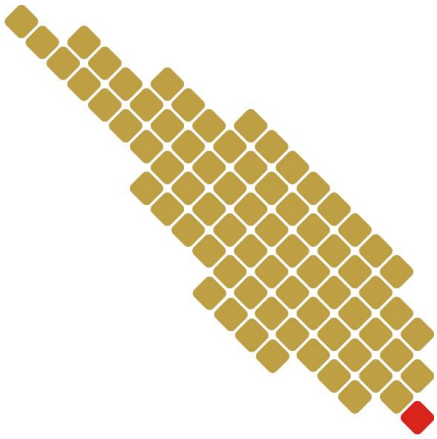
Tugas 2

Rangkuman

Pengembangan Aplikasi Mobile



DISUSUN OLEH

M. Anwar Ibrahim (119140221)

KELAS

RB

INSTITUT TEKNOLOGI SUMATERA

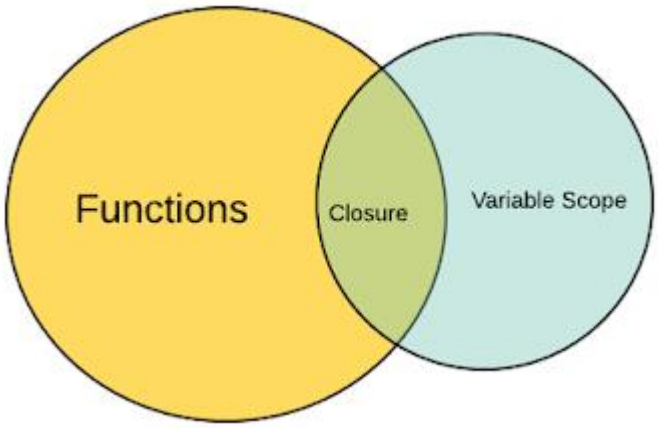
JURUSAN TEKNIK ELEKTRO, INFORMATIKA DAN SISTEM FISIKA

PROGRAM STUDI TEKNIK INFORMATIKA

2022

A. Closure

Closure adalah sebuah fungsi yang dapat memanggil fungsi induk walaupun fungsi induk telah di tutup dalam javascript fungsi ini memungkinkan memiliki variabel privat. Selain itu closure juga mengadopsi konsep Lamda calculus sehingga fungsi tersebut dapat di panggil oleh fungsi lainnya atau *nested function*.

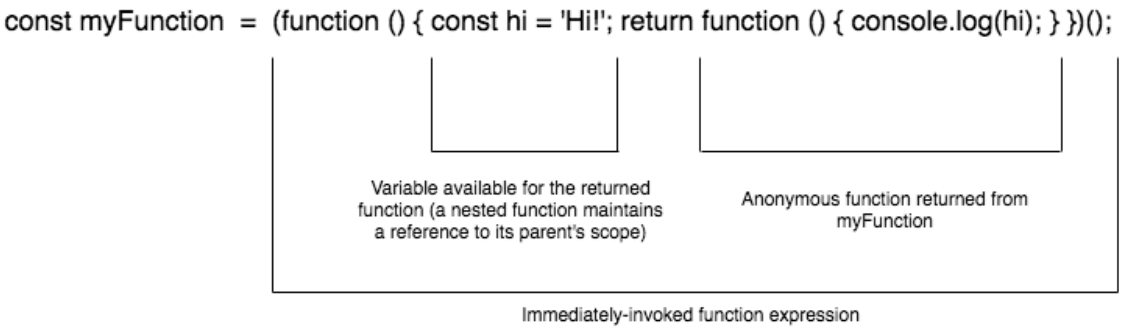


Contoh

```
// global scope

function parentFunction() {

  //local scope untuk fungsi parentFunction

    ....

  //lexical scope untuk fungsi childFunction

  return childFunction() {

    //local scope untuk fungsi childFunction

  }

}
```

B. Immediately Invoked Function Expression

salah satu cara untuk mengatasi kekurangan variable global dengan menggunakan Immediately Invoked Function Expression(IIFE), IFFE dapat menyembunyikan suatu kode yang kompleks serta variabel di dalamnnya selain itu dengan IFFE juga kita dapat memanggil fungsi tanpa menyimpannya.



Contoh sintaks

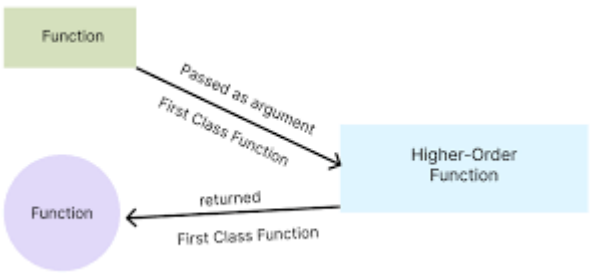
```
(function sayHi(){
   alert('Hi there!');
}());

// alerts 'Hi there!'
```

C. First-class function

Adalah fungsi yang dapat :

1. Mendukung fungsi tanpa nama
2. Mendukung pengiriman fungsi sebagai argumen
3. Fungsi return dari fungsi lain
4. Menyimpan fungsi sebagai stuktur data

Secara garis besar perbedaan dengan High Order Fucnction



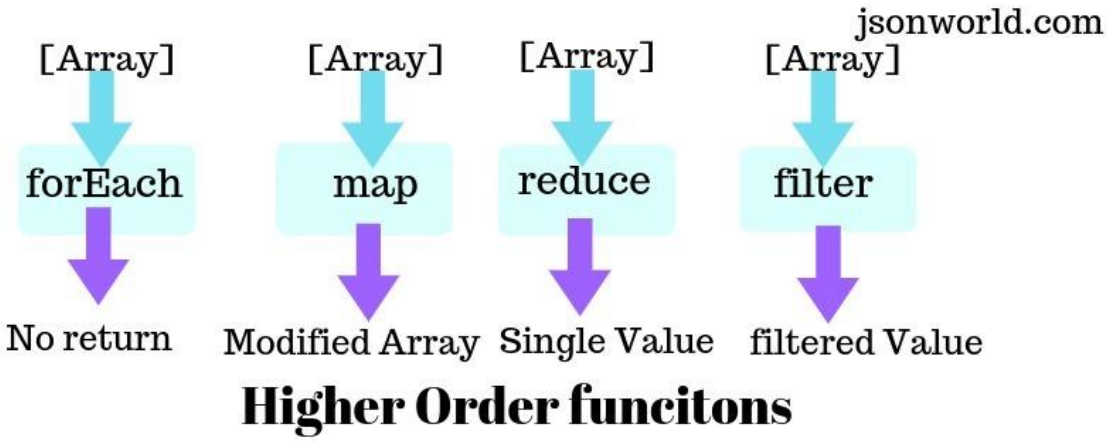
Contoh :

```
<html>
<head>
<title>Variable Function</title>
</head>
```

```
<body>
<script type="text/javascript">
  var fungsikuadrat = function (x) {
x = x * x;
return x;
};
alert(fungsikuadrat(3));
</script>
</body>
   </html>
```

D. Higher-order function

Adalah fungsi yang dapat di tempat kan dan di operasikan di dalam fungsi lain sebagai argumen atau nilai kembaliannya sehingga bisa di katakan bahwa fungsi ini menerima fungsi sebagai argumen atau ouput nya mengeluarkan nilai fungsi.



Contoh sintaks

```
const tahunLahir = [1998, 1999, 2000, 2001, 2002]
const umur = []

for (let i = 0; i < tahunLahir.length; i++) {
  umur.push(new Date().getFullYear() - tahunLahir[i])
}

alert(umur) // hasil: [21,20,19,18,17]
```

E. Execution Context

Adalah Konteks atau environment untuk mengeksekusi javascript yang dapat membungkus sebuah kode untuk di jalankan,Execution konteks memiliki 2 jenis yang bedakan dengan cakupannya yaitu local dan global, di javascipt sendiri terdapat 2 fase execution context yaitu creation phase dan execution phase.

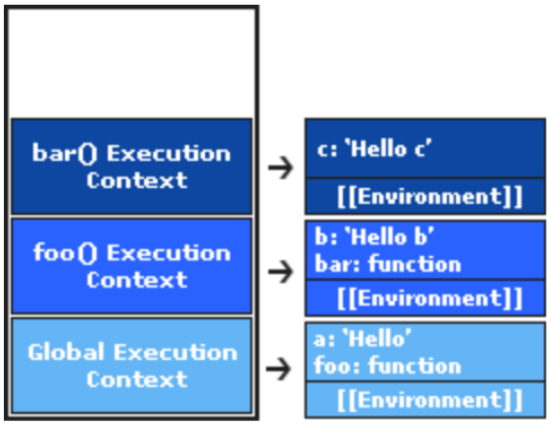
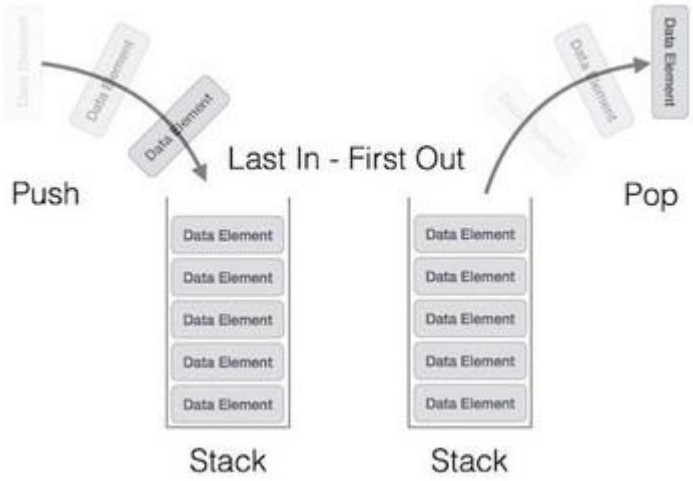


Diagram for Lexical Environment how they works

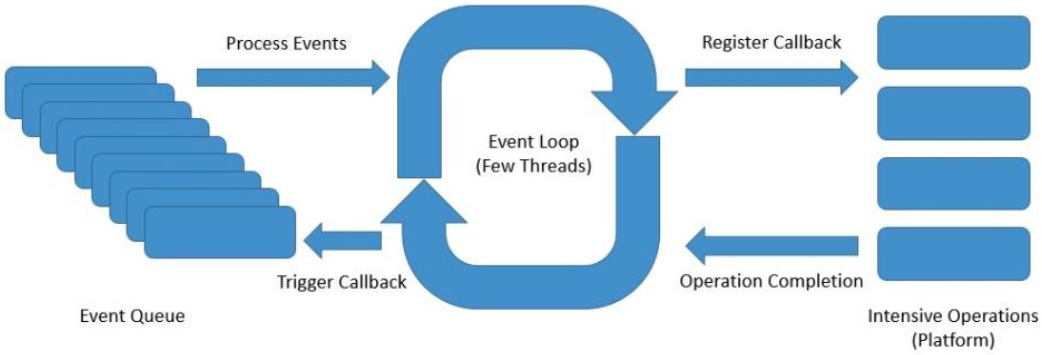
F. Execution Stack

Adalah sebuah tempat penumpukan event pada sebuah javascript interpreter dalam browser, tumpukan tersebut menggunakan metode Last in First Out



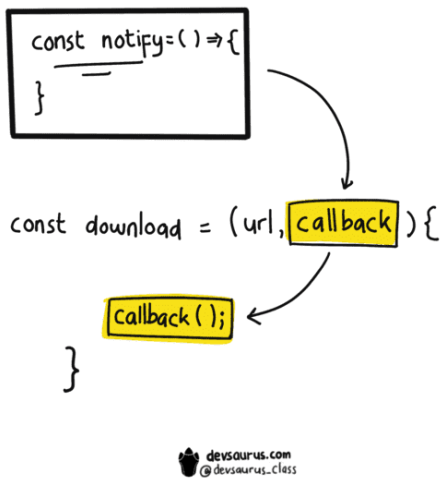
G. Event Loop

Adalah suatu proses yang hanya memiliki satu thread dengan banyak perulangan bahkan terus menerus dan tak terhingga, sehingga proses ini kita bisa gunakan untuk manjalankan banyak request dengan melakukan thread safe.



H. Callbacks

Adalah fungsi yang di kirimkan sebagai parameter pada fungsi lainnya, yang dapat mengakses pada poin tertentu.



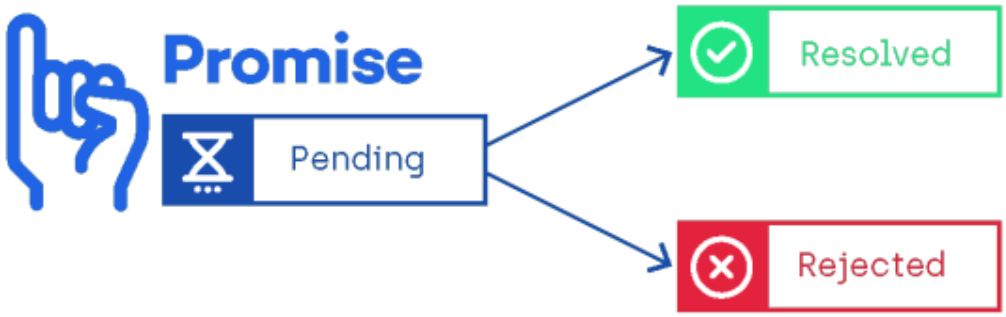
Contoh

```
function
main(param1,param2,callBack){
                                  console.log(param1, param2)
                                  callBack()
                                }

                                function myCallback(){
                                  console.log ('hello callback')
                                }

                                main(1,2,myCallback)

                                /* ==================
```

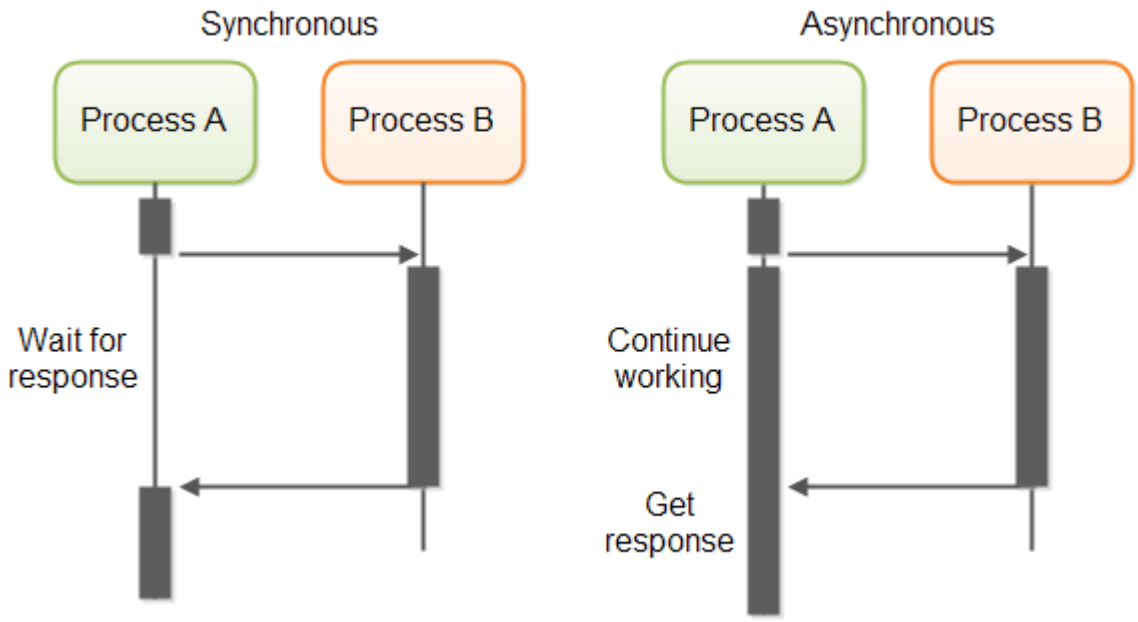
```
Output :
 1 2
 hello callback
 */
```

I. Promises dan Async/Await

Promises di analogikan sebagai janji yang apat di tepati dan tidak, ada 3 kondisi yang dapat terjadi pada promise yaitu Panding atau sedang dalam proses , Resolved atau janji di tepati atau berhasil dan Rejected / gagal atau janji di batalkan. Untuk menangani sebuah promise kita bisa menggunkan then dan catch.



Async adalah sebuah metode untuk mengekseksui kode tertentu tanpa menunggu kode sebelumnya di eksekusi.

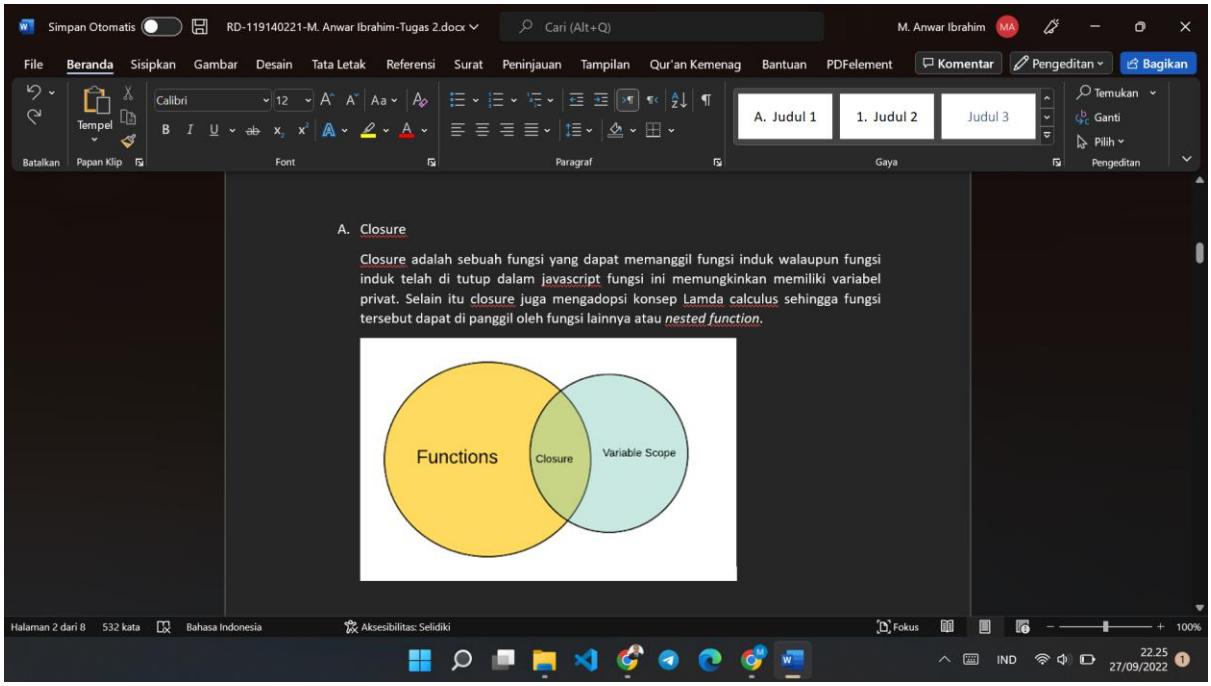


Await digunakan untuk menunggu eksekusi kode tertentu

J. Link Github :

https://github.com/MuhammadAnwarIbrahim/Tugas-2-PAM

K. Screen Shoot



L. Rerefernsi

https://dosenit.com/javascript/javascript-closure#JavaScript_Closure

https://www.shinta.dev/2020/10/apa-itu-closure.html

https://icalrn.id/immediately-invoked-function-expression/

https://jiewonchang1.gitbooks.io/udacity-summary/content/06-immediately-invoked-function-expressions-iife.html

https://sites.google.com/a/kursusinternet.com/kursusinternet-com/javascript/apa-itu-first-class-function

https://towardsdev.com/advanced-concepts-in-python-ii-5589faff6366

https://gading.dev/id/blog/mengenal-higher-order-function-di-javascript

https://jsonworld.com/blog/higher-order-function-in-javascript

https://stackoverflow.com/questions/56806807/does-a-js-execution-context-persist-in-memory-after-it-completes-if-a-reference

https://muhammadfahri.com/hoisting-execution-scope/

https://medium.com/@fadhlanfm/stack-dalam-javascript-f8136744ae9

https://medium.com/the-legend/concurrency-vs-event-loop-5648882ad668#:~:text=Event%20Loop%20adalah%20proses%20yang,tidak%20terhingga%20atau%20terus%20menerus.

https://id.quora.com/Apa-arti-callback-dalam-Javascript

https://devsaurus.com/javascript-asynchronous/

https://medium.com/codeacademia/belajar-callback-promise-dan-async-await-dalam-5-menit-fa1564956ab0