Laporan Praktikum Pemrograman

**20**

Modul IV – Object & Matrik

Rizky Saputra/22103001007

Dosen : Achmad Arif Munaji, ST., M.Kom

Tanggal praktikum: 15 Nopember 2022

Ikytopup21@gmail.com

Teknik Komputer

Institut Teknologi dan Sains Nahdlatul Ulama Kalimantan

***Abstrak*— Praktikum ini secara umum bertujuan untuk mempelajari programing mengenai dunia coding melalui methode praktikum. Secara khusus percobaan ini di maksud untuk mengasah kemampuan kita untuk bermain logika dan daya ingat kita dalam mempelajari materi pemrograman yang telah di pelajari, juga memahami cara kerja suatu program, dan mengenali dasar-dasar Bahasa pemrograman salah satu saat ini yang di pelajari yaitu JavaScript. Pada praktikum ini Kembali mengerjakan basic JavaScript pada freeCodeCamp tentang object di mulai dari build JavaScript object sampai Accessing nesting objects.**

***Kata kumci— bahasa pemrograman JavaScript***

1. Pendahuluan

Object adalah sekumpulan list dari tipe data primitif (terkadang juga tipe data reference) yang menyimpan nilai dengan konsep berpasangan name-value. Tiap item (yang lebih dikenal dengan nama variabel) disebut dengan property, dan function disebut dengan method, dan Matrix adalah objek yang melilit JavaScript biasa Array, menyediakan fungsi utilitas untuk manipulasi matriks yang mudah.

Di praktikum pertemuan ke 4 ini dengan materi object & matriks di kerjakan pada freeCodeCamp di mana kami akan memecahkan atau mengerjakan poin tugas yang ada di situs tersebut sebanyak 11 kasus tentang object dan 4 tentang matriks.

1. Tinjauan Pustaka
   1. Object pada JavaScript

Nilai-nilai selain tipe data dasar secara otomatis dianggap sebagai objek. Objek dalam Javascript didefinisikan sebagai mutable properties collection, yang artinya adalah sekumpulan properti (ciri khas) yang dapat berubah nilainya, Objek sebenarnya adalah sebuah variabel yang menyimpan nilai (properti) dan fungsi (method),Keberadaan objek dalam struktur suatu kalimat memiliki fungsi antara lain, membentuk kalimat dasar pada kalimat berpredikat transitif; memperjelas makna kalimat; serta membentuk kesatuan atau kelengkapan pikiran.

* 1. Matriks pada JavaScript

Matriks adalah objek yang melilit JavaScript biasa (Array), menyediakan fungsi utilitas untuk manipulasi matriks yang mudah seperti subset, size, , resize, clone, dan lainnya.Matriks dapat dibuat dari array menggunakan fungsi math.matrix. Array yang disediakan dapat berisi array bersarang untuk membuat matriks multidimensi. Saat dipanggil tanpa argumen, matriks kosong akan dibuat.

Matriks dapat berisi berbagai jenis nilai: angka, bilangan kompleks, satuan, atau string. Jenis yang berbeda dapat dicampur bersama dalam satu matriks.

1. Langkah Praktikum

Adapun urutan pengerjaan dalam praktikum ini dimulai dengan free test tentang function, setelah selesai kemudian di lanjut kan dengan membuka wibsiteyang digunakan dalam praktium (FreeCodeCamp), setelah FreeCodeCamp terbuka klik >> Get started>>klik Continue with google >> setelah itu klik JavaScript algorithms and Data Structures Certification >> Setelah klik kemudian melakukan perintah untuk mengerjakan poin-poin perulangan pada JavaScript pada website tersebut sebanyak 8 soal atau perintah pada situs wibesite tersebut.

1. hasil dan analisis

Di mana pratikum di mulai free test tentang function , sebagai berikut :

Tulis kode program sederhana menggunakan FUNCTION untuk melakukan proses Aritmatika 2 variabel atau nilai dan panggil Function tersebut.

setelah free test di lanjutkan dengan mengerjakan kuis atau perintah pada website FreeCodeCamp tentang function sebagai berikut :

1. Build JavaScript Objects

Objek serupa dengan arrays, kecuali bahwa alih-alih menggunakan indeks untuk mengakses dan memodifikasi datanya, Anda mengakses data dalam objek melalui apa yang disebut properties.

Objek berguna untuk menyimpan data dengan cara terstruktur, dan dapat mewakili objek dunia nyata.

myDog = {

// Only change code below this line

// Only change code above this line

};

let myDog = {

name: "helly",

legs: 3,

tails: 1,

friends: ["Water", "Dogs"]

};

1. Accessing Object Properties with Dot Notation

Ada dua cara untuk mengakses properti objek: notasi titik ( .) dan notasi braket ( []), mirip dengan array.

Notasi titik adalah apa yang Anda gunakan saat Anda mengetahui nama properti yang Anda coba akses sebelumnya.

// Setup

const testObj = {

"hat": "ballcap",

"shirt": "jersey",

"shoes": "cleats"

};

// Only change code below this line

const hatValue = testObj.hat; // Change this line

const shirtValue = testObj.shirt; // Change this line

1. Accessing Object Properties with Bracket Notation

Cara kedua untuk mengakses properti objek adalah notasi braket ( []). Jika properti objek yang Anda coba akses memiliki spasi di namanya, Anda perlu menggunakan notasi braket.

Namun, Anda masih bisa menggunakan notasi braket pada properti objek tanpa spasi.

// Setup

const testObj = {

"an entree": "hamburger",

"my side": "veggies",

"the drink": "water"

};

// Only change code below this line

const entreeValue = testObj["an entree"]; // Change this line

const drinkValue = testObj["the drink"]; // Change this line

1. Accessing Object Properties with Variables

Kegunaan lain dari notasi braket pada objek adalah untuk mengakses properti yang disimpan sebagai nilai variabel. Ini bisa sangat berguna untuk iterasi melalui properti objek atau saat mengakses tabel pencarian.

// Setup

const testObj = {

12: "Namath",

16: "Montana",

19: "Unitas"

};

// Only change code below this line

const playerNumber = 16; // Change this line

const player = testObj[playerNumber]; // Change this line

1. Updating Object Properties

Setelah Anda membuat objek JavaScript, Anda dapat memperbarui propertinya kapan saja seperti Anda memperbarui variabel lainnya. Anda dapat menggunakan notasi titik atau tanda kurung untuk memperbarui.

// Setup

const myDog = {

"name": "Coder",

"legs": 4,

"tails": 1,

"friends": ["freeCodeCamp Campers"]

};

// Only change code below this line

// Setup

myDog.name = "Happy Coder";

1. Add New Properties to a JavaScript Object

Anda dapat menambahkan properti baru ke objek JavaScript yang ada dengan cara yang sama seperti Anda memodifikasinya.

const myDog = {

"name": "Happy Coder",

"legs": 4,

"tails": 1,

"friends": ["freeCodeCamp Campers"]

};

myDog["bark"] = "floo"

1. Delete Properties from a JavaScript Object

Kami juga dapat menghapus properti dari objek seperti ini:

// Setup

const myDog = {

"name": "Happy Coder",

"legs": 4,

"tails": 1,

"friends": ["freeCodeCamp Campers"],

"bark": "woof"

};

// Only change code below this line.

delete myDog.tails;

1. Using Objects for Lookups

Objek dapat dianggap sebagai penyimpanan kunci/nilai, seperti kamus. Jika Anda memiliki data tabular, Anda bisa menggunakan objek untuk mencari nilai daripada switchpernyataan atau if/elserangkaian. Ini sangat berguna ketika Anda mengetahui bahwa data input Anda terbatas pada rentang tertentu.

function phoneticLookup(val) {

var result = "";

var lookup = {

"alpha": "Adams",

"bravo": "Boston",

"charlie": "Chicago",

"delta": "Denver",

"echo": "Easy",

"foxtrot": "Frank"

};

// After converting our case statements into object properties you can make use of the variable `result` to let the function return the correct value.

result = lookup[val];

// Only change code above this line

return result;

}

1. Testing Objects for Properties

Terkadang berguna untuk memeriksa apakah properti dari objek yang diberikan ada atau tidak. Kita dapat menggunakan .hasOwnProperty(propname)metode objek untuk menentukan apakah objek tersebut memiliki nama properti yang diberikan. .hasOwnProperty()kembali trueatau falsejika properti ditemukan atau tidak.

function checkObj(obj, checkProp) {

if (obj.hasOwnProperty(checkProp)) {

return obj[checkProp];

} else {

return "Not Found";

}

}

1. Manipulating Complex Objects

Terkadang Anda mungkin ingin menyimpan data dalam Struktur Data yang fleksibel . Objek JavaScript adalah salah satu cara untuk menangani data yang fleksibel. Mereka mengizinkan kombinasi sewenang-wenang string , angka , boolean , array , fungsi , dan objek .

var myMusic = [

{

artist: "Billy Joel",

title: "Piano Man",

release\_year: 1973,

formats: ["CD", "8T", "LP"],

gold: true

},

// Add record here

{

artist: "Deep Purple",

title: "Smoke on the water",

release\_year: 1976,

formats: ["CD", "8T", "LP"]

}

];

1. Accessing Nested Objects

Sub-properti objek dapat diakses dengan menggabungkan notasi titik atau tanda kurung.

const myStorage = {

"car": {

"inside": {

"glove box": "maps",

"passenger seat": "crumbs"

},

"outside": {

"trunk": "jack"

}

}

};

const gloveBoxContents = myStorage.car.inside["glove box"];

* **Analisis**

pada praktikum ini untuk Object & Matrik dalam mengerjakan object tidak ada kesulitan karna sebelum nya sudah mempelajari object di freecodecamp pada pertemuan pemrograman. Sehingga membantu untuk mengerjakan poin-poin pada object freeCodeCamp namun pada matrix sama sekali tidak mengerti karna pada sebelum tidak ada materi tentang matrix.

1. Kesimpulan

Object adalah sekumpulan list dari tipe data primitif (terkadang juga tipe data reference) yang menyimpan nilai dengan konsep berpasangan name-value. Dan Matriks adalah objek yang melilit JavaScript biasa (Array), menyediakan fungsi utilitas untuk manipulasi matriks yang mudah seperti subset, size, , resize, clone, dan lainnya.

Apa yang saya kerjakan pada website FreeCodeCamp tentang object ini saya mulai mampu memahami cara kerja object, dan sudah mampu memahami bekerja dengan object. Untuk matriks saya akan mendalami lagi untuk belajar matriks.

Daftar Pustaka

*https://www.petanikode.com/javascript-objek/*