Jurnal Modul 7

Doanta Aloycius Ginting (21104009)

Membuat file JSON

```
{
   "firstName": "Doanta",
   "lastName": "Ginting",
   "gender": "male",
   "age": 21,
   "address": {
       "streetAddress": "Purwokerto",
       "city": "Banyumas",
       "state": "Centra Java"
   },
   "courses": [
       { "code": "CRI2C4", "name": "Konstruksi Perangkat
Lunak" },
       { "code": "CCK4DAA4", "name": "Tugas Akhir" }
   ]
}
```

File DataMahasiswa dengan class dan function untuk membaca file JSON

```
using System.Text.Json;
namespace jurnal_modul7_21104009;

// Class yang merepresentasikan alamat
public class Address
{
    public string StreetAddress { get; set; }
    public string City { get; set; }
    public string State { get; set; }
}

// Class yang merepresentasikan mata kuliah
public class Course
{
    public string Code { get; set; }
    public string Name { get; set; }
}
```

```
// Class yang merepresentasikan informasi beserta alamat dan mata kuliahnya
public class Person
{
    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 List<Course> Courses { get; set; }
}

// Class utama untuk membaca dan menampilkan data dari file JSON
public class DataMahasiswa21104009
{
    // Metode untuk membaca data JSON dari file dan menampilkan informasi
    public void ReadJSON(string filePath)
    {
        try
        {
            // Membaca isi file JSON
            string jsonContent = File.ReadAllText(filePath);

            // Melakukan deserialisasi JSON ke dalam objek Person
            Person person = JsonSerializer.Deserialize<Person>(jsonContent, new JsonSerializerOptions { PropertyNameCaseInsensitive = true });

            // Menampilkan data hasil deserialisasi ke console
            Console.WriteLine("=== Deserialized JSON Object ===");
            Console.WriteLine($"Name: {person.FirstName} {person.LastName}");
            Console.WriteLine($"Gender: {person.Gender}");
            Console.WriteLine($"Age: {person.Age}");
            Console.WriteLine($"Address: {person.Address.StreetAddress}, {person.Address.City}, {person.Address.State}");
```

```
            Console.WriteLine("Courses:");
            foreach (var course in person.Courses)
            {
                Console.WriteLine($"  - {course.Code}:
{course.Name}");
            }
        }
        catch (Exception ex)
        {
            // Menangani kesalahan saat membaca atau memproses
file JSON
            Console.WriteLine($"Error reading JSON:
{ex.Message}");
        }
    }
}
```

Output:

```
=== Deserialized JSON Object ===
Name: Doanta Ginting
Gender: male
Age: 21
Address: Purwokerto, Banyumas, Centra Java
Courses:
  - CRI2C4: Konstruksi Perangkat Lunak
  - CCK4DAA4: Tugas Akhir
```

```
File JSON Team Members:
{
  "members" : [
    {
        "firstName": "Doanta",
        "lastName": "Ginting",
        "gender": "male",
        "age": 21,
        "nim": "21104009"
    },
```

```
    {
        "firstName": "James",
        "lastName": "Smooth",
        "gender": "male",
        "age": 32,
        "nim": "5678568567"
    },
    {
        "firstName": "Emily",
        "lastName": "Kelies",
        "gender": "female",
        "age": 24,
        "nim": "456754675"
    }
  ]
}
```

File TeamMembers yang berisi class dan method untuk membaca file JSON

```
using System.Text.Json;

// Class yang merepresentasikan anggota tim
public class TeamMember
{
    public string NIM { get; set; }
    public string FirstName { get; set; }
    public string LastName { get; set; }
    public string Gender { get; set; }
    public int Age { get; set; }
}

// Class yang merepresentasikan tim yang berisi daftar anggota
public class Team
{
    public List<TeamMember> Members { get; set; }
}

// Class utama untuk membaca dan menampilkan data anggota tim
dari file JSON
```

```
class TeamMembers21104009
{
    // Metode untuk membaca data JSON dari file dan menampilkan
informasi anggota tim
    public void ReadJSON(string filePath)
    {
        try
        {
            // Membaca isi file JSON
            string jsonContent = File.ReadAllText(filePath);

            // Melakukan deserialisasi JSON ke dalam objek Team
            Team team =
JsonSerializer.Deserialize<Team>(jsonContent, new
JsonSerializerOptions { PropertyNameCaseInsensitive = true });

            // Menampilkan daftar anggota tim ke console
            Console.WriteLine("Team member list:");
            foreach (var member in team.Members)
            {
                Console.WriteLine($"{member.NIM}
{member.FirstName} {member.LastName} ({member.Age}
{member.Gender})");
            }
        }
        catch (Exception ex)
        {
            // Menangani kesalahan saat membaca atau memproses
file JSON
            Console.WriteLine($"Error reading JSON:
{ex.Message}");
        }
    }
}
```

Output:

```
Team member list:
21104009 Doanta Ginting (21 male)
5678568567 James Smooth (32 male)
456754675 Emily Kelies (24 female)
```

File JSON Glossary Item

```
{
    "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 Glossary Item yang berisi class dan method untuk membaca file JSON

```
using System.Text.Json;
namespace jurnal_modul7_21104009;

// Class yang merepresentasikan definisi glosarium
public class GlossDef
{
    public string Para { get; set; }
    public List<string> GlossSeeAlso { get; set; }
}
```

```
// Class yang merepresentasikan entri glosarium
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; }
}

// Class yang merepresentasikan daftar entri glosarium
public class GlossList
{
    public GlossEntry GlossEntry { get; set; }
}

// Class yang merepresentasikan bagian glosarium
public class GlossDiv
{
    public string Title { get; set; }
    public GlossList GlossList { get; set; }
}

// Class yang merepresentasikan struktur utama glosarium
public class Glossary
{
    public string Title { get; set; }
    public GlossDiv GlossDiv { get; set; }
}

// Class root untuk glosarium dalam JSON
public class GlossaryRoot
{
    public Glossary Glossary { get; set; }
}

// Class utama untuk membaca dan menampilkan data glosarium dari
file JSON
```

```
class GlossaryItem21104009
{
    // Metode untuk membaca data JSON dari file dan menampilkan
informasi glosarium
    public void ReadJSON(string filePath)
    {
        try
        {
            // Membaca isi file JSON
            string jsonContent = File.ReadAllText(filePath);

            // Melakukan deserialisasi JSON ke dalam objek
GlossaryRoot
            GlossaryRoot glossaryRoot =
JsonSerializer.Deserialize<GlossaryRoot>(jsonContent,
                new JsonSerializerOptions
{ PropertyNameCaseInsensitive = true });

            // Mendapatkan entri glosarium
            var glossEntry =
glossaryRoot?.Glossary?.GlossDiv?.GlossList?.GlossEntry;

            // Menampilkan data hasil deserialisasi ke console
jika entri ditemukan
            if (glossEntry != null)
            {
                Console.WriteLine("=== GlossEntry Details ===");
                Console.WriteLine($"ID: {glossEntry.ID}");
                Console.WriteLine($"SortAs:
{glossEntry.SortAs}");
                Console.WriteLine($"GlossTerm:
{glossEntry.GlossTerm}");
                Console.WriteLine($"Acronym:
{glossEntry.Acronym}");
                Console.WriteLine($"Abbrev:
{glossEntry.Abbrev}");
                Console.WriteLine($"GlossDef:
{glossEntry.GlossDef.Para}");
                Console.WriteLine($"GlossSee:
{glossEntry.GlossSee}");
```

```
            }
            else
            {
                Console.WriteLine("Error: GlossEntry tidak
ditemukan dalam JSON.");
            }
        }
        catch (Exception ex)
        {
            // Menangani kesalahan saat membaca atau memproses
file JSON
            Console.WriteLine($"Error reading JSON:
{ex.Message}");
        }
    }
}
```

Output:

```
=== GlossEntry Details ===
ID: SGML
SortAs: SGML
GlossTerm: Standard Generalized Markup Language
Acronym: SGML
Abbrev: ISO 8879:1986
GlossDef: A meta-markup language, used to create markup languages such as DocBook.
GlossSee: markup
```