Laporan Praktikum Pemrograman

**20**

Modul IV – Object & Matrik

Nellyana Safitri/22103001011

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

Tanggal praktikum: 09 Nopember 2022

Nellyana2@gmail.com

Teknik Komputer

Institut Teknologi dan Sains Nahdlatul Ulama Kalimantan

***Abstrak*—Materi utama pada praktikum ini adalah mengenai objek dan matrik pada JavaScript. Suatu objek dapat diartikan dalam arti luas sebagai penyimpan nilai (properti). Dan matriks secara umum dapat diartikan sebagai himpunan yang berisi bilangan real atau bilangan kompleks. Pada praktikum ini tujuan utamanya adalah untuk memahami materi yang diberikan dan diimplementasikan dalam metode praktikum. Selain itu, praktikum ini menguji sejauh mana pemahaman materi yang diberikan pada pertemuan sebelumnya mampu menyelesaikan permasalahan terkait materi yang diberikan pada website freecodecamp dan Visual Studio Code.**

***Kata kumci— Object dan Matrik pada JavaScript***

1. Pendahuluan

Pada pembelajaran praktikum pemrograman yang keempat ini, materi yang diberikan yaitu mengenai object dan matrik pada Javascript yang diakses melalui website freecodecamp dan visual studio code. Pada praktikum ini, sama halnya dengan praktikum sebelumnya yaitu memecahkan masalah coding yang pada kali ini materinya adalah object dan matrik pada javascript. Soal yang diberikan di freecodecamp pada kali ini adalah sebanyak 11 soal yang berkaitan dengan object dan 4 soal di visual studio code yang berkaitan dengan matrik. 11 soal tersebut adalah yang pertama Build javascript object, yang kedua adalah Accessing object properties with dot notation, yang ketiga Accessing Object Properties with Bracket Notation , yang keempat Accessing oobject properties with variables, yang kelima Updating object properties, yang keenam Add new properties to a javascript object , yang ketujuh Delete properties from a javascript object, yang kedelapan using object for lookups, yang ke sembilan testing object for properties, yang kesepuluh manipulating complex object dan yang terakhir Accessing nested object.

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

Pertama-pertama pada praktikum ini diawali dengan soal freetest mengenai object dan matrik. Setelah selesai mengerjakan soal freetest tersebut, lalu dilanjutkan dengan membuka google chrome tentang website freecodecamp dan aplikasi visual studio code. Setelah itu mulai menjawab soal-soal pada website freecodecamp. Soal tersebut sebanyak 11 soal dan setiap soal terdapat solusi (bantuan). Selain itu, ada 4 soal yang dijawab dan diselesaikan melalui aplikasi visual studio code yang berkaitan dengan matrik.

1. hasil dan analisis

Sebelum dimulainya pratikum pertama-tama dimulai free test tentang object dan matrik,. setelah selesai dilanjutkan dengan mengerjakan kuis atau perintah pada website FreeCodeCamp tentang object dan pada aplikasi visual studio code. berikut rincinya:

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 dan matrik pada JavaScript saya sedikit mulai memahami dikarenakan sebelumnya sudah mempelajari materi object dan matrik pada pertemuan pemrograman dan dilanjutkan diajarkan oleh teman-teman Sehingga membantu saya untuk mengerjakan poin-poin pada freeCodeCamp.

1. Kesimpulan

Hasil dari praktikum keempat ini dapat disimpulkan bahwa penggunaan object dan matrik pada javascript sangat penting. Selain itu, object dan matrik juga sangat perlu untuk dipelajari oleh para programmer karena banyak kaitannya object dan matrik pada dunia coding.

Daftar Pustaka

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