

Konstruksi Perangkat Lunak

PERTEMUAN 6

Grammar-Based Input Processing



Ekspektasi Luaran

Mampu melakukan parsing terhadap file JSON dan XML.



Outline

- Dasar Parsing
 - JSON
 - Penutup
- 

Grammar-Based Input Processing

Dasar Parsing

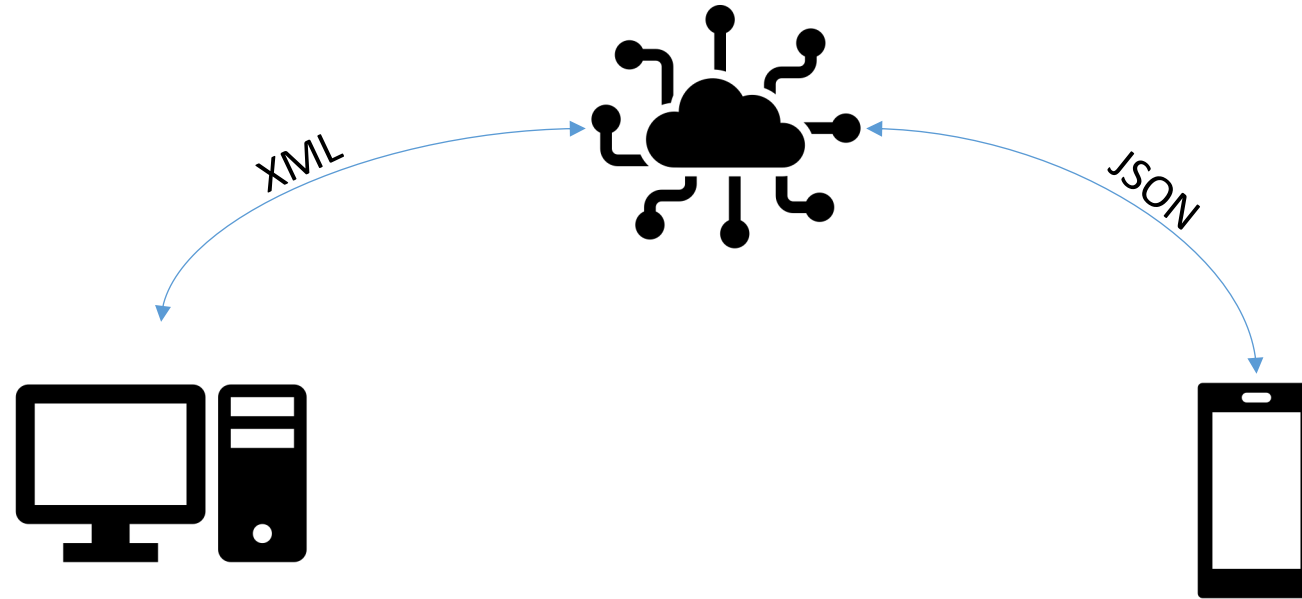


Apa itu Parsing?

“to **determine the syntactic structure of a language unit** by decomposing it into more elementary subunits and establishing the relationships among the subunits.”

Mengapa Parsing Penting

Software dan komponennya saling berkomunikasi lewat text!



Software dan komponennya saling berkomunikasi lewat text!

Parser Machine



“software tool that parses computer programs or other text, often as the first step of assembly, compilation, interpretation, or analysis.”

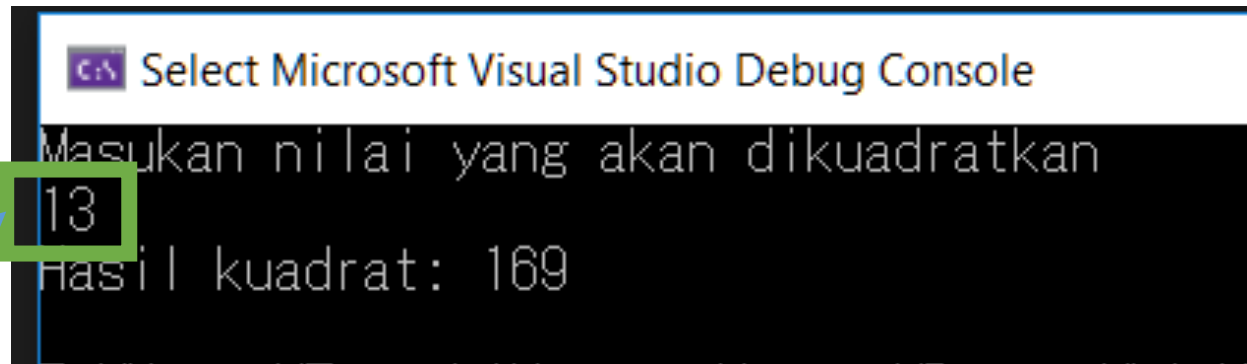
Contoh Parsing Sederhana

Parsing dari String ke integer menggunakan mesin parser `int.Parse`

```
public static void ParseInteger()
{
    Console.WriteLine("Masukan nilai yang akan dikuadratkan");
    String input = Console.ReadLine();
    int hasilParse = int.Parse(input);
    //Input tidak dapat dikuadratkan jika tidak dijadikan integer
    int hasilKuadrat = hasilParse * hasilParse;
    Console.WriteLine("Hasil kuadrat: " + hasilKuadrat);
}
```


Contoh Parsing Sederhana

Program ketika dijalankan

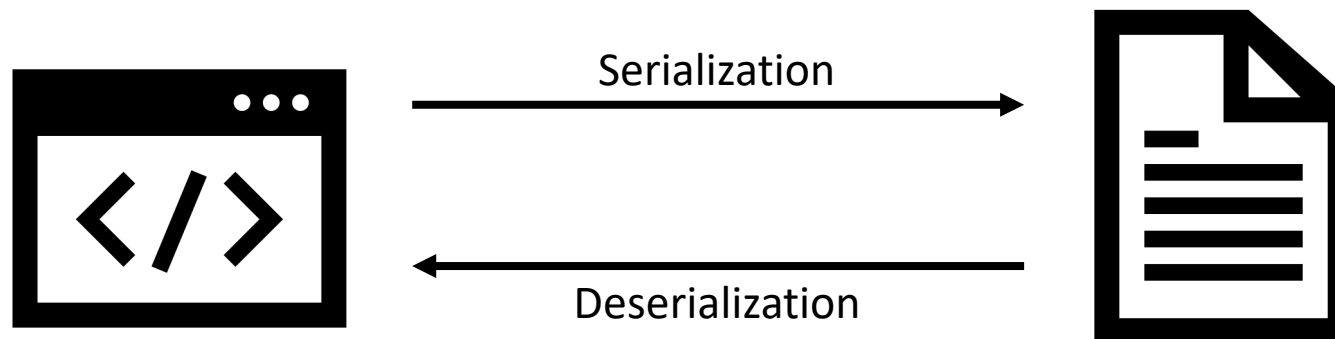


```
Select Microsoft Visual Studio Debug Console
Masukan nilai yang akan dikuadratkan
13
Hasil kuadrat: 169
```

Nilai yang dimasukan secara default bertipe string

Serialization dan Deserialization

Serialization adalah proses mengubah sebuah object/struktur data menjadi bentuk data yang dapat disimpan (Umumnya string atau byte).



Sebaliknya, deserialization adalah proses mengubah kembali data tersebut menjadi object/struktur data asalnya.

Grammar-Based Input Processing

JSON



JavaScript Object Notation (JSON)

JSON adalah format data-interchange yang didasari oleh standar bahasa pemrograman JavaScript.

```
{  
  "fruit": "Apple",  
  "size": "Large",  
  "color": "Red"  
}
```

JSON

```
class Item  
{  
    public String fruit { get; set; }  
    public String size { get; set; }  
    public String color { get; set; }  
  
    //Methods  
}
```

Class dari object yang ada
di JSON

Serializing JSON

Contoh class dari object yang akan diserialize

```
class ContohClassMahasiswa
{
    public String nama { get; set; }
    public String nim { get; set; }
    public double ipk { get; set; }

    public ContohClassMahasiswa(String nama, String nim, double ipk)
    {
        this.nama = nama;
        this.nim = nim;
        this.ipk = ipk;
    }
}
```

Inisiasi dari object yang akan diserialize

```
ContohClassMahasiswa mahasiswa = new ContohClassMahasiswa("ASDFG",  
"12345", 3.14);
```

Serializing JSON

Mengaktifkan pretty-print (whitespace dan formatting) untuk JSON

```
//[Optional] Merapikan format JSON
JsonSerializerOptions options = new JsonSerializerOptions()
{
    WriteIndented = true
};
```

Melakukan serialization menggunakan JsonSerializer

```
//Melakukan perubahan object menjadi string(Serialization)
String jsonString =
JsonSerializer.Serialize(mahasiswa, options);
```

Serializing JSON

Membuat file JSON berisi hasil serialization

```
//Membuat file JSON yg berisi string hasil serialization  
File.WriteAllText("E://Tes nulis.json", jsonString);
```

*File akan disimpan di lokasi sesuai dengan string pada parameter pertama



Serializing JSON

```
public static void WriteJSON()
{
    //Contoh object yang akan ditulis dalam JSON
    ContohClassMahasiswa mahasiswa = new ContohClassMahasiswa("ASDFG", "12345", 3.14);

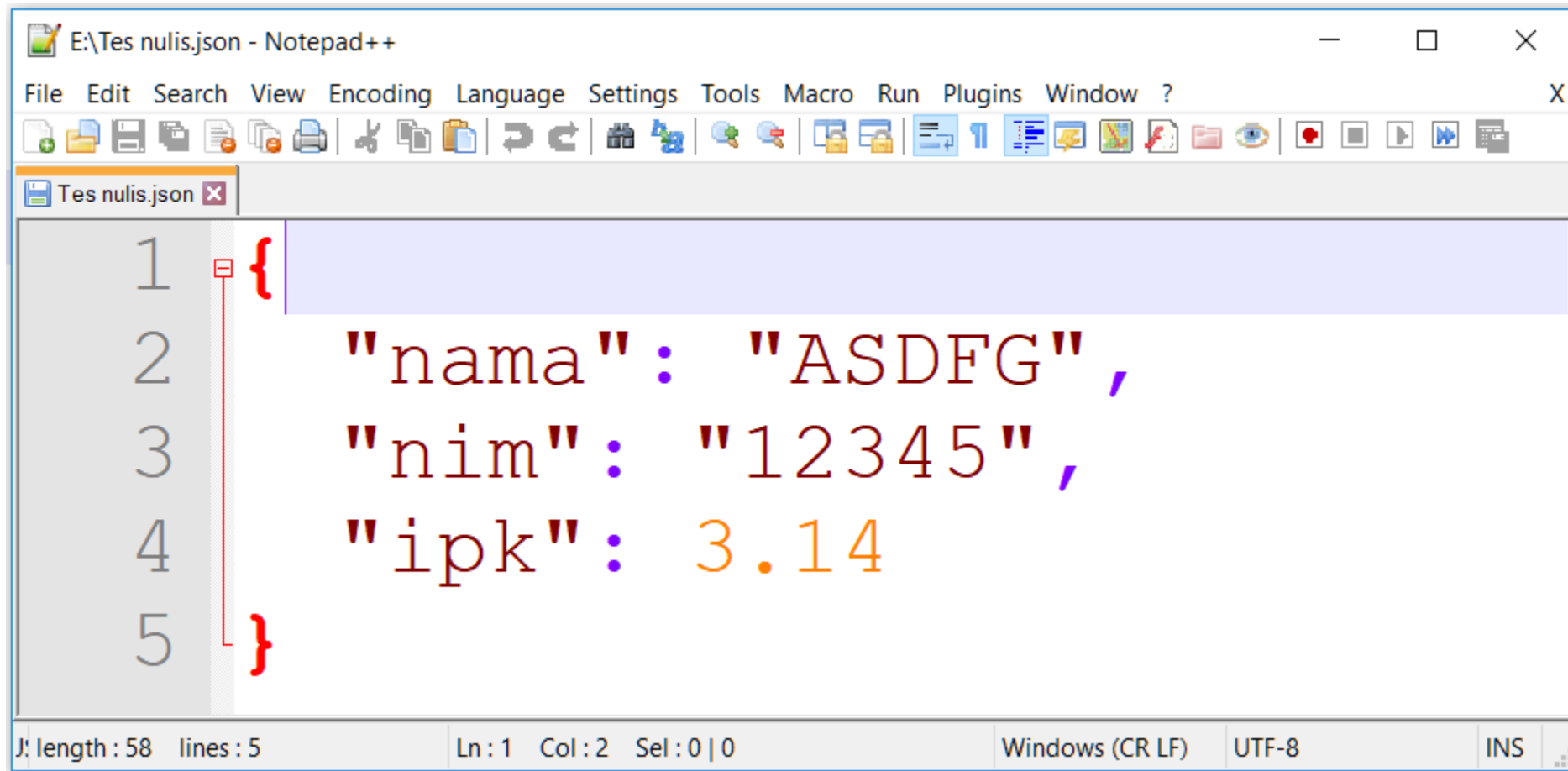
    //[Optional] Merapikan format JSON
    JsonSerializerOptions options = new JsonSerializerOptions()
    {
        WriteIndented = true
    };

    //Melakukan perubahan object menjadi string(Serialization)
    String jsonString = JsonSerializer.Serialize(mahasiswa, options);
    Console.WriteLine(jsonString);

    //Membuat file JSON yg berisi string hasil serialization
    File.WriteAllText("E://Tes nulis.json", jsonString);
}
```


Serializing JSON

JSON yg dihasilkan



```
E:\\Tes nulis.json - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
Tes nulis.json x
1 {
2     "nama": "ASDFG",
3     "nim": "12345",
4     "ipk": 3.14
5 }
J: length : 58 lines : 5
Ln : 1 Col : 2 Sel : 0 | 0
Windows (CR LF) UTF-8 INS
```


Deserializing JSON

Baca file JSONnya sesuai dengan lokasi file yg diinginkan

```
String jsonString = File.ReadAllText("E://Tes nulis.json");
```

Deserialize dengan menggunakan JsonSerializer

```
ContohClassMahasiswa mahasiswa =  
JsonSerializer.Deserialize<ContohClassMahasiswa>(jsonString);
```



Deserializing JSON

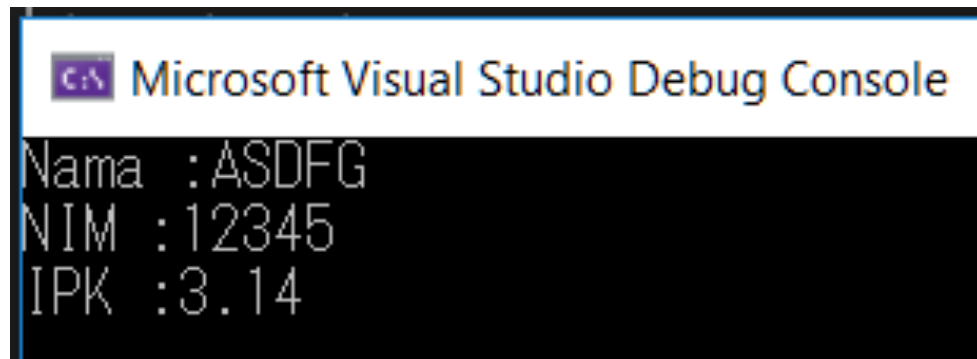
```
public static void ReadJSON()
{
    //Baca file Json yang diinginkan
    String jsonString = File.ReadAllText("E://Tes nulis.json");

    //Deserialize file json menjadi object
    ContohClassMahasiswa mahasiswa =
    JsonSerializer.Deserialize<ContohClassMahasiswa>(jsonString);

    Console.WriteLine("Nama :" + mahasiswa.nama);
    Console.WriteLine("NIM :" + mahasiswa.nim);
    Console.WriteLine("IPK :" + mahasiswa.ipk);
}
```

Deserializing JSON

Hasil program setelah dirunning.



```
Microsoft Visual Studio Debug Console  
Nama :ASDFG  
NIM :12345  
IPK :3.14
```




Grammar-Based Input Processing

Final takes



Final takes

- Gunakan JSON atau jenis file markup lainnya (XML, YAML dll.) untuk mengirimkan data antar satu komponen software dengan komponen lainnya.
 - Gunakan JSON atau jenis file markup lainnya (XML, YAML dll.) untuk membangun runtime configuration file (akan dibahas di minggu berikutnya).
 - Jangan gunakan JSON atau jenis file markup lainnya (XML, YAML dll.) untuk penyimpanan data dalam jumlah dan frekuensi akses yang besar, gunakan sistem database yang sesuai.
- 

Q&A

Any questions or comments?



Further reading

- SWEBOK V3.0
 - ISO/IEC/IEEE 24765:2017 Systems and software engineering - vocabulary
 - <https://docs.microsoft.com/en-us/dotnet/standard/serialization/system-text-json-how-to?pivots=dotnet-5-0>
 - <https://www.json.org/json-en.html>
- 