

Novita Syahwa Tri Hapsari

2311104007

SE07-01

Codingan:

```
const fs = require('fs');

class DataMahasiswa2311104007 {
  static ReadJSON() {
    const path = './jurnal7_1_2311104007.json';
    try {
      const data = fs.readFileSync(path, 'utf-8');
      const json = JSON.parse(data);

      console.log("==== DATA MAHASISWA =====");
      console.log(`Nama      : ${json.firstName} ${json.lastName}`);
      console.log(`Gender    : ${json.gender}`);
      console.log(`Usia      : ${json.age}`);
      console.log();
      console.log("Alamat:");
      console.log(`  Jalan    : ${json.address.streetAddress}`);
      console.log(`  Kota     : ${json.address.city}`);
      console.log(`  Provinsi : ${json.address.state}`);
      console.log();
      console.log("Mata Kuliah:");
      json.courses.forEach((course) => {
        console.log(`  - ${course.code}: ${course.name}`);
      });

    } catch (err) {
      console.error("Gagal membaca file JSON:", err.message);
    }
  }
}

module.exports = DataMahasiswa2311104007;
```

```
const DataMahasiswa123456789 = require('./DataMahasiswa2311104007.js');

DataMahasiswa123456789.ReadJSON();
```

```
{
  "firstName": "Novita",
  "lastName": "Syahwa",
  "gender": "female",
  "age": 20,
  "address": {
    "streetAddress": "Teluk",
    "city": "Purwokerto",
    "state": "Central Java"
  },
}
```

```

    "courses": [
      { "code": "CRI2C4", "name": "Konstruksi Perangkat Lunak" },
      { "code": "CRI2XX", "name": "Nama Mata Kuliah" }
    ]
  }
}

```

Penjelasan codingan:

Kode di atas berfungsi untuk membaca dan menampilkan isi dari file JSON berisi data mahasiswa. Method `ReadJSON()` di dalam class ini menggunakan module `fs` dari Node.js untuk membaca file `jurnal7_1_2311104007.json`, kemudian melakukan parsing dari format teks JSON menjadi objek JavaScript menggunakan `JSON.parse()`. Setelah berhasil diparsing, data seperti nama lengkap, gender, usia, alamat (yang berada di dalam objek `address`), serta daftar mata kuliah ditampilkan ke console dengan format yang rapi. Di akhir file, class ini diekspor menggunakan `module.exports` agar bisa digunakan di file lain seperti `index.js`.

Hasil output :

```

PS N:\TELKOM UNIVERSITY\SEM 4\KPL_Novita Syahwa Tri Hapsari_2311104007_SE07-01\07_Parsing\TP> node .\index.js
===== DATA MAHASISWA =====
Nama      : Novita Syahwa
Gender    : female
Usia      : 20

Alamat:
  Jalan    : Teluk
  Kota     : Purwokerto
  Provinsi : Central Java

Mata Kuliah:
- CRI2C4: Konstruksi Perangkat Lunak
- CRI2XX: Nama Mata Kuliah

```

Codingan :

```

const fs = require('fs');

class TeamMembers2311104007 {
  static ReadJSON() {
    const path = './jurnal7_2_2311104007.json';

    try {
      const data = fs.readFileSync(path, 'utf-8');
      const json = JSON.parse(data);

      console.log("Team member list:");

      json.members.forEach(member => {
        const fullName = `${member.firstName} ${member.lastName}`;
        console.log(`${member.nim} ${fullName} (${member.age} ${member.gender})`);
      });

    } catch (err) {
      console.error("Gagal membaca file JSON:", err.message);
    }
  }
}

```

```
module.exports = TeamMembers2311104007;
```

```
const TeamMembers2311104007 = require('./TeamMembers2311104007');
TeamMembers2311104007.ReadJSON();

{
  "members" : [
    {
      "firstName": "Novita",
      "lastName": "Syahwa",
      "gender": "female",
      "age": 20,
      "nim": "2311104007"
    },
    {
      "firstName": "Prajna",
      "lastName": "paramitha",
      "gender": "female",
      "age": 19,
      "nim": "2311104016"
    },
    {
      "firstName": "Mei ",
      "lastName": "Sari",
      "gender": "female",
      "age": 19,
      "nim": "23111040012"
    },
    {
      "firstName": "Matryatus ",
      "lastName": "Sofia",
      "gender": "female",
      "age": 19,
      "nim": "2311104003"
    },
    {
      "firstName": "Tri ",
      "lastName": "Mylani",
      "gender": "female",
      "age": 19,
      "nim": "2311104001"
    }
  ]
}
```

Penjelasan coding:

Code tersebut menampilkan daftar anggota tim dari file JSON. Di dalam class TeamMembers2311104007, terdapat method ReadJSON() yang menggunakan modul bawaan fs untuk membaca isi file jurnal7\_2\_2311104007.json. Setelah file dibaca, isinya diubah menjadi object JavaScript menggunakan JSON.parse(). Data anggota tim berada di dalam array members, dan masing-masing anggota akan ditampilkan menggunakan forEach() dengan

format: NIM, nama lengkap, usia, dan jenis kelamin. Hasil akhirnya akan ditampilkan ke console dengan rapi, dimulai dari teks "Team member list:"

Hasil output:

```
JURNAL 2> node .\index.js
Team member list:
2311104007 Novita Syahwa (20 female)
2311104016 Prajna paramitha (19 female)
23111040012 Mei Sari (19 female)
2311104003 Matryatus Sofia (19 female)
2311104001 Tri Mylani (19 female)
```

```
const fs = require('fs');

class GlossaryItem2311104007 {
  static ReadJSON() {
    const path = './jurnal7_3_2311104007.json';

    try {
      const data = fs.readFileSync(path, 'utf-8');
      const json = JSON.parse(data);

      const entry = json.glossary.GlossDiv.GlossList.GlossEntry;

      console.log("=== GlossEntry Details ===");
      console.log(`ID          : ${entry.ID}`);
      console.log(`SortAs       : ${entry.SortAs}`);
      console.log(`GlossTerm    : ${entry.GlossTerm}`);
      console.log(`Acronym      : ${entry.Acronym}`);
      console.log(`Abbrev       : ${entry.Abbrev}`);
      console.log(`Definition   : ${entry.GlossDef.para}`);
      console.log(`See Also    : ${entry.GlossDef.GlossSeeAlso.join(', ')}`);
      console.log(`GlossSee    : ${entry.GlossSee}`);

    } catch (err) {
      console.error("Gagal membaca file JSON:", err.message);
    }
  }
}

module.exports = GlossaryItem2311104007;
```

```
const GlossaryItem2311104007 = require('./GlossaryItem2311104007');
GlossaryItem2311104007.ReadJSON();
```

```

{
  "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"
        }
      }
    }
  }
}

```

Penejelasan coding :

Kode untuk membaca file JSON dan menampilkan bagian `GlossEntry` saja. File dibaca dengan modul `fs`, lalu diubah menjadi object dengan `JSON.parse()`. Setelah itu, bagian `GlossEntry` diakses dan isinya seperti ID, GlossTerm, Acronym, dan definisi ditampilkan ke console. Kode ini menunjukkan cara sederhana mengambil dan menampilkan data dari file JSON.

Hasil running:

```

=== GlossEntry Details ===
ID       : SGML
SortAs   : SGML
GlossTerm : 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

```