

Journal Mandiri Modul 7

Nama: Satria Ramadhan

Nim: 2311104026

Kelas: SE0701

1. Source Code

a. Javascript

```
07_Parsing > JM > DataMahasiswa_2311104026.js > ...
1  const fs = require("fs");
2
3  function ReadJson(filePath) {
4    try {
5      const data = fs.readFileSync(filePath, "utf8");
6      const jsonData = JSON.parse(data);
7
8      console.log("Hasil Deserialisasi:");
9      for (const key in jsonData) {
10       console.log(`${key}: ${JSON.stringify(jsonData[key], null, 2)}`);
11     }
12
13     return jsonData;
14   } catch (error) {
15     console.error("Error reading or parsing JSON file:", error.message);
16     return null;
17   }
18 }
19
20 const filePath = "./jurnal7_1_2311104026.json";
21 const jsonData = ReadJson(filePath);
22
23 if (jsonData) {
24   console.log("Data JSON berhasil dibaca.");
25 } else {
26   console.log("Gagal membaca file JSON.");
27 }
28
```

b. Json

```
07_Parsing > JM > jurnal7_1_2311104026.json > ...
1  {
2    "firstName": "Satria",
3    "lastName": "Ramadhan",
4    "gender": "male",
5    "age": 21,
6    "address": {
7      "streetAddress": "Karangwangkal",
8      "city": "Banyumas",
9      "state": "Purwokerto Utara"
10   },
11   "courses": [
12     { "code": "CRI2C4", "name": "Konstruksi Perangkat Lunak" },
13     { "code": "CRI2C5", "name": "Basis Data" }
14   ]
15 }
16
```

2. Output

```
> pwsh > JM > main > ?4 ~1 -28 38ms
>> node DataMahasiswa_2311104026.js
Hasil Deserialisasi:
firstName: "Satria"
lastName: "Ramadhan"
gender: "male"
age: 21
address: {
  "streetAddress": "Karangwangkal",
  "city": "Banyumas",
  "state": "Purwokerto Utara"
}
courses: [
  {
    "code": "CRI2C4",
    "name": "Konstruksi Perangkat Lunak"
  },
  {
    "code": "CRI2C5",
    "name": "Basis Data"
  }
]
Data JSON berhasil dibaca.
> pwsh > JM > main > ?4 ~1 -28 110ms
```

3. Penjelasan

Kode ini berfungsi untuk membaca dan mendeserialisasi file JSON menggunakan Node.js. Fungsi `ReadJson` menerima parameter `filePath` yang menunjukkan lokasi file JSON yang ingin dibaca. Di dalam fungsi, digunakan `fs.readFileSync` untuk membaca

isi file secara sinkron dalam format UTF-8, kemudian hasil bacaan tersebut diubah menjadi objek JavaScript dengan `JSON.parse`. Setelah berhasil, data hasil deserialisasi akan dicetak ke konsol dengan perulangan `for...in`, menampilkan setiap properti dan nilainya dalam format yang rapi menggunakan `JSON.stringify`. Jika terjadi error saat membaca atau parsing (misalnya file tidak ditemukan atau format JSON tidak valid), maka pesan error akan ditampilkan dan fungsi mengembalikan `null`. Pada bagian akhir kode, dipanggil fungsi `ReadJson` dengan path file `./jurnal7_1_2311104026.json`. Jika data berhasil dibaca, akan muncul pesan “Data JSON berhasil dibaca.”, jika gagal akan muncul “Gagal membaca file JSON.”. Kode ini berguna untuk memuat data dari file JSON secara sederhana dan menangani kesalahan dengan aman.