**JAVASCRIPT LANJUTAN | 3.1 Arrow Function**

Apa sih Arrow function itu, sebelum nya kita liat dulu kode berikut ini :

function tampilPesan(*nama*) {

    alert('halo' + *nama*);

}

tampilPesan('Sandhika');

Nah disini kita mendeklarasikan function seperti bahasa” pemrograman lain ya, jadi kita deklarasikan dulu sebelum dipanggil gaada masalah ya, ( **function Declaration** ).

Nah di javascript itu kita juga bisa membuat function menggunakan yg nama nya **function expression,** yg dimana function nya itu di simpan dulu ke dalam sebuah variabel, gini :

let tampilPesan = function (*nama*) {

    alert('halo ' + *nama*);

}

tampilPesan('Keyla');

Jadi kita bikin variabel let atau const di isi dengan function tanpa nama atau anonymous function, jadi di buat nya dalam bentuk expression jadi ada 2 jenis function.

**ARROW FUNCTION**

‘Bentuk lain yang lebih ringkas dari function Expression’

**- MDN**

Nah jadi ini adalah cara lain untuk kita menuliskan sebuah function tapi menjadi lebih ringkas, kalo tadikan kita punya function Expression seperti itu, nah kalo kita ringkas kode tersebut jadi arrow function gini :

let tampilPesan = (*nama*) => {

    alert('halo ' + *nama*);

}

tampilPesan('Keyla');

Jadi kyword function nya hilang hanya ada kurung buka dan kurung tutup untuk menyimpan argument lalu tanda panah, walaupun tanda panah itu gabungan dari tanda sama dengan ( = ) dan lebih besar dari ( > ), isi nya sama bisa menggunakan kurung kurawal jadi simpel nya begitu untuk mengubah penulisan function Expression untuk lebih ringkas lagi, tapi nanti kita liat perbedaan nya ya.

**Membuat ARROW FUNCTION**

Sekarang saya sudah punya coding nya :

const tampilNama = function (*nama*) {

    return `halo ${*nama*}`;

}

console.log(tampilNama('Sandhika'));

Saya punya function Expression yg ada argument nama terus return menggunakan string literal, string literal nanti akan kita bahas diselanjutnya, Nah sekarang cara merubah expression itu menjadi arrow gimana.

Kita bikin variabel dulu tapi sekarang tidak perlu kyword function langsung aja kurung buka untuk menyimpan argument lalu tanda panah dan lebih besar dari =>, lalu kurung kurawal gini :

const tampilNama = (*nama*) => { return `halo ${*nama*}`; }

console.log(tampilNama('Doddy Ferdiansyah'));

Jadi kita bisa bikin lebih ringkas seperti itu, dan cara penulisan arrow function ini ada beberapa bentuk, itu cara nulis nya kalo cuma satu parameter, nah kalo misalnya parameter nya lebih dari satu kalian bisa tulis seperti ini :

const tampilNama = (*nama*, *waktu*) => {

    return `Selamat ${*waktu*}, ${*nama*}`;

}

console.log(tampilNama('Erik', 'Malam'));

Nah selanjutnya kalo parameternya satu seperti sebelum nya sebenernya kalian gaperlu pake kurung di bagian parameter nya, gini :

const tampilNama = *nama* => { return `halo ${*nama*}`; }

console.log(tampilNama('Doddy Ferdiansyah'));

Jadi nya sama aja, kalo satu ga perlu pake kurung kalo dua atau lebih parameternya harus pake kurung, lalu kalo misalnya isi dari function nya hanya return aja seperti di atas kalian bahkan gaperlu nulis return nya, dan bahkan gaperlu pake kurung kurawal nya lagi, gini :

const tampilNama = *nama* => `Halo ${*nama*}`;

console.log(tampilNama('Doddy Ferdiansyah'));

Sama aja hasilnya tapi penulisan nya lebih ringkas dan ini di sebut nya dengan implisit return jadi return nya ga usah di tulis secara implisit, jadi itu otomatis menjadi return, satu parameter yg isi nya return aja ga ada isi yg lain gitu.

Dan kalo misalnya tanpa parameter di function nya itu wajib menuliskan kurung buka dan tutup nya, begini :

const tampilNama = () => 'Hello World!';

console.log(tampilNama());

Jadi hasilnya tampil hello world gitu, jadi itulah **beragam cara membuat arrow function pada javascript.**

Nah sekarang pada kenyataan nya kalian ga akan membuat fungsi sesederhana itu, kita buat contoh lain dari arrow function misalnya menggunakan function map pada javascript map itu bagian dari array, untuk memetakan sebuah function kedalam array misalkan.

Contoh simpel nya misalkan saya punya sebuah array namany mahasiswa yg isi nya ada 3 orang mahasiswa, ceritanya kalian ingin menghitung jumlah huruf dari masing” nama mahasiswa nya, lalu nanti jumlah huruf nya di simpan ke dalam array lagi baru ditampilkan ke dalam console nya.

Nah kalo kalian lakukan itu dengan function biasa, kalian punya variabel jumlahHuruf misalnya lalu didalam nya kita ambil array nya kita petakkan kedalam sebuah function, nanti parameternya akan mengambil elemen nama, masing” dari array nya kita anggap sebagai nama lalu kita return nama.length buat mengetahui ada berapa panjang dari masing” element array nya gini :

let mahasiswa = ['Sandhika Galih', 'Doddy Ferdiansyah', 'Erik'];

let jumlahHuruf = mahasiswa.map(function (*nama*) {

    return *nama*.length;

});

console.log(jumlahHuruf);

Maka hasilnya akan ada array baru yg isi nya 14 itu jumlah dari sandhika galih, 17 jumlah dari doddy ferdiansyah, dan 4 jumlah dari Erik, dan ceritanya sebelum nya kita punya itu. Nah sekarang gimana kalo ingin menggunakan arrow function. Gini :

let mahasiswa = ['Sandhika Galih', 'Doddy Ferdiansyah', 'Erik'];

let jumlahHuruf = mahasiswa.map(*nama* => *nama*.length);

console.log(jumlahHuruf);

Nah sekarang menggunakan arrow function jadi lebih sangat ringkas, Nah kalo misal kalian mau mengembalikan nya dalam bentuk object bukan array lagi, nah berartikan kita punya variabel yg sama lalu kita petakan.

Nah kalo mau membuat object didalam arrow function itu kita harus pake kurung lagi baru kurung kurawal karena kan buat object pake kurung kurawal ya {}, nah sekarang didalam nya kita buat object nya, misalnya yg ingin saya kembalikan itu bukan Cuma panjang nya tapi nama nya juga.

Jadi saya mau bikin property nya nama di isi dengan nama yg di isi dari argument nya lalu saya bikin property jmlHuruf yg diambil dari nama.length, Jadi cerita nya yg dikembalikan berupa object, gini :

let mahasiswa = ['Sandhika Galih', 'Doddy Ferdiansyah', 'Erik'];

let jumlahHuruf = mahasiswa.map(*nama* => ({

    nama: *nama*,

    jmlHuruf: *nama*.length

}));

console.log(jumlahHuruf);

Nah jadi yg kembali bukan array melainkan object yg isi nya namanya siapa jmlHuruf nya berapa gitu. Dan ini kalo kalian mau tampil hasil di console nya lebih rapih kalian bisa ganti console.log nya jadi console.table, gini :

let mahasiswa = ['Sandhika Galih', 'Doddy Ferdiansyah', 'Erik'];

let jumlahHuruf = mahasiswa.map(*nama* => ({

    nama: *nama*,

    jmlHuruf: *nama*.length

}));

console.table(jumlahHuruf);

Dan ini kalo di javascript baru sekarang kalo kalian mau mengembalikan object yg property nya sama dengan nilai nya atau argument nya, kalian gaperlu tulis 22 nya kalian cukup tulis satu aja udah jalan gini :

let mahasiswa = ['Sandhika Galih', 'Doddy Ferdiansyah', 'Erik'];

let jumlahHuruf = mahasiswa.map(*nama* => ({ nama, jmlHuruf: *nama*.length }));

console.table(jumlahHuruf);

Nah gitu, jadi kalo bingung pas baca kodenya knp kok ada kurung buka dan tutup yg ini (), nah lal knp kok kurung kurawal nya ada didalam kurung biasa nah itu membuat object ya bukan kurung kurawal untuk function nya, karena disitu function nya gapake kurung kurawal dan tidak pake tulisan return juga jadi dia itu langsung karena hanya satu baris program jadi bisa.

Kaya kode ini, beda nya yg atas itu pake object kalo ini hanya simpel aja dan tidak memakai argument juga :

const tampilNama = () => 'Hello World!';

console.log(tampilNama());

Itu gambaran kode sederhananya biar ga bingung pas baca nya, kalo masih bingung baca lagi atau ulang video nya.