

Laporan Praktikum “Pemrograman 1”

Modul 4 – Object & Matrik

Muhammad iqbal/ 22103001003

Dose : Achmad Arif Munaji., S.T., M.Kom

Tanggal praktikum: Selasa, 15 november 2022

Alamat email : mhmmdiqbal20210@gmail.com

Fakultas Teknik– Jurusan Teknik Komputer
ITSNU Kalimantan

5+3+8+7+12+3+2 =

40

5 Abstrak— Tujuan dari praktek ini biasanya untuk belajar pemrograman dari dunia pemrograman melalui metode praktis. Secara khusus, tujuan dari ujian ini adalah untuk meningkatkan kemampuan kita dalam bermain logika dan memori sambil mempelajari materi pemrograman yang dipelajari, serta memahami pengoperasian program dan mengenal dasar-dasar salah satu bahasa pemrograman yang digunakan saat ini. dipelajari yaitu JavaScript. Dalam latihan ini, kita kembali ke JavaScript dasar bekerja dengan objek di freeCodeCamp, dimulai dengan membuat objek JavaScript dan menggunakan objek bersarang.

3 I. PENDAHULUAN

Objek adalah daftar tipe data primitif (terkadang disebut tipe data referensi) yang menyimpan nilai menggunakan konsep pasangan nama-nilai. Setiap objek (lebih dikenal sebagai variabel) disebut properti dan fungsi disebut metode, dan matriks adalah objek yang membungkus array JavaScript standar dan menyediakan fungsi yang berguna untuk menyederhanakan matriks.

Pada pertemuan praktik ke-4 dengan materi objek dan matriks, kami melakukannya di freeCodeCamp, di mana kami menyelesaikan atau mengerjakan tugas di halaman hingga 11 kasus untuk objek dan 4 untuk matriks..

8 II. TINJAUAN PUSTAKA

A. Object pada javascript

Nilai selain tipe data fundamental secara otomatis diperlakukan sebagai objek. Dalam Javascript, sebuah objek didefinisikan sebagai kumpulan properti yang dapat berubah, yang berarti bahwa itu adalah sekumpulan properti (properti) yang nilainya dapat berubah. Sebuah objek sebenarnya adalah variabel yang menyimpan nilai (properti) dan fungsi (metode). objek dalam struktur kalimat memiliki fungsi antara lain , yang membentuk prinsip dalam klausa predikat transitif; Jelaskan arti kalimat; dan merupakan kesatuan atau kesempurnaan roh.

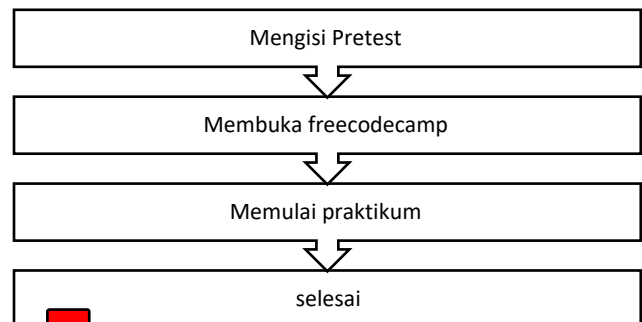
B. Matriks pada JavaScript

Array adalah objek JavaScript standar terbungkus (array) yang menyediakan fungsi berguna untuk manipulasi array sederhana, seperti subset B., ukuran, , ukuran, klon, dan lainnya. Matriks dapat dibuat dari array menggunakan fungsi math.matrix. Array yang disediakan dapat berisi array bersarang untuk membuat array multidimensi. Saat dipanggil tanpa argumen, array kosong dibuat.

Array dapat berisi nilai yang berbeda:

Bilangan, bilangan kompleks, satuan atau string karakter. Jenis yang berbeda dapat dicampur dalam satu matriks.

7 III. METODE PRAKTIKUM



Praktikum ini menganalisis dan memahami serta menyelesaikan tentang kode-kode javascript didalam situs web freecodecamp yang di akses menggunakan google chrome. Disini akan menyelesaikan 7 soal (kasus) pertanyaan function disana dengan menggunakan laptop yang terakses dengan internet setiap 1 soal memiliki 1 solusi dan solusi tersebut di download serta akan di cantumkan di laporan ini. Waktu pengerjaan selama 50 menit dikelas dan dijaga oleh dosen pengampu praktikum pemrograman 1.

12 IV. HASIL DAN ANALISIS

A. accessing-object-properties-with-bracket-notation.

```
// 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
```

B. accessing-nested-arrays

```
const myPlants = [
  {
    type: "flowers",
    list: [
      "rose",
      "tulip",
      "dandelion"
    ]
  },
  {
    type: "trees",
    list: [
      "fir",
      "pine",
      "birch"
    ]
  }
];
const secondTree = myPlants[1].list[1];
```

C. accessing-nested-objects

```
const myStorage = {
  "car": {
    "inside": {
      "glove box": "maps",
      "passenger seat": "crumbs"
    },
    "outside": {
      "trunk": "jack"
    }
  }
};
const gloveBoxContents = myStorage.car.inside["glove box"]
```

D. accessing-object-properties-with-dot-notation

```
// 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
```

E. manipulating-complex-objects

F. accessing-object-properties-with-variables

```
// 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
```

G. add-new-properties-to-a-javascript-object

```
const myDog = {
  "name": "Happy Coder",
  "legs": 4,
  "tails": 1,
  "friends": ["freeCodeCamp Campers"]
};
myDog.bark = 10;
```

H. build-javascript-objects

```
const myDog = {
  // Only change code below this line
  "name" : "iqbal",
  "legs" : 4,
  "tails" : 1,
  "friends" : ["aku", "kamu"]
  // Only change code above this line
};
```

I. delete-properties-from-a-javascript-object

```
// Setup
const myDog = {
  "name": "Happy Coder",
  "legs": 4,
  "tails": 1,
  "friends": ["freeCodeCamp Campers"],
  "bark": "woof"
};
// Only change code below this line
delete myDog.tails;
```

J. testing-objects-for-properties

```
function checkObj(obj, checkProp) {
  if (obj.hasOwnProperty(checkProp)) {
    return obj[checkProp];
  } else {
    return "Not Found";
  }
}
```

K. updating-object-properties

```
// Setup
```

```
const myDog = {
  "name": "Coder",
  "legs": 4,
  "tails": 1,
  "friends": ["freeCodeCamp Campers"]
};
```

// Only change code below this line

```
myDog.name = "Happy Coder";
```

L. using-objects-for-lookups

```
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;
```

3 V. KESIMPULAN

Fungsi dalam bahasa pemrograman adalah kode program yang dimaksudkan untuk melakukan tugas tertentu dan merupakan bagian dari program utama. Kita dapat membuat fungsi kita sendiri atau menggunakan fungsi yang dibuat oleh pengembang lain.[3]

Dari apa yang saya lakukan di FreeCodeCamp tentang fungsi dalam JavaScript, saya memahami cara kerja fungsi atau disebut juga fungsi dan saya mengerti cara bekerja dengan fungsi.

2 DAFTAR PUSTAKA

Buku Sakti Pemrograman Web: HTML, CSS, PHP, MySQL & Javascript Oleh Didik Setiawan

JavaScript: Dari A Sampai Z Oleh Vivian Siahaan, Rismon Hasiholan Sianipar

Dasar Pemrograman Internet dengan XHTML / CSS / Javascript / DHTML Oleh Rismon Hasiholan Sianipar