Nama : Raja Saputera

NIM : 120140228

Kelas : Pengembangan Aplikasi Mobile RA

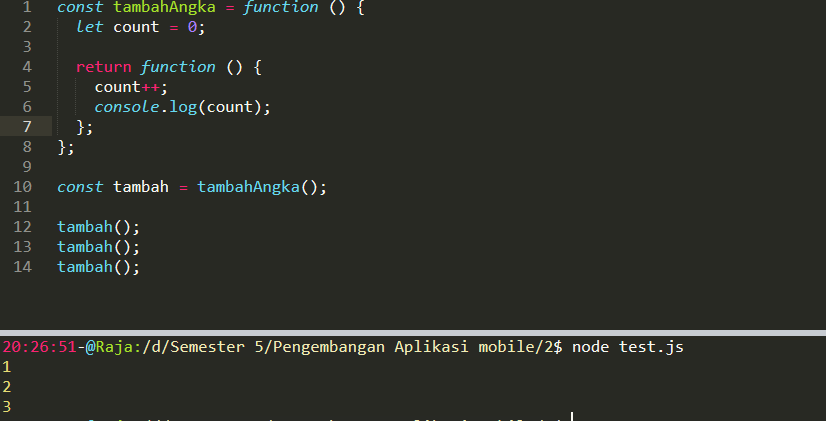
Link github : https://github.com/rajastra/Tugas-PAM

Tugas Individu 2

**1.Closure**

fungsi yang dieksekusi oleh fungsi lainnya(nested functions) sehingga fungsi tersebut memiliki akses ke variabel tempat fungsi tersebut lahir atau di buat.

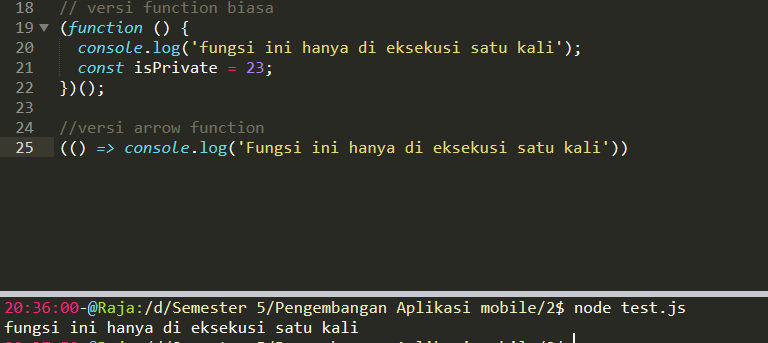
contoh :



Terlihat bawasannya fungsi tambah angka sebagai fungsi yang mengembalikan fungsi baru kemudian fungsi tersebut di letakkan ke dalam suatu variabel ketika variabel itu di panggil atau fungsi, fungsi yang di kembalikan bisa mengakses vairabel count dan juga bisa menambahnya hal ini bisa di lakukan karena adanya closure walaupun fungsi tambahAngka sudah selesai di eksekusi.

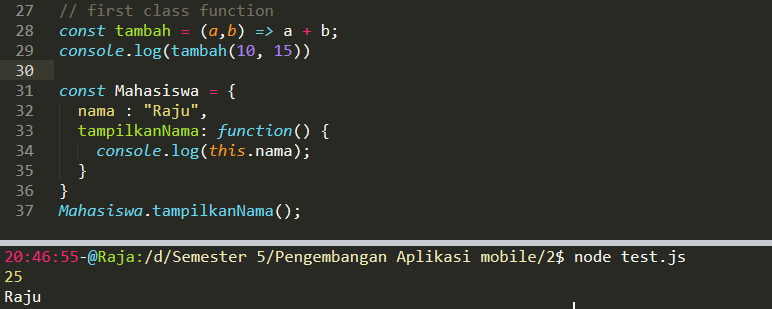
**2.Immediately Invoked Function Expression**

.Immediately Invoked Function Expression adalah fungsi yang akan menghilang apabila sudah di eksekusi satu kali. Contoh :



**3.First-class function**

First class function berararti bawassanya bahasa javascript memperlakukan fungsi hanya lah sebagai suatu nilai atau value dan fungsi juga hanya suatu tipe dari objek. Contoh :

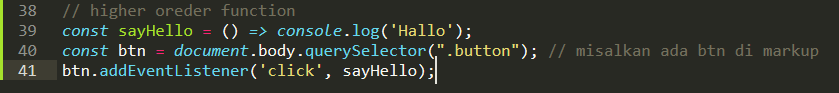


Terlihat kita bisa menyimpan function atau fungsi kedalam suatu variabel dan propertu dari suatu objek.

**4.Higher-order function**

Higher oreder function adalah suatu fungsi yang menerima fungsi lain sebagai argument atau yang mengembalikan fungsi baru .

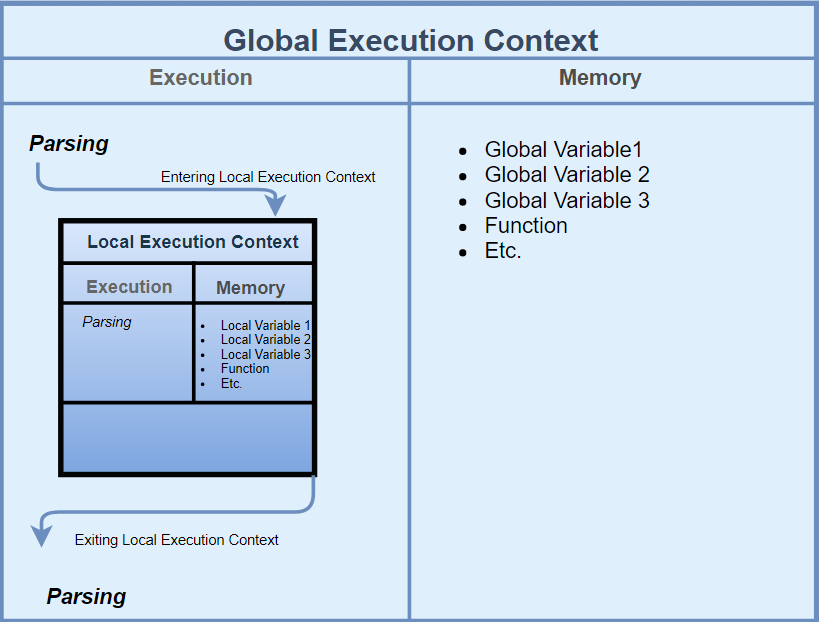
Contoh :



Terlihat bawasannya yang menjadi higher oreder function yaitu addEventListener, yang menerima argument sayhello yang merupakan function juga.

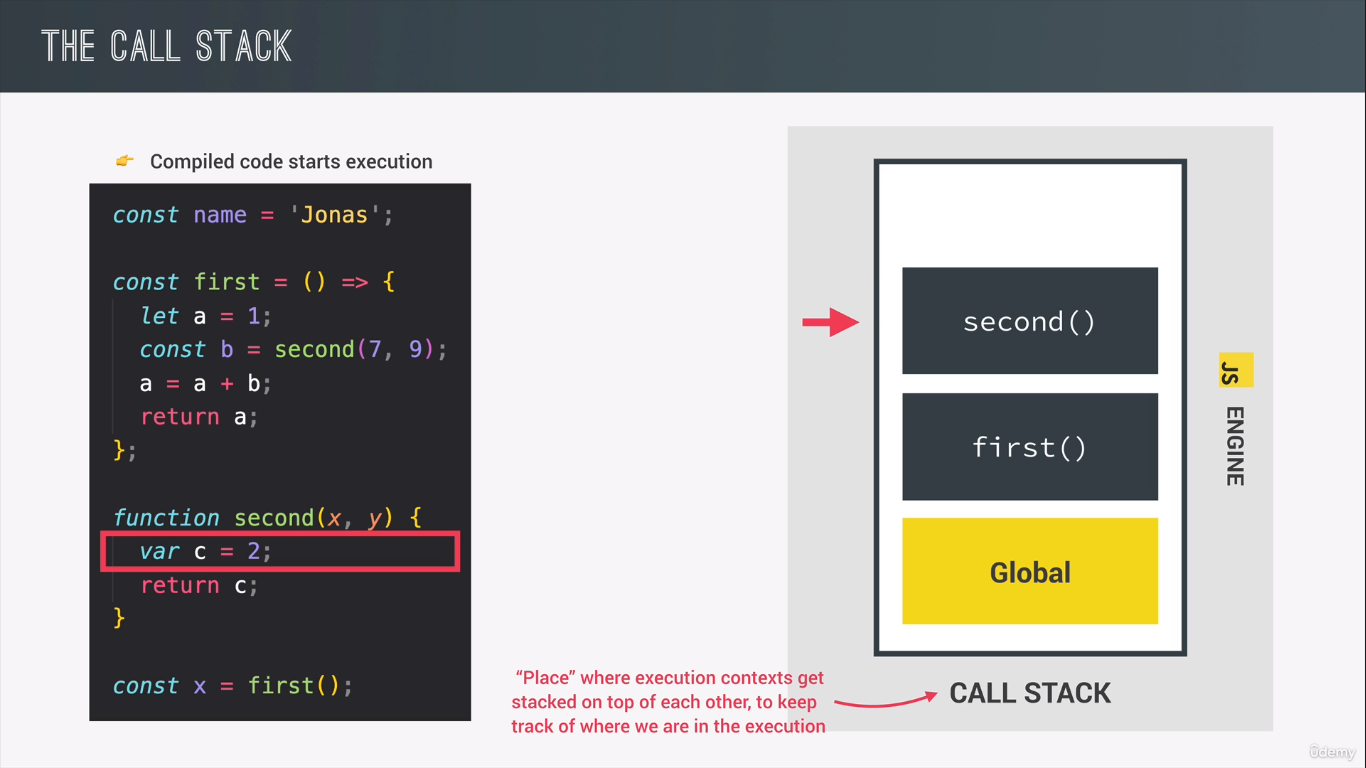
**5.Execution Context**

Execution Context adalah Environtment yang merupakan bagian dari javascript yang di eksekusi. Menyimpan semua informasi yang diperlukan untuk sutau kode menjadi di eksekusi. Seperti lokal variabel, argument yang di kirim untuk suatu fungsi



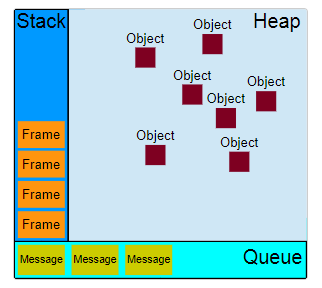
**6.Execution Stack**

Tempat di mana execution context ditumpuk, agar dapat melacak diamana sekarang di tempat eksekusinya.



**7.Event Loop**

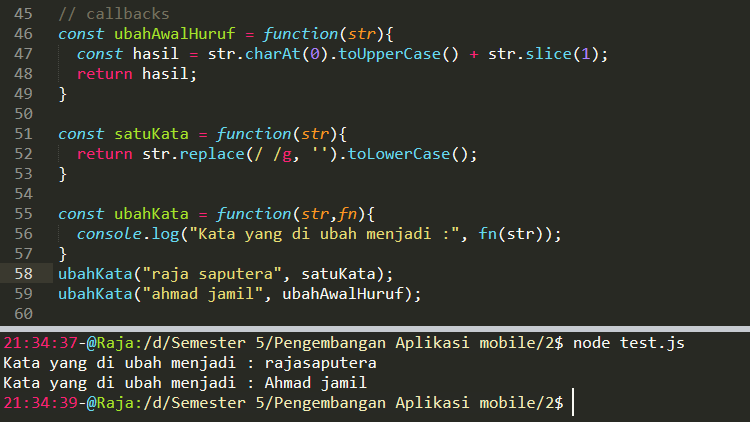
JavaScript memiliki model runtime berdasarkan event loop, yang bertanggung jawab untuk mengeksekusi kode, mengumpulkan dan memproses peristiwa, dan mengeksekusi sub-tugas yang diantrekan. Model ini sangat berbeda dari model dalam bahasa lain seperti C dan Java.



**8.Callbacks**

Callback pada dasarnya adalah sebuah function. Callback hanya sebuah istilah untuk function yang di passing ke dalam function lain sebagai argument, yang kemudian di eksekusi oleh function yang membungkus function callback tersebut.

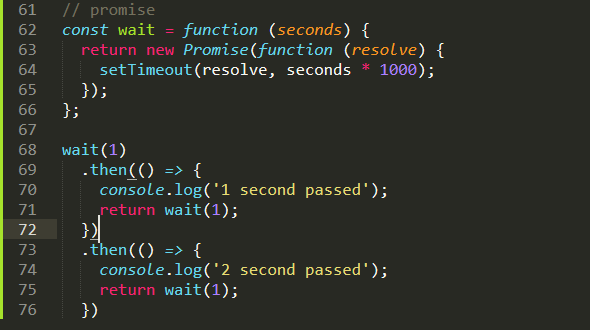
Contoh :



**9.Promises dan Async/Await**

-Promise adalah promise adalah Sebuah mekanisme baru pada fitur javascript / ES6 yang merepresentasikan sebuah object request pengolahan data yang dilakukan secara asynchronous seperti ajax, dan promise ini mewakili sebuah operasi yang belum selesai, tetapi diharapkan di masa mendatang.

Contoh :

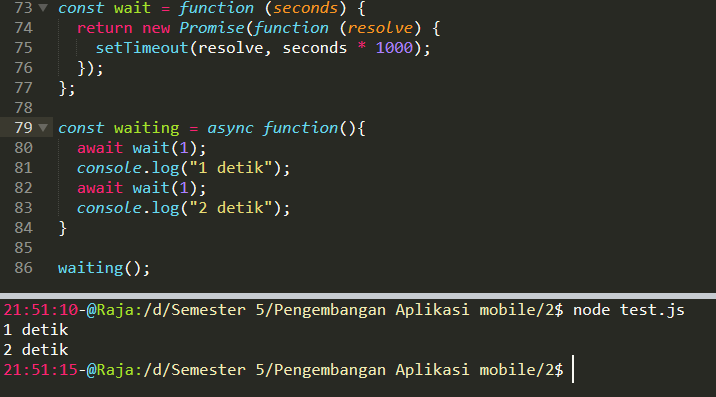


Jadi disini membuat promise untuk async code yaitu menunggu waktu 1 detik kemudian di eksekusi dan dapat di hasilkan hasilnya menggunakan then.

-Async / await

Async/await adalah fitur yang hadir sejak ES2017. Fitur ini mempermudah kita dalam menangani proses asynchronous.Async/Await merupakan sebuah syntax khusus yang digunakan untuk menangani Promise agar penulisan code lebih efisien dan rapih

Contoh :



Terlihat dari contoh bawasannya fungsi wait yang merupakan suatu promise kemudian dengan fungsi async kita dapat menggunakan fitur await di dalam fungsi tersebut sehingga dapat menjalan perintah asynchronous

Referensi :   
<https://www.udemy.com/course/the-complete-javascript-course/>

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

<https://medium.com/@aryarifqipratama/mengupas-tuntas-mengenai-apa-itu-callback-ce7bafc72173>

https://codesyariah122.github.io/javascript-promise/#:~:text=Sooo%20jadi%20promise%20adalah%20Sebuah,tetapi%20diharapkan%20di%20masa%20mendatang.