

- Source Code

- Data Mahasiswa

Kode JavaScript di atas menggunakan modul fs/promises untuk membaca file JSON secara asynchronous, kemudian mendeserialisasi kontennya dan menampilkan informasi pengguna ke konsol. Fungsi ReadJSON membaca file jurnal7_1_2311104013.json dengan encoding UTF-8, mem-parsing kontennya menjadi objek JavaScript menggunakan JSON.parse, lalu mencetak nama lengkap, jenis kelamin, usia, dan alamat (jalan, kota, dan provinsi) dari objek tersebut. Jika terjadi kesalahan saat membaca atau parsing file, program akan menampilkan pesan error ke konsol. Fungsi ini kemudian dipanggil di bagian akhir agar langsung dieksekusi saat file dijalankan.

```
07_Parsing > JURNAL_2311104013 > JS DataMahasiswa_2311104013.js > ReadJSON > then() callback
1  import { readFile } from 'fs/promises';
2
3  function ReadJSON() {
4      readFile('jurnal7_1_2311104013.json', 'utf8')
5      .then(data => {
6          const obj = JSON.parse(data);
7
8          console.log("==== HASIL DESERIALISASI =====");
9          console.log(`Nama: ${obj.firstName} ${obj.lastName}`);
10         console.log(`Jenis Kelamin: ${obj.gender}`);
11         console.log(`Usia: ${obj.age}`);
12         console.log("Alamat:");
13         console.log(`  Jalan: ${obj.address.streetAddress}`);
14         console.log(`  Kota: ${obj.address.city}`);
15         console.log(`  Provinsi: ${obj.address.state}`);
16         console.log("=====");
17     })
18     .catch(err => {
19         console.error('Gagal membaca atau parsing file:', err);
20     });
21 }
22
23 ReadJSON();
24
```

Dengan json

```
07_Parsing > JURNAL_2311104013 > {} jurnal7_1_2311104013.json > {} ad
1  {
2      "firstName": "Hafizh",
3      "lastName": "Dwi Andhika Faruq",
4      "gender": "male",
5      "age": 19,
6      "address": {
7          "streetAddress": "Tanjunganom",
8          "city": "Banjarnegara",
9          "state": "Central Java"
10     }
11 }
```

➤ Glossary Entry

Kode JavaScript tersebut menggunakan modul fs/promises untuk membaca file JSON jurnal7_3_2311104013.json secara asynchronous, kemudian melakukan parsing data JSON menjadi objek JavaScript untuk mengekstrak dan menampilkan informasi detail dari entri glossary. Fungsi readJSON mengakses properti berlapis hingga mencapai objek GlossEntry, kemudian mencetak informasi seperti ID, SortAs, GlossTerm, Acronym, Abbrev, definisi (dari GlossDef para), daftar referensi tambahan (GlossSeeAlso), dan GlossSee. Jika file tidak ditemukan atau tidak dapat diparse, maka program akan menangani error dan mencetak pesan kesalahan ke konsol. Fungsi ini langsung dipanggil di akhir skrip agar dijalankan saat file dibuka.

```
07_Parsing > JURNAL_2311104013 > JS GlossaryEntry_2311104013.js > readJSON > then() callback
1  import { readFile } from 'fs/promises';
2
3  function readJSON() {
4      readFile('jurnal7_3_2311104013.json', 'utf8')
5          .then(data => {
6              const obj = JSON.parse(data);
7              const entry = obj.glossary.GlossDiv.GlossList.GlossEntry;
8
9              console.log("==== GLOSSARY ENTRY =====");
10             console.log(` ID      : ${entry.ID}`);
11             console.log(` SortAs   : ${entry.SortAs}`);
12             console.log(` Term     : ${entry.GlossTerm}`);
13             console.log(` Acronym  : ${entry.Acronym}`);
14             console.log(` Abbrev   : ${entry.Abbrev}`);
15             console.log(` Definition: ${entry.GlossDef.para}`);
16             console.log(` See Also  : ${entry.GlossDef.GlossSeeAlso.join(", ")}`);
17             console.log(` GlossSee  : ${entry.GlossSee}`);
18             console.log("=====");
19         })
20         .catch(err => {
21             console.error("Gagal membaca atau parsing file:", err);
22         });
23     }
24
25     readJSON();
26
```

Dengan json

```
07_Parsing > JURNAL_2311104013 > { jurnal7_3_2311104013.json > {} glossary > {} GlossDiv > {} GlossList > {} GlossEntry > Abbrev
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 }
```

➤ Team Members

Kode JavaScript ini memanfaatkan modul fs/promises untuk membaca file JSON bernama jurnal7_2_2311104013.json secara asynchronous, lalu mendeserialisasi isinya dan menampilkan daftar anggota tim ke konsol. Dalam fungsi readJSON, data JSON yang dibaca diparse menjadi objek JavaScript, lalu diakses melalui properti members. Setiap elemen dalam array members akan dicetak dengan format <nim> <firstName> <lastName> (<age> <gender>) menggunakan perulangan forEach. Jika terjadi kesalahan saat membaca atau parsing file, pesan error akan ditampilkan. Fungsi ini dipanggil langsung agar eksekusinya berjalan saat file dijalankan.

```
07_Parsing > JURNAL_2311104013 > JS TeamMembers_2311104013.js > ...
1  import { readFile } from 'fs/promises';
2
3  function readJSON() {
4    readFile('jurnal7_2_2311104013.json', 'utf8')
5      .then(data => {
6        const obj = JSON.parse(data);
7        const members = obj.members;
8
9        console.log("Team member list:");
10       members.forEach(member => {
11         console.log(`${member.nim} ${member.firstName} ${member.lastName} (${member.age} ${member.gender})`);
12       });
13     })
14     .catch(err => {
15       console.error("Gagal membaca atau parsing file:", err);
16     });
17   }
18
19   readJSON();
20
```

Dengan json

```
07_Parsing > JURNAL_2311104013 > {} jurnal7_2_2311104013.json > [ ] members > {}
1  {
2    "members" : [
3      {
4        "firstName": "Andika",
5        "lastName": "Rifki Pratama",
6        "gender": "male",
7        "age": 20,
8        "nim": "2311104011"
9      },
10     {
11       "firstName": "Ardhian",
12       "lastName": "Dwi Saputra",
13       "gender": "male",
14       "age": 20,
15       "nim": "2311104013"
16     },
17     {
18       "firstName": "Farrel",
19       "lastName": "Izaz Yuwono",
20       "gender": "male",
21       "age": 20,
22       "nim": "2311104014"
23     },
24     {
25       "firstName": "Haikal",
26       "lastName": "Fadhilah Mufid",
27       "gender": "male",
28       "age": 20,
29       "nim": "2311104027"
30     }
31   ]
32 }
```

- Output

- Data Mahasiswa

```
PS E:\KULIAH\Semester 4\Konstruksi Perangkat Lunak\Praktikum\KPL_Hafizh Dwi Andhika Faruq_2311104013.js
===== HASIL DESERIALISASI =====
Nama: Hafizh Dwi Andhika Faruq
Jenis Kelamin: male
Usia: 19
Alamat:
    Jalan: Tanjunganom
    Kota: Banjarnegara
    Provinsi: Central Java
=====
PS E:\KULIAH\Semester 4\Konstruksi Perangkat Lunak\Praktikum\KPL_Hafizh Dwi Andhika Faruq_2311104013.js
```

- Glossary Entry

```
PS E:\KULIAH\Semester 4\Konstruksi Perangkat Lunak\Praktikum\KPL_Hafizh Dwi Andhika Faruq_2311104013_SE-07-01\07_Parsing\3111104013.js
===== 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
=====
PS E:\KULIAH\Semester 4\Konstruksi Perangkat Lunak\Praktikum\KPL_Hafizh Dwi Andhika Faruq_2311104013_SE-07-01\07_Parsing\3111104013.js
```

- Team Members

```
PS E:\KULIAH\Semester 4\Konstruksi Perangkat Lunak\Praktikum\KPL_Hafizh Dwi Andhika Faruq_2311104013.js
Team member list:
2311104011 Andika Rifki Pratama (20 male)
2311104013 Ardhian Dwi Saputra (20 male)
2311104014 Farrel Izaz Yuwono (20 male)
2311104027 Haikal Fadhilah Mufid (20 male)
PS E:\KULIAH\Semester 4\Konstruksi Perangkat Lunak\Praktikum\KPL_Hafizh Dwi Andhika Faruq_2311104013.js
```