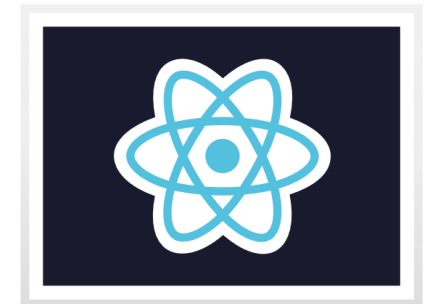
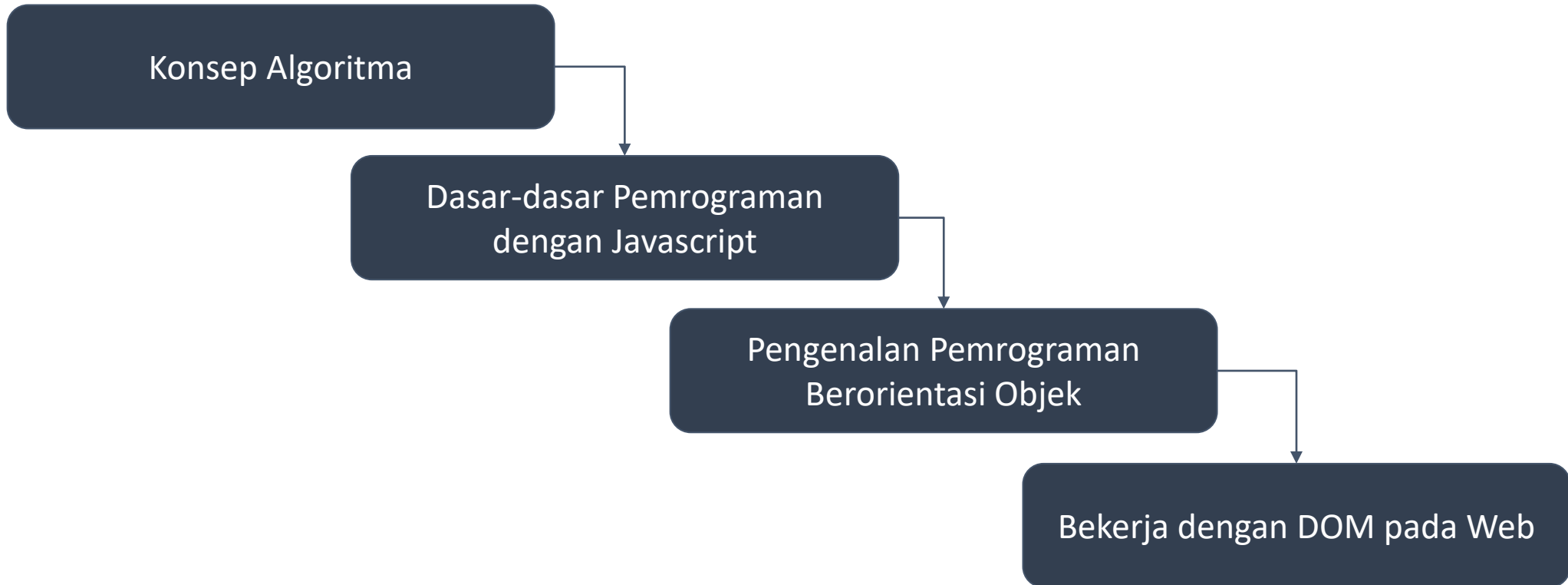


# PEMROGRAMAN JAVASCRIPT



- **KODE** : MK03
- **MATAKULIAH** : PEMROGRAMAN JAVASCRIPT
- **SKS** : 4 SKS
- **SEMESTER** : 1 (SATU)



“Gantungkan cita-citamu setinggi langit! Bermimpilah setinggi langit. Jika engkau jatuh, engkau akan jatuh di antara bintang-bintang.”

**- Ir. Soekarno -**



# MODULAR PROGRAM

## Module

Modular programming adalah teknik pada software design yang memisahkan fungsi-fungsi pada program ke dalam modul-modul (modules) yang independen dan interchangeable.

## Module

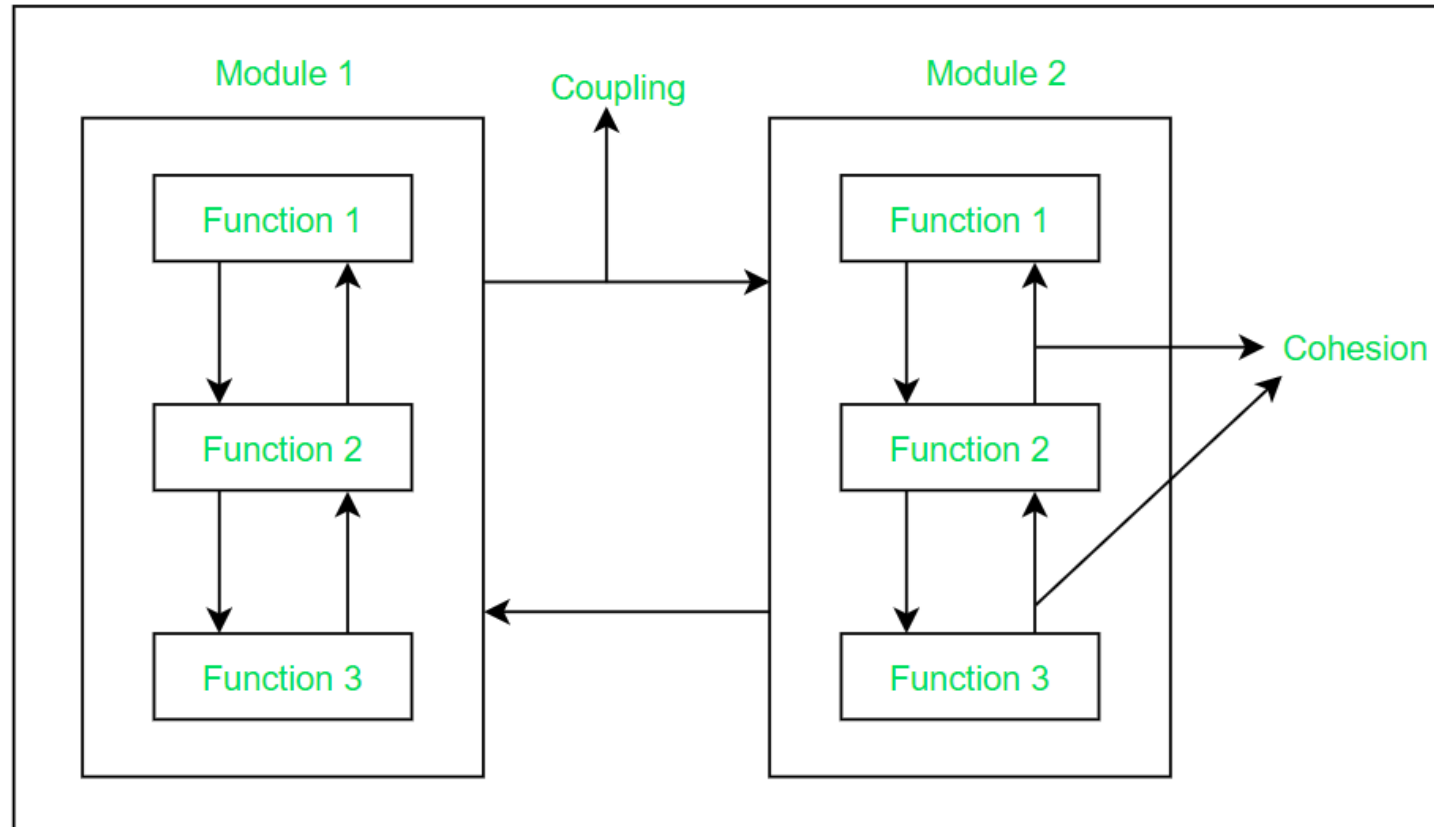
Awalnya javascript berdiri tanpa syntax untuk module. Hal tersebut tidak masalah karena biasanya script memang kecil dan sederhana. Tetapi seiring dengan perkembangan kompleksitas script yang digunakan, maka dibutuhkan cara untuk mengakomodir hal tersebut. **Module menjadi standard di javascript di tahun 2015.**

## Module

Module adalah file Javascript yang di dalamnya terdapat value dari objects, functions, dan variables. Kemudian file tersebut dapat diexport dan diimport oleh file lain. Yang mana file lain yang mengimportnya dapat menggunakan values yang ada di file tersebut.



# PENDAHULUAN



## Beberapa benefit menggunakan modules adalah :

- Maintainability. Artinya source kita dapat dimaintain secara mudah. Jika ada perubahan maka kiat tidak perlu melakukan perubahan di semua file namun hanya merubah pada file module tersebut.

## Beberapa benefit menggunakan modules adalah :

- Namespacing. Artinya adalah di dalam module kita dapat menampung banyak object, variable dan function yang sejenis atau yang saling berkaitan. Mereka di bungkus dalam satu file yang sama.

## Beberapa benefit menggunakan modules adalah :

- Reusability. Artinya adalah file modules kita dapat digunakan atau diimport di dalam file yang lain.

## **Perbedaan export default dengan tidak menggunakan default**

Dalam menuliskan keyword export kadang kala kita menemukan yang diimbui dengan keyword default. Ketika kita menggunakan imbuhan default maka ketika import kita tidak perlu menggunakan symbol bracket {}. Namun sebaliknya jika kita tidak menggunakan imbuhan default maka kita harus menggunakan symbol bracket {}.

## Export dan Import

Module dapat mengakses satu sama lain dengan memanfaatkan directive export dan import (untuk memanggil fungsi dari modul lain).

**export** : variabel dan fungsi yang ada pada module dapat diakses oleh module lain

**import** : untuk mengakses fungsi dan variabel dari module yang dituju

# EXPORT IMPORT

```
1 // 📁 sayHi.js
2 export function sayHi(user) {
3   alert(`Hello, ${user}!`);
4 }
```

Memanggil fungsi dari  
sayHi.js

```
1 // 📁 main.js
2 import {sayHi} from './sayHi.js';
3
4 alert(sayHi); // function...
5 sayHi('John'); // Hello, John!
```

## Export JavaScript

Directive ini diberikan di modul yang fungsi/variabelnya ingin digunakan oleh modul lain. Export dapat diberikan dengan cara :

- export yang dituliskan sebelum deklarasi

```
1 // export an array
2 export let months = ['Jan', 'Feb', 'Mar', 'Apr', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec'];
```

- export yang terpisah dari deklarasi

```
1 // say.js
2 function sayHi(user) {
3   alert(`Hello, ${user}!`);
4 }
5
6 function sayBye(user) {
7   alert(`Bye, ${user}!`);
8 }
9
10 export {sayHi, sayBye}; // a list of exported variables
```



## Import JavaScript

Fungsi atau variabel yang diimport dituliskan dalam kurung kurawal. Jika ingin semua fungsi dari modulnya diimport, maka digunakan tanda bintang ( \* ) tanpa kurung kurawal.

```
1 // main.js
2 import {sayHi, sayBye} from './say.js';
3
4 sayHi('John'); // Hello, John!
5 sayBye('John'); // Bye, John!
```

```
1 // main.js
2 import * as say from './say.js';
3
4 say.sayHi('John');
5 say.sayBye('John');
```

Note : import fungsi yang dibutuhkan saja agar waktu pembacaannya optimal

## Module dalam JavaScript

Module dapat dimasukkan ke dalam script dan diintegrasikan dengan HTML dengan menambahkan type="module" di tag scriptnya.

**Note** : module bekerja lewat http, tidak secara lokal, jadi simpan modulnya dalam webserver (baik lokal maupun di domain tertentu) dan panggil melalui alamat urlnya.

```
1 <!doctype html>
2 <script type="module">
3   import {sayHi} from './say.js';
4
5   document.body.innerHTML = sayHi('John');
6 </script>
```

## Module dalam JavaScript

Kode pada modul hanya dievaluasi sekali, yaitu pada saat diimport pertama kali.

Jika diimport lagi, maka import yang kedua tidak dieksekusi, melainkan hasil eksportnya yang diberikan. **Note** : Jika ada dua script eksternal dengan source yang sama, akan dieksekusi hanya sekali.

```
1 <!-- the script my.js is fetched and executed only once -->
2 <script type="module" src="my.js"></script>
3 <script type="module" src="my.js"></script>
```

## Module dalam JavaScript

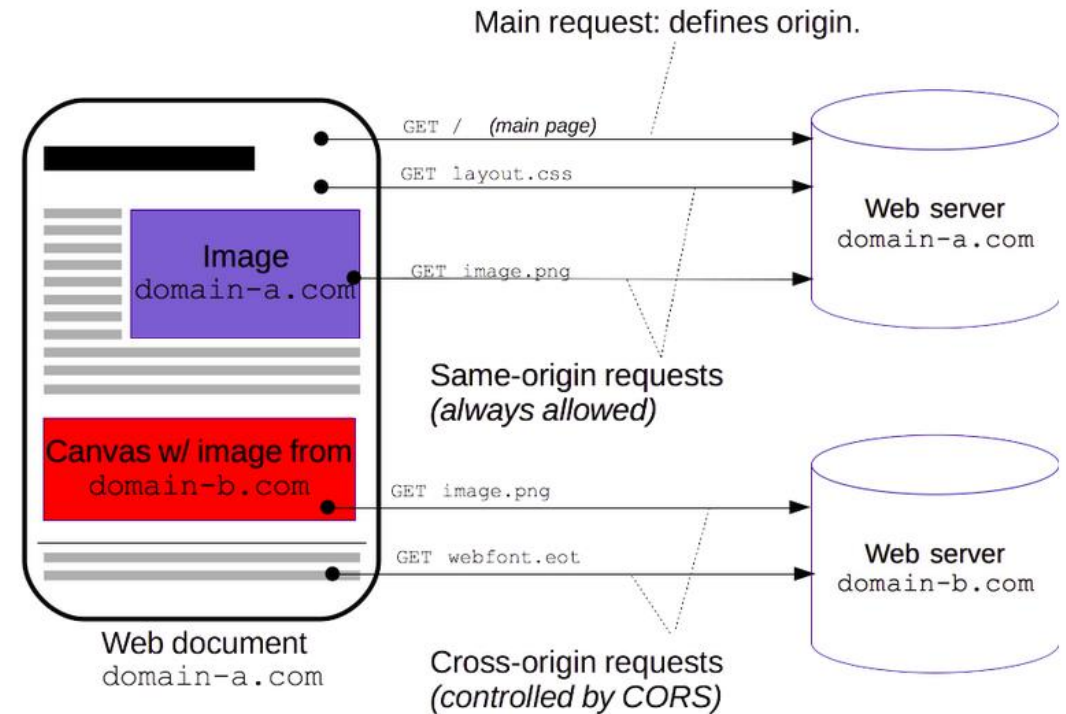
Dua tipe module yang sering digunakan dalam javascript :

- Modul yang mengandung sekumpulan fungsi (library)
- Modul yang mendeklarasikan sebuah entitas, misalnya user

- Module selalu bekerja dalam mode strict secara otomatis.
- Mode strict diterapkan mulai ES5. Dalam mode strict, user tidak dapat menggunakan variabel yang tidak dideklarasikan terlebih dahulu (undeclared), menghapus objek, dll.
- Untuk menjalankan mode strict ke script yang dibuat, maka ditambahkan string “use strict”; di awal filenya. Hal ini memastikan keamanan javascript.

## CORS

Script yang dipanggil dari domain yang berbeda butuh header Cross Origin Resource Sharing untuk menandakan bahwa browser mengizinkan membuka file tersebut. Hal ini dilakukan untuk alasan keamanan.



## CORS

CORS merupakan protokol penghubung antara browser dan web-service yang memberitahu browser bahwa itu adalah "OK" untuk mengeksekusi kode Javascript dari panggilan lintas domain.

Anda bisa melakukan transfer data antara domain. Metode ini mirip seperti API (application Programming Interface), anda bisa menggunakan json sebagai service-nya. Tentunya antara keduanya berbeda (api ataupun cors).

“ Jika kita menggunakan API biasanya anda harus mendapatkan sebuah token agar anda bisa menggunakan API tersebut, token itu berfungsi sebagai kunci akses yang di berikan untuk mengakses sebuah API.

\*) maaf kalau penjelasanku ini salah hehe ,silahkan ralat via komentar :) ”

“ Tetapi jika menggunakan CORS anda hanya cukup mengatur Access-Control-Allow-Origin yang biasanya di letakan di bagian header.

\*) maaf kalau penjelasanku ini salah hehe ,silahkan ralat via komentar :) ”



## Mengatur CORS agar semua bisa mengaksesnya

```
Access-Control-Allow-Origin: *  
Access-Control-Allow-Methods: GET, POST
```

## Mengatur CORS agar hanya beberapa yang dapat mengaksesnya

```
Access-Control-Allow-Origin: http://nama_domain.com  
Access-Control-Allow-Methods: GET, POST
```

# TUGAS JAVASCRIPT

---

1. Buatlah program kodingan yang terdiri dari import dan export
2. Gunakan Teknik DOM yang telah kalian pelajari
3. Simpan dalam file index.js
4. Upload pada akun github kalian dengan nama repository “Pertemuan-14-JS”



**Terima Kasih  
Syukron Katsiron  
Arigatou Gozaimasu**