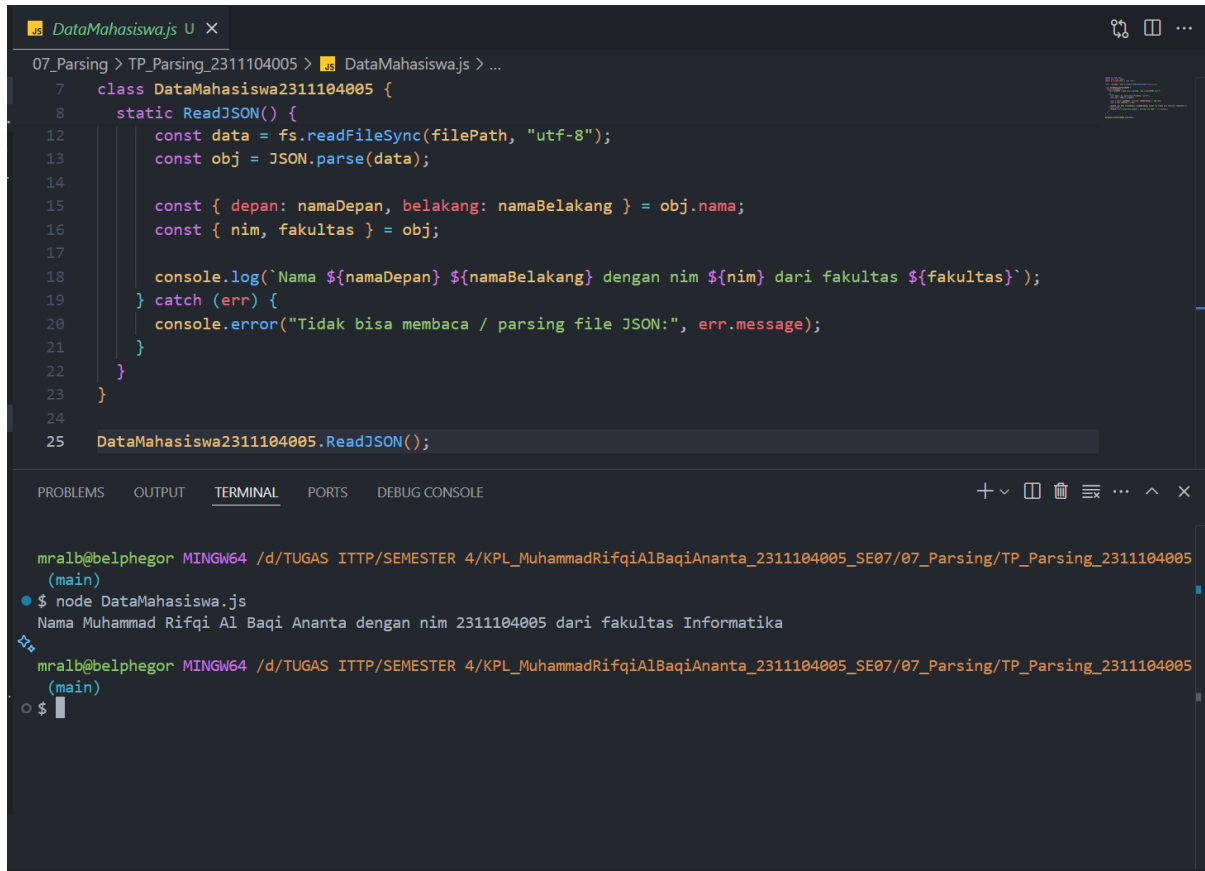


Tugas Pendahuluan

1. Soal 1



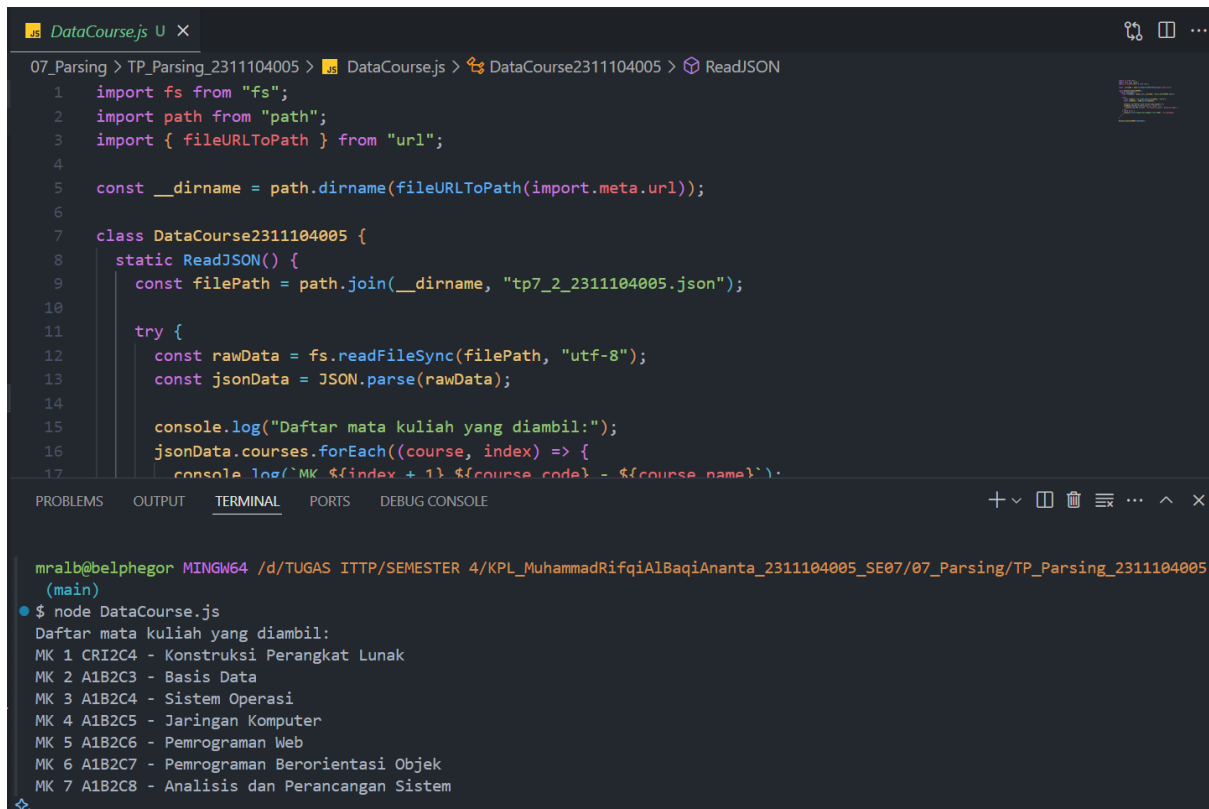
```
07_Parsing > TP_Parsing_2311104005 > DataMahasiswa.js > ...
7   class DataMahasiswa2311104005 {
8     static ReadJSON() {
12      const data = fs.readFileSync(filePath, "utf-8");
13      const obj = JSON.parse(data);
14
15      const { depan: namaDepan, belakang: namaBelakang } = obj.nama;
16      const { nim, fakultas } = obj;
17
18      console.log(`Nama ${namaDepan} ${namaBelakang} dengan nim ${nim} dari fakultas ${fakultas}`);
19    } catch (err) {
20      console.error("Tidak bisa membaca / parsing file JSON:", err.message);
21    }
22  }
23 }
24
25 DataMahasiswa2311104005.ReadJSON();
```

```
mralb@belphegor MINGW64 /d/TUGAS ITTP/SEMESTER 4/KPL_MuhammadRifqiAlBaqiAnanta_2311104005_SE07/07_Parsing/TP_Parsing_2311104005
(main)
• $ node DataMahasiswa.js
Nama Muhammad Rifqi Al Baqi Ananta dengan nim 2311104005 dari fakultas Informatika
mralb@belphegor MINGW64 /d/TUGAS ITTP/SEMESTER 4/KPL_MuhammadRifqiAlBaqiAnanta_2311104005_SE07/07_Parsing/TP_Parsing_2311104005
(main)
○ $
```

Penjelasan:

Kode di atas merupakan sebuah script Node.js yang membaca file JSON berisi data mahasiswa menggunakan modul `fs` dan `path`. Kelas `DataMahasiswa2311104005` memiliki metode statis `ReadJSON` yang akan mencari file bernama `tp7_1_2311104005.json` di direktori yang sama dengan file script ini berada. Setelah file ditemukan, isi file dibaca secara sinkron dan diubah menjadi objek JavaScript. Script ini kemudian mengambil nama depan, nama belakang, NIM, dan fakultas dari objek tersebut, lalu menampilkannya ke konsol dalam format kalimat yang rapi. Jika terjadi error saat membaca atau parsing file JSON, maka pesan error akan ditampilkan di konsol.

2. Soal 2



```
07_Parsing > TP_Parsing_2311104005 > DataCourse.js > ReadJSON
1  import fs from "fs";
2  import path from "path";
3  import { fileURLToPath } from "url";
4
5  const __dirname = path.dirname(fileURLToPath(import.meta.url));
6
7  class DataCourse2311104005 {
8      static ReadJSON() {
9          const filePath = path.join(__dirname, "tp7_2_2311104005.json");
10
11          try {
12              const rawData = fs.readFileSync(filePath, "utf-8");
13              const jsonData = JSON.parse(rawData);
14
15              console.log("Daftar mata kuliah yang diambil:");
16              jsonData.courses.forEach((course, index) => {
17                  console.log(`MK ${index + 1} ${course.code} - ${course.name}`);
18              });
19          } catch (error) {
20              console.error("Error reading JSON file:", error);
21          }
22      }
23  }
24
25  DataCourse2311104005.ReadJSON();
```

PROBLEMS OUTPUT **TERMINAL** PORTS DEBUG CONSOLE

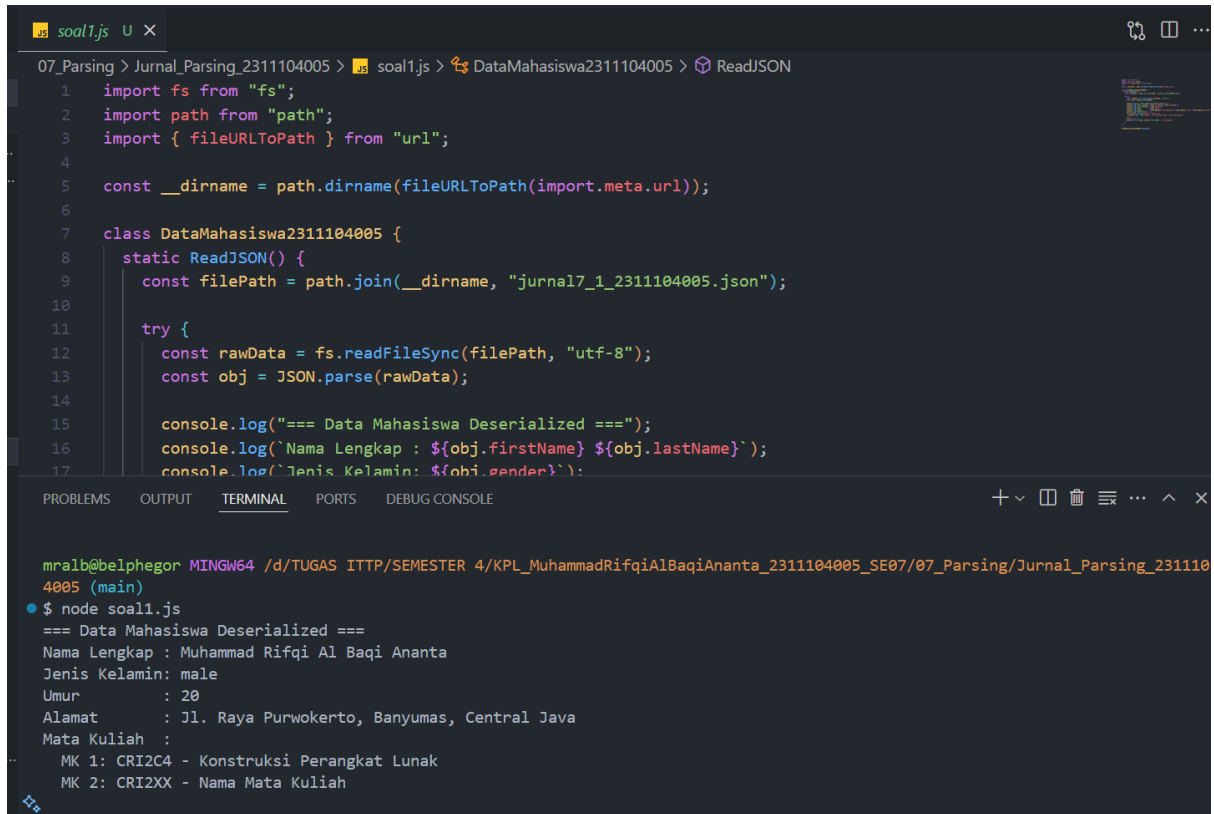
```
mralb@belphegor MINGW64 /d/TUGAS ITTP/SEMESTER 4/KPL_MuhammadRifqiAlBaqiAnanta_2311104005_SE07/07_Parsing/TP_Parsing_2311104005
(main)
$ node DataCourse.js
Daftar mata kuliah yang diambil:
MK 1 CRI2C4 - Konstruksi Perangkat Lunak
MK 2 A1B2C3 - Basis Data
MK 3 A1B2C4 - Sistem Operasi
MK 4 A1B2C5 - Jaringan Komputer
MK 5 A1B2C6 - Pemrograman Web
MK 6 A1B2C7 - Pemrograman Berorientasi Objek
MK 7 A1B2C8 - Analisis dan Perancangan Sistem
```

Penjelasan:

Kode di atas adalah script Node.js yang digunakan untuk membaca dan menampilkan daftar mata kuliah dari sebuah file JSON bernama `tp7_2_2311104005.json` yang berada di folder yang sama dengan script tersebut. Dengan memanfaatkan modul `fs` dan `path`, script ini membuka dan membaca file JSON, lalu mengubah isinya menjadi objek JavaScript. Selanjutnya, script ini menampilkan daftar mata kuliah ke konsol satu per satu, lengkap dengan kode dan nama mata kuliah. Jika terjadi kesalahan dalam proses pembacaan file, pesan error akan dicetak ke konsol.

Jurnal

1. Soal 1



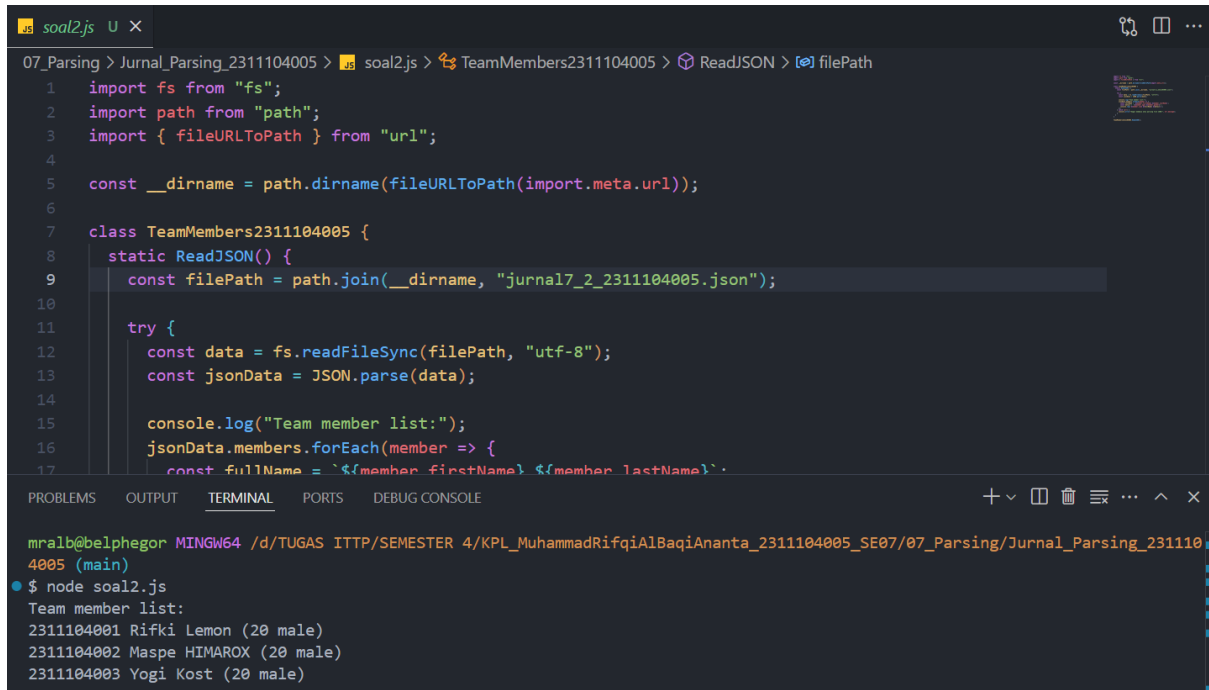
```
07_Parsing > Jurnal_Parsing_2311104005 > soal1.js > DataMahasiswa2311104005 > ReadJSON
1  import fs from "fs";
2  import path from "path";
3  import { fileURLToPath } from "url";
4
5  const __dirname = path.dirname(fileURLToPath(import.meta.url));
6
7  class DataMahasiswa2311104005 {
8      static ReadJSON() {
9          const filePath = path.join(__dirname, "jurnal7_1_2311104005.json");
10
11          try {
12              const rawData = fs.readFileSync(filePath, "utf-8");
13              const obj = JSON.parse(rawData);
14
15              console.log("=== Data Mahasiswa Deserialized ===");
16              console.log(`Nama Lengkap : ${obj.firstName} ${obj.lastName}`);
17              console.log(`Jenis Kelamin: ${obj.gender}`);
18          } catch (error) {
19              console.log("Error: ", error);
20          }
21      }
22  }
23
24  DataMahasiswa2311104005.ReadJSON();
```

```
mralb@belphegor MINGW64 /d/TUGAS ITTP/SEMESTER 4/KPL_MuhammadRifqiAlBaqiAnanta_2311104005_SE07/07_Parsing/Jurnal_Parsing_2311104005 (main)
$ node soal1.js
=== Data Mahasiswa Deserialized ===
Nama Lengkap : Muhammad Rifqi Al Baqi Ananta
Jenis Kelamin: male
Umur      : 20
Alamat    : Jl. Raya Purwokerto, Banyumas, Central Java
Mata Kuliah :
MK 1: CRI2C4 - Konstruksi Perangkat Lunak
MK 2: CRI2XX - Nama Mata Kuliah
```

Penjelasan:

Kode di atas adalah script Node.js yang digunakan untuk membaca dan menampilkan data lengkap seorang mahasiswa dari file JSON bernama jurnal7_1_2311104005.json yang berada di direktori yang sama dengan script. Dengan menggunakan modul fs dan path, script ini membaca file secara sinkron, lalu mengubah isinya menjadi objek JavaScript. Data yang ditampilkan ke konsol meliputi nama lengkap, jenis kelamin, umur, alamat lengkap, dan daftar mata kuliah yang diambil beserta kode dan nama mata kuliah. Jika proses pembacaan atau parsing file gagal, maka pesan error akan muncul di konsol.

2. Soal 2



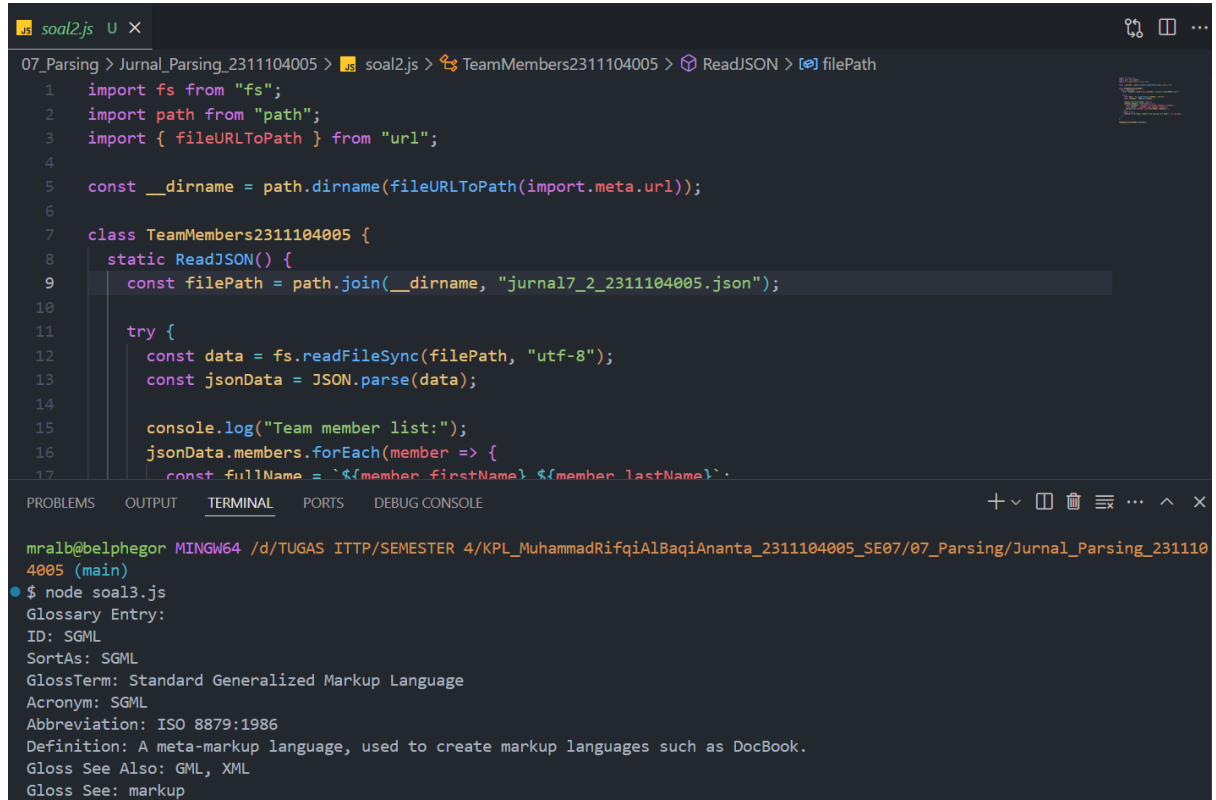
```
07_Parsing > Jurnal_Parsing_2311104005 > soal2.js > TeamMembers2311104005 > ReadJSON > filePath
1  import fs from "fs";
2  import path from "path";
3  import { fileURLToPath } from "url";
4
5  const __dirname = path.dirname(fileURLToPath(import.meta.url));
6
7  class TeamMembers2311104005 {
8      static ReadJSON() {
9          const filePath = path.join(__dirname, "jurnal7_2_2311104005.json");
10
11          try {
12              const data = fs.readFileSync(filePath, "utf-8");
13              const jsonData = JSON.parse(data);
14
15              console.log("Team member list:");
16              jsonData.members.forEach(member => {
17                  const fullName = `${member.firstName} ${member.lastName}`;
```

```
mralb@belphegor MINGW64 /d/TUGAS IITP/SEMESTER 4/KPL_MuhammadRifqialBaqiAnanta_2311104005_SE07/07_Parsing/Jurnal_Parsing_2311104005 (main)
$ node soal2.js
Team member list:
2311104001 Rifki Lemon (20 male)
2311104002 Maspe HIMAROX (20 male)
2311104003 Yogi Kost (20 male)
```

Penjelasan:

Kode di atas merupakan script Node.js yang digunakan untuk membaca dan menampilkan daftar anggota tim dari file JSON bernama `jurnal7_2_2311104005.json` yang berada di direktori yang sama dengan script. Dengan bantuan modul `fs` dan `path`, script ini membaca file JSON, mengubah isinya menjadi objek JavaScript, lalu menampilkan daftar anggota tim ke konsol satu per satu. Setiap anggota ditampilkan dengan format NIM, nama lengkap, umur, dan jenis kelamin. Jika terjadi kesalahan saat membaca atau parsing file, pesan error akan muncul di konsol.

3. Soal 3



```
07_Parsing > Jurnal_Parsing_2311104005 > soal2.js > TeamMembers2311104005 > ReadJSON > filePath
```

```
1 import fs from "fs";
2 import path from "path";
3 import { fileURLToPath } from "url";
4
5 const __dirname = path.dirname(fileURLToPath(import.meta.url));
6
7 class TeamMembers2311104005 {
8   static ReadJSON() {
9     const filePath = path.join(__dirname, "jurnal7_2_2311104005.json");
10
11     try {
12       const data = fs.readFileSync(filePath, "utf-8");
13       const jsonData = JSON.parse(data);
14
15       console.log("Team member list:");
16       jsonData.members.forEach(member => {
17         const fullName = `${member.firstName} ${member.lastName}`;
```

PROBLEMS OUTPUT TERMINAL PORTS DEBUG CONSOLE

```
mralb@belphegor MINGW64 /d/TUGAS ITTP/SEMESTER 4/KPL_MuhammadRifqiAlBaqiAnanta_2311104005_SE07/07_Parsing/Jurnal_Parsing_2311104005 (main)
$ node soal3.js
Glossary Entry:
ID: SGML
SortAs: SGML
GlossTerm: Standard Generalized Markup Language
Acronym: SGML
Abbreviation: ISO 8879:1986
Definition: A meta-markup language, used to create markup languages such as DocBook.
Gloss See Also: GML, XML
Gloss See: markup
```

Penjelasan:

Kode di atas adalah script Node.js yang digunakan untuk membaca dan menampilkan data entri glossary dari file JSON bernama `jurnal7_3_2311104005.json` yang berada di folder yang sama dengan script ini. Dengan menggunakan modul `fs` dan `path`, script ini membuka dan membaca file JSON, lalu mengonversinya ke objek JavaScript. Kemudian, script mengambil salah satu entri glossary dan menampilkan berbagai detail seperti ID, istilah, akronim, singkatan, definisi, serta daftar istilah terkait ke konsol. Jika terjadi kesalahan dalam membaca atau parsing file, maka pesan error akan muncul di konsol.