1. Explain how Object Oriented Programming works with a thorough understanding of the keyword this and the new keyword
2. What is the new class syntax and how to create instance methods, class methods?
3. Give an example of how to implement inheritance in ES2015 using extends and super
4. Imagine refactoring an ES5 application to use ES2015, how would you go about it?
5. Give an example of how you structure applications with design patterns using closure and modules
6. What are your preferred ways of testing your web application?
7. Which web server do you use? Why? Explain pros and cons of your choice.
8. What is your preferred production deployment process?
9. Give an example of clean README.md documentation.

**JAWABAN :**

1. cara kerja Object Oriented Programming ( OOP ) atau Pemrograman Berorientasi Objek adalah dengan membuat sebuah "class" yang merupakan sebuah identitas dari sebuah object yang akan di buat, contohnya "Barang", dan di dalam "class" terdapat beberapa propety yang merupakan part atau bagian dari sebuah object dan ada juga part atau bagian class yang udah secara default ada seperti constructor, analoginya barang itu udah pasti ada bentuknya, nah di dalam bentuknya itu ada beberapa point atau property yang dapat di tentukan, analoginya barangnya memiliki height atau width, dan constructur itu lah ya yg dapat menampung data atau property yang telah di berikan, jadi ketika membuat sebuah class kita bisa menentukan data yang akan di terima oleh sebuah class di dalam constructor dan dapat membuat sebuah property baru di dalam sebuah class,

dan untuk dapat memberikan nilai dari sebuah object class adalah dengan menggunakan keyword "new", dimana ketika memanggil keyword tersdebut harus di ikuti dengan nama class dan default value dari kelas tersebut, contohnya : new Barang(100, 200)

untuk dapat menggunakan value yang di berikan untuk sebuah class, kita menggunakan this, yang dimana keyword tersebut merupakan sebuah keyword khusus yang merujuk pada pada objek pemiliknya. Maksudnya adalah nilai dari this sangat bergantung pada di mana keyword this ini diletakkan atau di panggil. seperti contohnya : new Barang(100, 200), jadi this akan menggambil value 100 dan juga 200

1. new class syntax adalah sebuah bagian dari cara mendeklarasikan atau menginisialkisasikan sebuah object dimana kita membutuhkan initial value untuk sebuah object yang dapat di gunakan atau object baru dari sebuah class yang dapat jg di sebut instance.

bagaimana cara membua sebuah object baru di dalam class atau instance method ? caranya dalah dengan membuat sebuah class dimana di dalamnya kita insitialisatin kan dengan sebuah constructor dan menentukan value atau parameter yang dapat di terima, contohnya :

// buat class

class Barang{

constructor(width, height){

this.widh = width

this.height = height

}

}

// Inistialisation new Object or Isntance methide

const meja = new Barang(100, 200)

dan di dalam sebuah class kita juga dapat menambahkan sebuah custome methode atau methode baru caranya dengan menambahkan sebuah methiode atau function ke dalam kelas, dan utuk memanggil methiode tersebut kita tinggal menginsialisasi class nya seperti contoh di atas, dan dapat memanggil variable atau object barunya dan di ikuti dengan nama methiodenya contohnya : saya telah menabahkan methode dengan nama "tampilkanData", kita tinggal memanggil meja.tampilkanData()

1. dsinin saya memberikan contoh dimana saya membuat 2 buah object yaitu "Barang" dan "Ukuran", dimana class "Ukuran" akan mewariskan propertynya ke class "Barang" :

class Ukuran{

constructor(width, height){

this.widh = width

this.height = height

}

}

class Barang extends People{

constructor(width, height, name){

super(width, height)

this.name = name

}

}

1. ketika saya mebayangkannya, saya kepikiran untuk menggunakan tools yg dapat mengkonversikannya dari ES5 ke ES2015 secara instan, dan saya kepikiran untuk mengrefactornya menggunakan Guide dari beberapa company yang ada contohnya AirBNB, FaceBook dan masih banyak lagi yang merupakan sebuah best partice
2. to be honest saya terlalu familiar dengan kedua pattern tersebut di real project saya, tpi saya agak sedikit paham dengan closure, yaitu dimana sebuah function yang dapat mereturn function lgi, contohnya :

function calc(num1) {

var num2 = 10;

return function(num3) {

var sum = num1 + num2 + num3;

console.log(sum);

}

}

var add = calc(5)(15);

1. kalau saya pribadi belum pernah mengimplementasikan outomatision testing, jdi lebih prefer ke testing manual atau tdd secara manual yang biasa saya pake adalah integration testing dengan melakukan testing main function dari sebuah app yang saya develop
2. kalau di suruh memilih, mungkin saya tidak dapat meilihnya di karenakan saya kurang familiar dangan itu, tpi biasanya tim saya menggunakan apache web server
3. kalau untuk deployment saya pernah menggunakan netlify, vercel dan heroku, dan saya lebih prefer ke vercel
4. kalau di dokumentasi saya ada beberapa dokumntasi yang rapi tetapi semunya bersifat bserfifat private di karenakan itu di company saya sebelumnya kalau untuk secara pribadi saya ada https://github.com/MochIsrin068/flutter\_login-register\_mysql yang merupakan salah satu repository github saya yang populer dan udh banyak di baca orang