**Nama : Vika Alpiana**

**Program : React and React Native for Front End Developer**

**Kode Kelas : RCTN-KS05-004**

* **Class**

Untuk mendefinisikan class biasanya nama class dimulai dengan huruf kapital., contoh: Employee, Person. Kemudian didalamnya kitab isa mendefinisikan methods, constructors ataupun properties

* **Constructor**

Mirip dengan mendeklarasikan method, perbedaannya constuctor secara otomatis akan dijalankan pada saat inisialisasi class dengan propertis keyword this

* **Class Inheritance**

Menggunakan sintaks "extends" yang membuat semua method dan properties dari parent - childnya untuk class yang dibuat menggambarkan sebuah class merupakan turunan dari class lain (inheritance).

* **Method Super**  
  Digunakan untuk keyword super untuk menghubungkan dengan leluhur dari class Employee yang ada di class person. Kita tinggal memanggil method super dengan parameter yang mau kita assign.
* **JavaScript Modules**

Dengan JavaScript Modules bisa membuat kose JS lebih modular yaitu dengan Menggunakan IIFE atau immediately-invoked function expression, sebuah fungsi yang dibungkus tanda kurung untuk langsung dieksekusi. Kemudian muncul commons yang lazim digunakan di NodeJS dengan sintaks require dan module.exports. Selanjutnya Menggunakan AMD atau Requires.js, Lalu ES6 memberika solusi dengan menggunakan sintaks import dan export.

* **Promise**

Promise adalah sebuah objek yang menghasilkan sebuah nilai dimasa yang akan datang. Hasil Promise berisi 2 hal yaitu Resolve (disertakan dengan nilai atau value), Reject ( untuk kasus error).

Cara penggunaan promise cukup sederhana dengan merangkai fungsi promise dengan method then() = untuk resolve dan method.catch() = untuk rejection. Dengan syarat fungsi method atau librerary harus sudah mendukung promise misalnya fungsi fetch yang sudah tersedia di berbagai browser modern.

* **New promise**

New Promise digunakan untuk membuat sebuah function dapat menggunakan Promise. Jika fungsi promise belum mendukung, kita harus membungkus fungsi tersebut kedalam sebuah promise baru.

Cara penggunaan setTimeout dengan setInterval kurang lebih sama. setTimeout menerima dua parameter: fungsi callback, dan durasi beberapa lama wakti yang sudah ditundah sampai fungsi callback dipanggil. Bedanya adalah setTimeout akan selesai Ketika fungsi callback dipanggil, sementara setInterval akan terus menjalankan fungsi callback hingga dilakukan clearInterval().

* **Async/Await**

Dengan async/await, sudah dapat menjalankan kode asynchronous diatas dengan gaya synchronous. dengan syarat, browser atau nodejs support async/await. Dan async/await hanya berjalan didalam fungsi karena harus menandai mana fungsi yang akan menjalankan kode async mana fungsi biasa. Solusinya mudah yaitu bungkus kode yang tidak berada didalam fungsi menjadi sebuah fungsi.

**Async/Await**

Dengan fitur async/await kita dapat menjalankan kode asynchronous dengan gaya synchronous. Dan fitur ini kompetibel dengan Promise, dengan syarat browser atau Node.js support async/await. Async/await hanya berjalan didalam fungsi karena kita harus menandai mana fungsi yang akan menjalankan kode async mana fungsi biasa. Solusinya dengan membungkus kode yang tidak berada didalam fungsi menjadi sebuah fungsi.