|  |  |
| --- | --- |
| Materi | Nilai |
| Javascript 21-30 #3 | 93 |

**Saya Sudah Belajar dan Mengerti dan Saya BIS**

1. Array.map fungsinya untuk membuat array baru dari array yg sudah ada, sesuai ayng di pilihnya.

let nilai=[

    {nama:"rizal",ipa:82,bahasa:90,mat:74},

    {nama:"joni",ipa:80,bahasa:70,mat:84},

    {nama:"tejo",ipa:76,bahasa:82,mat:90},

    {nama:"jina",ipa:90,bahasa:86,mat:94},

];

let siswa = nilai.map(function (a) {

    return a.nama

})

console.log(siswa);

cara pertama yaitu membuat nama array baru dengan menggunakan var agar array,map mengetahui nama array nya melalui nama var nya. Lalu map akan membuat array baru dari array nilai dengan penulisan nilai.map. lalu akan me return function yg ber parameter a.nama yg artinya map membuat array baru dengan memilih objek nama dari array nilai.

1. Cara ke 2 yaitu menggunkan arrow function dan mengambil 2 elemen dari objek a.

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

console.log(siswa);

artinya sama seperti penjelasan tertapi memakai arrow function. Dan jika ingin mengambil dari array nilai lebih dari 1 elemen maka harus di tambahkan [] tanda array agar tidak error, jadi nilai.map akan mengambil a.nama, a.ipa,a.mat, a.bahasa. menjadi array baru yg bernama siswa. Jika tidak ada [] maka akan error atau cuman mengambil 1 objek.

1. Array.sort digunakan untuk mengurutkan

mapel.sort();

console.log(mapel);

berawal dari

let mapel=["ipa","bahasa","mat"];

akan menjadi hasilnya adalah

mapel = [bahasa, ipa,mat];

1. Array reduce yang memiliki 2 params a dan b. A digunakan untuk menyimpan atau accumulator, dan params b digunakan untuk menentukan kolom yg akan di jumlahkan.

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

  return (a=a+b.ipa)

},0);

console.log(hasil);

artinya ia akan mengembalikan nilai a+b.ipa maka akan menjumlahkan semua hasil objek ipa yg ada pada array nilai. Dan 0 di luar kurung kurawa itu artinya dimulai dari 0.

1. Cara ke 2 yaitu menggunakan arrow function.

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

console.log(hasil);

artinya params a dan b akan di isi oleh a+=b.ipa yang berarti menambahkan seluruh hasil nilai ipa.

1. Memulai membuat shoping cart menggunakan program java, di dalam html,css dan img nya saya mengcopy dari bekas program html-css toko online yang dulu. Tetapi saya hanya menggunakan sebagian program saja yaitu class produk.
2. Pertama tama membuat tblmenu yang sama seperti database, meskipun masih belum terhubung ke database tetapi agar programnya mirip. Lalu membuat 5 data di dalam tbl menu menggunakan objek yg ada di dalam array dan mendeskribsikan kolom masing masing. (ini hanya 2 contoh)

let tblmenu=[

    {

        idmenu:10,

        idkategori:33,

        menu:"jeruk manis",

        gambar:"jeruk.jpg",

        harga:5000

    },

    {

        idmenu:11,

        idkategori:33,

        menu:"apel",

        gambar:"apel.jpg",

        harga:6000

    },

];

1. Lalu memindahkan isi class produk ke dalam var tampil yg memiliki function untuk mengambil data array nya dan menggunakan array.map untuk membuat array baru sesuai data yg di minta. Dan menambahkan fungsi backtik.

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

    return `<div class="produk-content">

    <div class="img">

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

    </div>

    <div class="title">

      <h2>${kolom.menu}</h2>

    </div>

    <div class="harga">

     ${kolom.harga}

    </div>`;

});

Artinya array tblmenu akan terbaca oleh array tampil dan mengerjakan function yg ber params kolom. Lalu me return isi dari class produk menggunakan backtik. Lalu agar array terbaca pada tipe backtiki ini maka digunakan ${kolom.gambar} agar dapat membaca isi kolom yang ada pada gambar yg ada 5 data. Lalu kolom.menu dan kolom.harga pun sama.

1. Untuk menampilkannya yaitu menggunakan

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

isi.innerHTML=tampil;

agar datanya muncul di html maka data dari file javascript ini akan mengirim ke class produk yang ada di html agar menerima data dari let tampil dari file javascript.

1. Membuat fungsi button agar dapat mengambil id menu dari masing2 produk. Membuat css btn-beli > button agar button memiliki bungkus den merapikannya.

<div class="btn-beli">

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

    </div>

Mendambahkan fungsi data-idmenu agar data yang ada di kolom.idmenu dapat dibacah oleh tombol. Lalu mendeklarasikan var baru yaitu btn beli yang isinya.

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

1. Membuat fungsi agar tombol dapat membaca idmenu dari produk masing2.

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

let cart=[];

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

    btnbeli[index].onclick=function () {

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

    }

};

console.log(cart);

artinya adalah, membuat looping agar data btnbeli yang terhubung dengan data dari kolom.idmenu.

<div class="btn-beli">

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

    </div>

Dapat di baca oleh looping nya, jadi dapat dilakukan karena terdapat indeks yang sesuai. Lalu jika memang sudah sesuai maka btnbeli[indeks] akan mengambil idmenu angka ke berapa yang user klik. semisal ia meng klik apel yang idmenu nya 11. Maka btnbeli akan mengambil indeks 11 (btnbeli[11]). Lalu akan menjalankan function yang isinya cart.push agar apa yang di klik masuk ke bagian array cart=[]. Lalu yang dimasukan adalah btnbeli[11].dataset[“idmenu”] yang artinya mengambil data idmenu 11 dan memasukannya ke array cart sebagai id menu.

1. Meggunakan fungsi array.filter dan sedikit mengubah pendeklarasian dan pemanggilannya.

tblmenu.filter(function (a) {

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

                cart.push(a);

                console.log(cart);

            } });

Filter mengambil dari tblmenu, lalu di dalamnya ada function ber pasams a yang akan menjalankan pengkondisian bila a.idmenu atau id dari produk sama dengan data idmenu asli dari tbl menu. Maka nilai a akan menambahkan ke array cart, misal 1 kali akan mengambil 1 data dari idmenu yg sesuai, bahkan 2 kali,a akan berhasil diambil dan di simpan ke array cart, yang tujuannya agar dapat mengisi array cart dengan objek pada tblmenu yang sesuai produk yang di beli.

1. Localstorage digunakan untuk menyimpan data key dan value ke penyimpanan lokal program dan dapat dibuka apa application pada inspect. Sifatnya permanen di dalam browser.

localStorage.setItem("key / nama",12);

console.log(localStorage.getItem("key / nama"));

berikut kodingan untuk menyimpan datanya.lalu untuk memanggilnya hanya perlu dengan localstorage.getitem dan di dalam kurungnya menuliskan key nya. Maka akan menghasilkan value nya atau hasilnnya. Set untuk mendeklarasikan dan get untuk mengambil / menampilkan.

1. Karena sifatnya permanen maka harus dihapus menggunakan.

localStorage.removeItem("lahir tgl");

agar localstorage hilang maka auser harus menghapusnya menggunakan codingan ini menggunakan nama key nya. Meskipun codingannya hilang, tetapi localstorage yang anda kirim ke application tidak akan bisa hilang kecuali di hapus menggunakan codingan tersebut.

1. Codingan ini digunakan untuk menghapus semua localstorage secara kelsuluruhan.

localStorage.clear();

1. Codingan ini digunakan untk memanggil nama key dari pendeklarasian localstorage pada indeks ke 1.

localStorage.setItem("key / nama","hasil / value");

localStorage.setItem("lahir tgl",12);

console.log(localStorage.key(1));

Artinya adalah iya akan menampilkan nama key dari indeks ke 1 pada application bukan pada codingan. Yaitu lahir tgl.

1. Fungsi ini digunakan untuk mengetahui berapakah jumlah key localstorage yang ada dalam application.

console.log(localStorage.length);

1. Untuk menggunakn kodingan session kita menggunakan session storage unutk menyimpan session. Perbedaan antara session storage dan localstorage adalah jika local ia akan permanen namun harus diapus menggunakan remove item. Jika session ia bisa hilang jika codingannya di hapus lalu di refresh halaman browsernya atau di exit dan juga dapat menggunakan removeitem.

sessionStorage.setItem("coba",100);

console.log(sessionStorage.getItem("coba"));

pendeklarasian dan pemanggilannya hampir sama seperti localstorage. Session ini jiga memiliki function function yang sama pada localstorage contohnya removeitem,clear,index,length. Jadi pembahasannya bisa lihat di atas, cuman perbedaannya adalah ini session dan itu local.

1. JSON adalah

* JavaScript Object Nation
* Format data yang digunakan untuk pertukaran antar aplikasi
* Penulisannya mengikuti format javascript
* Mudah dibaca dan ditulis
* Sering digunakan pada AJAX
* Bisa digunakan berbagai bahasa pemrograman

1. Atura penulisan JSON

* Harus ditulis berpasangan antara KEY dan VALUE
* KEY harus diapit oleh petik 2 (“KEY”)
* VALUE harus menunjukan tipe data yang digunakan. Misal string atau number atau yang lainnya
* Nama file mempunyai ekstensi.JSON

1. Membuat codingan json.

let tblmenu={

        "idmenu":10,

        "idkategori":33,

        "menu":"jeruk manis",

        "harga":5000,

        "warna":["merah","kuning","hijau"],

        "stok":true,

        "keterangan":null

}

Dapat di isi berbagai Tipe data seperti string,int,array,boolean, dll. Dan dapat di chek melalui situs jsonlint.com

1. Agar dapat memuat lebih dari 1 objek maka harus diperlukan array, karena dalam satu var harus ada 1 objek, tetapi jika var itu memiliki array, maka anda dapat menaruh objek berapapun itu dalam array.

let tblmenu=[

{

        "idmenu":10,

        "idkategori":33,

        "menu":"jeruk manis",

        "harga":5000,

        "warna":["merah","kuning","hijau"],

        "stok":true,

        "keterangan":null

},

{

        "idmenu":10,

        "idkategori":33,

        "menu":"jeruk manis",

        "harga":5000,

        "warna":["merah","kuning","hijau"],

        "stok":true,

        "keterangan":null

},

]

1. untuk menampilkannya bisa menggunakan

console.log(tblmenu);

sebagai file js, bukan file bentuk json.

1. Membuat file tblmenu.json yang diisi kodingan dari deklarasi d json.js tadi. Agar tersimpan dengan tipe file yang cocok yaitu .json
2. Membuat document.queryselector untuk mengisi id klik dan menambahkan add event listener agar jika di klik maka akan menampilkan key click dan menghasilksan value isi dari function tampil.

document.querySelector("#klik").addEventListener("click",tampil);

function tampil() {

    fetch('https://jsonplaceholder.typicode.com/todos/1')

    .then(response => response.json())

    .then(json => console.log(json))

    document.querySelector("#isi").innerHTML="<h1>Data Menu</h1>";

}

Codingan fetch di atas merupakan contoh oengambilan data secara AIP menggunakan JSON.

Yang ada pada web jsonPlaceholder. Maka hasilnya akan sama jika di klik tombolnya maka akan mengeluarkan **Data menu bersifat** <h1> dan di dalam consol keluar.

{userId: 1, id: 1, title: "delectus aut autem", completed: false}

completed: false

id: 1

title: "delectus aut autem"

userId: 1

\_\_proto\_\_: Object

1. Mengganti isi dari function tampil.

function tampil() {

    let url="js/tblmenu.json";

    fetch(url)

    .then(response => response.json())

    .then(function (data) {

        console.log(data);

    })

    document.querySelector("#isi").innerHTML="<h1>Data Menu</h1>";

}

Artinya yaitu saya membuat var baru yaitu url agar saat memanggil fetch lebih efektif. Lalu di dalam url terdapat link yang mengarah ke file js/tblmenu.json agar dapat terpanggil.lalu setelah fetch(url) sudah mengarah ke sana, kita memanggilnya menggunakan arrow function untuk baris pertama dan arrow anonymous untuk baris ke 2. Baris pertama digunakan agar file yang di panggil dijadikan data json. Lalu baris ke 2 menampilkan data json ini ke consol.log. jadi ada 3 proses yaitu ambil url, jika sudah mengarah maka ubah isi url menjadi data json, lalu tampilkan.

1. Mengubah function pemanggilan pada function baris ke 2 , untuk menampilkan data json

    .then(response => response.json())

    .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;

    })

Artinya function yang ber params data akan mengerjakan tugas berikut. Pertama, menyimpan let output yg isinya ada h1,tabel yang masih terbuka dan th. Lalu ia akan menjadikan params data yang menjadi wadah dari array yang ada di file tblmenu.json , me foreach dan menaruhnya ke wadah element secara 1 per 1 objek. Lalu akan mengerjakan arrowfunction yang isinya output += <tr> maksudnya agar let output yang tersimpan tadi, akan dikeluarkan dan terhubung oleh <tr> dan akan menjadi tabel. Isinya adalah <td> yang ter isi oleh element.menu yaitu isi dari bagian objek pada array tblmenu yang ada pada file tblmenu.json. lalu setelah itu memanggil output += lagi agar tabel nya tertutup oleh </table>. Dan lalu menampilkan nya ke halaman html dengan menambahkan queryselector ke #isi untuk memanggil data output menggunakan innerhtml. Hasilnya adalah jika di klik tombol tadi maka langsung akan menampilkan array dari tblmenu yang ada di file json yang tersusun seperti tabel.

1. Membuat file baru fetchdata.js lalu mengisi codingan Mengambil data dari url <http://jsonplaceholder.typicode.com/todos> yang memiliki banyak isi ke program.

document.querySelector("#klik").addEventListener("click",tampil);

function tampil() {

    let url="http://jsonplaceholder.typicode.com/todos";

    fetch(url)

    .then(res => res.json())

    .then(data => {

        // console.log(data)

        data.forEach(element => {

            console.log(element.title);

        });

    });

}

Artinya yaitu jika tombol sudah di klik maka akan menjalankan function tampil yang isinya untuk mengarahkan ke web url lalu mengubahnya isi url menjadi file json dan memanggil title dari isi array dalam url tersebut melalui data.foreach.

1. Jika ingin menampilkannya dalam dom/di halaman web, maka kita harus ubah dan menambahkan di .then bagian pemanggilan.

.then(data => {

        // console.log(data)

        out = "<ul>";

        data.forEach(element => {

            out += `<li>${element.title}<li>`

        });

        out+="</ul>";

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

    });

Codingan ini sama seperti codingan sepertni no 27,penjelasanya pun hampir sama, coman beda kode nya dan yang di panggil. Singkatnya var data sebagai wadah dari isi url tersebut lalu memforeach biar bisa di pecah lagi objeknya. Wadahnya yaitu element. Lalu untuk menampilkannya menggunakan document query yang mengisi id isi dan memanggil out. yang isinya adalah gabungan dari

   out = "<ul>"; , out += `<li>${element.title}<li>`, dan  out+="</ul>";

1. Async adalah sintaks khusus untuk bekerja dengan promise dengan cara yang lebih nyaman. Saya buat file async.js dan menambahkan fungsi dom event juga yang dari tombol #klik dan memanggil function tampil. Yang isinya mengarah ke url <http://jsonplaceholder.typicode.com/users>.

document.querySelector("#klik").addEventListener("click",tampil);

async function tampil() {

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

    const res= await fetch(url);

    const data = await res.json();

    console.log(data);

}

fetch menggunakan async ini dapat menyederhanakan penggunaan pemanggilan. Jadi setelah const data =await res.json(); dibawahnya langsung dapat mendeklarasikan apapun tidak perlu ada .then.

1. Untuk await secara harfiah menangguhkan eksekusi fungsi hingga promise diselesaikan, lalu melanjutkannya dengan hasil promise. Itu tidak memerlukan biaya sumber daya CPU, karena mesin JavaScript dapat melakukan pekerjaan lain untuk sementara: menjalankan skrip lain, menangani acara, dll.
2. Membuat program yang ditambahi async.

document.querySelector("#klik").addEventListener("click",tampil);

async function tampil() {

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

    const res= await fetch(url);

    const data = await res.json();

    let out="<ul>";

    data.forEach(element => {

        out+=`<li>${element.name}</li>`;

        console.log(element.name);

    });

    out +="</ul>";

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

}

Penjelasannya hampir sama pada penjelasan sebelumnya cuman ini menggunakan tipe async juga agar di sederhanakan.

**Saya Belum Mengerti**