|  |  |
| --- | --- |
| **Materi** | **Nilai** |
| JavaScript Semester-4 Minggu ke-3 Video 21-30 | 95 |

**Saya Sudah Belajar dan Mengerti dan Saya BISA**

1. Array method pada bagian array map, array map ini yang mengambil bagian dari array dan outputnya juga array. Misalkan terdapat array yang menyimpan banyak object dan setiap object nya memiliki indeks dengan nama bertype string, dengan membuatkan variable yang diisi dengan nama array.map lalu ditambahkan (function (a){return a.nama *string indeks yang akan diambil*}) maka hasilnya dia akan mereturn isi indeks yang dipanggil dari array yang dituliskan dengan map nya, ketika di console.log(siswa) maka hasilnya akan muncul tulisan array(jumlah isi array) dengan panah yang jika dijabarkan akan memperlihatkan isi array nya. Ada cara lain untuk menggunakan array map seperti ini

let siswa = nilai.map(a => [a.nama, a.ipa, a.bahasa]);

yang berarti variable siswa diisi dengan array bernama nilai lalu diambil data indeks yang bertuliskan string nama, ipa, dan bahasa. Bedanya yang kedua tidak menggunakann return sehingga tampilan console yang pertama sudah ada tulisan isi hasil dari array map nya.

2. Array method pada bagian array sort(fn) yang berfungsi untuk mengurutkan abjad dari isi array yang dipilih. Misalkan ada array bernama abc = [b, c, a]; jika diurutkan menggunakan array sort(fn) maka hasilnya isi array menjadi a, b, c.

3. Array method pada bagian array reduce yang berfungsi untuk menjumlahkan isi atau element dari array yang bertype number agar bisa dijumlahkan. Contoh pertama

let hasil = nilai.reduce(function (a, b) {

    return (a = a + b.ipa);

}, 0);

console.log(hasil);

yang berarti variable hasil menyimpan array bernama nilai yang direduce lalu dibuatkan function bernama a dan b. isi funcionnya adalah mereturn a = a + b.ipa yang berarti a menyimpan a ditambah number yang diambil dari indeks bernama ipa lalu hasilnya diulanginya lagi ke a = a+b.ipa lalu ada ,0 yang berarti penjumlahan dimulai dari 0. Cara kedua yaitu

let hasil = nilai.reduce((a, b) => (a = a + b.ipa),0);

console.log(hasil);

sama seperti cara kesatu cara kerjanya, bedanya yang kedua tidak menggunakan return.

4. Project Shopping Cart, yang pertama adalah bagian menampilkan menu di dalam browsernya. Untuk menampilkan sebuah menu, yang pertama adalah membuatkan sebuah wadah di dalam file html nya seperti <div class=”produk”></div> setelah itu menyiapkan file style.css yang isinya hasil copas dari style.css pada took-online lalu menyiapkan folder images yang berisi gambar. Jika sudah, file html ini diarahkan ke dalam file javascript nya. di dalam file js yang pertama adalah diisi dengan sebuah variable yang menyimpan array dan array ini berisi banyak objek yang menyimpan data, jadi setiap menu disediakan 1 objek. Setelah itu membuat variable yang menyimpan array map seperti ini

let tampil = tblmenu.map(function (kolom) {

return `<div class="produk-content">

    <div class="image">

        <img src="images/${kolom.gambar}" alt="" />

    </div>

    <div class="title">

        <h2>${kolom.menu}</h2>

    </div>

    <div class="harga">

        <h2>Rp. ${kolom.harga}</h2>

    </div>

</div>`;

});

Variable nya bernama tampil yang menyimpan array map yang diambil dari array bernama tblmenu, setelah itu berisi function anonymous yang parameternya bernama kolom. Di dalam function ini mereturn sebuah wadah seperti div class produk-content yang diisi dengan image, title, dan harga. Untuk mengisi images yang diambil dari objek menggunakan ${kolom.gambar} karena nama parameternya kolom dan bagian indeks dalam objek yang mau diambil adalah gambar, gambar yang diambil adalah semua indeks gambar dari semua objek itu. Begitu juga ketika mengambil nama menu dan harganya menggunakan cara yang sama. Terakhir untuk mengisi wadah yang sudah dibuat dalam html dengan ini tadi menggunakan

let isi = document.querySelector(".produk");

isi.innerHTML = tampil;

wadah yang dipakai yaitu class produk (ada tanda titik didepan produk) lalu variable isi atau wadahnya tadi diisi dengan variable tampil.

5. Project Shopping Cart kedua yaitu mengambil idmenu nya menggunakan button, yang pertama yaitu mengganti div class produk berada di dalam div class content > div class main lalu menambahkan

<div class="btn-beli">

        <button data-idmenu="${kolom.idmenu}">Beli</button>

</div>

Ke dalam return dari variable tampil seperti div class sebelumnya. Lalu mengatur style css agar letaknya tepat didalam product content. Untuk mengaktifkan button atau membuat button bekerja menggunakan

let btnbeli = document.querySelectorAll(".btn-beli > button");

let cart = [];

for (let index = 0; index < btnbeli.length; index++) {

  btnbeli[index].onclick = function () {

    // console.log(btnbeli[index].dataset["idmenu"]);

    cart.push(btnbeli[index].dataset["idmenu"]);

  };

}

console.log(cart);

Yang berarti btnbeli untuk menyimpan data yang diambil dari class btn-beli bagian buttonnya, lalu ada cart yang berisi array untuk menyimpan data nanti, ada for yang berfungsi untuk melooping button yang ditekan dari browsernya (karena ada onclick sebagai tanda nya itu) lalu hasilnya akan dimasukkan ke dalam variable cart yang berfungsi menyimpan data menggunakan array. Terakhir console akan menampilkan data yang disimpan oleh cart yaitu button apa saja yang ditekan dan dengan button ini yang tampil adalah bagian idmenu dari produk nya.

6. Shopping Cart pada bagian memfilter, pada bagian cart.push(btnbeli[index].dataset["idmenu"]); diubah menjadi

tblmenu.filter(function (a) {

      if (a.idmenu == btnbeli[index].dataset["idmenu"]) {

        cart.push(a);

        console.log(cart);

      }

    });

Yang berarti memfilter variable bernama tblmenu lalu dibuatkan function dengan param a sebagai wadah untuk mengambil data dari tblmenu nya lalu diuji apabila a.idmenu sama dengan button yang diklik yang berarti mencocokkan dari idmenu yang diklik dengan idmenu yang ada di tblmenu lalu data dari produk beridmenu ini dimasukkan ke dalam cart. Terakhir ketika cart di console, maka tampilanya adalah data dari produk yang diklik, dan akan terlooping data nya sesuai dengan apa saja yang diklik dari buttonnya.

7. Menyimpan nilai ke dalam localstorage yang dibagi menjadi bagian key dan value nya menggunakan setItem, seperti ini localStorage.setItem(“coba”, 100); yang berarti key nya adalah coba dan value nya adalah 100.

8. Menampilkan localStorage dengan menggunakan getItem, contohnya seperti ini console.log(localStorage.getItem(“namakey”)); maka hasil pada console nya adalah value dari key yang dipanggil.

9. Menghapus item di localStorage menggunakan removeItem. Contohnya localStorage.removeItem(“namakey”); maka item yang memiliki key yang dipanggil akan hilang atau terhapus dari browser localStorage nya.

10. Menghapus semua item di localStorage dengan menggunakan clear, contohnya localStorage.clear(); maka semua item yang ada di localStorage akan terhapus.

11. Memanggil nama key menggunakan urutan indeksnya, misalkan console.log(localStorage.key(0)); maka yang tampil pada console nya adalah nama key yang ke-0.

12. Melihat jumlah item yang ada di localStorage nya dengan menggunakan length seperti ini contohnya console.log(localStorage.length); maka yang tampil pada console nya adalah jumlah dari item yang ada pada localStorage misalnya 3.

13. LocalStorage akan terus ada pada browsernya, meskipun browsernya ditutup dia akan tetap ada sampai ada perintah untuk menghapus localStorage nya. sessionStorage berbeda dengan localStorage, sessionStorage akan hilang ketika browsernya ditutup, jadi dia hanya ada ketika browsernya menyala.

14. Untuk bagian memanggil, membuat, menghapus item pada sessionStorage semuanya sama caranya dengan localStorage.

15. JSON adalah singkatan dari JavaScript Obejct Notation yang merupakan format data yang digunakan untuk pertukaran antar aplikasi, penulisannya mengikuti format javascript, mudah dibaca dan ditulis, sering digunakan pada AJAX dan bisa digunakan pada berbagai bahasa pemrograman.

16. Aturan penulisan pada JSON harus ditulis berpasangan antara KEY dan VALUE, key harus diapit pada petik dua (“KEY”) dan Value harus menunjukkan tipe data yang digunakan. Misal string atau number atau yang lainnya termasuk array dan object. Nama file mempunyai extensi.json

17. Untuk mengecek apakah code sebuah json valid atau tidak, bisa menggunakan jsonlint.com

18. Menggunakan Fetch Data, fetch ini berfungsi untuk mengambil url data, setelah diambil ada .then yang isinya untuk mengatur data ini akan dibentuk apa misalkan dibentuk json, lalu ada .then lagi untuk mengatur setelah data menjadi bentuk json akan diapakan misalkan akan ditampilkan ke dalam console.log contohnya seperti ini

fetch(url)

  .then(function (res) {

   return res.json();

  })

  .then(function (data) {

   console.log(data);

  });

Jadi urutannya adalah ambil url -> ubah data -> tampilkan

19. Menampilkan fetch ke dalam html. Contohnya

.then(function (data) {

   let output =

    "<h1>DATA MENU</h1><table><th>MENU</th><th>HARGA</th><th>WARNA</th>";

   data.forEach(element => {

    output += `<tr>

            <td>${element.menu}</td>

            <td>${element.harga}</td>

            <td>${element.warna[0]}</td>

            </tr>`;

   });

   output += "</table>";

   document.querySelector("#isi").innerHTML = output;

  });

Yang berarti pertama membuat sebuah function dengan parameter data dulu untuk menyimpannya, kemudian membuat variable bernama output yang berisi h1 dan open table juga th nya, setelah itu ada foreach untuk melooping data dari file jsonnya misalkan di dalam file json nya berisi 3 objek dan tiap objek memiliki element, nah element ini yang diambil untuk ditampilkan isinya cara memanggilnya yaitu dengan element => ada td karena dibentuk seperti table lalu ${element.menu} yang berarti memanggil element bernama menu sedangkan yang element warna ada [0] berarti dia bentuknya array dan diambil yang indeks ke 0 nya. setelah itu ditutup lagi dengan close table dan variable output ini akan dimasukkan ke dalam div yang memiliki id = isi untuk ditampilkan ke dalam htmlnya.

20. Async await hampir sama seperti fetch data yang berfungsi untuk mengambil url lalu mengubahnya dan menampilkan hasilnya, untuk penulisan pada async contohnya seperti ini

async function tampil() {

 const url = "https://jsonplaceholder.typicode.com/users";

 const res = await fetch(url);

 const data = await res.json();

 let out = "<ul>";

 data.forEach(element => {

  console.log(element.name);

  out += `<li>${element.name}</li>`;

 });

 out += "</ul>";

 document.querySelector("#isi").innerHTML = out;

}

jadi async Ini seperti membuatkan function dulu lalu diisi dengan const url yang menyimpan sebuah url lalu ada const res = await fetch(url); yang berarti await fetch(url) pada const res ini adalah sebagai tempat mengambil url dari const bernama url, lalu ada const data yang berisi await (.then) untuk diubah datanya menjadi json, dan untuk menampilkan hasilnya menggunakan variable out dengan melooping hasil dari const data nya.

**Saya Belum Mengerti**

1.