

Muhammad Fathi Rafa

2311104022

SE0701

Jurnal 07 Parsing

Soal1

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

function ReadJSON() {
  try {
    const data = fs.readFileSync('jurnal7_1_2311104022.json', 'utf8');
    const mahasiswa = JSON.parse(data);

    console.log('=== Data Mahasiswa ===');
    console.log(`Nama: ${mahasiswa.firstName} ${mahasiswa.lastName}`);
    console.log(`Jenis Kelamin: ${mahasiswa.gender}`);
    console.log(`Umur: ${mahasiswa.age}`);
    console.log(`Alamat: ${mahasiswa.address.streetAddress}, ${mahasiswa.address.city}, ${mahasiswa.address.state}`);
    console.log('Mata Kuliah:');
    mahasiswa.courses.forEach((course, index) => {
      console.log(`  ${index + 1}. ${course.code} - ${course.name}`);
    });
  } catch (err) {
    console.error('Terjadi kesalahan saat membaca file JSON:', err.message);
  }
}

ReadJSON();
```

Outputnya:

```
Nama: Muhammad Fathi Rafa
Jenis Kelamin: male
Umur: 19
Alamat: Rinjani, Cilacap, Central Java
Mata Kuliah:
  1. CRI2C4 - Konstruksi Perangkat Lunak
  2. CRI2XX - Nama Mata Kuliah
```

Penjelasan:

Fungsi ReadJSON() didefinisikan untuk melakukan proses utama. Di dalam fungsi ini, pertama-tama program mencoba membaca file jurnal7_1_2311104022.json secara sinkron menggunakan fs.readFileSync(), kemudian mengubah isi file JSON menjadi objek JavaScript dengan JSON.parse(). Setelah itu, program mencetak detail mahasiswa seperti nama lengkap (gabungan firstName dan lastName), jenis kelamin, umur, serta alamat lengkap yang terdiri dari streetAddress, city, dan state. Kemudian, program menampilkan daftar mata kuliah yang diambil mahasiswa menggunakan forEach() pada array courses, dan mencetak setiap mata kuliah dengan format nomor urut, kode mata kuliah, dan nama mata kuliah. Jika terjadi kesalahan saat

membaca file atau parsing JSON, error akan ditangkap oleh blok catch dan ditampilkan ke konsol. Terakhir, fungsi ReadJSON() dipanggil agar program dapat dijalankan langsung saat file dieksekusi. Program ini cocok digunakan untuk menampilkan data mahasiswa dari file JSON dalam format yang terstruktur dan mudah dibaca.

Soal2

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

class TeamMembers_2311104022 {
  static ReadJSON() {
    try {
      const data = fs.readFileSync('jurnal7_2_2311104022.json', 'utf8');
      const obj = JSON.parse(data);
      const members = obj.members;

      console.log("Team member list:");
      members.forEach(member => {
        const fullName = `${member.firstName} ${member.lastName}`;
        console.log(`${member.nim} ${fullName} (${member.age} ${member.gender})`);
      });
    } catch (err) {
      console.error("Terjadi kesalahan saat membaca file JSON:", err.message);
    }
  }
}

// Panggil method ReadJSON
TeamMembers_2311104022.ReadJSON();
```

Output:

```
Team member list:
2311104022 Eduardo Bagus (20 male)
2311104022 Muhammad Mahrus Ali (20 male)
2311104039 Fajar Budiawan (22 male)
2311104035 Reza Irawan (20 male)
```

Penjelasan:

Class ini memiliki satu method statis bernama ReadJSON(), yang pertama-tama menggunakan fs.readFileSync() untuk membaca isi file secara sinkron. Kemudian, data JSON diparsing menjadi objek JavaScript menggunakan JSON.parse(). Objek tersebut diasumsikan memiliki properti members, yaitu sebuah array yang berisi data setiap anggota tim. Program kemudian

mencetak judul "Team member list:" ke konsol. Selanjutnya, dilakukan iterasi pada setiap elemen array members dengan `forEach()`, dan tiap anggota ditampilkan dalam format: `<nim>` `<nama lengkap>` (`<umur>` `<gender>`), di mana nama lengkap merupakan gabungan `firstName` dan `lastName`. Jika file tidak ditemukan atau terdapat kesalahan saat parsing, maka pesan kesalahan akan ditampilkan melalui blok `catch`. Method ini kemudian dipanggil di akhir file agar dapat langsung menampilkan data saat program dijalankan.

Soal3

```
class GlossaryItem_2311104022 {
  static ReadJSON() {
    try {
      const data = fs.readFileSync('jurnal7_3_2311104022.json', 'utf8');
      const obj = JSON.parse(data);

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

      console.log("=== Glossary Entry ===");
      console.log(`ID      : ${entry.ID}`);
      console.log(`Sort As   : ${entry.SortAs}`);
      console.log(`Gloss Term : ${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(`Gloss See  : ${entry.GlossSee}`);
    } catch (err) {
      console.error("Terjadi kesalahan saat membaca file JSON:", err.message);
    }
  }
}

// Panggil method untuk membaca JSON
GlossaryItem_2311104022.ReadJSON();
```

Output:

```
PS D:\KPL_MuhammadFathiRafa_2311104022_SE0701\07_Parsing\Jurnal_Parsing_2311104022> no
=== Glossary Entry ===
ID      : SGML
Sort As   : SGML
Gloss 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
Gloss See  : markup
```

Penjelasan:

File ini berisi sebuah class bernama GlossaryItem2311104022 yang memiliki method statis bernama ReadJSON(). Method ini bertugas untuk membaca file glossary.json menggunakan modul fs (file system) bawaan Node.js. Setelah membaca file, isi file tersebut diparsing menjadi objek JavaScript menggunakan JSON.parse(). Kemudian, program mengambil informasi dari objek JSON bagian GlossEntry dan mencetak semua informasi penting seperti ID, istilah, akronim, singkatan, deskripsi, daftar referensi, dan rujukan terkait ke console. Program ini menampilkan informasi yang relevan secara rapi dalam format output yang mudah dibaca.