



Jurusan Teknologi Informasi - Politeknik Negeri Malang

Jobsheet-3: Javascript

Mata Kuliah Desain dan Pemrograman Web

Pengampu: Tim Pengajar Desain dan Pemrograman Web

September 2024

Topik

- Pengenalan konsep dasar Javascript
- Tipe Data, Operator, Fungsi di dalam Javascript
- Javascript didalam HTML

Tujuan

Mahasiswa diharapkan dapat:

1. Memahami konsep Javascript
2. Memahami tipe data, operator, dan fungsi di Javascript
3. Mahasiswa mampu menjalankan Javascript pada HTML

Pendahuluan

JavaScript adalah bahasa pemrograman web yang bersifat *Client-Side Programming Language*. *Client-Side Programming Language* adalah tipe bahasa pemrograman yang pemrosesannya dilakukan oleh *client*. Aplikasi *client* yang dimaksud merujuk kepada *web browser* seperti **Google Chrome** dan **Mozilla Firefox**. Bahasa pemrograman *Client Side* berbeda dengan bahasa pemrograman *Server Side* seperti PHP, dimana untuk *server side* seluruh kode program dijalankan di sisi server.

Untuk menjalankan **JavaScript**, kita hanya membutuhkan aplikasi *text editor* dan *web browser*. **JavaScript** memiliki fitur: *high-level programming language, client-side, loosely typed* dan berorientasi objek. **JavaScript** pada awal perkembangannya berfungsi untuk membuat interaksi antara user dengan situs web menjadi lebih cepat tanpa harus menunggu pemrosesan di *web server*. Sebelum *JavaScript*, setiap interaksi dari user harus diproses oleh *web server*.

Bayangkan ketika kita mengisi *form registrasi* untuk pendaftaran sebuah situs web, lalu men-klik tombol *submit*, menunggu sekitar 20 detik untuk website memproses isian form tersebut, dan mendapati halaman yang menyatakan bahwa terdapat kolom form yang masih belum diisi. Untuk keperluan seperti inilah **JavaScript** dikembangkan. Pemrosesan untuk mengecek apakah seluruh form telah terisi atau tidak, bisa dipindahkan dari *web server* ke dalam *web browser*.

Dalam perkembangan selanjutnya, *JavaScript* tidak hanya berguna untuk *validasi form*, namun untuk berbagai keperluan yang lebih modern. Berbagai animasi untuk mempercantik

halaman web, fitur chatting, efek-efek modern, games, semuanya bisa dibuat menggunakan *JavaScript*. Ada 3 macam cara penulisan tag **javascript**, yaitu:

1. Menulis tag dengan `<script type="text/javascript">` di awal dan diakhiri dengan `</script>`. Atribut memberi tahu browser bahwa skrip dalam tag adalah JavaScript dalam format teks.
2. Menulis tag dengan `<script language="javascript">` di awal dan diakhiri dengan `</script>`. Atribut ini digunakan untuk menentukan versi JavaScript yang digunakan. Misalnya, `<script language="javascript1.2">` menunjukkan bahwa versi JavaScript yang digunakan adalah 1.2.
3. Menulis tag dengan `<script language="javascript" type="text/javascript">` di awal dan diakhiri dengan `</script>`. Metode campuran ini menggabungkan cara menulis lama dan baru, memungkinkan kompatibilitas untuk browser web yang mendukung JavaScript tetapi mungkin belum mendukung HTML sepenuhnya.

Praktikum Bagian 1 : Membuat Program Javascript

Langkah	Keterangan
1	Buat folder baru dengan nama minggu3. Pada folder tersebut, buat file baru bernama <code>hello_world.html</code>
2	<p>Ketikkan kode program di bawah ini :</p> <pre> <!DOCTYPE html> <html> <head> <title>Hello World Javascript</title> </head> <body> <script> console.log("Saya belajar Javascript"); document.write("Hello World!"); </script> </body> </html> </pre>
3	Simpan dengan nama <code>hello_world.html</code> , kemudian buka file tersebut dengan web browser.
4	Amati apa yang terjadi pada browser, kemudian catat hasil pengamatanmu (Pertanyaan No. 1)
5	Sekarang coba buka console javascript, kemudian lihat pada <code>Inspect Elements > Console</code>

6	Amati apa yang terjadi pada tab Console, kemudian catat hasil pengamatanmu! (Pertanyaan No. 2)
7	Tadi kita menuliskan perintah : <pre>console.log("Saya belajar Javascript");</pre> Menurut Anda, mengapa perintah tersebut tidak ditampilkan? (Pertanyaan No. 3)

Praktikum Bagian 2 : Cara Menulis Kode Javascript di HTML

Pada Praktikum Bagian 1, kita sudah menulis kode JavaScript di dalam HTML, cara tersebut merupakan cara penulisan *embedded* (ditempel). Beberapa cara lain yang perlu kita ketahui diantaranya:

1. *Embed* (Kode Javascript ditempel langsung pada HTML)
2. *Inline* (Kode Javascript ditulis pada atribut HTML)
3. *Eksternal* (Kode Javascript ditulis dengan terpisah dengan file HTML)

1. Penulisan Kode Javascript dengan Embed	
Langkah	Keterangan
1	Pada cara ini, kita menggunakan tag <code><script></code> untuk menempelkan (<i>embed</i>) kode Javascript pada HTML. Tag ini dapat ditulis didalam tag <code><head></code> dan <code><body></code>
2	Buat file <code>embed.html</code> . Ketikkan kode program di bawah ini : <pre> <!DOCTYPE html> <html> <head> <title>Belajar Javascript dari Nol</title> <script> // ini adalah penulisan kode javascript // di dalam tag <head> console.log("Hello JS dari Head"); </script> </head> <body> <p>Tutorial Javascript untuk Pemula</p> <script> // ini adalah penulisan kode javascript // di dalam tag <body> console.log("Hello JS dari body"); </script> </body> </html> </pre>

3	Amati apa yang terjadi pada browser? Catat hasil pengamatanmu (Pertanyaan No. 4)
4	Menurut Anda manakah yang lebih baik, dituliskan didalam tag <code><head></code> atau <code><body></code> ? (Pertanyaan No. 5)

2. Penulisan Kode Javascript <i>Inline</i>	
Langkah	Keterangan
1	Pada cara ini, kita akan menulis kode javascript di dalam atribut HTML. Cara ini biasanya digunakan untuk memanggil suatu fungsi pada event tertentu. Salah satu contohnya pada saat di-klik.
2	<p>Buatlah file <code>inline.html</code>. Ketikkan kode program di bawah ini :</p> <pre>Klik aku!</pre> <p>Atau bisa juga seperti ini :</p> <pre>Klik aku!</pre>
3	Amati apa yang terjadi pada browser! Catat hasil pengamatanmu (Pertanyaan No. 6)
4	Apa yang membedakan dari kedua kode program tersebut (Pertanyaan No. 7)

3. Penulisan Kode Javascript <i>Eksternal</i>	
Langkah	Keterangan
1	Pada cara ini, kita akan menulis kode javascript secara terpisah dengan file HTML. Cara ini biasanya digunakan pada proyek-proyek besar, karena diyakini dengan cara ini dapat lebih mudah mengelola kode project.
2	<p>Mari kita coba, buat dua file yaitu file HTML dan Javascript.</p> <pre>├── kode-program.js └── index.html</pre>
3	<p>Isi dari file <code>kode-program.js</code>:</p> <pre>alert("Hello, ini adalah program JS eksternal!");</pre>
4	Isi dari file <code>index.html</code> :

	<pre> <!DOCTYPE html> <html> <head> <title>Belajar Javascript dari Nol</title> </head> <body> <p>Tutorial Javascript untuk Pemula</p> <!-- Menyisipkan kode js eksternal --> <script src="kode-program.js"></script> </body> </html> </pre>
5	Amati apa yang terjadi pada browser! Catat hasil pengamatanmu (Pertanyaan No. 8)
6	<p>Pada percobaan tersebut, kita menulis kode javascript terpisah dengan kode HTML. Lalu pada kode HTML kita menyisipkan atribut src pada tag <script></p> <pre> <!-- Menyisipkan kode js eksternal --> <script src="kode-program.js"></script> </pre> <p>Maka apapun yang ada di dalam file kode-program.js akan dapat dibaca dari file index.html</p>
7	<p>Pindahkan file kode-program.js pada folder lain, apa yang akan terjadi jika file javascript berada di folder yang berbeda?</p> <p>Amati dan catat hasil pengamatanmu (Pertanyaan No. 9)</p>
8	<p>Misalkan kita memiliki struktur folder seperti ini:</p> <pre> ├── js/ │ └── kode-program.js └── index.html </pre> <p>Maka untuk menyisipkan file kode-program.js ke dalam HTML, kita bisa menuliskan kode berikut :</p> <pre> <script src="js/kode-program.js"></script> </pre>

Praktikum Bagian 3: Jendela Dialog

Jendela dialog merupakan jendela yang digunakan untuk berinteraksi dengan pengguna. Ada tiga macam jendela dialog pada Javascript:

1. Jendela dialog **alert()** ;
2. Jendela dialog **confirm()** ;

3. Jendela dialog `prompt()`;

Langkah	Keterangan
1	Buat File baru berana <code>alert_javascript.html</code>
2	Ketikkan kode program di bawah ini
	<pre> <html> <head> <script type="text/javascript"> function message() { alert("This alert box was called with the onload event") } </script> </head> <body onload="message()"> </body> </html> </pre>
3	Amati apa yang muncul pada browser
4	Catat hasil pengamatanmu (Pertanyaan No. 10)
5	Buat file baru bernama <code>confirm_javascript.html</code>
6	Ketikkan kode program di bawah ini
	<pre> <!DOCTYPE html> <html> <head> <title>Dialog Confirm</title> </head> <body> <script> var yakin = confirm("Apakah kamu yakin akan mengunjungi polinema?"); if (yakin) { window.location = "https://www.polinema.ac.id"; } else { document.write("Baiklah, tetap di sini saja ya :)"); } </script> </body> </html> </pre>
7	Amati apa yang muncul pada browser
8	Catat hasil pengamatanmu (Pertanyaan No. 11)
9	Ketikkan kode program di bawah ini

	<pre> <!DOCTYPE html> <html> <head> <title>Dialog Promp</title> </head> <body> <script> var nama = prompt("Siapa nama kamu?", ""); document.write("<p>Hello " + nama + "</p>"); </script> </body> </html> </pre>	
10	Amati apa yang muncul pada browser	
11	Catat hasil pengamatanmu (Pertanyaan No. 12)	

Praktikum Bagian 4: Variabel

Cara membuat variabel yang umum digunakan di JavaScript adalah menggunakan kata kunci `var` lalu diikuti dengan nama variabel dan nilainya.

Contoh: `var title = "Belajar Pemrograman Javascript";`

Menampilkan Isi Variabel

Untuk menampilkan isi variabel, kita bisa memanfaatkan fungsi-fungsi untuk menampilkan output seperti:

- Fungsi `console.log()` mengembalikan output ke konsol JavaScript;
- Fungsi `document.write()` mengembalikan output ke dokumen HTML;
- dan `alert()` mengembalikan output ke jendela dialog.

Langkah	Keterangan
1	Buat file baru bernama <code>variable_javascript.html</code>
2	Ketikkan kode program di bawah ini

	<pre> <!DOCTYPE html> <html lang="en"> <head> <title>Belajar Variabel dalam Javascript</title> <script> // membuat variabel var name = "Javascript"; var visitorCount = 50322; var isActive = true; // menampilkan variabel ke jendela dialog (alert) alert("Selamat datang di " + name); // menampilkan variabel ke dalam HTML document.write("Nama Situs: " + name + "
"); document.write("Jumlah Pengunjung: " + visitorCount + "
"); document.write("Status Aktif: " + isActive + "
"); </script> </head> <body> </body> </html> </pre>
3	Amati apa yang muncul pada browser
4	Catat hasil pengamatanmu (Pertanyaan No. 13)

Deleting Variables

Penghapusan variabel dalam Javascript memang jarang dilakukan. Namun, untuk program yang membutuhkan ketelitian dalam alokasi memori, penghapusan variabel perlu dilakukan agar penggunaan memori lebih optimal. Penghapusan variabel dapat dilakukan dengan kata kunci **delete**.

Contoh:

```

var bookTitle = "Learn Javascript Programming";
delete bookTitle;

```

Maka variabel bookTitle akan menghilang dari memori.

Praktikum 6 : FUNGSI

Fungsi adalah sub-program yang bisa digunakan kembali baik di dalam program itu sendiri, maupun di program yang lain. Fungsi di dalam Javascript adalah sebuah objek. Karena memiliki properti dan juga *method*.

Langkah	Keterangan
1	Buat File baru bernama <code>function_javascript.html</code>
2	Cara memanggil fungsi didalam kode Javascript biasanya ditulis dengan : <code>namaFungsi () ;</code>
3	Ketikkan kode program berikut ini
	<pre><!DOCTYPE html> <html> <head> <script> // membuat fungsi var sayHello = () => alert("Hello World!"); </script> </head> <body> <!-- Memanggil fungsi saat link diklik --> Klik Aku! </body> </html></pre>
4	Amati apa yang muncul pada browser
5	Catat hasil pengamatanmu (Pertanyaan No.14)
6	Parameter adalah variable yang menyimpan nilai untuk proses di dalam fungsi Cara memanggil parameter dalam javascript adalah:
	<pre>function kali(a, b){ hasilKali = a * b; console.log("Hasil kali a*b = " + hasilKali); }</pre>
7	Ketikkan kode program berikut ini

	<pre> <html> <head> <script type="text/javascript"> function total(numberA,numberB) { return numberA + numberB } </script> </head> <body> <script type="text/javascript"> document.write(total(2,3)) </script> </body> </html> </pre>
8	Amati apa yang muncul pada browser
9	Catat hasil pengamatanmu (Pertanyaan No.15)

Praktikum Bagian 6: Tipe Data

Tipe data adalah jenis-jenis data yang bisa kita simpan di dalam variabel. Ada beberapa tipe data dalam pemrograman Javascript:

- String (teks)
- Integer atau Number (bilangan bulat)
- Float (bilangan Pecahan)
- Boolean
- Object

Javascript adalah bahasa yang bersifat *dynamic typing*, artinya kita tidak harus menuliskan tipe data pada saat pembuatan variabel seperti pada bahasa [C](#), [C++](#), [Java](#), dsb. yang bersifat *static typing*.

Ada beberapa aturan penulisan variabel dalam Javascript:

- Penamaan variabel **tidak boleh** menggunakan angka di depannya.

Contoh:

```

// salah
var 123name = "Polinema";

// benar
var name123 = "Polinema";

```

- Penamaan variabel **boleh** menggunakan awal underscore.

Contoh:

```

var _nama = "Polinema";

```

- Penamaan variabel **dianjurkan** menggunakan **camelCase** apabila terdiri dari dua suku kata.

Contoh:

```
var _fullName = "Polinema";
```

- Penamaan variabel **dianjurkan** menggunakan bahasa inggris

Contoh:

```
var _postTitle = "Javascript Tutorials";
```

Langkah	Keterangan
1	Buat File baru bernama tipedata_javascript.html
2	<p>Ketikkan kode program berikut ini</p> <pre> <!DOCTYPE html> <html> <body> <h2>JavaScript Data Types</h2> <p>Contoh Javascript Data Types </p> <p id="demo"></p> <script> var x; // Now x is undefined x = 5; // Now x is a Number x = "John"; // Now x is a String document.getElementById("demo").innerHTML = x; </script> </body> </html> </pre>
3	Amati apa yang muncul pada browser
4	Catat hasil pengamatanmu (Pertanyaan No. 16)
5	Ketikkan program di bawah ini dan simpan dengan nama string_javascript.html

	<pre> <!DOCTYPE html> <html> <body> <h2>JavaScript Strings</h2> <p>Membuat Javascript String</p> <p id="demo"></p> <script> var answer1 = "It's alright"; var answer2 = "He is called 'Dilan'"; var answer3 = 'He is called "Dilan"'; document.getElementById("demo").innerHTML = answer1 + "
" + answer2 + "
" + answer3; </script> </body> </html> </pre>
6	Amati apa yang muncul pada browser
7	Catat hasil pengamatanmu (Pertanyaan No. 17)
8	<p>Ketikkan program di bawah ini dan simpan dengan nama <u>boolean_javascript.html</u></p> <pre> <!DOCTYPE html> <html> <body> <h2>JavaScript Booleans</h2> <p>Booleans hanya memiliki nilai true dan false</p> <p id="demo"></p> <script> var x = 5; var y = 5; var z = 6; document.getElementById("demo").innerHTML = (x == y) + "
" + (x == z); </script> </body> </html> </pre>
9	Amati apa yang muncul pada browser
10	Catat hasil pengamatanmu (Pertanyaan No. 18)

11	<p>Ketikkan program di bawah ini dan simpan dengan nama array_javascript.html</p> <pre> <!DOCTYPE html> <html> <body> <h2>JavaScript Arrays</h2> <p>Array</p> <p id="demo"></p> <script> var cars = ["Satu","Dua","Tiga"]; document.getElementById("demo").innerHTML = cars[0]; </script> </body> </html> </pre>
12	Amati apa yang muncul pada browser
13	<p>Catat hasil pengamatanmu (Pertanyaan No. 19)</p>

Praktikum Bagian 7: Operator

Operator adalah simbol yang digunakan untuk melakukan operasi pada suatu nilai dan variabel. Operator dalam pemrograman terbagi dalam 6 jenis:

1. Operator aritmatika;
2. Operator Penugasan (Assignment);
3. Opeartor relasi atau perbandingan;
4. Operator Logika;
5. Operator Bitwise;
6. Operator Ternary;

Operator aritmatika merupakan operator untuk melakukan operasi aritmatika seperti penjumlahan, pengurangan, pembagian, perkalian, dsb. Operator aritmatika terdiri dari:

Nama Operator	Simbol
Penjumlahan	+
Pengurangan	-
Perkalian	*
Pemangkatan	**
Pembagian	/
Sisa Bagi	%

Langkah	Keterangan
1	Buat File baru bernama <code>operator_javascript.html</code>
2	Ketikkan program di bawah ini <pre> <!DOCTYPE html> <html> <body> <h2>JavaScript Operators</h2> <p>x = 5, y = 2, menghitung z = x + y, dan tampil z:</p> <p id="demo"></p> <script> var x = 5; var y = 2; var z = x + y; document.getElementById("demo").innerHTML = z; </script> </body> </html> </pre>
3	Amati apa yang muncul pada browser
4	Catat hasil pengamatanmu (Pertanyaan No.20)

Praktikum Bagian 8: Percabangan

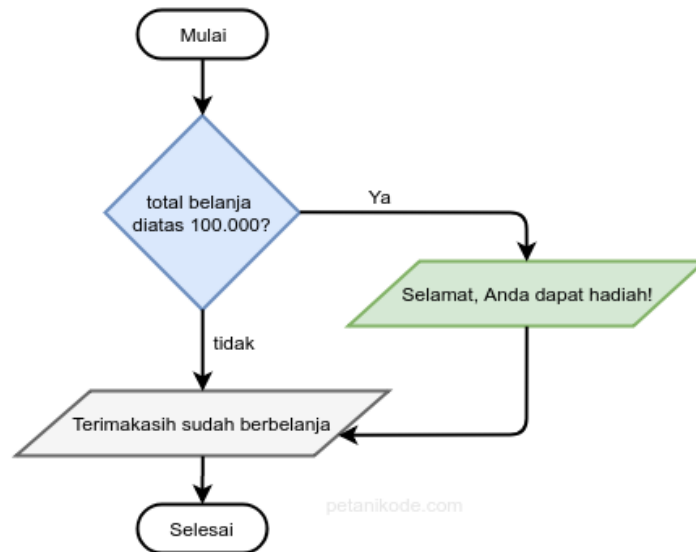
Dapat dikatakan bahwa percabangan dan *looping* merupakan salah satu inti metode dalam semua bahasa pemrograman yang ada di dunia, karena dengan percabangan dan *looping* akan dihasilkan sebuah program yang dinamis, dan bukan program yang linear serta bersifat statik. Karena JavaScript merupakan salah satu cara dalam melakukan pemrograman web di sisi client, maka JavaScript juga memiliki kemampuan ini.

Beberapa fungsi percabangan:

- Gunakan **if** untuk menentukan blok kode yang akan dieksekusi, jika kondisi yang ditentukan adalah true
- Gunakan **else** untuk menentukan blok kode yang akan dieksekusi, jika kondisi yang sama adalah false
- Gunakan **else if** untuk menentukan kondisi baru yang akan diuji, jika kondisi pertama adalah false
- Gunakan **switch** untuk menentukan banyak blok kode alternatif yang akan dieksekusi

Percabangan IF

Percabangan *if* merupakan percabangan yang hanya memiliki **satu blok pilihan** saat kondisi bernilai benar. Coba perhatikan *flowchart* berikut ini:

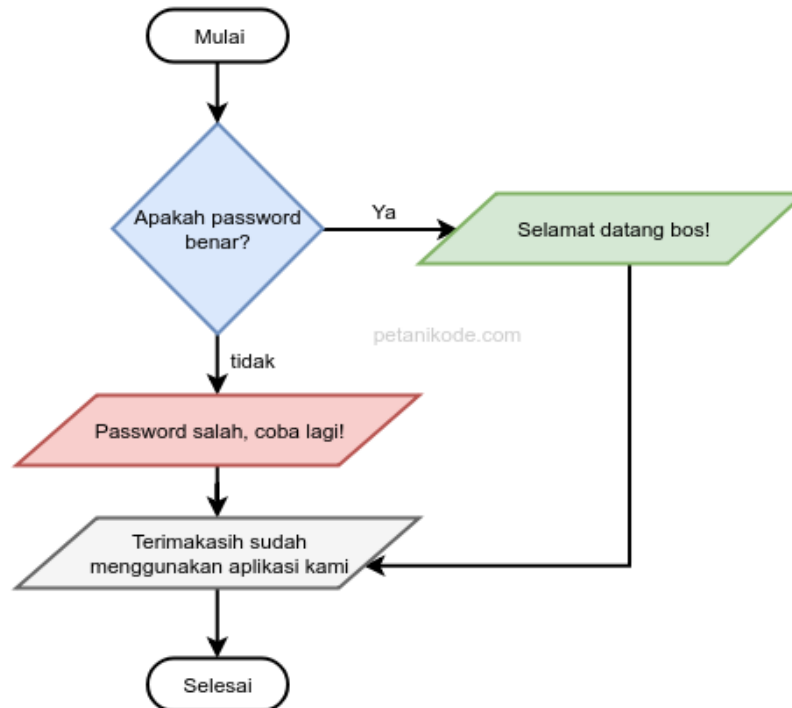


- Jika total belanja lebih besar dari Rp 100.000, maka tampilkan pesan “Selamat, Anda dapat hadiah!”.
- Jika di bawah Rp 100.000, maka pesan “Selamat, Anda dapat hadiah!” tidak ditampilkan.

Langkah	Keterangan
1	Buat File baru bernama if_javascript.html
2	Ketikkan program di bawah ini <pre><!DOCTYPE html> <html lang="en"> <head> <title>Percabangan if</title> </head> <body> <script> var totalBelanja = prompt("Total belanja?", 0); if(totalBelanja > 30000){ document.write("<h2>Selamat Anda dapat hadiah</h2>"); } document.write("<p>Terimakasih sudah berbelanja di toko kami</p>"); </script> </body> </html></pre>
3	Amati apa yang muncul pada browser
4	Catat hasil pengamatanmu (Pertanyaan No. 21)

Percabangan IF ELSE

Percabangan *if/else* merupakan percabangan yang memiliki **dua blok pilihan**. Pilihan pertama untuk kondisi **benar**, dan pilihan kedua untuk kondisi **salah** (*else*). Coba perhatikan flowchart ini:



Ini adalah flowchart untuk memeriksa password.

- Apabila password benar, pesan yang ada pada blok hijau akan ditampilkan: **“Selamat datang bos!”**
- Jika salah, maka pesan yang ada di blok merah yang akan ditampilkan: **“Password salah, coba lagi!”**

Langkah	Keterangan
1	Buat File baru bernama <code>ifelse_javascript.html</code>
2	Ketikkan program di bawah ini <pre><!DOCTYPE html> <html> <head> <title>Percabangan if/else</title> </head> <body> <script> var password = prompt("Password:");</pre>

	<pre> if(password == "teh"){ document.write("<h2>Selamat datang !</h2>"); } else { document.write("<p>Password salah, coba lagi!</p>"); } document.write("<p>Terima kasih sudah menggunakan aplikasi ini!</p>"); </script> </body> </html> </pre>
3	Amati apa yang muncul pada browser
4	Catat hasil pengamatanmu (Pertanyaan No. 22)

Percabangan SWITCH CASE

switch/case adalah bentuk alternatif dari struktur percabangan **if/else/if**. Dalam pernyataan **switch/case**, alih-alih mengevaluasi beberapa kondisi if, program mengevaluasi nilai variabel atau ekspresi dan membandingkannya dengan beberapa kemungkinan kasus. Setiap kasus mewakili nilai potensial, dan ketika kecocokan ditemukan, blok kode yang sesuai dieksekusi. Jika tidak ada kasus yang cocok, kasus **default** dieksekusi (jika disediakan), mirip dengan blok **else** dalam pernyataan **if/else**.

Struktur **switch/case** dapat membuat kode lebih mudah dibaca dan terorganisir, terutama saat berhadapan dengan beberapa kondisi berdasarkan satu variabel. Strukturnya terlihat seperti ini:

```

switch(variabel){
    case <value>:
        // blok kode
        break;
    case <value>:
        // blok kode
        break;
    default:
        // blok kode
}

```

Langkah	Keterangan
1	Buat File baru bernama switchcase_javascript.html
2	Ketikkan program di bawah ini

	<pre> <!DOCTYPE html> <html> <head> <title>Percabangan switch/case</title> </head> <body> <script> var jawab = prompt("Kamu beruntung! Silahakn pilih hadiahmu dengan memasukan angka 1 sampai 5"); var hadiah = ""; switch(jawab){ case "1": hadiah = "Tisu"; break; case "2": hadiah = "1 Kotak Kopi"; break; case "3": hadiah = "Sticker"; break; case "4": hadiah = "Minyak Goreng"; break; case "5": hadiah = "Uang Rp 50.000"; break; default: document.write("<p>Opps! anda salah pilih</p>"); } if(hadiah == ""){ document.write("<p>Kamu gagal mendapat hadiah</p>"); } else { document.write("<h2>Selamat kamu mendapatkan " + hadiah + "</h2>"); } </script> </body> </html> </pre>
3	Amati apa yang muncul pada browser
4	Catat hasil pengamatanmu (Pertanyaan No. 23)

Percabangan Bersarang

Percabangan Bersarang mengacu pada kondisi di mana satu pernyataan percabangan (seperti **jika**, **else**, **switch**, dll.) ditempatkan di dalam pernyataan percabangan lainnya. Hal ini memungkinkan proses pengambilan keputusan yang lebih kompleks di mana beberapa kondisi perlu dievaluasi pada tingkat yang berbeda. Dalam percabangan bersarang, hasil dari satu kondisi dapat bergantung pada hasil kondisi lainnya, memberikan kontrol yang lebih halus atas alur program.

Misalnya, Anda dapat menumpuk pernyataan **if** di dalam pernyataan **if** lain untuk terlebih dahulu memeriksa satu kondisi dan kemudian, berdasarkan itu, memeriksa kondisi kedua dalam blok pertama.

Langkah	Keterangan
1	Buat file baru bernama <code>nestedif_javascript.html</code>
2	Ketikkan program di bawah ini

	<pre> <!DOCTYPE html> <html> <head> <title>Percabangan Ternary</title> </head> <body> <script> var username = prompt("Username:"); var password = prompt("Password:"); if(username == "mahasiswa"){ if(password == "kopi"){ document.write("<h2>Selamat datang </h2>"); } else { document.write("<p>Password salah, coba lagi!</p>"); } } else { document.write("<p>Anda tidak terdaftar!</p>"); } </script> </body> </html> </pre>
3	Amati apa yang muncul pada browser
4	Catat hasil pengamatanmu (Pertanyaan No.24)

Praktikum Bagian 9: Perulangan

Perulangan akan membantu kita mengeksekusi kode yang berulang-ulang, berapapun yang kita mau. Ada lima macam bentuk perulangan di Javascript. Secara umum, perulangan ini dibagi dua yaitu: *counted loop* dan *uncounted loop*. Perbedaanannya adalah:

- **Counted Loop** merupakan perulangan yang jelas dan sudah tentu banyak perulangannya.
- **Uncounted Loop**, merupakan perulangan yang tidak jelas berapa kali ia harus mengulang.

Perulangan yang termasuk dalam *Counted Loop*:

1. Perulangan For
2. Perulangan Foreach
3. Perulangan Repeat

Perulangan yang termasuk dalam *Uncounted Loop*:

1. Perulangan While
2. Perulangan Do/While

Perulangan FOR

Perulangan for merupakan perulangan yang termasuk dalam *counted loop*, karena sudah jelas berapa kali ia akan mengulang. Bentuknya seperti ini:

```
for(let i = 0; i < 10; i++){  
    document.write("<p>Perulangan ke-" + i + "</p>")  
}
```

Langkah	Keterangan
1	Buat file baru bernama <code>for_javascript.html</code>
2	Ketikkan program di bawah ini <pre><!DOCTYPE html> <html> <body> <h2>JavaScript Loops</h2> <p id="demo"></p> <script> var text = ""; var i; for (i = 0; i < 5; i++) { text += "The number is " + i + "
"; } document.getElementById("demo").innerHTML = text; </script> </body> </html></pre>
3	Amati apa yang muncul pada browser
4	Catat hasil pengamatanmu (Pertanyaan No. 25)

Perulangan WHILE

Perulangan while merupakan perulangan yang termasuk dalam perulangan uncounted loop. Perulangan while juga dapat menjadi perulangan yang counted loop dengan memberikan counter di dalamnya.

Langkah	Keterangan
1	Buat File baru bernama <code>while_javascript.html</code>
2	Ketikkan program di bawah ini

	<pre> <!DOCTYPE html> <html> <body> <h2>JavaScript while</h2> <p id="demo"></p> <script> var text = ""; var i = 0; while (i < 10) { text += "
The number is " + i; i++; } document.getElementById("demo").innerHTML = text; </script> </body> </html> </pre>
3	Amati apa yang muncul pada browser
4	Catat hasil pengamatanmu (Pertanyaan No. 26)

Perulangan DO WHILE

Perulangan do/while adalah variasi dari perulangan while di JavaScript. Perbedaan utama di antara keduanya adalah bahwa perulangan do/while akan selalu mengeksekusi kode di dalam perulangan **setidaknya sekali**, terlepas dari apakah kondisinya benar atau salah. Ini karena kondisi dievaluasi **setelah** blok kode dieksekusi, bukan sebelumnya, seperti dalam perulangan while standar:

```

do {
    // blok kode yang akan diulang
} while (<kondisi>);

```

Karakteristik utama:

- Kode di dalam blok do berjalan terlebih dahulu, lalu kondisinya diperiksa.
- Jika kondisinya benar, perulangan berulang; Jika false, loop berhenti.

Jenis perulangan ini memastikan bahwa kode di dalam perulangan dieksekusi setidaknya sekali, bahkan jika kondisinya salah sejak awal.

Langkah	Keterangan
1	Buat file baru bernama dowhile_javascript.html
2	Ketikkan program di bawah ini

	<pre> <!DOCTYPE html> <html> <body> <h2>JavaScript do ... while</h2> <p id="demo"></p> <script> var text = "" var i = 0; do { text += "
The number is " + i; i++; } while (i < 10); document.getElementById("demo").innerHTML = text; </script> </body> </html> </pre>
3	Amati apa yang muncul pada browser
4	Catat hasil pengamatanmu (Pertanyaan No.27)

Referensi:

- 1) Jason Beaird, The principles of Beautiful Web Design
- 2) Rian Ariona, Belajar HTML dan CSS (Tutorial Fundamental dalam mempelajari HTML dan CSS)
- 3) Adi Hadisaputra, HTML dan CSS Fundamental dari Akar menuju Daun
- 4) John Duckett, HTML and CSS design and build websites