|  |  |
| --- | --- |
| Materi | Nilai |
| Pembelajaran Javascript Video ke 1 - 10  (Tugas semester 4 #1) | 95 |

1. Pembuatan program mobile dapat menggunakan 2 cara, bisa menggunakan javascript (pembelajaran dasar), react native.
2. Mendownload ekstensi codesnippets yaitu JavaScript (ES6) dan mendownload ekstensi code runner.
3. Membuat file index.html di dalam folder javascript.
4. Mengaktifkan live server menggunakan ekstensi live server lalu mengklik go live di bagian pojok kanan bawah, dan apabila tulisan go live berubah menjadi Port : 5500 artinya live server sudah menyala.

(Karena live server saya tidak mau membuka chrome, maka di dalam chrome saya tuliskan manual untuk menyalakan live server, yaitu angka 127.0.0.1:5500/index.html. setelah itu, otomatis live server akan menyala, dan ketika terdapat perubahan di dalam file, maka tampilan web akan otomatis berubah tanpa direfersh terlebih dahulu).

1. **Cara penulisan pertama dari javascript, internal,** yaitu dengan memasukkan isi ke dalam tag <script> di dalam <body>.

Cth :

<body>

    <script>

      alert("belajar javascript");

    </script>

  </body>

Artinya, di dalam tag <body> terdapat tag <script> digunakan untuk menulis script yang di dalamnya berisi alert() dan digunakan untuk menampilkan pesan informasi yang berisi tulisan “belajar javascript”.

1. Tag <script> digunakan untuk menulis script.
2. Jendela dialog merupakan jendela yang digunakan untuk berinteraksi dengan pengguna. Jendela dialog ada 3 macam, yaitu :

* Jendela dialog *alert();*
* Jendela dialog *confirm();*
* Jendela dialog *promp();*

1. Dialog alert() berfungsi untuk menampilkan sebuah pesan peringatan / informasi.

Cth :

alert("belajar javascript");

Artinya, terdapat alert yang digunakan untuk menampilkan sebuah informasi dengan tulisan “belajar javascript”. Jadi di dalam web ini akan dimunculkan jendela dialog alert() yang akan menampilkan pesan “belajar javascript”.

1. **Cara penulisan kedua dari javascript, file eksternal,** yaitu dengan menambahkan src yang digunakan untuk memanggil file javascript, penulisan src diletakkan di dalam tag <script> yang ada pada posisi atas dari </head>.

Cth :

Di dalam file index.html :

<script src="js/javascript.js"></script>

Artinya, di dalam tag <script> terdapat src yang digunakan untuk menentukan link yang memanggil file eksternal / file di luar dari file index.html, link yang dipanggil adalah js/javascript.js, berarti file javascript.js berada di dalam folder js.

1. Src digunakan untuk menentukan link yang merujuk pada sumber file script eksternal.
2. Apabila di dalam file index.html terdapat 2 tag <script> yaitu tag yang pertama dengan menggunakan src (eksternal) untuk memanggil file javascript dan tag yang kedua dengan memasukkan alert (internal) ke dalam tag <body> untuk menampilkan isi. Karena metode yang dipakai pertama adalah metode penulisan eksternal, maka jendela dialog yang ditampilkan pertama adalah tag <script> yang memanggil file javascript, selanjutnya jendela dialog kedua akan ditampilkan, yaitu tag <script> yang diletakkan di dalam tag <body>. Karena program dijalankan dari atas ke bawah.
3. Javascript selalu membutuhkan HTML untuk menampilkan ke dalam browser. Javascript harus ada di dalam halaman dari HTML karena javascript hanya bisa tampil di halaman HTML.
4. Cara penulisan variable di dalam javascript :

* Penulisan nama variable bisa menggunakan huruf, angka underscore, dan tanda dollar.
* Penulisan nama variable harus diawali dengan huruf.
* Penulisan nama variable di dalam javascript harus memperhtaikan huruf besar dan huruf kecil (case sensitive).
* Nama variable tidak boleh mengandung spasi.
* Awalan variabel diawali dengan let.

1. let digunakan untuk menetapkan variable pada javascript.
2. console.log() adalah fungsi yang digunakan untuk menampilkan teks ke console javascript.
3. Menampilkan variable menggunakan console.log();

Cth :

Di dalam file index.html :

<script src="js/variabel.js"></script>

Artinya, di dalam tag <script> terdapat src yang digunakan untuk menentukan link yang memangggil file eksternal / file variabel.js di dalam folder js yang berada di luar dari file index.html.

Di dalam file variabel.js :

let angka = 10;

console.log(angka);

artinya, terdapat variabel bernama angka yang mempunyai nilai 10, penulisan variabel menggunakan let untuk menetapkan variabel. Setelah itu, console.log(angka) digunakan untuk menampilkan teks ke console javascript dan teks yang ditampilkan adalah teks yang memanggil variabel angka.

1. const digunakan untuk mendeklarasikan variabel, tetapi bukan berarti variabel tetap, hanya variabel tidak bisa dideklarasikan ulang, baik keyword let cdan const / menginisialisasi ulang nilainya. Tetapi konten variabel nya masih bisa diubah.
2. Konstanta harus diisi ketika dideklarasikan, apabila tidak diisi maka akan menghasilkan error. Tetapi jika variabel itu bisa dideklarasikan dulu setelah itu diisi kemudian.

Cth :

let angka;

const data = 1000;

angka = 5;

console.log(angka);

console.log(data);

artinya, const harus diberi nilai setelah dideklarasikan, const dengan nama data mempunyai nilai 1000, dan ditampilkan menggunakan console.log(data). Lalu variabel angka hanya dideklarasikan saja, tetapi diisi setelah variabel dideklarasikan dan ditampilkan menggunakan console.log(angka).

1. Nilai dari konstanta tidak bisa diisi lagi, maka apabila nilai yang dideklarasikan pada kosntanta adalah 100, maka jika nama konstanta dipanggil lagi lalu diubah nilai nya menjadi 50, akan terjadi error.

Cth :

const data = 1000;

data = 2;

console.log(data);

Artinya, const dengan nama data berisi nilai 1000, dan nilai dari data diubah menjadi 2, maka apabila akan ditampilkan akan terjadi error, karena nilai dari konstanta itu tidak bisa diisi lagi.

1. Nilai dari variabel bisa diisi lagi / diganti setelah dideklarasikan, jadi apabila nilai dari variabel yang dideklarasikan adalah 50, dan jika nama konstanta dipanggil lagi lalu nilainya diubah menjadi 15, maka nilai nya akan menjadi 15 jika dipanggil.

Cth :

let angka = 100;

angka = 50;

console.log(angka);

Artinya, variabel dengan nama angka mempunyai nilai 100, dan nilai dari angka diisi kembali dengan nilai 50, maka jika variabel angka ditampilkan, maka akan menampilkan nilai 50.

1. typeof digunakan untuk mengecek tipe data dari sebuah variabel.
2. Mengecek tipe data dari sebuah variabel

Cth :

let isi = "jangan lupa senyum hehe";

console.log(typeof isi);

Artinya, variabel dengan nama isi mempunyai nilai tulisan jangan lupa senyum hehe, lalu mengecek tipe data pada variabel isi menggunakan typeof lalu menampilkannya ke dalam console.

1. Angka bulat dan pecahan jika dilihat tipe data nya, maka tipe data yang ditampilkan adalah number atau sama sama angka.
2. Infinity adalah hasil perhitungan matematika yang digunakan javascript untuk menampung nilai yang bukan angka / menghasilkan nilai tak terbatas. Hasil infinity akan didapat jika melakukan operasi :

* Jika sebuah angka melewati nilai maksimum angka yang bisa ditampung di dalam javascript / overflow.
* Jika sebuah angka lebih besar dari angka negatif yang bisa ditampung.
* Melakukan operasi aritmatika dengan infinity (a = 1+infinity).
* Melakukan operasi pembagian dengan nilai 0 (a= 1/0)

1. NaN (Not a number) adalah hasil perhitungan matematika yang digunakan javascript untuk menampung nilai yang bukan angka. Hasil Nan akan didapat jika melakukan operasi :

* Pembagian 0 dengan 0.
* Pembagian infinity dengan infinity.
* Akar kuadrat dari nilai negatif.
* Operasi aritmatika dengan nilai yang bukan angka.

1. Di dalam javascript, penulisan string dapat menggunakan 3 quote, yaitu :

* Double quotes : “Javascript”;

Apabila ingin menggabungkan kalimat dan variabel menggunakan double quotes, maka hanya dengan menambahkan tanda + di luar dari double quotes.

Cth :

console.log("variabel kata berisi : "+kata);

* Single quotes : ‘Javascript’;
* Backtiks : `Javascript`;

Backtiks memungkinkan kita menggabungkan beberapa variabel langsung, dan penulisan menggunakan backtiks : (`${nama variabel}`);

Cth :

console.log(`${angka} ${kata}`)

console.log(`hasil dari c adalah ${b/0}`);

1. Tipe data Boolean adalah tipe data yang hanya memounyai dua nilai, yaitu nilai benar (true) dan nilai salah (false).
2. Tipe data null berarti tipe data yang kosong / tidak memiliki nilai.
3. Tipe data undefined adalah tipe data yang belom diisi nilai hanya dideklarasikan nama variabelnya saja tetapi sudah digunakan.
4. Operand adalah nilai yang digunakan didalam proses operasi.
5. Operator matematika di dalam javascript ada 2, yaitu :

* Unary.
* Binary.

1. Operator unary adalah operator yang hanya memiliki 1 operand.

Cth :

let x = 2;

x = -x;

console.log(x);

Artinya, variabel tersebut memiliki operator unary, yaitu operator yang hanya memiliki satu operand, angka 5.

1. Operator binary adalah operator yang memiliki 2 operand.

Cth :

let p, l, t, y;

p = 12;

l = 10;

t = 6;

y = p\*l\*t;

console.log(y);

Artinya, terdapat beberapa variabel yaitu p, l, t, dan y. Variabel tersebut menggunakan aritmatika dengan operator binary. Variabel y berisi hasil operasi aritmatika dari p, l, dan t, dengan banyaknya operand yaitu 3, 12, 10, dan 6. Lalu variabel y ditampilkan menggunakan console.log(y).

1. Tanda yang digunakan untuk operator aritmatika di dalam javascript :

* + = Penjumlahan bilangan.
* - = Pengurangan bilangan.
* \* = Perkalian bilangan.
* / = Pembagian bilangan.
* % = Sisa pembagian.
* \*\* = Bilangan berpangkat.

1. String bisa digabungkan dengan angka menggunakan tanda +.

Cth :

console.log(13 + "5"); // hasil 135

console.log(10 + 6 + "5"); // hasil 165

console.log("5" + 1 + 6); // hasil 516

Artinya, pada console.log yang pertama berisi angka 13 digabungkan dengan 5 menjadi 135 bukan 18. Pada console.log kedua angka 10 ditambah 6 llau digabungkan dengan 5 menjadi 165 bukan 21. Lalu pada console.log ketiga, 5 adalah string digabungkan dengan 1 dan 6, karena yang depan adalah string, jadi otomatis angka belakang adalah string juga, maka hasil nya adalah 516.

1. Tanda + adalah satu – satunya tanda yang bisa menggabungkan string dengan angka, selain tanda plus, apabila terdapat string dan angka maka string akan diubah menjadi number dan akan melakukan operasi artimatika seperti biasa, jadi tidak bisa digabung.
2. Chaining assignments adalah metode yang digunakan untuk menggabungkan beberapa variabel yang mempunyai satu value. Penilaian pada assignments ini dilakukan dari kanan ke kiri, yaitu dari menghitung angka setelah itu variabel mengambil nilai.

Cth :

let e = f = g = 5 +5;

console.log(e, f, g);

Artinya, terdapat variabel e, f, g yang menyambung, jadi angka nya terlebih dahulu dijumlahkan yaitu 5 + 5, berarti variabel e, f, dan g bernilai sama, yaitu 10 dari hasil penjumlahan 5 +5.

1. Increment mempunyai tanda (++) berfungsi untuk menambah nilai variabel sebanyak 1.

Cth :

let inc = 9;

inc ++;

console.log(inc);

Artinya, variabel inc mempunyai nilai 9, lalu terdapat tanda increment pada variabel inc yang bernilai tambah 1, jadi hasil dari inc adalah 9 + 1 yaitu 10.

1. Decrement mempunyai tanda (--) berfungsi untuk mengurangi nilai variabel sebanyak 1.

Cth :

let inc = 14;

inc ++;

console.log(inc);

inc --;

console.log(inc);

Artinya, variabel inc mempunyai nilai 9, lalu terdapat tanda increment pada variabel inc yang bernilai tambah 1, jadi hasil dari inc adalah 9 + 1 yaitu 10. Setelah itu, terdapat tanda decrement pada variabel inc yang bernilai kurang 1, jadi hasil variabel inc adalah 14 lalu ditambah 1 karena operator increment lalu dikurang 1 karena operator decrement, jadi hasilnya adalah 14.

1. Di dalam operator comma, semua bagian dijumlahkan tetapi yang muncul sebagai hasil adalah bagian belakang saja.

Cth :

let h = (1 + 2, 3 + 4);

console.log(h);

Artinya, di dalam variabel h terdapat operator comma, pada bagian 1 yaitu 1 + 2 = 3 dan bagian 2 yaitu 3 + 4 = 7, semua bagian sudah dihitung, tetapi yang muncul sebagai hasil hanyalah bagian yang terakhir, yaitu bagian 3 + 4 = 7.

1. Operator logika hanya menguji sebuah nilai apakah nilai tersebut true / false. Di dalam operator logika terdapat beberapa operator, yaitu :

* AND ( & ) = Operator akan bernilai true jika semua kondisi bernilai true.
* OR ( | ) = Operator akan bernilai true jika salah satu kondisi bernilai true.

1. Operator pembanding, yaitu :

* Lebih besar dari ( > )
* Kurang dari ( < )
* Sama dengan ( == ) = sama isinya.
* Identik ( === ) = sama tipe data dan isinya.
* Tidak sama dengan ( != )
* Lebih besar sama dengan ( >= )
* Kurang dari sama dengan ( <= )

1. Percabangan if adalah percabangan yang memiliki satu pilihan ketika kondisi bernilai true / benar. Apabila sebuah kondisi bernilai false, maka tidak akan masuk ke dalam pilihan.

Cth :

*if* (true) {

    console.log("Dijalankan jika benar");

}*else*{

    console.log("dijalankan jika salah");

}

Artinya, terdapat percabangan if, jika percabangan bernilai true, maka akan menampilkan tulisan “dijalankan jika benar”, tetapi jika kondisi salah, maka akan menjalankan tulisan “dijalankan jika salah”.

1. Melakukan pengujian terhadap lulus atau tidak sesuai dengan nilai.

Cth :

let nilai = 55;

let lulus = 80;

let batasatas = 100;

let batasbawah = 0;

let peringatan = "NILAI SALAH";

let baik = "LULUS";

let kurang = "REMIDI";

*if* (nilai <= batasatas && nilai >= batasbawah) {

*if* (nilai >= lulus) {

        console.log(baik);

    } *else* {

        console.log(kurang);

    }

} *else* {

    console.log(peringatan);

}

Artinya, terdapat variabel nilai yang berisi 65, variabel lulus yang berisi 80, variabel batasatas yang berisi 100, variabel batasbawah yang berisi 0, variabel peringatan yang berisi NILAI SALAH, variabel baik yang berisi LULUS, variabel kurang yang berisi REMIDI. Terdapat pengujian yang menguji kondisi menggunakan operator logika, yaitu apabila variabel nilai lebih kecil dari variabel batasatas dan variabel nilai lebih besar dari variabel batas bawah , maka akan menjalankan percabangkan kedua, yaitu apabila variabel nilai lebih besar sama dengan variabel lulus maka akan menjalankan variabel baik yang berisi LULUS. Tetapi jika variabel nilai kurang dari variabel lulus, maka akan menjalankan variabel kurang yang berisi REMIDI. Dan jika variabel nilai berisi lebih besar dari variabel batasatas dan lebih kecil dari variabel batasbawah, maka akan menampilkan variabel peringatan yang berisi NILAI SALAH.

1. Switch case digunakan untuk memilih kondisi dimana terdapat banyak pilihan yang diberikan, dan hasil percabangan akan dijalankan oleh case.
2. Memilih kondisi dengan banyak pilihan.

Cth :

const hari = 3;

let hasil;

*switch* (hari) {

*case* 1:

        hasil = "MINGGU"

*break*;

*case* 2:

        hasil = "SENIN"

*break*;

*case* 3:

        hasil = "SELASA"

*break*;

*case* 4:

        hasil = "RABU"

*break*;

*case* 5:

        hasil = "KAMIS"

*break*;

*case* 6:

        hasil = "JUMAT"

*break*;

*case* 7:

        hasil = "SABTU"

*break*;

*default*:

        hasil = "HARI TIDAK DITEMUKAN"

*break*;

}

console.log(hasil);

Artinya, terdapat const dengan nama hari yang bernilai 3, dan terdapat variabel dengan nama hasil yang masih belom diisi. Lalu terdapat switch yang menguji kondisi variabel hari dengan beberapa case, yaitu, jika case bernilai 1, maka akan menampilkan variabel hasil yang berisi MINGGU. Jika case bernilai 2, maka akan menampilkan variabel hasil yang berisi SENIN. Jika case bernilai 3, maka akan menampilkan variabel hasil yang berisi SELASA. Jika case bernilai 4, maka akan menampilkan variabel hasil yang berisi RABU. Jika case bernilai 5, maka akan menampilkan variabel hasil yang berisi KAMIS. Jika case bernilai 6, maka akan menampilkan variabel hasil yang berisi JUMAT. Jika case bernilai 7, maka akan menampilkan variabel hasil yang berisi SABTU. Tetapi jika isi dari const hari tidak terdapat di dalam semua case, maka akan menjankan default yang berisi variabel hasil dengan isi HARI TIDAK DITEMUKAN. Setelah menemukan case yang sesuai dengan isi dari konstanta hari, maka program akan berhenti sesuai dengan hasil yang didapat. Setelah mendapatkan pilihan yang didapat, maka akan menampilkan variabel hasil sesuai dengan pilihan.

1. Contoh pengulangan menggunakan while.

Cth :

let mulai = 0;

let selesai = 10;

*while* (mulai < selesai) {

    console.log(mulai);

    mulai++;

}

Artinya, terdapat variabel mulai dengan nilai 0, dan variabel selesai dengan nilai 10. Lalu di dalam while berisi kondisi yaitu apabila mulai lebih kecil dari selesai, maka pengulangan untuk variabel mulai diulang sampai nilai nya kurang dari variabel selesai. Setelah itu, variabel mulai ditampilkan dan mulai mempunyai nilai ++, artinya nilainya akan selalu bertambah 1 pada setiap perulangan.

1. Contoh pengulangan menggunakan do while.

Cth :

let start = 110;

let finish = 115;

*do* {

    console.log(start);

    start++;

} *while* (start < finish);

Artinya, terdapat variabel start yang berisi 110 dan variabel finish yang berisi 115. Setelah itu terdapat do yang menampilkan start dan nilai start akan ditambah 1 ketika selesai mengulang. Lalu while berisi kondisi apabila variabel start lebih kecil dari variabel finish.

1. Contoh pengulangan menggunakan for.

Cth :

let awal = 1001;

let akhir = 1005;

*for* (awal; awal < akhir; awal++) {

    console.log(awal);

}

Artinya, terdapat variabel awal dengan nilai 1001 dan variabel akhir dengan nilai 1005. Setelah itu, pengulangan menggunakan for, dengan mulai dari variabel awal lalu memeriksa kondisi apakah variabel awal lebih kecil dari variabel akhir, jika iya maka dilakukan pengulangan dengan penambahan 1 disetiap pengulangannya. Lalu nilai awal ditampilkan menggunakan console.log.

1. Function digunakan untuk menyimpan coding yang dapat dikeluarkan ketika kita akan membutuhkannya dan agar dapat digunakan berkali - kali. Ciri – ciri dari function adalah terdapat tanda () setelah nama dari function. Dan untuk memanggil function hanya dengan memanggil nama function lalu diberi ().
2. Return digunakan untuk mengeluarkan nilai yang sudah diproses.
3. Contoh dari function :

function lingkaran(r) {

    let luas = 3.14 \* r \* r;

*return* luas;

}

console.log(lingkaran(10));

Artinya, function dengan nama lingkaran yang mempunyai parameter r, di dalam function terdapat variabel luas yang berisi luas lingkaran, yaitu 3.14 dikalikan r dikalikan r. setelah itu, nilai dari variabel luas akan dikeluarkan. Function lingkaran akan ditampilkan dan mengisi parameter r dengan nilai 10.

Apabila nilai sudah dikeluarkan tetapi jika ingin menambahkan operator matematika, bisa ditambahkan di samping dari hasil.

1. Untuk variabel di dalam parameter dan di dalam function tidak perlu menggunakan let, let hanya digunakan ketika variabel berada di luar.
2. Penulisan variabel yang di dalamnya terdapat function.

Cth :

let fungsi = function () {

    console.log("Belajar function");

}

fungsi();

Artinya, terdapat variabel yang berisi function, di dalam function terdapat console.log yang akan menampilkan tulisan belajr function. Lalu variabel fungsi dipanggil, karena di dalam variabel fungsi terdapat function, maka jika akan memanggil variabel fungsi harus menggunakan () agar dapat ditampilkan.

1. Arrow function adalah arrow yang didalamnya terdapat function dengan penulisan yang lebih singkat.
2. Contoh arrowfunction tanpa parameter :

let belajar = () => {

    console.log("baris pertama");

    console.log("baris kedua");

    console.log("baris ketiga");

    console.log("baris selanjutnya");

}

belajar();

Artinya, terdapat variabel belajar yang di dalamya berisi function tanpa parameter, dan function berisi consol.log yang digunakan untuk menampilkan tulisan - tulisan.

1. Contoh penulisan arrowfunction dengan satu parameter :

let hasil = (a) => (a \* 2);

console.log(hasil(3));

Artinya, terdapat variabel hasil dengan parameter a dan function tersebut berisi operasi aritmatika yaitu a \* 2, setelah itu function ditampilkan menggunakan console.log dengan memanggil nama dari variabel hasil dan mengisi parameter a dengan 3, maka operasi aritmatika adalah 3 \* 2 = 6.

1. Contoh penulisan arrowfunction dengan dua parameter :

let plus = (c, d) => c + d;

console.log(plus(3, 4));

Artinya, terdapat variabel plus dengan parameter c dan d yang mengisi parameter, dan function tersebut berisi operasi aritmatika yaitu c + d, setelah itu, variabel plus ditmapilkan menggunakan console.log() dengan memanggil variabel plus dan mengisi parameter dengan angka 3 dan 4, sehingga operasi aritmatika bisa dijalankan, yaitu 3 + 4 = 7.

1. Melakukan pengujian di dalam arrowfunction :

let nilai = 9;

let uji = (nilai < 8) ? () => predikat="MENGULANGI LAGI" : () => predikat="LULUS";

console.log(uji());

Artinya, terdapat variabel nilai yang berisi 9. Variabel uji yang di dalamnya terdapat kondisi apabila nilai kurang dari 8, maka akan menjalankan variabel predikat dengan tulisan MENGULANGI LAGI, dan jika nilai lebih besar dari 8 maka menjalankan variabel predikat dengan tulisan LULUS. Lalu akan ditampilkan menggunakan console.log() yang memanggil variabel uji().

Tanda ? merupakan if dan tanda : merupakan else jika di dalam arrowfunction.

**Saya Belum Mengerti**