

Nama : Gelar Zulfikri  
Kelas : TI22F  
NIM : 20220040055

## LAPORAN SINGKAT APLIKASI NODE JS

### Konsep Modularisasi

Modularisasi adalah pendekatan dalam pengembangan perangkat lunak di mana kode program dibagi menjadi modul atau berkas-berkas terpisah. Setiap modul berisi fungsi-fungsi atau komponen-komponen yang memiliki tanggung jawab spesifik. Konsep ini membantu dalam pemisahan logika, memudahkan pemeliharaan, meminimalkan konflik, dan meningkatkan reusabilitas kode.

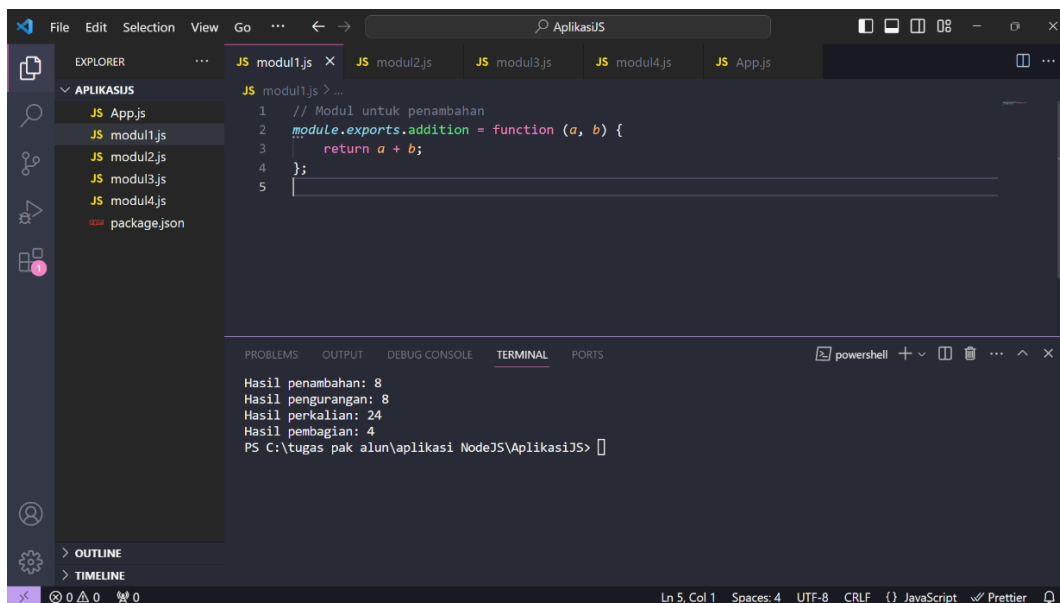
### Struktur Proyek

Struktur proyek adalah cara Anda mengorganisasi berkas-berkas dan direktori dalam proyek Anda. Dalam contoh Anda, proyek memiliki struktur sederhana yang mencakup berkas-berkas modul dan berkas utama app.js. Struktur proyek yang baik membantu Anda menjaga keteraturan dan kemudahan dalam mengelola kode

### Bagaimana Modul-modul Bekerja Bersama (Operasi Matematika Dasar)

Dalam proyek yang telah saya buat untuk operasi matematika dasar (penambahan, pengurangan, perkalian, dan pembagian). Berikut adalah contoh modul-modul tersebut bekerja bersama.

**module1.js:** Berisi fungsi *addition* untuk melakukan penambahan



The screenshot shows the Visual Studio Code interface. The Explorer sidebar on the left displays the project structure under the name 'APLIKASIJS', containing files: JS App.js, JS modul1.js, JS modul2.js, JS modul3.js, JS modul4.js, and package.json. The main editor window is open to 'JS modul1.js', showing the following code:

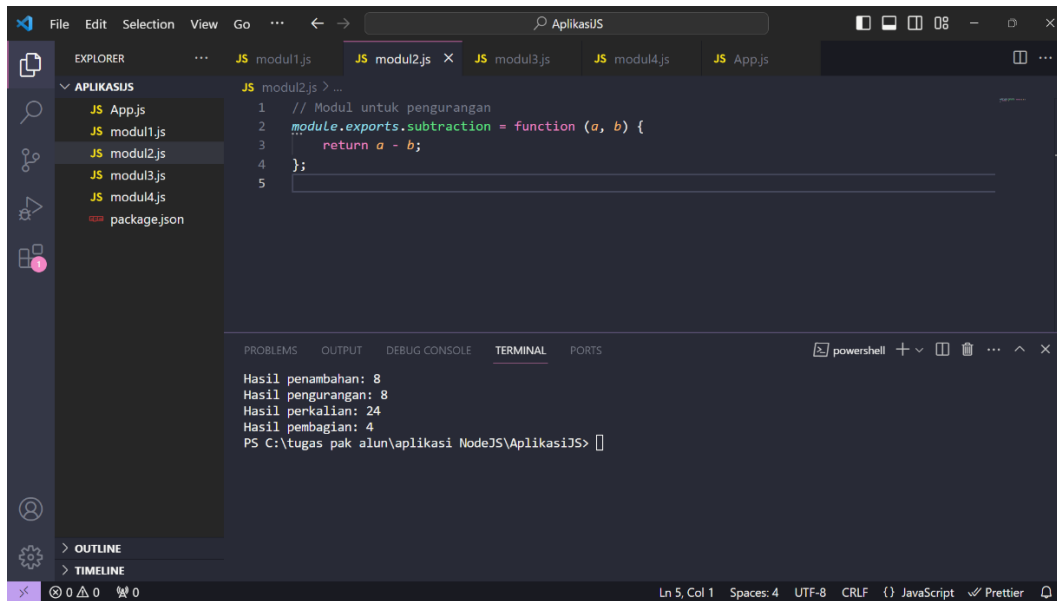
```
1 // Modul untuk penambahan
2 module.exports.addition = function (a, b) {
3     return a + b;
4 };
5
```

At the bottom, the TERMINAL panel shows the output of a Node.js script execution:

```
Hasil penambahan: 8
Hasil pengurangan: 8
Hasil perkalian: 24
Hasil pembagian: 4
PS C:\tugas pak alun\aplikasi NodeJS\AplikasiJS>
```

The status bar at the bottom indicates the current position is 'Ln 5, Col 1' with 'Spaces: 4', 'UTF-8' encoding, 'CRLF' line endings, and 'JavaScript' language mode with 'Prettier' formatting.

**module2.js:** Berisi fungsi *subtraction* untuk melakukan pengurangan.



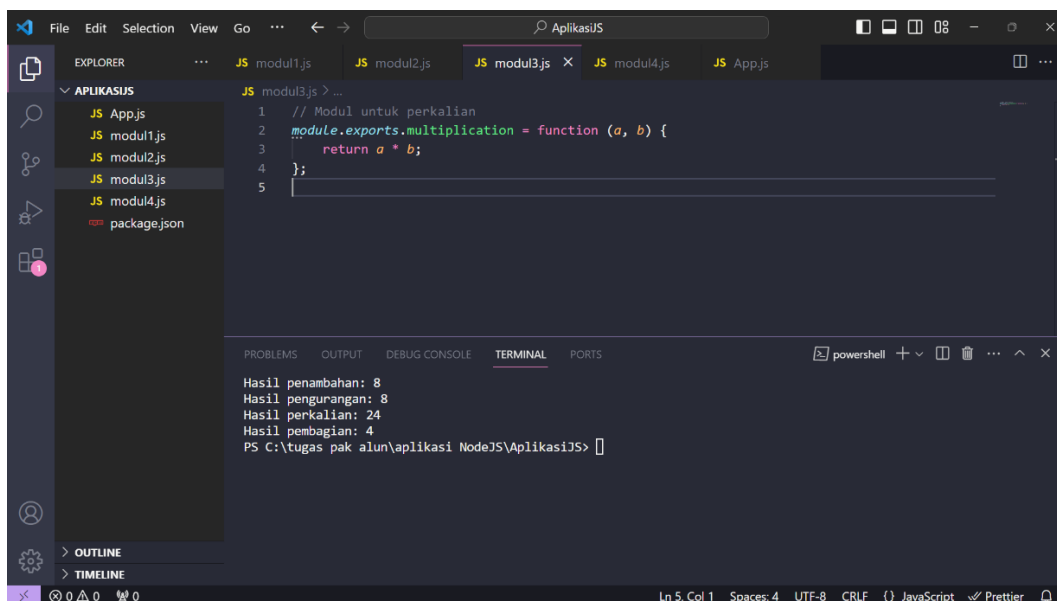
The screenshot shows the Visual Studio Code editor with a project named 'AplikasiJS'. The Explorer sidebar on the left lists the files: App.js, modul1.js, modul2.js (selected), modul3.js, modul4.js, and package.json. The main editor window displays the content of 'modul2.js', which contains a JavaScript function for subtraction:

```
1 // Modul untuk pengurangan
2 module.exports.subtraction = function (a, b) {
3   return a - b;
4 };
5
```

Below the editor, the TERMINAL panel shows the output of a Node.js script execution:

```
Hasil penambahan: 8
Hasil pengurangan: 8
Hasil perkalian: 24
Hasil pembagian: 4
PS C:\tugas pak alun\aplikasi NodeJS\AplikasiJS>
```

**module3.js:** Berisi fungsi *multiplication* untuk melakukan perkalian.



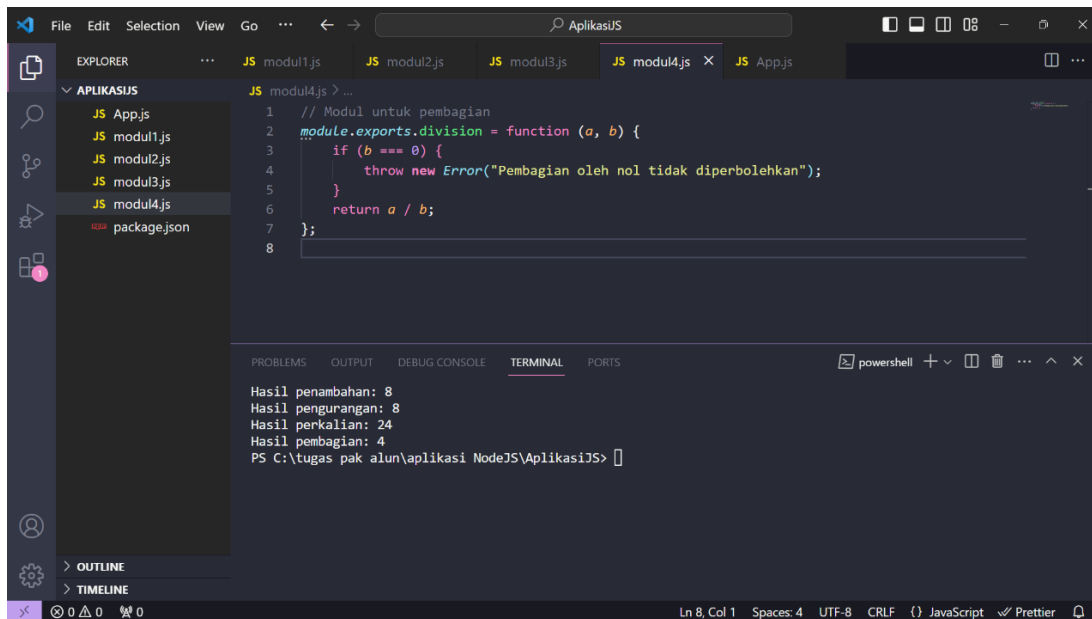
The screenshot shows the Visual Studio Code editor with the same project 'AplikasiJS'. The Explorer sidebar lists the files, with 'modul3.js' now selected. The main editor window displays the content of 'modul3.js', which contains a JavaScript function for multiplication:

```
1 // Modul untuk perkalian
2 module.exports.multiplication = function (a, b) {
3   return a * b;
4 };
5
```

The TERMINAL panel below shows the updated output after running the script again:

```
Hasil penambahan: 8
Hasil pengurangan: 8
Hasil perkalian: 24
Hasil pembagian: 4
PS C:\tugas pak alun\aplikasi NodeJS\AplikasiJS>
```

**module4.js:** Berisi fungsi *division* untuk melakukan pembagian.



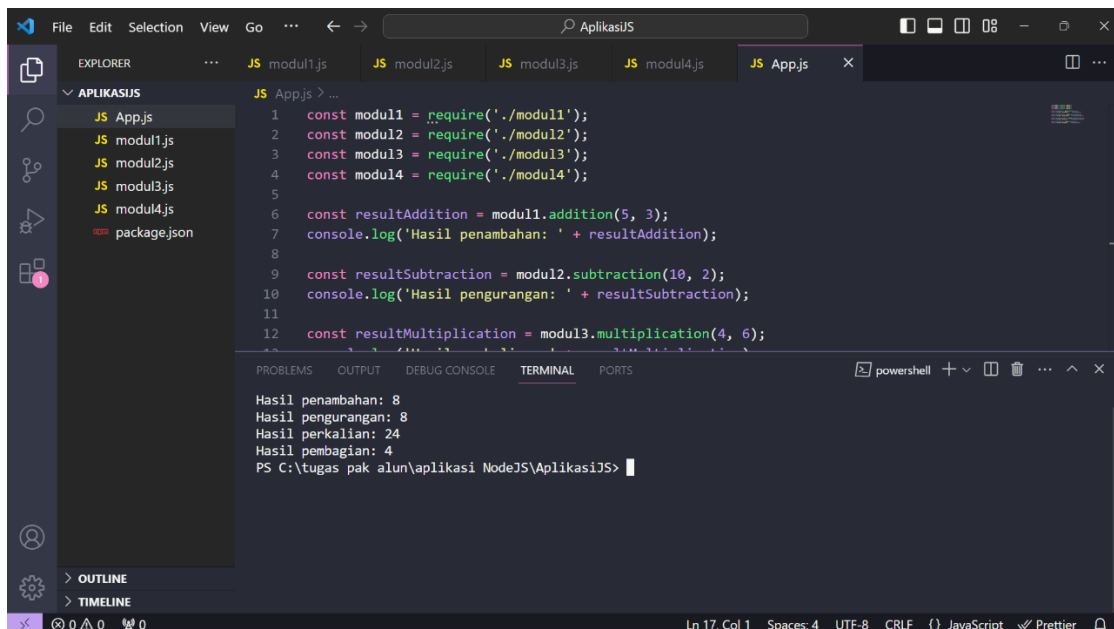
The screenshot shows the Visual Studio Code editor with the file explorer on the left displaying a project named 'APLIKASIJS' containing files: App.js, modul1.js, modul2.js, modul3.js, modul4.js, and package.json. The main editor window shows the content of 'modul4.js' with the following code:

```
1 // Modul untuk pembagian
2 module.exports.division = function (a, b) {
3     if (b === 0) {
4         throw new Error("Pembagian oleh nol tidak diperbolehkan");
5     }
6     return a / b;
7 };
8
```

Below the editor, the TERMINAL panel shows the output of a Node.js command:

```
Hasil penambahan: 8
Hasil pengurangan: 8
Hasil perkalian: 24
Hasil pembagian: 4
PS C:\tugas pak alun\aplikasi NodeJS\AplikasiJS>
```

**app.js:** Berkas utama yang mengimpor dan menggunakan modul-modul di atas. Dalam berkas ini, Anda mengimpor modul-modul yang diperlukan dan menggunakan fungsi-fungsi tersebut untuk melakukan operasi matematika dasar.



The screenshot shows the Visual Studio Code editor with the file explorer on the left displaying the same project. The main editor window shows the content of 'App.js' with the following code:

```
1 const modul1 = require('./modul1');
2 const modul2 = require('./modul2');
3 const modul3 = require('./modul3');
4 const modul4 = require('./modul4');
5
6 const resultAddition = modul1.addition(5, 3);
7 console.log('Hasil penambahan: ' + resultAddition);
8
9 const resultSubtraction = modul2.subtraction(10, 2);
10 console.log('Hasil pengurangan: ' + resultSubtraction);
11
12 const resultMultiplication = modul3.multiplication(4, 6);
13 console.log('Hasil perkalian: ' + resultMultiplication);
14
```

Below the editor, the TERMINAL panel shows the output of a Node.js command:

```
Hasil penambahan: 8
Hasil pengurangan: 8
Hasil perkalian: 24
Hasil pembagian: 4
PS C:\tugas pak alun\aplikasi NodeJS\AplikasiJS>
```