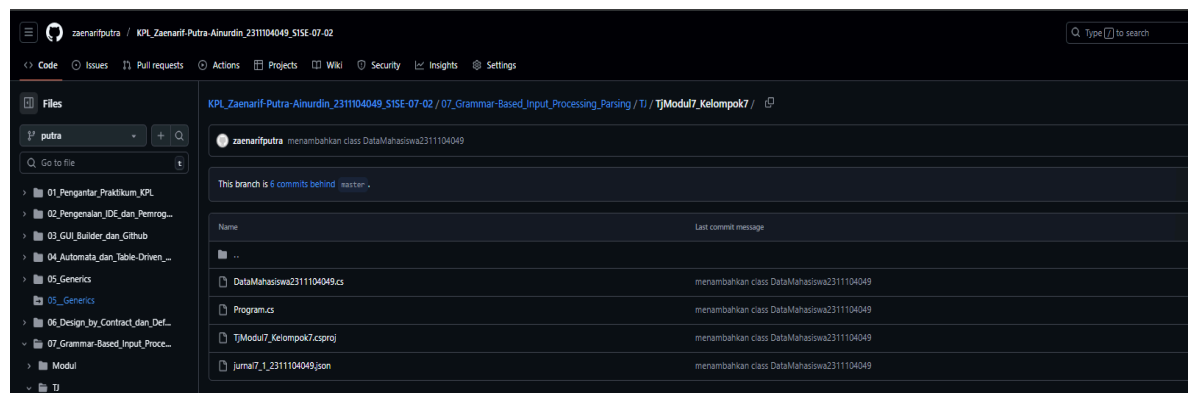
https://github.com/zaenarifputra/KPL_Zaenarif-Putra-Ainurdin_2311104049_S1SE-07-02/tree/1a07bb8542100d8c1e27aa6ea200414e56281c04/07_Grammar-Based_Input_Processing_Parsing/TJ

II. Alur Pengerjaan

1. Membuat Project Awalan Dengan ConsoleApp, dan diberi nama *TjModul7_Kelompok7*



2. Setelah Selesai membuat project selanjutnya adalah membuat branch yang diberi nama putra, yang dimana isi dari branch putra adalah class DataMahasiswa2311104049.cs, Program.cs dan jurnal7_1_2311104049.json yang dimana untuk commitnya adalah “menambahkan class DataMahasiswa2311104049, Berikut bukti dari branch putra beserta penjelasan syntax secara sederhana dari semua class yang ada :



Class DataMahasiswa2311104049

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

namespace TjModul7_Kelompok7
{
    public class Address
    {
        public string streetAddress { get; set; }
        public string city { get; set; }
        public string state { get; set; }
    }

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

    public class DataMahasiswa2311104049
    {
        public string firstName { get; set; }
        public string lastName { get; set; }
        public string gender { get; set; }
        public int age { get; set; }
        public Address address { get; set; }
        public Course[] courses { get; set; }

        public void ReadJSON()
        {
            string fileName = "jurnal7_1_2311104049.json";

            if (!File.Exists(fileName))
            {
                Console.WriteLine("File tidak ditemukan: " + fileName);
                return;
            }

            string jsonString = File.ReadAllText(fileName);

            var mahasiswa =
                JsonSerializer.Deserialize<DataMahasiswa2311104049>(jsonString);

            if (mahasiswa == null)
            {
                Console.WriteLine("Gagal melakukan deserialisasi JSON.");
                return;
            }
        }
    }
}
```

```
Console.WriteLine($"Nama: {mahasiswa.firstName} {mahasiswa.lastName}");
Console.WriteLine($"Gender: {mahasiswa.gender}");
Console.WriteLine($"Age: {mahasiswa.age}");

if (mahasiswa.address != null)
{
    Console.WriteLine($"Address: {mahasiswa.address.streetAddress},
{mahasiswa.address.city}, {mahasiswa.address.state}");
}

if (mahasiswa.courses != null)
{
    Console.WriteLine("Courses:");
    foreach (var course in mahasiswa.courses)
    {
        Console.WriteLine($"- {course.code}: {course.name}");
    }
}
}
```

Pada syntax berikut merupakan syntax sederhana yang dibangun dengan C# untuk membaca data mahasiswa dari file JSON yang bernama jurnal7_1_2311104049.json. Dalam program ini terdapat tiga kelas utama: Address, Course, dan DataMahasiswa2311104049.

Kelas Address digunakan untuk menyimpan informasi alamat mahasiswa, termasuk nama jalan, kota, dan provinsi. Sementara itu, kelas Course berfungsi untuk menyimpan data mengenai mata kuliah yang diambil oleh mahasiswa, yang mencakup kode dan nama mata kuliah. Kelas utama, DataMahasiswa2311104049, berfungsi sebagai representasi keseluruhan data mahasiswa, dengan properti seperti nama depan, nama belakang, jenis kelamin, umur, alamat (dari kelas Address), dan daftar mata kuliah (dari array Course[]).

Fungsi ReadJSON() yang terdapat dalam kelas DataMahasiswa2311104049 bertugas untuk membaca file JSON tersebut. Pertama-tama, program akan memeriksa keberadaan file. Jika file tidak ditemukan, program akan memberikan informasi bahwa file tersebut tidak ada. Namun, jika file ada, isinya akan dibaca dan diupayakan untuk dikonversi (deserialisasi) menjadi objek DataMahasiswa2311104049. Jika proses deserialisasi berhasil, data mahasiswa seperti nama, jenis kelamin, umur, alamat, dan daftar mata kuliah akan ditampilkan di konsol.

Class Program.cs

```
using System;
using TjModul7_Kelompok7;

class Program
{
```

```
static void Main(string[] args)
{
    Console.WriteLine("=== [1] Menampilkan Parsing Data Mahasiswa ===");
    var dataMahasiswa = new DataMahasiswa2311104049();
    dataMahasiswa.ReadJSON();
    Console.ReadLine();
}
}
```

Pada program di atas merupakan file Main, yang bisa dianggap sebagai awal dari eksekusi program. Di dalam fungsi Main(), akan muncul tulisan di konsol: "=== [1] Menampilkan Parsing Data Mahasiswa ===", yang menandakan bahwa program akan menampilkan data mahasiswa yang diambil dari file JSON.

Selanjutnya, dibuatlah sebuah objek dari kelas DataMahasiswa2311104049 yang sebelumnya telah didefinisikan dalam namespace TjModul7_Kelompok7. Objek ini kemudian digunakan untuk memanggil metode dari ReadJSON(), yang berfungsi untuk membaca file JSON, mengonversi isinya menjadi objek C#, dan menampilkan informasi mahasiswa seperti nama, umur, jenis kelamin, alamat, serta daftar mata kuliah ke konsol.

Class jurnal7_1_2311104049.json

```
{
  "firstName": "Zaenarif",
  "lastName": "Putra",
  "gender": "Ainurdin",
  "age": 20,
  "address": {
    "streetAddress": "Lengkong",
    "city": "Bandung",
    "state": "West Java"
  },
  "courses": [
    { "code": "CRI2C4", "name": "Konstruksi Perangkat Lunak" },
    { "code": "CCK2HAB4", "name": "Basis Data" }
  ]
}
```

Pada class JSON berikut, yang dimana berisi data seorang mahasiswa bernama Zaenarif Putra menunjukkan bahwa ia berusia 20 tahun. Namun, ada keanehan pada data gendernya yang tercatat sebagai "Ainurdin", yang mungkin diisi tidak sesuai dengan format yang umum, di mana biasanya gender dicantumkan sebagai "Male" atau "Female". Zaenarif tinggal di Lengkong, Kota Bandung, Provinsi Jawa Barat.

Selain informasi pribadi, data dalam format JSON ini juga mencakup daftar mata kuliah yang diambil oleh Zaenarif. Ia terdaftar dalam dua mata kuliah, yaitu "Konstruksi Perangkat Lunak" dengan kode CRI2C4 dan "Basis Data" dengan kode CCK2HAB4. Struktur data ini sangat sesuai untuk digunakan dalam aplikasi C# yang telah kita bahas sebelumnya, karena dapat dengan mudah di-parsing menjadi objek

DataMahasiswa2311104049. Dengan demikian, JSON ini memberikan gambaran yang sederhana namun lengkap mengenai profil dan jadwal kuliah seorang mahasiswa.

Selanjutnya setelah menyelesaikan semua class dan menyelesaikan semua syntax, bisa melanjutkan commit yang dimana untuk pesan pada commitnya adalah “menambahkan class DataMahasiswa2311104049” untuk buktinya sebagai berikut :



Name	Last commit message	Last commit date
...		
DataMahasiswa2311104049.cs	menambahkan class DataMahasiswa2311104049	2 days ago
Program.cs	menambahkan class DataMahasiswa2311104049	2 days ago
TjModul7_Kelompok7.csproj	menambahkan class DataMahasiswa2311104049	2 days ago
jurnal7_1_2311104049.json	menambahkan class DataMahasiswa2311104049	2 days ago

- Selanjutnya melanjutkan branch selanjutnya yaitu putra2 sama dengan branch putra yang dimana terdapat 3 class juga yaitu class TeamMembers2311104049, Program.cs dan jurnal7_2_2311104049.json. Selanjutnya penjelasan singkat mengenai syntax pada branch putra2 dan juga commit yang dilakukan pada branch putra2 :

Class TeamMembers2311104049

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

namespace TjModul7_Kelompok7
{
    public class Member
    {
        public string firstName { get; set; }
        public string lastName { get; set; }
        public string gender { get; set; }
        public int age { get; set; }
        public string nim { get; set; }
    }

    public class TeamMembers2311104049
    {
        public Member[] members { get; set; }

        public void ReadJSON()
        {
            try
            {
                string jsonString = File.ReadAllText("jurnal7_2_2311104049.json");

                TeamMembers2311104049 team =
                JsonSerializer.Deserialize<TeamMembers2311104049>(jsonString);
            }
            catch { }
        }
    }
}
```

```
        if (team == null || team.members == null)
        {
            Console.WriteLine("Data JSON kosong atau tidak sesuai format.");
            return;
        }

        Console.WriteLine("Team member list:");
        foreach (var member in team.members)
        {
            Console.WriteLine($"{member.nim} {member.firstName}
{member.lastName} ({member.age} {member.gender})");
        }
    }
    catch (FileNotFoundException)
    {
        Console.WriteLine("File JSON tidak ditemukan.");
    }
    catch (JsonException ex)
    {
        Console.WriteLine($"Terjadi kesalahan saat membaca JSON:
{ex.Message}");
    }
    catch (Exception ex)
    {
        Console.WriteLine($"Error tak terduga: {ex.Message}");
    }
}
}
```

Program ini digunakan untuk membaca data pada anggota tim (team member) dari file JSON yang bernama jurnal7_2_2311104049.json. Data JSON ini berisi informasi beberapa anggota, seperti nama depan (firstName), nama belakang (lastName), gender, usia, dan NIM. Di dalam kode, ada dua class utama: Member, yang berfungsi menyimpan data masing-masing anggota. TeamMembers2311104049, yang berisi sebuah array dari Member dan mempunyai fungsi

ReadJSON() digunakan untuk membaca dan menampilkan datanya. Ketika program dijalankan, dia akan mencoba membaca file JSON tadi. Kalau file-nya ada dan formatnya benar, maka semua data anggota akan ditampilkan satu per satu di console dalam format yang rapi: NIM, nama lengkap, usia, dan gender-nya. Tapi kalau file-nya nggak ketemu atau formatnya salah, program ini juga sudah siap menampilkan pesan error yang jelas, jadi kita tahu di mana masalahnya.

Class Program.cs

```
using System;
using TjModul7_Kelompok7;

class Program
{
    static void Main(string[] args)
```

```
{  
    Console.WriteLine("=== [2] Menampilkan Parsing Team Members Mahasiswa  
===");  
    var teamMembers = new TeamMembers2311104049();  
    teamMembers.ReadJSON();  
    Console.ReadLine();  
}  
}
```

Program yang digunakan untuk menampilkan data anggota tim mahasiswa dari sebuah file JSON. Di dalam Main, pertama-tama program akan mencetak judul ke layar: "Menampilkan Parsing Team Members Mahasiswa". Lalu, dibuatlah objek dari kelas TeamMembers2311104049, yang kemungkinan besar berisi fungsi-fungsi untuk membaca dan menampilkan data anggota tim. Setelah objek dibuat, program memanggil fungsi ReadJSON() yang bertugas membaca data dari file JSON dan mungkin langsung menampilkannya ke layar. Terakhir, Console.ReadLine() dipakai agar konsol tidak langsung tertutup, jadi pengguna bisa lihat dulu hasil output-nya sebelum program selesai.

Class jurnal7_2_2311104049.json

```
{  
  "members": [  
    {  
      "firstName": "Zaenarif",  
      "lastName": "Putra",  
      "gender": "male",  
      "age": 20,  
      "nim": "2311104049"  
    },  
    {  
      "firstName": "Izzaty",  
      "lastName": "Zahara",  
      "gender": "female",  
      "age": 20,  
      "nim": "2311104049"  
    },  
    {  
      "firstName": "Emily",  
      "lastName": "Kelies",  
      "gender": "female",  
      "age": 24,  
      "nim": "2311104090"  
    },  
    {  
      "firstName": "Mc",  
      "lastName": "Wizz",  
      "gender": "male",  
      "age": 22,  
      "nim": "2311104080"  
    }  
  ]  
}
```


Data json berikut digunakan sebagai tempat nyimpen informasi tentang beberapa anggota tim mahasiswa. Di dalamnya ada satu array besar bernama members, yang isinya adalah beberapa objek—masing-masing mewakili satu orang anggota tim. Tiap anggota punya data lengkap seperti firstName (nama depan), lastName (nama belakang), gender (jenis kelamin), age (umur), dan nim (Nomor Induk Mahasiswa). Misalnya, ada Zaenarif Putra yang berumur 20 tahun dan punya NIM 2311104049, dan ada juga Emily Kelies yang berumur 24 tahun dengan NIM 2311104090.

Commit Branch putra2



4. Sama dengan sebelumnya selanjutnya berpindah menuju branch baru yaitu putra3 yang dimana untuk class pada branch putra3 yaitu GlossaryItem2311104049.cs, Program.cs dan jurnal7_3_2311104049.json kemudian akan dijelaskan secara sederhana mengenai 3 class yang tersedia. Kemudian melakukan commit “menambahkan class GlossaryItem2311104049” berikut penjelasan serta commitnya

Class GlossaryItem2311104049.cs

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

namespace TjModul7_Kelompok7
{
    public class GlossDef
    {
        public string para { get; set; }
        public string[] GlossSeeAlso { get; set; }
    }

    public class GlossEntry
    {
        public string ID { get; set; }
        public string SortAs { get; set; }
        public string GlossTerm { get; set; }
        public string Acronym { get; set; }
        public string Abbrev { get; set; }
    }
}
```

```
public GlossDef GlossDef { get; set; }
public string GlossSee { get; set; }
}

public class GlossList
{
    public GlossEntry[] GlossEntry { get; set; } // <- sudah jadi array
}

public class GlossDiv
{
    public string title { get; set; }
    public GlossList GlossList { get; set; }
}

public class Glossary
{
    public string title { get; set; }
    public GlossDiv GlossDiv { get; set; }
}

public class GlossaryItem2311104049
{
    public Glossary glossary { get; set; }

    public void ReadJSON()
    {
        string jsonString = File.ReadAllText("jurnal7_3_2311104049.json");
        GlossaryItem2311104049 glossaryItem =
        JsonSerializer.Deserialize<GlossaryItem2311104049>(jsonString);

        var entries = glossaryItem.glossary.GlossDiv.GlossList.GlossEntry;

        Console.WriteLine("GlossEntry List:");
        foreach (var entry in entries)
        {
            Console.WriteLine("-----");
            Console.WriteLine($"ID: {entry.ID}");
            Console.WriteLine($"SortAs: {entry.SortAs}");
            Console.WriteLine($"GlossTerm: {entry.GlossTerm}");
            Console.WriteLine($"Acronym: {entry.Acronym}");
            Console.WriteLine($"Abbrev: {entry.Abbrev}");
            Console.WriteLine($"Definition: {entry.GlossDef.para}");
            Console.WriteLine("See Also: " + string.Join(" ",
entry.GlossDef.GlossSeeAlso));
            Console.WriteLine($"See: {entry.GlossSee}");
        }
    }
}
}
```

Syntax berikut digunakan untuk menampilkan isi dari file JSON yang dimana strukturnya cukup kompleks, khususnya tentang entri kamus atau glossary. Kelas-kelas seperti GlossDef, GlossEntry, GlossList, GlossDiv, dan Glossary dibikin untuk menyesuaikan struktur isi JSON-nya. Misalnya, satu entri kamus (GlossEntry) punya ID, istilah, akronim, singkatan, definisi (dalam GlossDef), dan daftar referensi lain.

Nah, untuk kelas utama di sini adalah GlossaryItem2311104049 yang punya method ReadJSON(). Di dalam method ini, file jurnal7_3_2311104049.json dibaca, lalu isinya diubah jadi objek C# pake JsonSerializer. Setelah itu, program bakal looping tiap entri glossary dan menampilkannya ke konsol, lengkap dari ID sampai referensi.

Class Program.cs

```
using System;
using TjModul7_Kelompok7;

class Program
{
    static void Main(string[] args)
    {
        Console.WriteLine("=== [3] Menampilkan Parsing Glossary Item Mahasiswa  
===");
        var teamMembers = new GlossaryItem2311104049();
        teamMembers.ReadJSON();
        Console.ReadLine();
    }
}
```

Pada syntax ini berfungsi buat nampilin data glossary (istilah-istilah kamus) mahasiswa dari file JSON. Di dalam Main(), pertama-tama ditampilkan dulu judul: "Menampilkan Parsing Glossary Item Mahasiswa" ke layar biar pengguna tahu apa yang sedang dilakukan program. Lalu dibuatlah objek dari kelas GlossaryItem2311104049, yang sebelumnya udah didefinisikan buat membaca dan menampilkan isi glossary. Setelah objeknya siap, program memanggil method ReadJSON() buat membaca file JSON dan menampilkan isinya satu per satu. Console.ReadLine() dipakai biar jendela konsol nggak langsung nutup dan pengguna bisa baca hasil output-nya dulu.

Class jurnal7_3_2311104049.json

```
{
  "glossary": {
    "title": "Advanced Glossary of Markup Languages",
    "GlossDiv": {
      "title": "Markup Concepts",
      "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 define markup languages. SGML is
the parent of both HTML and XML.",
      "GlossSeeAlso": [ "GML", "XML" ]
    },
    "GlossSee": "markup"
  },
  {
    "ID": "XML",
    "SortAs": "XML",
    "GlossTerm": "Extensible Markup Language",
    "Acronym": "XML",
    "Abbrev": "W3C Recommendation",
    "GlossDef": {
      "para": "A flexible text format derived from SGML. XML is widely used for
data exchange on the web.",
      "GlossSeeAlso": [ "HTML", "XHTML" ]
    },
    "GlossSee": "markup"
  },
  {
    "ID": "HTML",
    "SortAs": "HTML",
    "GlossTerm": "HyperText Markup Language",
    "Acronym": "HTML",
    "Abbrev": "HTML5",
    "GlossDef": {
      "para": "The standard markup language for creating web pages and
applications. HTML defines the structure of web content.",
      "GlossSeeAlso": [ "CSS", "JavaScript" ]
    },
    "GlossSee": "markup"
  }
] } } }

```

Pada data json jurnal7_3_2311104049 berisi informasi tentang istilah-istilah penting dalam dunia markup language, atau bahasa penandaan yang sering dipakai buat bikin dan atur struktur dokumen di web. Di dalam objek utama glossary, ada judul "Advanced Glossary of Markup Languages", dan bagian di dalamnya disebut GlossDiv yang punya judul "Markup Concepts". Di dalamnya ada daftar istilah (GlossEntry) yang masing-masing menjelaskan satu teknologi markup, seperti SGML, XML, dan HTML.

Setiap entri nyimpen informasi penting: ID-nya, istilah lengkap (GlossTerm), akronim, singkatan resmi (Abbrev), definisi (GlossDef), dan daftar istilah lain yang masih berhubungan lewat GlossSeeAlso. Misalnya, SGML dijelaskan sebagai "induknya" HTML dan XML, sedangkan XML dijelasin sebagai format fleksibel untuk pertukaran data, dan HTML adalah bahasa standar buat bikin halaman web.

Commit branch putra3

KPL_Zaenarif-Putra-Ainurdin_2311104049_S1SE-07-02 / 07_Grammar-Based_Input_Processing_Parsing / TJ / TJModul7_Kelompok7 /

zaenarifputra menambahkan class GlossaryItem2311104049

This branch is 4 commits behind master .

Name	Last commit message
..	
GlossaryItem2311104049.cs	menambahkan class GlossaryItem2311104049
Program.cs	menambahkan class GlossaryItem2311104049
TJModul7_Kelompok7.csproj	menambahkan class DataMahasiswa2311104049
jurnal7_3_2311104049.json	menambahkan class GlossaryItem2311104049

- Setelah menyelesaikan semua dari ke-3 branch yang sudah dilakukan selanjutnya kita akan pull request dan juga merge menyatukan semua syntax yang ada pada branch 1 hingga 3 ke branch master yang dimana ketika sudah disatukan semuanya maka semua class akan disatukan kedalam branch master seperti berikut :

KPL_Zaenarif-Putra-Ainurdin_2311104049_S1SE-07-02 / 07_Grammar-Based_Input_Processing_Parsing / TJ / TJModul7_Kelompok7 /

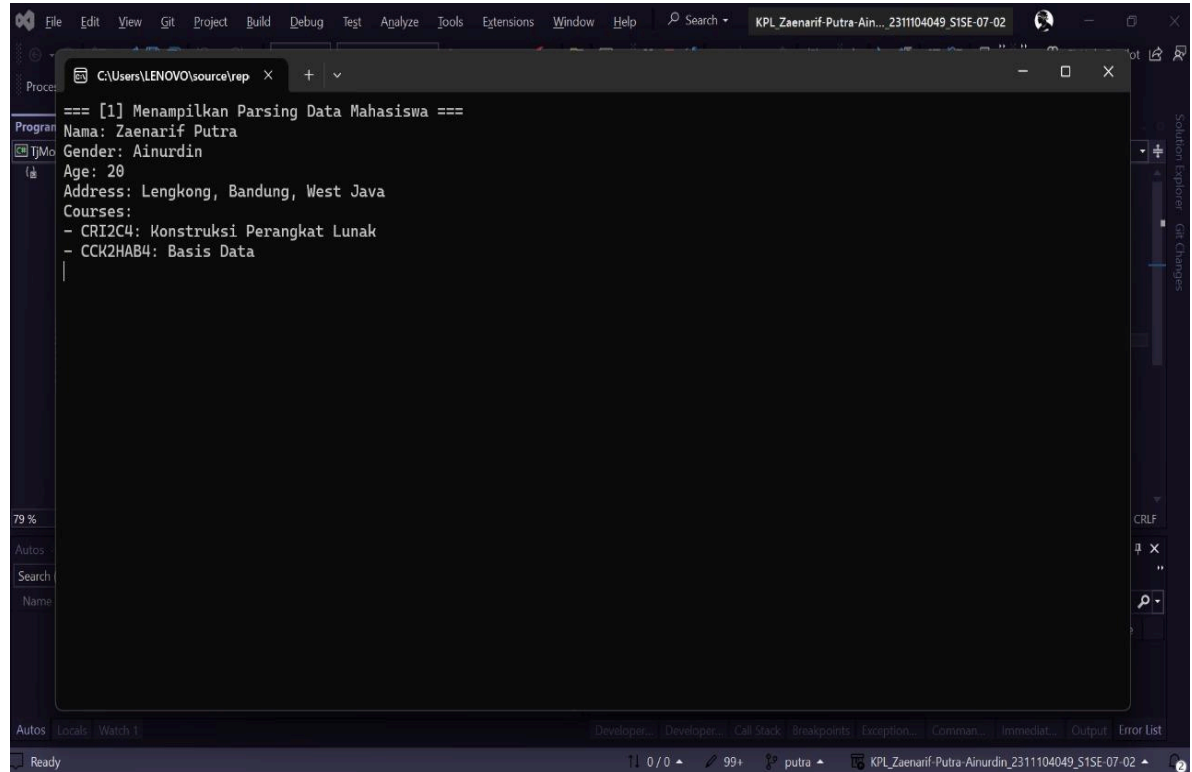
zaenarifputra menyelesaikan conflict merge putra dan putra2

Name	Last commit message
..	
DataMahasiswa2311104049.cs	menyelesaikan conflict merge putra dan putra2
GlossaryItem2311104049.cs	menambahkan class GlossaryItem2311104049
Program.cs	menyelesaikan conflict merge putra dan putra2
TeamMembers2311104049.cs	menyelesaikan conflict merge putra dan putra2
TJModul7_Kelompok7.csproj	menambahkan class DataMahasiswa2311104049
jurnal7_1_2311104049.json	menyelesaikan conflict merge putra dan putra2
jurnal7_2_2311104049.json	menyelesaikan conflict merge putra dan putra2
jurnal7_3_2311104049.json	menambahkan class GlossaryItem2311104049

Setelah semua program dan sudah merge menuju branch master modul7 sudah berhasil di selesaikan

III. Hasil Running

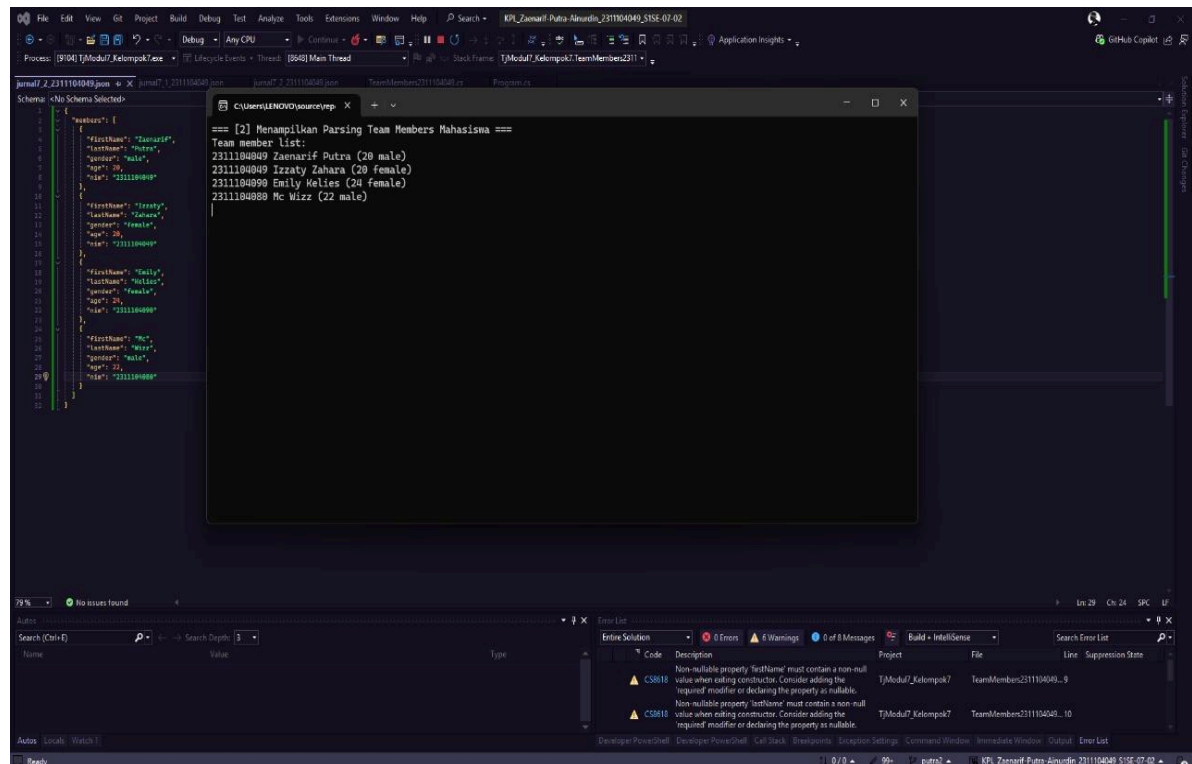
1. Hasil Running Pada Branch putra



```

C:\Users\LENOVO\source\rep x + v
=== [1] Menampilkan Parsing Data Mahasiswa ===
Nama: Zaenarif Putra
Gender: Ainurdin
Age: 20
Address: Lengkong, Bandung, West Java
Courses:
- CRI2C4: Konstruksi Perangkat Lunak
- CCK2HAB4: Basis Data
  
```

2. Hasil Running Pada Branch putra2



```

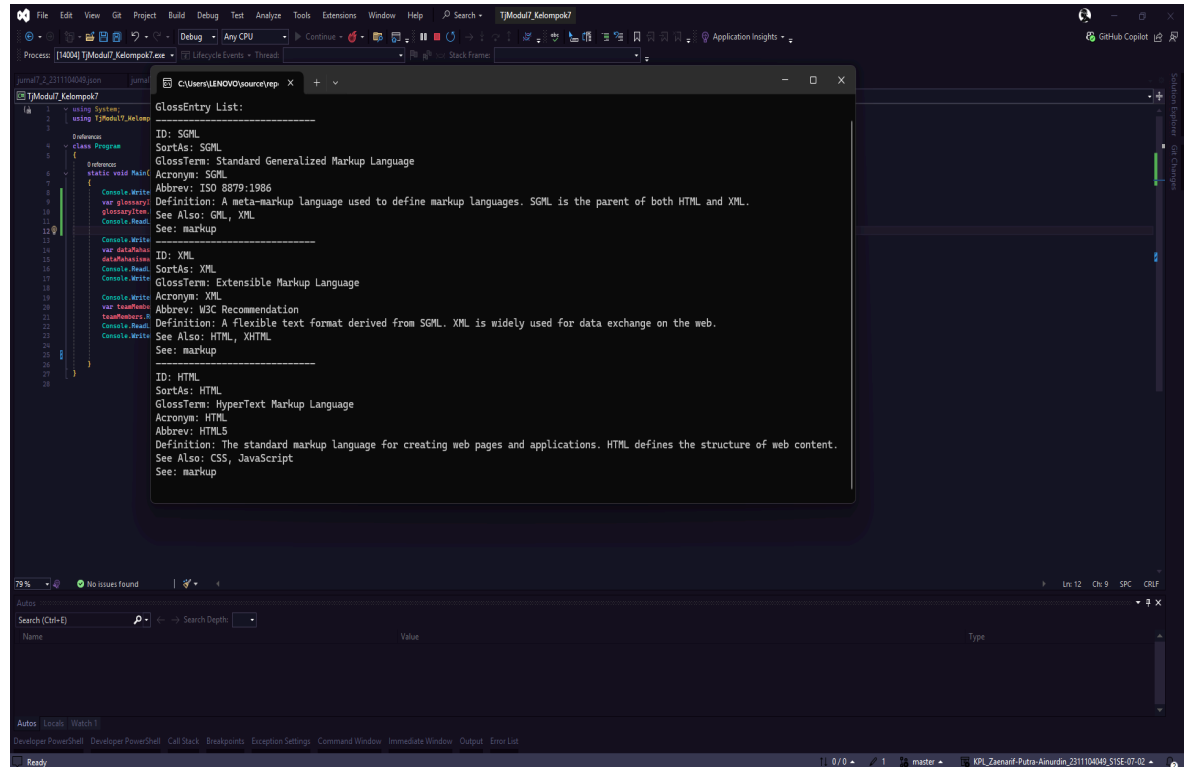
C:\Users\LENOVO\source\rep x + v
=== [2] Menampilkan Parsing Team Members Mahasiswa ===
Team member List:
2311104049 Zaenarif Putra (20 male)
2311104049 Izzaty Zahara (20 female)
2311104090 Emily Helies (20 female)
2311104088 Mc Wizz (22 male)
  
```

```

{
  "members": [
    {
      "first_name": "Zaenarif",
      "last_name": "Putra",
      "gender": "male",
      "age": 20,
      "id": "2311104049"
    },
    {
      "first_name": "Izzaty",
      "last_name": "Zahara",
      "gender": "female",
      "age": 20,
      "id": "2311104090"
    },
    {
      "first_name": "Emily",
      "last_name": "Helies",
      "gender": "female",
      "age": 20,
      "id": "2311104088"
    },
    {
      "first_name": "Mc",
      "last_name": "Wizz",
      "gender": "male",
      "age": 22,
      "id": "2311104088"
    }
  ]
}
  
```

Code	Description	Project	File	Line	Suppression State
CSRI1	Non-nullable property 'first_name' must contain a non-null value when calling constructor. Consider adding the 'required' modifier or declaring the property as nullable.	TJModul7_Kelompok7	TeamMembers2311104049_9	9	
CSRI1	Non-nullable property 'last_name' must contain a non-null value when calling constructor. Consider adding the 'required' modifier or declaring the property as nullable.	TJModul7_Kelompok7	TeamMembers2311104049_10	10	

3. Hasil Running Pada Branch putra3



```
using System;
using TModul7_Kelompok7;

namespace TModul7_Kelompok7
{
    class Program
    {
        static void Main()
        {
            Console.WriteLine("GlossEntry List:");
            var glossary = new Glossary();
            glossary.AddEntry("SGML");
            Console.WriteLine(glossary.ToString());
            Console.WriteLine("-----");
            var dataXml = new XmlDocument();
            dataXml.Load("data.xml");
            Console.WriteLine(dataXml.ToString());
            Console.WriteLine("-----");
            var dataHtml = new HtmlDocument();
            dataHtml.Load("data.html");
            Console.WriteLine(dataHtml.ToString());
        }
    }
}
```

GlossEntry List:

ID: SGML
SortAs: SGML
GlossTerm: Standard Generalized Markup Language
Acronym: SGML
Abbrev: ISO 8879:1986
Definition: A meta-markup language used to define markup languages. SGML is the parent of both HTML and XML.
See Also: GML, XML
See: markup

ID: XML
SortAs: XML
GlossTerm: Extensible Markup Language
Acronym: XML
Abbrev: W3C Recommendation
Definition: A flexible text format derived from SGML. XML is widely used for data exchange on the web.
See Also: HTML, XHTML
See: markup

ID: HTML
SortAs: HTML
GlossTerm: HyperText Markup Language
Acronym: HTML
Abbrev: HTML5
Definition: The standard markup language for creating web pages and applications. HTML defines the structure of web content.
See Also: CSS, JavaScript
See: markup

Kesimpulan dari pengerjaan Modul 7 ini adalah kita belajar gimana cara parsing atau membaca data dari file JSON ke dalam program C# dengan berbagai macam struktur—mulai dari data pribadi mahasiswa, daftar anggota tim, sampai istilah-istilah teknis dalam bentuk glossary. Semua data itu diproses dan ditampilkan lewat class-class yang dibuat sesuai struktur JSON-nya. Intinya, modul ini ngajarin cara kerja dengan data terstruktur, cara ngelola file JSON, dan pentingnya desain program yang rapi supaya parsing data bisa berjalan lancar dan mudah dipahami.