Lampiran Jawaban **(MODUL DAN PAKET)**

Nama : Julhan Abdul Malik
NIM : 20220040126
Kelas : TI22 A
Program Studi : Teknik Informatika

**- Lampiran -**

- **Rincian Tugas:**

**TUGAS PERTEMUAN 5**

Buat aplikasi Node.js dengan fungsi gratis yang mengimplementasikan modularisasi. Modularisasi adalah praktik penting dalam pengembangan perangkat lunak yang membagi kode menjadi modul atau file terpisah, sehingga lebih mudah untuk dikelola dan dipelihara.

Petunjuk Tugas:

1. Buat proyek Node.js baru dengan struktur dasar menggunakan perintah npm init untuk membuat package.json.
2. Dalam proyek, buat beberapa modul terpisah yang mewakili berbagai bagian fungsionalitas aplikasi. Jumlah modul adalah 4 file.
3. Setiap modul harus berisi setidaknya satu fungsi atau metode yang dapat diimpor dan digunakan dalam file utama aplikasi.
4. Buat file utama (misalnya app.js) yang mengimpor dan menggunakan modul yang dibuat.
5. Di file utama, buat logika untuk menjalankan beberapa tugas sederhana yang melibatkan penggunaan modul yang telah dibuat.
6. Pastikan untuk menggunakan module.exports dan perlu mengimpor modul Anda.
7. Jalankan aplikasi Node.js dan pastikan semua modul berfungsi dengan benar. Siswa harus dapat menjalankan file utama (indeks/utama) tanpa kesalahan.
8. Tulis laporan singkat yang menjelaskan konsep modularisasi, struktur proyek, dan bagaimana modul bekerja sama. Lampirkan screenshot aplikasi yang sedang berjalan sebagai bukti.

Penyerahan Tugas:

1. Kumpulkan file proyek Node.js, termasuk semua modul dan file utama.
2. Lampirkan laporan singkat dalam bentuk teks atau file PDF yang menjelaskan langkah-langkah yang Anda ambil dalam proyek dan cara menerapkan modularisasi.

- **Jawaban:**

1. Langkah 1

   Membuat project Node.js yang akan langsung menghasilkan file ‘package.json’ dengan menggunakan perintah npm.init

package.json > ...

```
{
  "name": "nodejs",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  Debug
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [
    "Julhan",
    "Abdul"
  ],
  "author": "Julhan",
  "license": "ISC"
}
```

2. Langkah 2

   Membuat buat beberapa modul terpisah kita akan membuat empat modul:

   calculator.js, greet.js, validator.js, dan logger.js

   - Modul calculator.js

     calculator.js
     ```
     module.exports = {
       add: (a, b) => a + b,
       subtract: (a, b) => a - b,
     };
     ```

   - Modul greet.js

     greet.js
     ```
     module.exports = {
       greetUser: (name) => `Hello, ${name}!`,
     };
     ```

   - Modul validator.js

     validator.js
     ```
     module.exports = {
       isEmailValid: (email) => {
         // Validating email (simplified for example)
         return /^[^\s@]+@[^\s@]+\.[^\s@]+$/.test(email);
       },
     };
     ```

   - Modul logger.js

     logger.js
     ```
     module.exports = {
       logMessage: (message) => console.log(message),
     };
     ```

3. Langkah 3:

   Membuat file utama dengan nama ‘app.js’ sekaligus mengimport beberapa module yang telah kita buat

app.js

```
module.exports

const calculator = require('./calculator');
const greet = require('./greet');
const validator = require('./validator');
const logger = require('./logger');

const result = calculator.add(5, 3);
const greeting = greet.greetUser('julhanamalik');
const isEmailValid = validator.isEmailValid('julhanamalik@gmail.com');

logger.logMessage(`Addition result: ${result}`);
logger.logMessage(greeting);
logger.logMessage(`Is email valid? ${isEmailValid ? 'Yes' : 'No'}`);
```

4. Bagaimana Modul ini bekerja

   Dalam proyek ini, modul-modul terpisah yang mencakup berbagai fungsionalitas, diimpor untuk digunakan pada aplikasi utama yaitu ‘app.js’ dengan menggunakan perintah require.

5. Lampiran

   Link Github Code Program nya:

   ➔ https://github.com/JulhanAbdulMalik-TI22A/Pemrograman-Berbasis-Platform---Julhan-Abdul-Malik/tree/8c4a23331a40f38c11f3a437286dc8f7b1a584ca/Tugas_Sesi%20ke%205_Julhan%20Abdul%20Malik

6. Kesimpulan

   Konsep modularisasi memungkinkan pemisahan kode menjadi modul-modul terpisah, yang membantu meningkatkan pemahaman, pemeliharaan, dan pengelolaan kode. Dalam proyek ini, saya menciptakan modul-modul terpisah kemudian mengimpornya ke aplikasi utama. Hal tersebut memungkinkan saya untuk mengorganisasi dan menjalankan berbagai fungsi dengan efisien. Dengan modularisasi, saya dapat menciptakan kode yang lebih bersih dan lebih mudah dipelihara.