

Nama : Denny Prayoga Setiawan Halim
NIM : 120140012
Kelas : PAM RA

Link Github Tugas 2 : <https://github.com/DennyPrayogaSetiawanHalim/Tugas-2-Individu-PAM>

1. Closure adalah sebuah fungsi yang menggunakan variabel dalam luar fungsi tersebut sebagai referensi. Closure sendiri biasanya menggunakan fungsi yang digabung bersama untuk menjadi bagian luar dan dalamnya sehingga closure memberikan akses dari variabel luar ke dalam fungsi yang ada di dalam.

```
1 function greeting() {  
2     let message = 'Hi';  
3  
4     function sayHi() {  
5         console.log(message);  
6     }  
7     return sayHi;  
8 }  
9 let hi = greeting();  
10 hi();
```

Sumber : <https://www.javascripttutorial.net/javascript-closure/> dan
<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Closures>

2. Immediately Invoked Function Expression atau (IIFE) adalah fungsi yang akan langsung run setelah didefinisikan dalam kode. Immediately Invoked Function Expression memiliki banyak fungsi seperti:
 - Membuat closure
 - Membuat variable dan method yang public maupun private
 - Digunakan pada Library JQuery

```
1 // Regular Function.  
2 function Greet()  
3 {  
4     console.log("Welcome to GFG!");  
5 };  
6 // Execution of Regular Function.  
7 Greet();  
8  
9 // IIFE creation and execution.  
10 (function() { console.log("Welcome to GFG!"); })();
```

Sumber : <https://www.geeksforgeeks.org/javascript-immediately-invoked-function-expressions-iife/>
dan <https://developer.mozilla.org/en-US/docs/Glossary/IIFE>

3. First class function adalah fungsi yang dapat dianggap sebagai variabel contohnya fungsi tersebut dapat digunakan sebagai argumen dalam fungsi lainnya.

```
1 function sayHello() {  
2   return "Hello, ";  
3 }  
4 function greeting(helloMessage, name) {  
5   console.log(helloMessage() + name);  
6 }  
7 // Pass `sayHello` as an argument to `greeting` function  
8 greeting(sayHello, "JavaScript!");  
9 // Hello, JavaScript!
```

Sumber : https://developer.mozilla.org/en-US/docs/Glossary/First-class_Function

4. Higher order function adalah fungsi yang menerima fungsi lain sebagai argumen atau sebagai hasil output dari return.

```
1 const arr1 = [1, 2, 3];  
2 const arr2 = arr1.map(function(item) {  
3   return item * 2;  
4 }); console.log(arr2);
```

Sumber : <https://blog.bitsrc.io/understanding-higher-order-functions-in-javascript-75461803bad>

5. Execution Context adalah environment yang digunakan untuk mengurus proses transformasi dan eksekusi dari kode yang sedang dijalankan. Terdapat 2 tipe Execution Context yaitu Global Execution Context yaitu Execution Context yang tidak ada di dalam fungsi dan Function Execution Context yaitu Execution Context yang terjadi saat fungsi dipanggil.

Sumber : <https://www.freecodecamp.org/news/execution-context-how-javascript-works-behind-the-scenes/>

6. Execution Stack adalah sebuah data stack yang mencatat semua Execution Context yang terjadi dari awal hingga akhir pada kode yang dijalankan.

Sumber : <https://www.freecodecamp.org/news/execution-context-how-javascript-works-behind-the-scenes/>

7. Event Loop adalah desain pada javascript di mana akan menunggu saat stack fungsi kosong dan memasukkan hal yang ada pada event queue ke dalam Function Execution stack.

Sumber : <https://www.geeksforgeeks.org/what-is-an-event-loop-in-javascript/>

8. Callback adalah fungsi yang diberikan ke fungsi lain sebagai argumen, yang akan langsung dipanggil dalam fungsi luar untuk menyelesaikan sebuah aksi atau rutin.

```
1 function greeting(name) {  
2   alert(`Hello, ${name}`);  
3 }  
4  
5 function processUserInput(callback) {  
6   const name = prompt("Please enter your name.");  
7   callback(name);  
8 }  
9  
10 processUserInput(greeting);
```

Sumber : https://developer.mozilla.org/en-US/docs/Glossary/Callback_function

9. Promises dan Async/Await

Promise adalah janji yang tidak diketahui valuenya yang memiliki 3 tahap yaitu tahap pending di mana janji masih blm dipenuhi dan tidak dibatalkan, fulfilled di mana janji tersebut dapat dipenuhi dan rejected di mana janji tersebut gagal dipenuhi.

Async/Await adalah kata kunci yang digunakan untuk membuat janji yang asynchronous untuk ditulis dalam style yang lebih jelas dan jernih.

Sumber : https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/async_function dan https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Promise