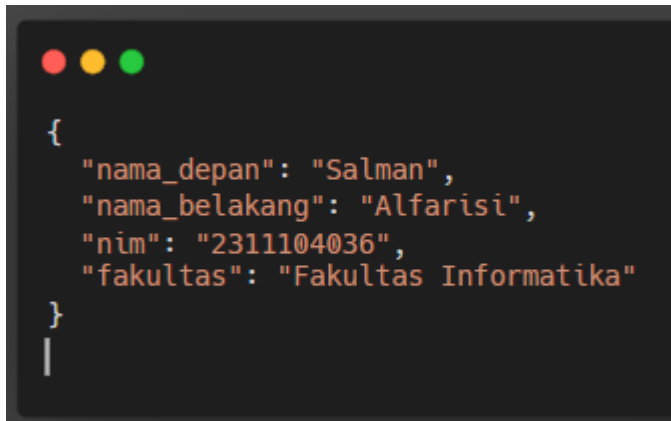


Nama : Salman Alfarisi

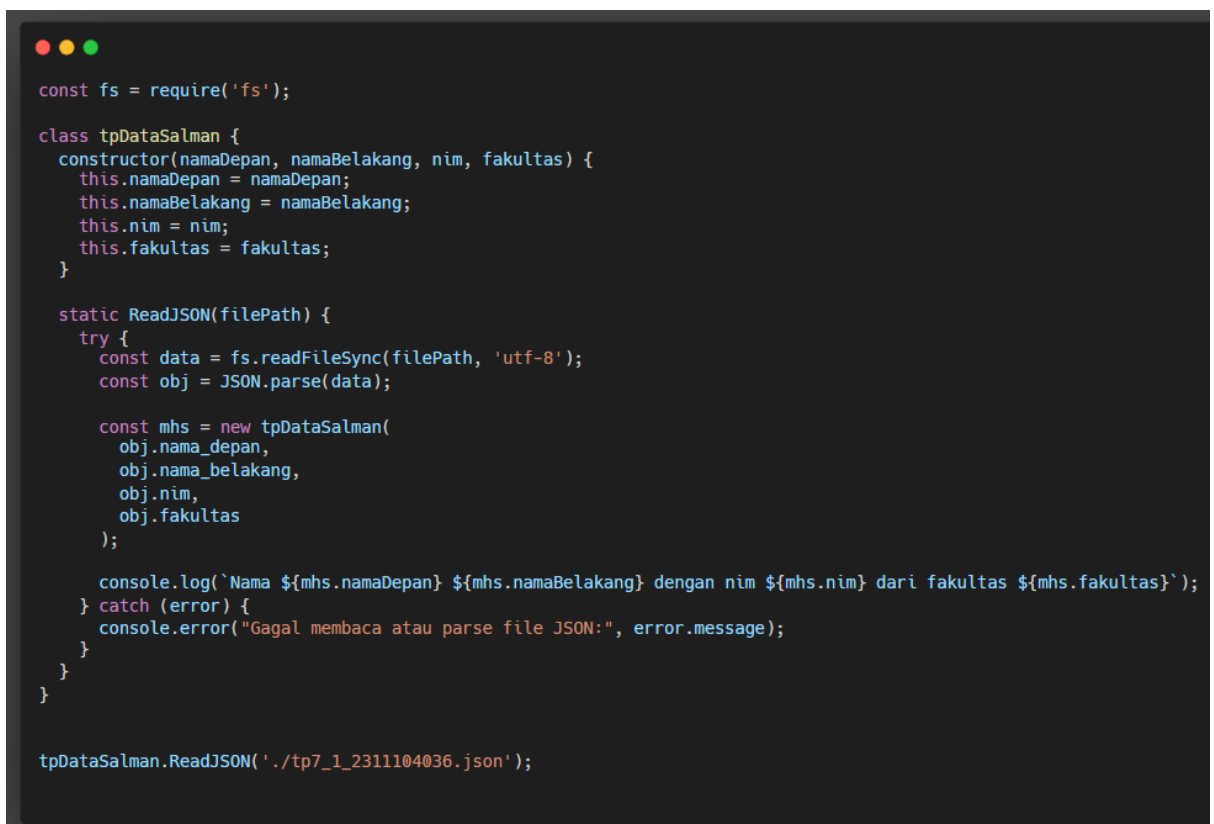
Kelas :S1SE-07-01

NIM: 2311104036

1.



```
{
  "nama_depan": "Salman",
  "nama_belakang": "Alfarisi",
  "nim": "2311104036",
  "fakultas": "Fakultas Informatika"
}
```



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

class tpDataSalman {
  constructor(namaDepan, namaBelakang, nim, fakultas) {
    this.namaDepan = namaDepan;
    this.namaBelakang = namaBelakang;
    this.nim = nim;
    this.fakultas = fakultas;
  }

  static ReadJSON(filePath) {
    try {
      const data = fs.readFileSync(filePath, 'utf-8');
      const obj = JSON.parse(data);

      const mhs = new tpDataSalman(
        obj.nama_depan,
        obj.nama_belakang,
        obj.nim,
        obj.fakultas
      );

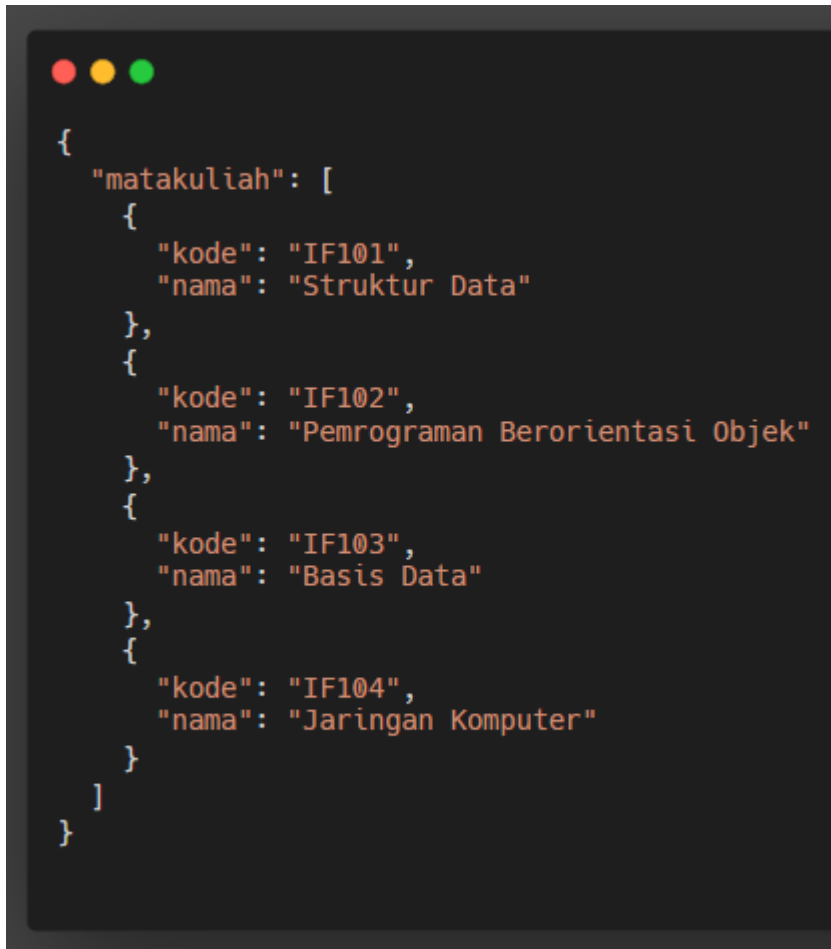
      console.log(`Nama ${mhs.namaDepan} ${mhs.namaBelakang} dengan nim ${mhs.nim} dari fakultas ${mhs.fakultas}`);
    } catch (error) {
      console.error("Gagal membaca atau parse file JSON:", error.message);
    }
  }
}

tpDataSalman.ReadJSON('./tp7_1_2311104036.json');
```

Pada bagian ini, class `tpDataSalman` berfungsi untuk membaca data pribadi mahasiswa dari file JSON yang berisi informasi nama depan, nama belakang, NIM, dan fakultas. Metode statis `ReadJSON` digunakan untuk membuka dan membaca isi file secara sinkron menggunakan modul `fs`. Setelah data berhasil dibaca, string JSON

diubah menjadi objek JavaScript menggunakan `JSON.parse()`. Data yang sudah diparsing kemudian dimasukkan ke dalam instance `tpDataSalman` melalui konstruktor. Terakhir, program menampilkan informasi mahasiswa secara terformat dalam satu kalimat lengkap yang mencakup nama lengkap, nomor induk mahasiswa, dan fakultas asal. Error handling juga diterapkan untuk mengantisipasi kegagalan dalam pembacaan file atau kesalahan format JSON, sehingga program tetap aman dari crash.

2.



```
{
  "matakuliah": [
    {
      "kode": "IF101",
      "nama": "Struktur Data"
    },
    {
      "kode": "IF102",
      "nama": "Pemrograman Berorientasi Objek"
    },
    {
      "kode": "IF103",
      "nama": "Basis Data"
    },
    {
      "kode": "IF104",
      "nama": "Jaringan Komputer"
    }
  ]
}
```

```

const fs = require('fs');

class MKSalman {
  constructor(matakuliah) {
    this.matakuliah = matakuliah;
  }

  static ReadJSON(filePath) {
    try {
      const data = fs.readFileSync(filePath, 'utf-8');
      const obj = JSON.parse(data);

      const mahasiswa = new MKSalman(obj.matakuliah);

      console.log("Daftar mata kuliah yang diambil:");
      mahasiswa.matakuliah.forEach((mk, index) => {
        console.log(`MK ${index + 1} ${mk.kode} - ${mk.nama}`);
      });
    } catch (err) {
      console.error("Gagal membaca atau parse JSON:", err.message);
    }
  }
}

MKSalman.ReadJSON('./tp7_2_2311104036.json');

```

Bagian kedua menggunakan class MKSalman yang bertugas membaca daftar mata kuliah dari sebuah file JSON yang berisi array objek matakuliah. Metode ReadJSON membaca file JSON, kemudian mem-parsing data dan menyimpannya ke dalam properti matakuliah pada objek class. Selanjutnya, program melakukan iterasi pada array tersebut menggunakan fungsi forEach untuk menampilkan setiap mata kuliah dengan nomor urut, kode mata kuliah, dan nama mata kuliah secara berurutan. Dengan cara ini, pengguna dapat melihat daftar lengkap mata kuliah yang diambil secara ringkas dan jelas. Sama seperti bagian sebelumnya, error handling dengan try...catch diterapkan untuk memastikan proses berjalan lancar meskipun terjadi kesalahan pembacaan file atau parsing.