

JAVASCRIPT

For

EDUCATOR

Revised Version

Femi Dena Juang

femijuang@gmail.com

2020

KATA PENGANTAR

Alhamdulillah, puji dan syukur penulis panjatkan kehadiran Allah swt yang telah memberikan segala rahmat, taufik, hidayah, nikmat dan karunia-Nya, sehingga penulis dapat menyelesaikan karya yang berjudul “Javascript Bagi Pendidik”. Shalawat beserta salam semoga senantiasa tercurahkan kepada junjungan kita Nabi Muhammad SAW beserta keluarganya, para sahabatnya, dan para pengikutnya hingga akhir zaman.

Buku sederhana ini hadir untuk mempermudah proses PBM pada mata kuliah Web Programming. Dalam penyusunan buku ini, penulis menyadari sepenuhnya bahwa masih terdapat banyak kekurangan dan keterbatasan ilmu pengetahuan yang penulis miliki, Namun berkat adanya dorongan dan bantuan dari berbagai pihak, akhirnya karya sederhana ini dapat terselesaikan

Semoga buku sederhana inii dapat bermanfaat bagi penulis khususnya dan bagi para pembaca pada umumnya. Apabila terdapat kekurangan dan kesalahan adalah semata-mata keterbatasan ilmu yang penulis miliki. Apabila terdapat kesempurnaan itu berasal dari Allah SWT.

Bandung, 10 Januari 2019

PENULIS

KATA PENGANTAR Edisi Revisi

Alhamdulillah, puji dan syukur penulis panjatkan kehadiran Allah swt yang telah memberikan segala rahmat, taufik, hidayah, nikmat dan karunia-Nya, sehingga penulis dapat menyelesaikan karya yang berjudul “Javascript For Educator”. Shalawat beserta salam semoga senantiasa tercurahkan kepada junjungan kita Nabi Muhammad SAW beserta keluarganya, para sahabatnya, dan para pengikutnya hingga akhir zaman.

Buku edisi revisi mengandung beberapa penyesuaian kosa kata agar lebih mudah dipahami pembaca. Koreksi beberapa kesalahan ketik. Perbaiki layout pada banyak bab agar terlihat lebih simpel namun efisien. Terdapat koreksi atas syntax sumber code di beberapa lokasi.

Semoga buku sederhana ini dapat bermanfaat bagi penulis khususnya dan bagi para pembaca pada umumnya. Apabila terdapat kekurangan dan kesalahan adalah semata-mata keterbatasan ilmu yang penulis miliki. Apabila terdapat kesempurnaan itu berasal dari Allah SWT.

Bandung, 17 Mei 2020

PENULIS

DAFTAR ISI

KATA PENGANTAR.....	1
KATA PENGANTAR Edisi Revisi.....	1
DAFTAR ISI.....	2
PERTEMUAN 1 Introduction.....	1
PERTEMUAN 2 Grammar & Types.....	6
PERTEMUAN 3 Arrays.....	10
PERTEMUAN 4 Conditional statements.....	14
PERTEMUAN 5 Loops and Iteration.....	33
PERTEMUAN 6 Function.....	39
PERTEMUAN 7 Sound Effect.....	45
PERTEMUAN 8 UTS.....	47
PERTEMUAN 9 Strings Methods.....	48
PERTEMUAN 10 Objek Document.....	50
PERTEMUAN 11 window.open method.....	53
PERTEMUAN 12 Event.....	54
PERTEMUAN 13 Event 2.....	58
PERTEMUAN 14 Event 3.....	60
PERTEMUAN 15 Latihan & Praktek.....	63
PERTEMUAN 16 UAS Index Juara Mahasiswa.....	69
PERTEMUAN 17 Manipulasi Gambar.....	71
PERTEMUAN 18 Array level 2.....	75
PERTEMUAN 19 Function + validation lv2.....	78
PERTEMUAN 20 self invoking function & OOP:inheritance (prototypal).....	80
PERTEMUAN 21 loop lv.2.....	82
PERTEMUAN 22 Data type conversion & manipulation.....	84
PERTEMUAN 23 Controlling the length of decimals.....	86

PERTEMUAN 24 UTS.....	88
PERTEMUAN 25 Latihan menunjukan waktu dengan function.....	89
PERTEMUAN 26 Object & Object Constructor.....	91
PERTEMUAN 27 try and catch.....	94
PERTEMUAN 28 Prompt the name (inputs).....	97
PERTEMUAN 29 Create element on Click !.....	99
PERTEMUAN 30 kombinasi alert dengan prompt.....	101
PERTEMUAN 31 Final Lesson, Materi pemanasan Game tebak angka.....	105
PERTEMUAN 32 UAS.....	108
REFERENSI.....	109

PERTEMUAN 1 Introduction

JavaScript distandarisasi di Ecma International - asosiasi Eropa untuk membakukan sistem informasi dan komunikasi (ECMA sebelumnya adalah akronim untuk Asosiasi Produsen Komputer Eropa) untuk memberikan bahasa pemrograman internasional berstandar internasional berdasarkan JavaScript.

Web Console sangat bagus untuk mengeksekusi satu baris JavaScript, tetapi meskipun Anda dapat mengeksekusi beberapa baris, itu tidak terlalu nyaman untuk itu, dan Anda tidak dapat menyimpan sampel kode Anda menggunakan Web Console. Jadi untuk contoh yang lebih kompleks, Scratchpad adalah alat yang lebih baik.

JavaScript dibuat pada tahun 1995 oleh Brendan Eich ketika ia adalah seorang insinyur di Netscape. JavaScript pertama kali dirilis dengan Netscape 2 pada awal tahun 1996. Pada awalnya akan disebut LiveScript, tetapi namanya diubah menjadi keputusan pemasaran yang bernasib buruk yang berusaha memanfaatkan popularitas bahasa Jawa Sun Microsystem - meskipun keduanya memiliki sangat sedikit bersama. Ini telah menjadi sumber kebingungan sejak itu.

Beberapa bulan kemudian, Microsoft merilis JScript dengan Internet Explorer 3. Itu adalah JavaScript yang sebagian besar kompatibel. Beberapa bulan setelah itu, Netscape mengirimkan JavaScript ke Ecma International, sebuah organisasi standar Eropa, yang menghasilkan edisi pertama dari standar ECMAScript tahun itu. Standar menerima pembaruan yang signifikan sebagai ECMAScript edisi 3 pada tahun 1999, dan itu telah tetap cukup stabil sejak saat itu. Edisi keempat ditinggalkan, karena perbedaan politik mengenai kompleksitas bahasa. Banyak bagian dari edisi keempat membentuk dasar untuk ECMAScript edisi 5, yang diterbitkan pada bulan Desember 2009, dan untuk edisi utama standar ke-6, yang diterbitkan pada bulan Juni 2015.

Tidak seperti kebanyakan bahasa pemrograman, bahasa JavaScript tidak memiliki konsep input atau output. Ini dirancang untuk dijalankan sebagai bahasa scripting di lingkungan host, dan tergantung pada lingkungan host untuk menyediakan mekanisme untuk berkomunikasi dengan dunia luar. Lingkungan host yang paling umum adalah browser,

tetapi penerjemah JavaScript juga dapat ditemukan di daftar besar tempat lain, termasuk Adobe Acrobat, Adobe Photoshop, gambar SVG, mesin Widget Yahoo, lingkungan sisi server seperti Node.js, database NoSQL seperti open source Apache CouchDB, komputer yang disematkan, lingkungan desktop yang lengkap seperti GNOME (salah satu GUI paling populer untuk sistem operasi GNU / Linux), dan lainnya.

contoh:

ketik dan jalankan script berikut pada scratchpad!

```
(function(){  
    "use strict";  
  
    var nama = "Chicken";  
    function sapaAku () {  
        alert('Yuhuu ' + nama);  
    }  
    sapaAku();  
  
})();
```

OPERATORS in JS

JavaScript memiliki jenis-jenis operator berikut. Bagian ini menjelaskan tentang operator dan berisi informasi tentang prioritas operator.

- ✓ Assignment operators
- ✓ Comparison operators
- ✓ Arithmetic operators
- ✓ Bitwise operators
- ✓ Logical operators
- ✓ String operators
- ✓ Conditional (ternary) operator

- ✓ Comma operator
- ✓ Unary operators
- ✓ Relational operators

contoh penggunaan operator tipe binary dan penyebutan dalam js:

operand1 operator operand2

For example, 3+4 or x*y.

Operator unary membutuhkan operan tunggal, baik sebelum atau sesudah operator

:

operator operand

or

operand operator

For example, x++ or ++x

Lihat penjelasan lebih lanjut berikut!

A.Compound assignment operators

Nama:	shorthand operator:	Meaning:
Assignment	x = y	x = y
Addition assignment	x += y	x = x + y
Subtraction assignment	x -= y	x = x - y
Multiplication assignment	x *= y	x = x * y
Division assignment	x /= y	x = x / y
Remainder assignment	x %= y	x = x % y
Exponentiation assignment	x **= y	x = x ** y
Left shift assignment	x <<= y	x = x << y
Right shift assignment	x >>= y	x = x >> y
Unsigned right shift assignment	x >>>= y	x = x >>> y
Bitwise AND assignment	x &= y	x = x & y

Bitwise XOR assignment	$x \wedge= y$	$x = x \wedge y$
Bitwise OR assignment	$x \mid= y$	$x = x \mid y$

latihan!

kerjakan pada web console:

-jumlahkan var a dan var b

-kalikan dan jumlahkan var a dan var b , contoh $var\ c = 2 * a + b$; lalu perhatikan bagian mana yg dikerjakan javascript!

B.Comparison operators

Operator:	Description:
Equal (==)	Returns true if the operands are equal.
Not equal (!=)	Returns true if the operands are not equal.
Strict equal (===)	Returns true if the operands are equal and of the same type.
Strict not equal (!==)	Returns true if the operands are of the same type but not equal, or are of different type.
Greater than (>)	Returns true if the left operand is greater than the right operand.
Greater than or equal (>=)	Returns true if the left operand is greater than or equal to the right operand.
Less than (<)	Returns true if the left operand is less than the right operand.
Less than or equal (<=)	Returns true if the left operand is less than or equal to the right operand.

Note: (=>) is not an operator, but the notation for Arrow functions.

C.Arithmetic operators

Operator:	Description & Example:
Remainder (%)	Binary operator. Returns the integer remainder of dividing the two operands. $12 \% 5$ returns 2.
Increment (++)	Unary operator. Adds one to its operand. If used as a prefix operator (++x), returns the value of its operand after adding one; if used as a

postfix operator (x++), returns the value of its operand before adding one. If x is 3, then ++x sets x to 4 and returns 4, whereas x++ returns 3 and, only then, sets x to 4.

Decrement (--) Unary operator. Subtracts one from its operand. The return value is analogous to that for the increment operator. If x is 3, then --x sets x to 2 and returns 2, whereas x-- returns 3 and, only then, sets x to 2.

Unary negation (-) Unary operator. Returns the negation of its operand. If x is 3, then -x returns -3.

Unary plus (+) Unary operator. Attempts to convert the operand to a number, if it is not already. + "3" returns 3. +true returns 1.

Exponentiation operator ()** Calculates the base to the exponent power, that is, base^{exponent}

D.Bitwise operators

E.Logical operators

Operator	Usage & Description
Logical AND (&&)	expr1 && expr2 ;Returns expr1 if it can be converted to false; otherwise, returns expr2. Thus, when used with Boolean values, && returns true if both operands are true; otherwise, returns false.
Logical OR ()	expr1 expr2 ;Returns expr1 if it can be converted to true; otherwise, returns expr2. Thus, when used with Boolean values, returns true if either operand is true; if both are false, returns false.
Logical NOT (!)	!expr Returns false if its single operand that can be converted to true; otherwise, returns true.

PERTEMUAN 2 Grammar & Types

1. Basic

Javascript bersifat case-sensitive, sehingga konten seperti misalnya:

Jengkol dan jengkol adalah dua entitas yang berbeda.

`var Jengkol` vs `var jengkol` is not the same!

a semicolon after every command is the best practice. Spaces, tabs and newline characters are called whitespace.

2. Comments

The syntax of comments is the same as in C++ and in many other languages:

```
// a one line comment
```

```
/* this is a longer,  
 * multi-line comment  
 */
```

```
/* You can't, however, /* nest comments */ SyntaxError */
```

Comments behave like white space and are discarded during script execution.

3. Declarations

There are three kinds of declarations in JavaScript.

`var`

Declares a variable, optionally initializing it to a value.

`let`

Declares a block-scoped, local variable, optionally initializing it to a value.

`const`

Declares a block-scoped, read-only named constant.

4.Variables

You use variables as symbolic names for values in your application. The names of variables, called identifiers, conform to certain rules.

A JavaScript identifier must start with a letter, underscore (`_`), or dollar sign (`$`); subsequent characters can also be digits (0-9). Because JavaScript is case sensitive, letters include the characters "A" through "Z" (uppercase) and the characters "a" through "z" (lowercase).

contoh:

```
jengkol
_jengkol
$jengkol
jengkol99
jengkol_100
```

You can declare a variable in three ways:

With the keyword `var`. For example, `var x = 42`. This syntax can be used to declare both local and global variables.

By simply assigning it a value. For example, `x = 42`. If this form is used outside of a function, it declares a global variable. It generates a strict JavaScript warning. You shouldn't use this variant.

With the keyword `let`. For example, `let y = 13`. This syntax can be used to declare a block-scope local variable. See Variable scope below.

5. mengevaluasi variable

Gunakan fasilitas web console (atau scratchpad) pada browser, kemudian cobalah setiap baris dibawah!

```
var a;
console.log('The value of a is ' + a); // The value of a is undefined
```

```
console.log('The value of b is ' + b); // The value of b is undefined
```

```
var b;
```

```
console.log('The value of c is ' + c); // Uncaught ReferenceError: c is not defined
```

```
let x;
```

```
console.log('The value of x is ' + x); // The value of x is undefined
```

```
console.log('The value of y is ' + y); // Uncaught ReferenceError: y is not defined
```

```
let y;
```

secara sederhana penggunaan pada web console dapat dilakukan seperti berikut:

ketik :

```
var a;
```

kemudian panggil: a (tekan enter)

latihan!

buat 4 variable terdiri atas dua jenis string dan dua jenis numerik

latihan!

variable isinya bisa seperti berikut:

```
var a = "aku adalah " + "pahlawan bertopeng"; perhatikan spasi tambahan diakhir kata "adalah"!
```

apa yang terjadi jika di tambah konten numerik?

latihan!

ketik pada web console, lalu panggil mereka dan perhatikan apa yang terjadi!

```
var a = new Date();
```

```
var b = Date();
```

Latihan!

membuat app perhitungan sederhana dgn user input

pd HTML, buatlah form, beri nama

buat dua input tipe number, beri nama kedua input tsb!. manfaatkan element <pre>!

buat satu input tipe button

buat satu input tipe reset

pada JS buat fungsi yang berisi dua variable untuk menyimpan dua inputan tipe number dari user. gunakan parseFloat pada kedua variable tsb!

buat variable tempat kalkulasi (sekaligus tempat hasil) kedua inputan tipe number diatas. gunakan 'alert' untuk menampilkan hasil

contoh lengkap app:

HTML code:

```
<form name ="form_isian">
```

```
<H1>App sederhana dgn HTML+CSS+JS</H1>
```

```
<pre>
```

```
Angka Pertama: <input type="number" name="angka1">
```

```
Angka Kedua : <input type="number" name="angka2">
```

```
</pre>
```

```
<input type="button" value="Jumlahkan" onclick="jumlah()">
```

```
<input type="reset" value="Reset">
```

```
</form>
```

JS code:

```
function jumlah(){  
    var a1  = parseFloat(document.form_isian.angka1.value);  
    var a2  = parseFloat(document.form_isian.angka2.value);  
    var hasil = a1 + a2;  
    alert("Hasil Penjumlahan = " + hasil);  
}
```

PERTEMUAN 3 Arrays

contoh array:

```
var a = ['jengkol', 'pete', 'palanding', 'sambel'];
```

cara memanggil array satu-persatu ialah seperti berikut:

```
var a =[0]; hasilnya=jengkol
```

```
var a = [1]; hasilnya=pete
```

```
dst.....
```

note:

perhitungan array dimulai dengan angka nol untuk item yang pertama!

latihan!

-buat sebuah array, dengan 5 sub item didalamnya, kemudian panggil satu persatu di web console!

-buatlah variable yang mengandung variable lain didalamnya!

-buatlah array yg mengandung array, lalu coba panggil keseluruhan sub array dan coba panggil satu persatu sub array tsb!

spesial karakter pd JS:

\n = membuat garis baru

\t = tab

\v = vertikal tab

\\ = backslash

\\" = double quote

latihan!

buat lah variable yang berisi:

"jengkol adalah "kacang dewa" di Indonesia!" Perhatikan kata yang bertanda petik!!

Array Methods

The Array object has the following methods:

`concat()` joins two arrays and returns a new array.

```
var myArray = new Array('1', '2', '3');  
myArray = myArray.concat('a', 'b', 'c');  
// myArray is now ["1", "2", "3", "a", "b", "c"]
```

`join(delimiter = ',')` joins all elements of an array into a string.

```
var myArray = new Array('Wind', 'Rain', 'Fire');  
var list = myArray.join(' - '); // list is "Wind - Rain - Fire"
```

`push()` adds one or more elements to the end of an array and returns the resulting length of the array.

```
var myArray = new Array('1', '2');  
myArray.push('3'); // myArray is now ["1", "2", "3"]
```

`pop()` removes the last element from an array and returns that element.

```
var myArray = new Array('1', '2', '3');  
var last = myArray.pop();  
// myArray is now ["1", "2"], last = "3"
```

`shift()` removes the first element from an array and returns that element.

```
var myArray = new Array('1', '2', '3');  
var first = myArray.shift();  
// myArray is now ["2", "3"], first is "1"
```

`unshift()` adds one or more elements to the front of an array and returns the new length of the array.

```
var myArray = new Array('1', '2', '3');  
myArray.unshift('4', '5');  
// myArray becomes ["4", "5", "1", "2", "3"]
```


slice(start_index, upto_index) extracts a section of an array and returns a new array.

```
var myArray = new Array('a', 'b', 'c', 'd', 'e');  
myArray = myArray.slice(1, 4); // starts at index 1 and extracts all elements  
// until index 3, returning [ "b", "c", "d"]
```

splice(index, count_to_remove, addElement1, addElement2, ...) removes elements from an array and (optionally) replaces them. It returns the items which were removed from the array.

```
var myArray = new Array('1', '2', '3', '4', '5');  
myArray.splice(1, 3, 'a', 'b', 'c', 'd');  
// myArray is now ["1", "a", "b", "c", "d", "5"]  
// This code started at index one (or where the "2" was),  
// removed 3 elements there, and then inserted all consecutive  
// elements in its place.
```

reverse() transposes the elements of an array, in place: the first array element becomes the last and the last becomes the first. It returns a reference to the array.

```
var myArray = new Array('1', '2', '3');  
myArray.reverse();  
// transposes the array so that myArray = ["3", "2", "1"]
```

sort() sorts the elements of an array in place, and returns a reference to the array.

```
var myArray = new Array('Wind', 'Rain', 'Fire');  
myArray.sort();  
// sorts the array so that myArray = ["Fire", "Rain", "Wind"]  
sort() can also take a callback function to determine how array elements are compared.
```

masih banyak method dari array yang tidak dibahas disini!

Latihan! -> mengulang kembali pembuatan app penjumlahan sederhana agar ngolotok!!

-> coba gunakan operator pengurangan dan perkalian!!

-> gunakan css untuk mempercantik tampilan!

membuat app perhitungan sederhana dgn user input

pd HTML, buatlah form, beri nama

buat dua input tipe number, beri nama kedua input tsb!.

buat satu input tipe button

buat satu input tipe reset

pada JS buat fungsi yang berisi dua variable untuk menyimpan dua inputan tipe number dari user. gunakan parseFloat pada kedua variable tsb!

buat variable tempat kalkulasi (sekaligus tempat hasil) kedua inputan tipe number diatas.

gunakan 'alert' untuk menampilkan hasil

Latihan!

modifikasi app diatas dgn menggunakan addEventListener!

pada HTML, ubah input tipe button, menjadi element <button>!

pd JS, tambahkan dibagian paling awal dokumen, varibel yang berisi tombol, menggunakan querySelector!

tautkan varibel tsb dgn event listener, tambahkan jenis event, dan fungsi yang aktif saat event muncul!

contoh code js:

```
var tombol = document.querySelector('button');
```

```
tombol.addEventListener('click', jumlah);
```

```
function jumlah(){
```

```
    var a1  = parseFloat(document.form_isian.angka1.value);
```

```
    var a2  = parseFloat(document.form_isian.angka2.value);
```

```
    var hasil = a1 + a2;
```

```
    alert("Hasil Penjumlahan = " + hasil);
```

```
}
```

PERTEMUAN 4 Conditional statements

conditional statements

A conditional statement is a set of commands that executes if a specified condition is true. JavaScript supports two conditional statements: if...else and switch.

Use the if statement to execute a statement if a logical condition is true. Use the optional else clause to execute a statement if the condition is false.

```
if (condition) {  
    statement_1;  
} else {  
    statement_2;  
}
```

If condition evaluates to true, statement_1 is executed; otherwise, statement_2 is executed. statement_1 and statement_2 can be any statement, including further nested if statements.

You may also compound the statements using else if to have multiple conditions tested in sequence, as follows:

```
if (condition_1) {  
    statement_1;  
} else if (condition_2) {  
    statement_2;  
} else if (condition_n) {  
    statement_n;  
} else {  
    statement_last;  
}
```

kita dapat mengisi lebih dari satu statement pada block statement, contoh:

```
if (condition) {  
    statement_1_runs_if_condition_is_true;  
    statement_2_runs_if_condition_is_true;  
} else {  
    statement_3_runs_if_condition_is_false;  
    statement_4_runs_if_condition_is_false;  
}
```

contoh praktikal fungsi logika/percabangan:

note:

Apabila melihat struktur script dibawah ini, kira-kira
element apa yg tepat untuk di tempatkan pd dokumen HTML-nya?

```
function tanyabilangan() {  
    var bil = parseFloat(document.fform.bilangan.value);  
    var jenis = " ";  
    if(isNaN(bil)) {  
        alert("Anda Belum memasukkan Bilangan");  
    }  
    else if (bil > 0 )  
        {jenis = " Adalah bilangan Positif";}   
    else if (bil < 0)  
        {jenis = " Adalah bilangan Negatif";}   
    else  
        {jenis = " Adalah Nol";}   
    alert (bil+" "+jenis);  
}
```

Switch statement

A switch statement allows a program to evaluate an expression and attempt to match the expression's value to a case label. If a match is found, the program executes the associated statement. A switch statement looks as follows:

```
switch (expression) {  
    case label_1:  
        statements_1  
        [break;]  
    case label_2:  
        statements_2  
        [break;]  
    ...  
    default:  
        statements_def  
        [break;]  
}
```

The program first looks for a case clause with a label matching the value of expression and then transfers control to that clause, executing the associated statements. If no matching label is found, the program looks for the optional default clause, and if found, transfers control to that clause, executing the associated statements. If no default clause is found, the program continues execution at the statement following the end of switch. By convention, the default clause is the last clause, but it does not need to be so.

The optional break statement associated with each case clause ensures that the program breaks out of switch once the matched statement is executed and continues execution at the statement following switch. If break is omitted, the program continues execution at the next statement in the switch statement.

contoh penggunaan switch:

```
switch (fruittype) {  
    case 'Oranges':
```

```

    console.log('Oranges are $0.59 a pound.');
```

break;

```

case 'Apples':
    console.log('Apples are $0.32 a pound.');
```

break;

```

case 'Bananas':
    console.log('Bananas are $0.48 a pound.');
```

break;

```

case 'Cherries':
    console.log('Cherries are $3.00 a pound.');
```

break;

```

case 'Mangoes':
    console.log('Mangoes are $0.56 a pound.');
```

break;

```

case 'Papayas':
    console.log('Mangoes and papayas are $2.79 a pound.');
```

break;

```

default:
    console.log('Sorry, we are out of ' + fruittype + '.');
```

}

```

console.log("Is there anything else you'd like?");
```

contoh penggunaan lebih riil:

HTML code:

 -----ganti sesuai dgn element yang anda gunakan!

JS Code:

```
<script>
```

```
var day;
```

```
switch (new Date().getDay()) {
```

```
    case 0:
```

```

    day = "<img src=hari/minggu.png style=height:80px;width:100px;align-
items:center;justify-content:center;display:flex;>";
    break;
case 1:
    day = "<img src=hari/senin.png style=height:80px;width:100px;align-
items:center;justify-content:center;display:flex;>";
    break;
case 2:
    day = "<img src=hari/selasa.png style=height:80px;width:100px;align-
items:center;justify-content:center;display:flex;>";
    break;
case 3:
    day = "<img src=hari/rabu.png style=height:80px;width:100px;align-
items:center;justify-content:center;display:flex;>";
    break;
case 4:
    day = "<img src=hari/kamis.png style=height:80px;width:100px;align-
items:center;justify-content:center;display:flex;>";
    break;
case 5:
    day = "<img src=hari/jumat.png style=height:80px;width:100px;align-
items:center;justify-content:center;display:flex;>";
    break;
case 6:
    day = "<img src=hari/sabtu.png style=height:80px;width:100px;align-
items:center;justify-content:center;display:flex;>";
    break;
}
document.getElementById("hari").innerHTML = day;
</script>

```

Latihan tentang switch akan kita lakukan nanti!

Latihan!

lanjutan app sederhana dgn radio button.----agar ngolotok

- buat form yg berisi beberapa element radio button

beri value dan nama setiap radio button, misal: pria, wanita, banci

- buat sebuah elemen <button>, beri value dan beri "onclick"!

- buat tombol reset

- buat JS code sesuai keperluan

buat beberapa variable yg diperlukan sebagai wadah data inputan HTML

kemudian buat fungsi logika untuk mengecek radio mana yg dipilih

contoh code HTML:

```
<form name ="form_isian">
```

```
<H1>App sederhana dgn HTML+CSS+JS</H1>
```

```
<pre>
```

```
<input type="radio" value="pria" name="pria">Pria</input>
```

```
<input type="radio" value="wanita" name="wanita">Wanita</input>
```

```
<input type="radio" value="waria" name="waria">Waria</input>
```

```
<button onclick="konfirmasi()">submit</button>
```

```
<input type="reset" value="Reset">
```

```
</pre>
```

```
</form>
```

contoh code JS:

```
function konfirmasi(){
```

```
    var a = document.form_isian.pria.checked;
```

```
    var b = document.form_isian.wanita.checked;
```

```
    if (a == true)
```



```

    {alert('Ternyata anda Pria!');}
else if (b == true)
    {alert('Ternyata anda Wanita!');}
else
    {alert('Anda Waria? ... Zizai sekali!');}

}

```

Code JS menyederhanakan fungsi logika:

```

function konfirmasi(){

    if (form_isian.pria.checked == true)
        {alert('Ternyata anda Pria!');}
    else if (form_isian.wanita.checked == true)
        {alert('Ternyata anda Wanita!');}
    else
        {alert('Anda Waria? ... Zizai sekali!');}

}

```

Latihan dengan checkbox:

Buat Html dan JS yang persis seperti app sblmnya!

Kali ini kita menggunakan element checkbox.

Tugas kita adalah menampilkan pilihan user atas checkbox yg ditawarkan.

Hasil yg ditampilkan hanya pada pilihan user saja!

-pada html buat tiga element input tipe checkbox, lalu beri nama!

-buat element tombol, beri nama submit

-buat tombol reset

- pada JS, manfaatkan variable kosong untuk menyimpan pilihan user
- gunakan fungsi logika serta alert untuk menyelesaikan latihan ini!
- pada bagian alert, gunakan spasi (karakter kosong) untuk memisahkan setiap pilihan user

contoh HTML:

```
<form name ="form_isian">
```

```
<H1>App sederhana dgn HTML+CSS+JS</H1>
```

```
<pre>
```

```
<input type="checkbox" value="on" name="pubg">Maen pubg</input>
```

```
<input type="checkbox" value="on" name="kill">suka nge-kill</input>
```

```
<input type="checkbox" value="on" name="dikill">suka di-kill</input>
```

```
<button>submit</button>
```

```
<input type="reset" value="Reset">
```

```
</pre>
```

```
</form>
```

contoh JS:

```
var tombol = document.querySelector('button');
```

```
tombol.addEventListener('click', konfirmasi);
```

```
function konfirmasi(){
```

```
    var a = "";
```

```
    var b = "";
```

```
    var c = "";
```

```
    if (form_isian.pubg.checked == true)
```

```
        {a = "Maen PUBG";}
```

```
    if (form_isian.kill.checked == true)
```

```

        {b = "Nge-Kill musuh dan teman";}
    if (form_isian.dikill.checked == true)
        {c = "Di-Kill musuh dan teman";}
    alert("Hobby Gue adalah " + a + " " + b + " " + c);

}

```

Latihan Medium level!!

Buat app penghitungan tiket!

contoh HTML:

```
<form name ="fform">
```

```
<table border="1" align="center"width="70%">
```

```
<tr>
```

```
<td colspan="2"><h2 align="center">PT. Selalu Ciken</H2></td>
```

```
</tr>
```

```
<tr>
```

```
<td width="50%">
```

```
<pre>
```

```
Nama      :<input type="text" size="20" name="inama">
```

```
Tujuan    :<select name="tujuan" size=1>
```

```
<option value="Jakarta">Jakarta</option>
```

```
<option value="Cirebon">Cirebon</option>
```

```
<option value="Tasikmalaya">Tasikmalaya</option>
```

```
</select>
```

```
Jumlah Tiket :<input type="text" size="11" name="ijumlah">
```

```
Member     :<input type="checkbox" name="imember">Ya</td>
```

```
</pre>
```

```
<td>
```

```

<pre>
  Harga Tiket :<input type="text" size="10" name="otiket">
  Sub Total   :<input type="text" size="10" name="osub">
  Diskon      :<input type="text" size="10" name="odiskon">
  Total Bayar :<input type="text" size="10" name="ototal"></td>
<pre>
</tr>

<tr>
  <td width="100%" colspan="2">
    <center>
      <input type="button" value="Hitung" onclick="hitungtotal()">
      <input type="reset" value="Reset">
    </center>
  </td>
</tr>

</table>

</form>

```

Contoh JS:

```

/*awal line fungsi hitungtotal(), sekaligus menyiapkan variable untuk data2 yg akan diolah*/
function hitungtotal() {
  var nama = (document.fform.inama.value);
  var tujuan = (document.fform.itujuan.value);
  var jumlahtiket = parseFloat(document.fform.ijumlah.value);
  var ht = "";
  var sub = "";

```

```

var diskon ="";
var total ="";

/*memeriksa tujuan destinasi user*/
if (tujuan=="Jakarta")
    {ht =100000;}
else if (tujuan=="Cirebon")
    {ht =150000;}
else
    {ht =200000;}

/*menghitung berapa jumlah tiket yg dipesan, dikalikan dgn harga tiket tsb*/
sub = jumlahtiket*ht;

/*memeriksa apakah user sudah jadi member atau belum*/
if (document.fform.imember.checked==true)
    {diskon =0.10*sub;}
else
    {diskon=0.0;}

/*menghitung total harga tiket, dikurangi diskon apabila user adalah member*/
total = sub-diskon;

/*menerapkan hasil pengolahan fungsi, pada variable yg telah disiapkan*/
document.fform.otiket.value=ht;
document.fform.osub.value=sub;
document.fform.odiskon.value=diskon;
document.fform.ototal.value=total;

/*line akhir fungsi hitungtotal()*/
}

```

Note:

Kita dapat menambahkan fungsi validasi pada jumlah tiket, agar user dipastikan memasukkan angka, bukan huruf!

contoh:

```
if (isNaN(jumlahtiket)) {  
    alert('Mohon Masukan Angka Pada Jumlah Tiket!');  
}
```

Note 2:

Coba tambahkan pemberitahuan akhir di bawah tabel, berisi nama, tujuan+jumlah tiket dan harga akhir!

Latihan Medium level 2!

Membuat app penilai IP mahasiswa.

Lihat tampilan pd proyektor, ikuti design tampilan tsb pada HTML semampu anda!

Kemudian design code JS nya!

contoh HTML:

<!--membuat form, lalu beri nama-->

<form name ="fform">

<!--Gunakan berbagai cara, yg penting rapi-enak dilihat. Form dibawah menggunakan element table

dgn lebar 100%, artinya ditampilkan selebar layar yg digunakan.-->

<table border="1" width="70%" align="center">

<!--tr dibawah menggunakan colspan=4, alias merger empat kolom-->

```

<tr>
  <td colspan="4"><h2 align="center">Menghitung Indeks Prestasi Ciken</h2></td>
</tr>

```

<!--membuat empat input untuk nilai absen, tugas, uts dan uas-->

```

<tr>
  <td width="25%">Absensi (10%) :<input type="text" size="10" name="iabsen"> </td>
  <td width="25%">Tugas (20%):<input type="text" size="10" name="itugas"> </td>
  <td width="25%">UTS(35%):<input type="text" size="10" name="iuts"> </td>
  <td width="25%">UAS (35%) :<input type="text" size="10" name="iuas"> </td>
</tr>

```

<!--membuat dua tombol dlm satu element td, tombol hitung dan reset dgn posisi rata tengah. td disini harus diberi colspan=4. Lalu, beri fungsi onclick pd tombol hitung untuk memanggil fungsi JS-->

```

<tr>
  <td width="100%" colspan="4" align="center">
    <INPUT TYPE="button" value="Hitung" onclick="hitungip()">
    <INPUT TYPE="reset" value="Ulang">
  </td>
</tr>

```

<!--Bagian trakhir, membuat wadah untuk hasil pengolahan IP beserta keterangannya. Kedua input disini wajib diberi nama, agar dapat digunakan oleh JS. Element td diberi colspan=4 dan rata tengah-->

```

<tr>
  <td width="100%" colspan="4" align="center">
    Indeks Prestasi : <input type="text" size="20" name="oip">
    Keterangan :<input type="text" size="25" name="oket">
  </td>
</tr>

```

```

</table>

```

</form>

contoh JS:

```
//mulai dgn membuat fungsi hitungip()
function hitungip() {

//persiapkan empat input variable: absen, tugas, uts dan uas
    var absen = parseFloat(document.fform.iabsen.value);
    var tugas = parseFloat(document.fform.itugas.value);
    var uts   = parseFloat(document.fform.iuts.value);
    var uas   = parseFloat(document.fform.iuas.value);

//persiapkan juga variable untuk output pengolahan ip dan keterangannya
    var ip    =" ";
    var ket    =" ";

//buat variable untuk penghitungan ke-4 input di awal
    var na    = (0.10*absen)+(0.20*tugas)+(0.35*uts)+(0.35*uas);

//buat fungsi logika untuk menentukan batas bawah dan batas atas nilai
//kemudian, hasilnya diisikan pada variable ip dan ket
    if ((na > 80) && (na <=100))
        {ip ="A"; ket="Lulus dengan Sangat Baik";}
    else if ((na > 68) && (na <=80))
        {ip ="B"; ket="Lulus dengan Baik";}
    else if ((na > 55) && (na <=68))
        {ip ="C"; ket="Lulus dengan Cukup";}
    else if ((na > 38) && (na <=55))
        {ip ="D"; ket="Lulus dengan Kurang";}
    else
        {ip ="E"; ket="Tidak Lulus";}
```


//proses akhir adalah menerapkan hasil pengolahan JS pada sub element form dalam HTML

//yakni, element input dgn name 'oip' dan 'oket'

```
document.fform.oip.value = "huruf: " + ip + " " + "angka: " + na;
```

```
document.fform.oket.value = ket;
```

```
//akhir kurung kurawal function
```

```
}
```

Code JS diatas akan menghasilkan output indeks prestasi : huruf: xx angka: xx

Coba kalian experiment, bagaimana agar menghasilkan output akhir seperti berikut:

Indeks Prestasi: B (78)

Jawabannya:

Hanya perlu memodifikasi bari code JS ini:

```
document.fform.oip.value = "huruf: " + ip + " " + "angka: " + na;
```

Menjadi:

```
document.fform.oip.value = ip + " " + "(" + na + ")";
```

Berikutnya, coba tampilkan juga angka hasil perhitungan pada input absensi, tugas, uts dan uas.

Apabila tombol hitung di tekan, ke-4 input tersebut juga turut menampilkan hasil perhitungannya.

Contoh Code JS:

```
//mulai dgn membuat fungsi hitungip()
function hitungip() {

    //persiapkan empat input variable: absen, tugas, uts dan uas
    var absen = parseFloat(document.fform.iabsen.value);
    var tugas = parseFloat(document.fform.itugas.value);
    var uts   = parseFloat(document.fform.iuts.value);
    var uas   = parseFloat(document.fform.iuas.value);

    //persiapkan juga variable untuk output pengolahan ip dan keterangannya
    var ip    =" ";
    var ket   =" ";

    //buat variable untuk wadah atas penghitungan ke-4 input dan untuk nilai IP/nilai akhir--
    >na

    //untuk input absen, kita harus menghitung berapa % total kehadiran mahasiswa terlebih
    dahulu
    //baru kemudian total % kehadiran tersebut dikalikan dgn total pertemuan, misal 14
    pertemuan
    var nabsen    = ((absen/14)*10);
    var nabsen_10 = (nabsen*1.4);
    var ntugas    = (0.20*tugas);
    var nuts      = (0.35*uts);
    var nuas      = (0.35*uas);
    var na        = nabsen_10 + ntugas + nuts + nuas;

    //buat fungsi logika untuk menentukan batas bawah dan batas atas nilai
    //kemudian, hasilnya diisikan pada variable ip dan ket
    if ((na > 80) && (na <=100))
        {ip ="A"; ket="Lulus dengan Sangat Baik";}
    else if ((na > 68) && (na <=80))
```

```

        {ip ="B"; ket="Lulus dengan Baik";}
    else if ((na > 55) && (na <=68))
        {ip ="C"; ket="Lulus dengan Cukup";}
    else if ((na > 38) && (na <=55))
        {ip ="D"; ket="Lulus dengan Kurang";}
    else
        {ip ="E"; ket="Tidak Lulus";}

```

//proses akhir adalah menerapkan hasil pengolahan JS pada sub element form dalam HTML

//yakni, element input dgn name iabsen, itugas, iuts, ouas, oip dan oket.

```

document.fform.iabsen.value= nabsen_10;
document.fform.itugas.value= ntugas;
document.fform.iuts.value = nuts;
document.fform.iuas.value = nuas;
document.fform.oip.value = ip + " " + "(" +na+ ")";
document.fform.oket.value = ket;

```

//akhir kurung kurawal function

```

}

```

Latihan switch:

Membuat app sederhana dengan switch

HTML:

```

<form name ="fform">
    <h2>Penggunaan Percabangan Switch</h2>
    <hr>
    <PRE>

```

```
Masukkan Nomor Bulan [1-12] :<input type="text" size="2" name="ibulan">
<INPUT TYPE="button" value="Hitung" onclick="tanyabulan()"> <INPUT TYPE="reset"
value="Ulang">
</PRE>
</form>
```

JS:

```
function tanyabulan() {
    var bulan = parseFloat(document.fform.ibulan.value);
    var namabulan=" ";

    switch (bulan) {
        case 1 : namabulan="Bulan ke 1 adalah = Januari";
        break;
        case 2 : namabulan="Bulan ke 2 adalah = Februari";
        break;
        case 3 : namabulan="Bulan ke 3 adalah = Maret";
        break;
        case 4 : namabulan="Bulan ke 4 adalah = April";
        break;
        case 5 : namabulan="Bulan ke 5 adalah = Mei";
        break;
        case 6 : namabulan="Bulan ke 6 adalah = Juni";
        break;
        case 7 : namabulan="Bulan ke 7 adalah = Juli";
        break;
        case 8 : namabulan="Bulan ke 8 adalah = Agustus";
        break;
        case 9 : namabulan="Bulan ke 9 adalah = September";
        break;
        case 10 : namabulan="Bulan ke 10 adalah = Oktober";
        break;
        case 11 : namabulan="Bulan ke 11 adalah = November";
```

```
break;
case 12 : namabulan="Bulan ke 12 adalah = Desember";
break;
default : namabulan="Anda salah mengisi";
}
alert(namabulan);
}
```

Latihan!

Ubahnya code switch diatas dengan fungsi logika if - else if - else !

PERTEMUAN 5 Loops and Iteration

Loops offer a quick and easy way to do something repeatedly. This chapter of the JavaScript Guide introduces the different iteration statements available to JavaScript.

There are many different kinds of loops, but they all essentially do the same thing: they repeat an action some number of times (and it's actually possible that number could be zero). The various loop mechanisms offer different ways to determine the start and end points of the loop. There are various situations that are more easily served by one type of loop over the others.

The statements for loops provided in JavaScript are:

- for statement
- do...while statement
- while statement
- labeled statement
- break statement
- continue statement
- for...in statement
- for...of statement

contoh loop dengan for statement:

```
var step;
for (step = 0; step < 5; step++) {
  // Runs 5 times, with values of step 0 through 4.
  console.log('Walking east one step');
}
```

contoh penggunaan secara penuh:

```
<form name="selectForm">
  <p>
    <label for="musicTypes">Choose some music types, then click the button
below:</label>
    <select id="musicTypes" name="musicTypes" multiple="multiple">
      <option selected="selected">R&B</option>
      <option>Jazz</option>
      <option>Blues</option>
      <option>New Age</option>
      <option>Classical</option>
      <option>Opera</option>
    </select>
  </p>
  <p><input id="btn" type="button" value="How many are selected?" /></p>
</form>

<script>
function howMany(selectObject) {
  var numberSelected = 0;
  for (var i = 0; i < selectObject.options.length; i++) {
    if (selectObject.options[i].selected) {
      numberSelected++;
    }
  }
  return numberSelected;
}

var btn = document.getElementById('btn');
btn.addEventListener('click', function() {
  alert('Number of options selected: ' + howMany(document.selectForm.musicTypes));
});
</script>
```

Latihan!

Membuat fungsi perulangan (loop) sederhana

-sisipkan script dibawah pada HTML!

-pelajari dgn seksama, pahami

-kemudian coba tulis ulang tanpa melihat contoh yg sudah ada tsb!

```
<script>
```

```
for (i=1; i<=10; i++) {
```

```
document.write("-----" + "<br/>");
```

```
document.write("Yuhuu...Belajar Ciken Yuk!" + "<br/>");
```

```
document.write("-----" + "<br/>");
```

```
}
```

```
</script>
```

Latihan 2!

Latihan!

Membuat fungsi perulangan (loop) sederhana

-sisipkan script dibawah pada HTML!

-pelajari dgn seksama komentar yg telah disediakan!

```
<script>
```

```
//membuat judul
```

```
document.write("<h2>Latihan Perulangan For</h2>");
```

```
//membuat garis pembatas. Secara otomatis berada pd garis baru, karena element h2  
telah dipanggil
```

```
//sebelumnya
```



```

document.write("-----");

//membuat fungsi loop. Setiap loop akan ditempatkan pd garis baru
//karena kita membubuhkan element <br/>
//kita juga memanfaatkan element <pre> untuk perataan setiap hasil
for (i=1; i<=10; i++)
{
    sq=i*i;
    document.write("<pre>");
    document.write("Angka:" + i + " -----> Kuadrat: " + sq + "<br/>");
    document.write("</pre>");
}
document.write("-----");
</script>

```

Latihan do-while!

Buat sebuah form, gunakan method 'post'.

Buat satu input tipe button di dlm form tsb.

Persiapkan/buat dokumen welcome.html, diperuntukan bagi user yg benar memasukan password.

Pada JS, buat sebuah function bernama pass().

buat satu variable untuk digunakan oleh while diakhir fungsi nanti.

desain fungsi do-while, untuk pengecekan inputan yg benar, inputan yg salah dan pengecekan sudah berapa kali percobaan inputan yg dilakukan user.

Code HTML:

```

<!--menggunakan form dgn method post untuk keamanan data-->
<form method="post">

```

```
<input type="button" value="Password" onclick="pass()">
</form>
```

Code JS:

```
function pass() {
    //mempersiapkan variable untuk fungsi while.
    var coba = 1;

    //mulai fungsi do - while
    do {
        //buat dulu variable untuk menyimpan inputan user
        p = prompt("Masukan Password ", "");
        if (p=="ciken")
            {alert("Selamat Datang Friends");
            //persiapkan file welome.html nya oleh anda!
            window.open("welcome.html");
            break;
        }
        else
            {alert("Password Salah !!! Ulangi lagi.");}

        if (coba==3) {
            alert("maaf, kesempatan anda hanya 3 kali");
            //script dibawah ini agar user dibawa ke halaman sebelumnya
            //misal halaman daftar ulang lagi
            //atau dpt menggunakan window.open untuk mengarahkan user pd halaman
            autentifikasi lainnya
            history.go(-1);
        }

        //setiap user mencentik tombol, maka nilai var coba ditambah 1
        coba=coba+1;
    }
```

```
}
```

```
//apabila var coba sudah bernilai lebih dari 3, user tidak diperbolehkan mencoba lagi
```

```
while (coba<=3)
```

```
}
```

PERTEMUAN 6 Function

Functions are one of the fundamental building blocks in JavaScript. A function is a JavaScript procedure—a set of statements that performs a task or calculates a value. To use a function, you must define it somewhere in the scope from which you wish to call it.

contoh function beserta penjelasannya:

```
function square(number) {  
  return number * number;  
}
```

The function square takes one parameter, called number. The function consists of one statement that says to return the parameter of the function (that is, number) multiplied by itself. The return statement specifies the value returned by the function.

Note:

variable juga dapat diisi dengan function, contoh:

```
var square = function(number) { return number * number; };  
var x = square(4); // x gets the value 16
```

nah, coba panggil variable x!

contoh lain, perhatikan scopenya!

```
// The following variables are defined in the global scope  
var num1 = 20,  
    num2 = 3,  
    name = 'jengkol';
```

```
// This function is defined in the global scope
```

```
function multiply() {  
    return num1 * num2;  
}
```

`multiply(); // Returns 60`

// A nested function example

//Variable yg ada pd scope spesifik, tidak dpt digunakan oleh fungsi pd level global!

//perhatikan dan pahami var num1 dan num2 dari fungsi multi[y()] dan getScore() dibawah.

```
function getScore() {  
    var num1 = 2,  
        num2 = 3;  
  
    function add() {  
        return name + ' scored ' + (num1 + num2);  
    }  
  
    return add();  
}
```

`getScore(); // Returns "jengkol scored 5"`

Objek Math

Math digunakan untuk menangani perhitungan matematis yang rumit. Bentuk penulisan :

`Math.metode(nilai)`

Berikut adalah daftar metode objek Math:

Metode : Keterangan :

`abs(a)` Nilai absolut dari a

acos(a) Nilai arc-kosinus dari a
asin(a) Nilai arc-sinus dari a
atan(a) Nilai arc-tan dari a
ceil(a) Membulatkan nilai ke integer di atasnya
cos(a) Nilai kosinus dari a
exp(a) Nilai E pangkat a
log(a) Nilai logaritma dari a
max(a,d) Nilai terbesar dari a dan d
min(a,d) Nilai terkecil dari a dan d
pow(a,d) Nilai dari a pangkat d
random(a) Nilai acak antara 0 dan 1
round(a) Membulatkan nilai a ke integer terdekat
sqrt(a) Nilai akar dari kuadrat a
sin(a) Nilai sinus dari a
tan(a) Nilai tangen dari a

contoh penggunaanya:

```
var a=10;  
var b=5;  
besar=Math.max(a,b);
```

Latihan!

Buat dua variable yg berisi data numerik.

Buat variable sbgi wadah untuk mengolah var sebelumnya dgn menggunakan metode.

Sisipkan script pada dokumen HTML

Tambahkan script untuk menampilkan hasil pengolahannya!

Contoh Code JS:

```
<script>
```

```
var a=10;
var b=5;
besar=Math.max(a,b);

//menampilkan hasil Math.max
document.write("antara " + a + " dan "+ b +" lebih besar" +besar+ "<br>");

//menampilkan hasil Math.pow
pangkat=Math.pow(b,a);
document.write(b+ " pangkat "+ a +" adalah " +pangkat+ "<br>");

//mempersiapkan wadah untuk Math.round
//var ran juga ditambah method Math.round, agar hasil dari Math.random dibulatkan
menjadi angka terdekat
//ketimbang full bentuk integer
var ran;
ran = Math.round(Math.random()*50000);
document.write("Anda adalah pengunjung yang ke " + ran + ".<br>");

//menampilkan hasil Math.sqrt
var akar;
akar =Math.sqrt(ran);
document.write ("Akar dari " + ran + " adalah "+akar);

</script>
```

Latihan new Date() dan methods!

Sisipkan code JS dibawah ini pada dokumen HTML!

Pelajari step per stepnya, baca komentar yg telah disiapkan!

Code JS:

<script>

```
//membuat dua array, hari dan bulan, untuk digunakan oleh methode getDay() dan
getMonth()
var hari = ["Minggu","Senin","Selasa","Rabu","Kamis","Jumat","Sabtu"];
var bulan =
["Januari","Februari","Maret","April","Mei","Juni","Juli","Agustus","September","Oktober","N
ovember","Desember"];

//membuat variable untuk menyimpan hasil fungsi new Date()
//new Date() menghasilkan output tanggal,hari,bulan,tahun,jam,menit dan detik
//semua itu dpt kita panggil satu persatu nantinya
var t = new Date();

//dari mulai var hari_ini s/d var detik, digunakan untuk menyimpan data spesifik atas hasil
fungsi new Date()
//yg ada pd var t.
//disini, setiap var mengambil bagian spesifik dari var t dgn menggunakan methods
//khusus untuk var hari_ini dan var bulan_ini, memanfaatkan var hari dan var bulan
sebelumnya
//agar hasil dari hari[t.getDay()]; dan bulan[t.getMonth()]; kita konversi menjadi string hari
dan bulan
//yg sudah kita siapkan
//tanpa var hari dan var bulan pun, JS msh berjalan normal, hanya nama hari akan
berbentuk angka (0 s/d 6)
//nama bulan pulan berbentuk angka (1-12)
var hari_ini=hari[t.getDay()];
var tanggal=t.getDate();
var bulan_ini=bulan[t.getMonth()];
var tahun=t.getFullYear(); //getFullYear() akan menampilkan tahun dalam format 4 digit
var jam =t.getHours();
var menit =t.getMinutes();
```



```
var detik =t.getSeconds();

//langkah selanjutnya menuliskan data pada setiap variabel diatas dgn menggunakan
//document.write("");
//dgn JS, kita juga bisa membuat dan menulis element HTML beserta CSS nya secara
inline
document.write("<font size=5 face=arial>");
document.write("<b><center>Sekarang adalah hari :"+hari_ini+",tanggal: "+ tanggal +" "+
bulan_ini +" " +tahun);
document.write("<hr width=700>");
document.write("</font>");
document.write("<font size=3 face=arial>");
document.write("<b><center>Jam sekarang = "+ jam + ":" +menit+ ":" +detik);
document.write("</font>");

</script>
```

Note:

Setelah berhasil menampilkan script JS diatas pada browser

Coba kalian tekan tekan refresh dgn cepat pada browser, dan keterangan Jam Sekarang berubah terus!

PERTEMUAN 7 Sound Effect

Pertemuan 7 ini anda akan berlatih membubuhkan efek suara pada element link <a>. Persipan yang mesti dilakukan adalah sebagai berikut:

Siapkan dua sample sound efek, atau anda dapat memasukan satu buah lagu full.

HTML Code:

```
<audio id="soundmenu" preload="auto">
  <source src="sound/menu.ogg"></source>
  <source src="sound/menu.mp3"></source>
  Browser dirimu tak suport sound efek coy!
</audio>
```

```
<audio id="soundmenu2" preload="auto">
  <source src="sound/link.ogg"></source>
  <source src="sound/link.mp3"></source>
  Browser dirimu tak suport sound efek coy!
</audio>
```

```
<div id="tombol" class="animated pulse">
  <a href="doc/home.html" onclick="soundmenu.play()" onhover="soundmenu2.play()"
target="kanvasnav"></img></a>
</div>
```

```
<div id="tombol" class="animated pulse">
  <a href="mahasiswa/datamahasiswa.html" onclick="soundmenu.play()"
target="kanvasnav"></img></a>
</div>
```

```
<div id="tombol" class="animated pulse">
  <a href="dosen/datadosen.html" onclick="soundmenu.play()" target="kanvasnav"></img></a>
```

</div>

JS code

<script>

```
    window.onload=function() {
```

```
        var sound1 = document.getElementById("soundmenu");
```

```
        var sound2 = document.getElementById("soundmenu2");
```

```
        sound1.onclick=function()
```

```
        {soundmenu.play();};
```

```
        sound2.onclick=function()
```

```
        {soundmenu2.play();}
```

```
    };
```

</script>

PERTEMUAN 8 UTS

-Kerjakan dlm waktu 60 menit -- !

1. Siapkan dokumen uts.html dan uts.js (uts.css opsional)
2. Pada HTML, buat sebuah <form>, yg mengandung empat element <input> yakni: nama, alamat, hobi, kontak
tipe input bebas, gunakan yg kalian kuasai saja !
3. Buat sebuah tombol dgn value: kirim atau submit
4. Buat sebuah tombol reset
5. Pada JS:
 - persiapkan variable yg diperlukan.
 - buat function untuk dipanggil ketika tombol kirim/submit di klik
 - gunakan "alert" untuk menampilkan hasilnya
 - tambahkan sedikit pengantar pd output alert, contoh:
Nama anda adalah : xxxxx
Alamat anda adalah : xxxxx
dst

-----Semoga Sukses !!-----

PERTEMUAN 9 Strings Methods

Strings Methods

Method :	Fungsi :
<code>big()</code>	Tercetak lebih besar
<code>blink()</code>	Efek berkedip aktif pada browser Netscape
<code>bold()</code>	Tercetak tebal
<code>charAt(n)</code>	Mengambil karakter ke -n dari string. Index string dimulai dari 0
<code>fixed()</code>	Tercetak fixed-pitch
<code>fontcolor('warna')</code>	Tercetak sesuai warna yang didefinisikan
<code>indexOf('char')</code>	Mengambil nilai indeks dari suatu karakter
<code>italic()</code>	Tercetak miring
<code>link('url')</code>	Menjadikan string hyperlink
<code>small()</code>	Tercetak lebih kecil
<code>strike()</code>	Tercetak dengan coretan
<code>sub()</code>	Tercetak subscript
<code>substring(a,b)</code>	Mengambil karakter dari posisi a sampai b-1
<code>sup()</code>	Tercetak superscript
<code>toLowerCase()</code>	Tercetak huruf kecil
<code>toUpperCase()</code>	Tercetak huruf besar
<code>split("")</code>	Menjadikan string diuraikan/dipisahkan berdasarkan tanda (""). Hasil dari split akan dihasilkan sebuah array dengan indeks 0 untuk string ke 1 dan seterusnya.

Contoh penggunaan methods diatas:

Sisipkan pada HTML script dibawah!

```
<script>
var nama ="Master Pubg";
var panjang=nama.length;
var n=nama.substring(1,4);
var besar=nama.toUpperCase()
```

```

var namakulink=nama.link('shafa.html');

document.writeln('Namaku adalah = '+ nama +'<BR>');
document.writeln('Panjang namaku adalah '+ panjang + ' karakter
<BR>');
document.writeln('method BIG = '+ nama.big() +'<BR>');
document.writeln('method SMALL = '+ nama.small() +'<BR>');
document.writeln('method SUB = '+ nama.sub() +'<BR>');
document.writeln('method SUP = '+ nama.sup() +'<BR>');
document.writeln('method BOLD = '+ nama.bold() +'<BR>');
document.writeln('method ITALIC = '+ nama.italics() +'<BR>');
document.writeln('method FONTCOLOR = '+ nama.fontcolor('red')
+'<BR>');
document.writeln('method LOWERCASE = '+nama.toLowerCase() +'<BR>');
document.writeln('method UPPERCASE = '+ besar +'<BR>');
document.writeln('method SUBSTRING = '+ n +'<BR>');
document.writeln('method STRIKE = '+ nama.strike() +'<BR>');
document.writeln('method CharAT = '+ nama.charAt(3) +'<BR>');
document.writeln('method Link = '+ namakulink +'<BR>');
document.writeln('Index Huruf c = '+ nama.indexOf("c") +'<BR>');
awal=nama.indexOf('V');
akhir=nama.length;
document.writeln('Kata Yang Tersisa = '+ nama.substring(awal,akhir)
+'<BR>');

</script>

```

PERTEMUAN 10 Objek Document

Objek Document

Objek ini digunakan untuk mengakses informasi mengenai dokumen HTML, tampilan output dan memanipulasinya.

Property:	Fungsi:
bgColor	Memberikan warna latar belakang
fgColor	Memberikan warna foreground atau warna huruf
link	Mengakses objek anchor/link(dapat digunakan nama objek anchor/link)
linkColor=warna	Memberikan warna link
alinkColor=warna	Memberikan warna pada active link
vlinkColor=warna	Memberikan warna pada visited link
title=judul window	Memberikan judul/title window
image	Mengakses objek image(dapat digunakan nama objek anchor/link)
forms	Mengakses objek form(dapat digunakan nama objek form)

Method dari objek document meliputi :

Method:	Fungsi:
open()	enciptakan/membuka document HTML
close()	Mengakhiri document HTML
write(output)	Memberikan output ke browser
writeln(output)	Memberikan output ke browser dengan menyertakan perpindahan baris

Contoh Program:

Sisipkan script dibawah pada dokumen HTML!

```
<script>
nama="Ciken";
document.bgColor="blue";
document.fgColor="white";
document.title="Belajar objek Document";
```

```
document.linkColor="red";
document.vlinkColor="green";
document.alinkColor="white";
namakulink=nama.link('ciken.html');
document.writeln('Warna Link = '+ namakulink +'<br/>');
</script>
```

Objek Window

Objek window merupakan objek tertinggi dalam objek Javascript. Objek ini digunakan untuk memanipulasi tampilan jendela dari document HTML.

Property pada Objek window:

Property:	Fungsi
length	Mengetahui jumlah frame pada window
location.href	Mengakses objek location untuk melakukan redirect atau berpindah ke alamat tertentu.
Status=nilai_status	Memberikan nilai status window

Metode-metode untuk Objek window

Method:	Fungsi:
alert(pesan)	Memunculkan messagebox sebuah pesan kesalahan/lainnya
confirm(pesan)	Memunculkan pesan konfirmasi. Method ini akan menghasilkan dua nilai kembalian yaitu true untuk Ok dan false untuk Cancel
prompt(pesan,nilai default)	Memunculkan pesan yang menunggu sebuah input
close()	Menutup jendela aktif
open(url file>windowname,feature)	Membuka jendela baru dengan feature meliputi : toolbar=yes no mengaktifkan toolbar tatus=yes no mengaktifkan window status menubar=yes no mengaktifkan menubar

scrollbars=yes|no mengaktifkan scrollbar
resizable=yes|no jendela resizable
width = ukuran lebar jendela
height = ukuran tinggi jendela

Contoh Program

Sisipkan script dibawah pada dokumen HTML!

```
<script>
```

```
window.status="Welcome";  
window.alert("Selamat Datang");  
var angka=window.prompt("Inputkan Angka ?",0);  
document.write("Angka vaforit anda adalah =<strong>" +angka+ "</strong><br>");  
var tampung=window.confirm("jenis kelamin anda Wanita ?");
```

```
if(tampung)  
    {document.write("Boleh Kenalan donk");}  
else  
    {document.write("Ok dech");}  
window.close();
```

```
</script>
```

contoh program selanjutnya akan dibahas pd pertemuan yg akan datang!

Latihan!

Lihat Dj2.5!

PERTEMUAN 11 window.open method

Contoh penggunaan perintah window.open dan window.location untuk membuka halaman web lain.

Code HTML:

```
<center>
<form method="post">
<input type="button" value="Kunci Jawaban UTS A" onclick="konek1()">
<input type="button" value="Kunci Jawaban UTS B" onclick="konek2()">
</form>
</center>
```

Code JS:

```
<script>

function konek1()
    {window.open("utsb.HTML");}

function konek2()
    {window.location="kunci_jawaban UTS.HTML";}

</script>
```

Latihan !

Buatlah sebuah tampilan web yang mengandung penggunaan objek window.location.href untuk membuka halaman web yang lain !

PERTEMUAN 12 Event

Kejadian (Event)

Even adalah sesuatu yang terjadi pada halaman HTML. Berikut ini terdapat beberapa bentuk kejadian yaitu jika pengguna memuat dokumen, pengguna memasukkan data, pengguna mengklik tombol dan sebagainya. Hal-hal tersebut diatur oleh even.

Semua kejadian pada Javascript dapat anda tangani dengan menentukan kejadiannya. Biasanya kejadian(event) adalah sebuah fungsi, tetapi pada beberapa kasus, kita dapat menuliskan pernyataan-pernyataannya secara langsung.

Berikut ini adalah daftar kejadian(event) pada JavaScript :

Kejadian:	Keterangan:
onClick	Kejadian yang dibangkitkan bila pengguna mengklik sebuah elemen form atau link.
onChange	Dibangkitkan bila informasi masukan pada sebuah elemen form (text, textarea, select) diubah oleh pengguna.
onBlur	Dibangkitkan ketika suatu elemen kehilangan focus masukan, yaitu ketika pengguna menekan tombol <tab> atau mengklik elemen lain form lainnya.
onFocus	Dibangkitkan bila sebuah elemen form menerima focus masukan; yaitu bila pengguna mengklik elemen form tersebut atau menekan tombol <tab> sehingga focus masukan berpindah ke elemen ini.
onAbort	Dibangkitkan bila pengguna menghentikan pemuatan citra (tag) yaitu bila pengguna menekan tombol stop atau mengklik link.
onError	Dibangkitkan bila terjadi kesalahan saat browser memuat dokumen atau citra.
onLoad	Dibangkitkan bila browser selesai memuat document
onUnload	Dibangkitkan bila pengguna keluar dari dokumen
onMouseOver	Dibangkitkan bila kursor mouse berada di atas sebuah link.
onMouseOut	Dibangkitkan bila kursor mouse keluar dari daerah link atau peta citra.
onReset	Dibangkitkan bila pengguna menekan tombol reset

onSelect	Kejadian yang dibangkitkan bila pengguna memilih sebagian atau seluruh teks pada elemen form yang berupa kotak teks
onSubmit	Dibangkitkan ketika pengguna menekan tombol submit.

Penanganan Kejadian (Event)

Berikut ini akan diberikan beberapa contoh program-program yang menggunakan kejadian-kejadian dalam aplikasinya.

Contoh program yang menggunakan even OnClick :

Code HTML:

```
<form>
<input type="button" value="Biru" onClick="warna('lightblue')">
<input type="button" value="Pink" onClick="warna('pink')">
<input type="button" value="Coklat" onClick="warna('burlywood')">
<input type="button" value="Kelabu" onClick="warna('darkgray')">
<input type="button" value="Oranye" onClick="warna('peachpuff')">
<input type="button" value="Putih" onClick="warna('white')">
</form>
```

Terapkan pada lat.js kalian!

Code JS:

```
function warna(pilihan) {
alert("Anda Memilih Warna " + pilihan);
document.bgColor=pilihan;
}
```

Contoh Event Focus dan Blur

Code HTML:

Note: entah kenapa, fungsi onfocus() blm berfungsi ?????? ha hai....

```
<h1 align="center">Latihan Event OnFocus dan OnBlur</h1>
```

```
<hr width="600" color="black" size="10">
```

```
<h3 align="center">R E G I S T R A S I</h3>
```

```
<hr width="600" color="black" size="2">
```

```
<center>
```

```
<form name="f" method="get">
```

```
<table>
```

```
<tr>
```

```
<td width="31%">NIM</td>
```

```
<td width="7%">:</td>
```

```
<td width="62%"><input type="text" name="inim" size="9"
```

```
onFocus="window.status='Silahkan Mengisi NIM Anda';" onBlur="masukannim()"></td>
```

```
</tr>
```

```
<tr>
```

```
<td width="31%">NAMA</td>
```

```
<td width="7%">:</td>
```

```
<td width="62%"><input type="text" name="inama" size="23"
```

```
onFocus="window.status='Silahkan Mengisi Nama Anda';"
```

```
onBlur="masukannama()"></td>
```

```
</tr>
```

```
<tr>
```

```
<td width="31%">ALAMAT</td>
```

```
<td width="7%">:</td>
```

```
<td width="62%"><input type="text" name="ialamat" size="34"
```

```
onFocus="window.status='Silahkan Mengisi Alamat Anda';"
```

```
onBlur="masukanalamat()"></td>
```

```

        </tr>
</table>
<hr width="600" color="black" noshade size="2">
<input type="BUTTON" value="Kirim" Onclick="terimakasih()">
<input type="reset" value="Reset">
</form>
</center>

```

Code JS:

```

function masukannim() {
    if (document.f.inim.value=="")
        {alert("anda belum memasukkan nim");}
}

function masukannama() {
    if (document.f.inama.value=="")
        {alert("anda belum memasukkan nama");}
}

function masukanalamat() {
    if (document.f.ialamat.value=="")
        {alert("anda belum memasukkan alamat");}
}

function terimakasih() {
    if ((document.f.inim.value!="")&&(document.f.inama.value!="") &&
(document.f.ialamat.value!=""))
        {alert("Terima Kasih Telah mengisi Data");}
    else
        alert("Mohon Data Dilengkapi");
}

```

PERTEMUAN 13 Event 2

Contoh program yang menggunakan even onLoad dan onUnload

Sisipkan fungsi onload() dan onunload pada <body>

Untuk mengaktifkan mereka pada saat jendela di load dan saat jendela ditutup.

Pertama-tama pelajari dan ikuti code dibawah, lalu coba pahami alur pemikirannya!

Code HTML:

```
<body onload="tanggal()">
```

```
<h1 align="center">Latihan Event OnLoad dan OnUnLoad</h1>
```

```
<hr width="600" color="black" noshade size="10">
```

```
<h3 align="center">SELAMAT DATANG</h3>
```

```
<center>
```

```
<form name="f">
```

```
  <h4 align="center">Sekarang Menunjukkan Jam</h4>
```

```
  <input type="Text" size="16" name="txtwaktu">
```

```
  <hr width="600" color="black" noshade size="2">
```

```
</form>
```

```
</center>
```

Code JS:

```
function tanggal() {  
  var d = new Date();  
  var y = d.getFullYear();  
  var m = d.getMonth() + 1;  
  var d = d.getDate();
```

```
var t = d + '/' + m + '/' + y + ' ';  
document.f.waktu.value = t;  
alert("Selamat Datang Teman \n Silahkan Masuk Ke Websiteku");  
}
```

Laihan new Date()!

- ambil tahun saja!
- ambil bulan saja!
- ambil tanggal saja!

Latihan!

Buatlah tampilan web yang menggunakan event onload, sehingga saat halaman dimuat akan menampilkan notifikasi hari, tanggal dan tahun !

PERTEMUAN 14 Event 3

Event on submit, seperti yg pernah kita praktekan di awal-awal pertemuan.

Kali ini kita akan mendesain inputan sedikit kompleks, namun jumlah itemnya tak terlalu banyak.

Kemudian mendesain script JS untuk validasi inputan user dan menambahkan action pd form menuju suatu alamat email.

Contoh Code HTML:

```
<form name="f" method="post" action="mailto:femijuang@gmail.com?subject=Form">
```

```
<table align="center">
```

```
<tr bgcolor="gray">
```

```
<td colspan="2" align="center"><h1>G u e s t B o o k</h1></td>
```

```
</tr>
```

```
<tr bgcolor="green">
```

```
<td>Nama Anda :</td>
```

```
<td><input type="text" value="" name="nama" size="30"></td>
```

```
</tr>
```

```
<tr bgcolor="green">
```

```
<td>Email Anda :</td>
```

```
<td><input type="text" value="" name="email" size="30"></td>
```

```
</tr>
```

```
<tr bgcolor="green">
```

```
<td>Komentar Anda :</td>
```

```
<td><textarea name="cs" rows="5" cols="50"></textarea></td>
```

```
</tr>
```

```

<tr bgcolor="orange">
  <td colspan="2" align="center">
    <input type="button" name="thesubmit" value="kirim" onclick="isiform(this.form)">
    <input type="reset" value="hapus">
  </td>
</tr>

<tr bgcolor="gray">
  <td colspan="2" align="center"><b>Terima Kasih Telah Mengisi Guest Book</b>
</td>
</tr>

</table>

</form>

```

JS Code:

```

function isiform(form) {
  isinama(form);
  isiemail(form);
  isikomentar(form);
  kosongkan(form);
}

function kosongkan(form) {
  if((isinama(form) && isiemail(form) && isikomentar(form)))
    {form.submit;}
  if((isinama(form)== false || isiemail(form)== false || isikomentar(form)== false))
    {salahisi(form);}
}

```

```
function salahisi(form) {
    var teks ="Ada Kesalahan Isian :";
    if (isinama(form)== false)
        {teks += "\nNama Anda";}
    if (isiemail(form)==false)
        {teks += "\nEmail Anda";}
    if (isikomentar(form)==false)
        {teks += "\nKomentar Anda";}
    alert(teks);
}
```

```
function isinama(form) {
    if (form.nama.value=="")
        {return false;}
    else
        {return true;}
}
```

```
function isiemail(form) {
    if((form.email.value==" " || form.email.value.indexOf('@',0)==-1) ||
form.email.value.indexOf('.')== -1)
        {return false;}
    else
        {return true;}
}
```

```
function isikomentar(form) {
    if(form.cs.value=="")
        {return false;}
    else
        {return true;}
}
```

-Coba klik kirim tanpa mengisi input apapun!

PERTEMUAN 15 Latihan & Praktek

Pelajari dan praktekan code HTML dan JS dibawah ini.

Tentukan dan analisa bahan atau file yang diperlukan, lalu lengkapi sampai aplikasi berjalan normal!

HTML Code:

```
<!DOCTYPE html>
<!--Halaman Otentifikasi-->
<html>
<head>
<title>Halaman Otentifikasi</title>
<meta char-set="utf-8 ">
<link rel="stylesheet" href="logincss.css">
</head>
<body>

<div class="kontainerutama">

<div class="kontainer1">


<!--form satu-->
<form name="login">
<select id="un">
  <option value="andrian">andrian</option>
</select>

<p>
Password :
<input
type="password"
id="pw"
```

```
size="15"
maxlength="15">
</p>
```

```
<input class="ratakiri" type="button" onclick="check(this.form)" value="masuk"/>
</form>
```

```
</div>
```

```
<div class="kontainer2">

```

```
<!--form dua-->
<form class="dua" name="login">
```

```
<select id="un">
  <option value="erwin">erwin</option>
</select>
```

```
<p>
Password :
<input
type="password"
id="pw"
size="15"
maxlength="15">
</p>
```

```
<input class="ratakanan" type="button" onclick="check(this.form)" value="masuk"/>
</form>
```

```
</div>
```

```
<div class="kontainer3">

```

```
<!--form tiga-->
<form class="tiga" name="login">
<select id="un">
  <option value="oila">oila</option>
</select>
```

```
<p>
Password :
<input
type="password"
id="pw"
size="15"
maxlength="15">
</p>
```

```
<input class="ratakiri" type="button" onclick="check(this.form)" value="masuk"/>
</form>
```

```
</div>
```

```
<div class="kontainer4">

```

```
<!--form empat-->
<form class="empat" name="login">
<select id="un">
  <option value="yustia">yustia</option>
</select>
```

```
<p>
```

```
Password :
```

```
<input
```

```
type="password"
```

```
id="pw"
```

```
size="15"
```

```
maxlength="15">
```

```
</p>
```

```
<input class="ratakanan" type="button" onclick="check(this.form)" value="masuk"/>
```

```
</form>
```

```
</div>
```

```
<div class="kontainer5">
```

```

```

```
<!--form empat-->
```

```
<form class="lima" name="login">
```

```
<select id="un">
```

```
<option value="gue">gue</option>
```

```
</select>
```

```
<p>
```

```
Password :
```

```
<input
```

```
type="password"
```

```
id="pw"
```

```
size="15"
```

```
maxlength="15">
```

```
</p>
```

```
<input class="ratakanan" type="button" onclick="check(this.form)" value="masuk"/>
</form>
```

```
</div>
```

```
<!--kontainer terluar-->
```

```
</div>
```

Javascript Code:

```
<script>
```

```
function check(form)
```

```
{
```

```
if (form.un.value=="gue" && form.pw.value=="tjil" ||
form.un.value=="andrian" && form.pw.value=="stmik" ||
form.un.value=="erwin" && form.pw.value=="stmik" ||
form.un.value=="oila" && form.pw.value=="stmik" ||
form.un.value=="yustia" && form.pw.value=="stmik"
```

```
)
```

```
{
```

```
(confirm("Anda Boleh Masuk"))
```

```
location.href="data/index.html";
```

```
}
```

```
else if (form.un.value=="gue" && form.pw.value!="stmik" ||
form.un.value=="andrian" && form.pw.value!="stmik" ||
form.un.value=="erwin" && form.pw.value!="stmik" ||
form.un.value=="oila" && form.pw.value!="stmik" ||
form.un.value=="yustia" && form.pw.value!="stmik"
```

```
)
```

```
{
```



```

alert("Password Anda Illegal coy!");
history.back();
}

else if (form.un.value!="gue" && form.pw.value!="stmik" ||
form.un.value!="andrian"    && form.pw.value!="stmik" ||
form.un.value!="erwin"      && form.pw.value!="stmik" ||
form.un.value!="oila"       && form.pw.value!="stmik" ||
form.un.value!="yustia"     && form.pw.value!="stmik"
)
{
alert("Nama dan Password Anda Illegal coy!");
history.back();
}

else
{
alert("Nama Anda Illegal coy!");
history.back();
}

}

</script>

</body>

</html>

```

Langkah selanjutnya adalah menambah CSS pada aplikasi diatas!

PERTEMUAN 16 UAS Index Juara Mahasiswa

-Kerjakan dlm waktu 90 menit -- -- !

1. Siapkan dokumen uas.html dan uas.js (uas.css opsional)

2. Pada HTML, buat sebuah <form>, isi dgn element2 dibawah ini!

yakni:

--nama (element <input>),

--negara (gunakan element <select>),

--perolehan juara (input tipe: number)

3. Buat sebuah tombol dgn value: kirim atau submit

4. Buat sebuah tombol reset

5. Siapkan Tiga buah inputan kosong, yg nanti isinya akan diisi setelah diproses oleh javascript

-- Nama

-- Negara

-- Rank

6. Pada JS:

-persiapkan variable yg diperlukan untuk mengambil data dari HTML!

-buat function untuk dipanggil ketika tombol kirim/submit di klik

-buat fungsi logika (if -else - else if) untuk mengecek inputan user.

-khusus inputan perolehan ciken, gunakan pengecekan untuk memastikan data yg user masukan

adalah angka (if (isNaN(xxx).

-rank inputan perolehan juara:

platinum: >50

gold : 30-50

silver : 20-29

bronze : 0-19

-Tampilkan hasil pengolahan data pada tiga inputan kosong yg sudah disiapkan pd HTML!

Note: setelah mengisi form, refresh dahulu, baru click submit!

-----Semoga Sukses!!-----

PERTEMUAN 17 Manipulasi Gambar

Materi lanjutan setelah pertemuan ke-16 pd dokumen pertemuan 1-16.txt

Untuk memuat suatu image, pada Javascript terdapat objek Image. Untuk membuat objek tersebut pendeklarasiannya adalah sebagai berikut :

```
img1 = new Image ()
```

```
img1.src = "pic1.gif"
```

contoh program slideshow sederhana:

Persiapkan sebuah dokumen HTML dan JS.

Persiapkan lima buah gambar (bebas).

Code HTML:

```
<!--propertis center untuk rata tengah, skrng sudah obsolete!!!-->
```

```
<center>
```

```
<!--persiapkan gambar default untuk ditampilkan ketika layar di load
```

```
lalu, beri nama!
```

```
-->
```

```
<p>
```

```
<!--element link kita isi dgn fungsi javascript, perhatikan format penulisannya!
```

```
(-1) berarti nilai dikurangi satu, (1) berarti nilai ditambah satu
```

```
-->
```

```
<a href="JavaScript:slideshow(-1)">Previous</A>
```

```
<a href="JavaScript:slideshow(1)">Next</A>
```

```
</center>
```

Code JS:

```
//persiapkan variable untuk menampung data yg akan di olah JS
var num = 1

//persipakan objeg image javascript
//jgn lupa sertakan url dan ekstensi image-nya!
img1 = new Image ()
img1.src = "image/aa.png"

img2 = new Image ()
img2.src = "image/bb.png"

img3 = new Image ()
img3.src = "image/cc.png"

img4 = new Image ()
img4.src = "image/dd.png"

img5 = new Image ()
img5.src = "image/ee.png"

//Lalu, buat fungsi untuk mengolah data diatas
//kita memasukan parameter (x) pd fungsi, yg mana jdi objek dlm fungsi tsb
//tidak mesti menggunakan "x", kalian bebas menggunakan huruf apeh ajeh!
function slideshow(x) {
    //variable num disinilah yg akan digunakan sbg output diakhir script
    num = num+x

    //karena disini kita hanya menyiapkan lima buah gambar
    //maka kita tentukan batas max atas dan batas max bawah
    //kita dapati bahwa, jumlah keseluruhan gambar ada lima buah, maka
    //batas max atas harus 6 (total +1), agar gambar terakhir turut diperlihatkan
    //pada output nanti. Hal itu disebabkan pd fungsi if pertama
    //num akan dikembalikan bernilai 1 apabila nilainya sudah sama dgn 6
```

```

//sebaliknya pd fungsi if yg kedua, karena terjadi pengurangan (-1), maka
//ketika nilai num = 0, num akan dikembalikan keposisi gambar terakhir, yakni
//gambar ke-5, sehingga num = 5
if (num == 6)
    {num = 1}
if (num == 0)
    {num = 5}

//kemudian terapkan hasil fungsi diatas pada element link
//yg sudah dipersiapkan pd dokumen HTML
document.mypic.src=eval("img"+num+".src")
}

```

Contoh slideshow disertai timer:

Tambahkan script dibawah pada dikahir bagian JS sebelumnya!
Pelajari dan pahami!

Code JS:

```

//membuat fungsi untuk auto slide atas gambar kita
function autoslide() {
    //Ingat, karena kita mempersiapkan lima buah gambar
    //maka perhitungan logikanya harus melibatkan angka diatas lima
    //hal ini agar seluruh gambar ditampilkan sebagai outputnya
    //apabila kita memasukan batas max atas = 5, maka salah satu gambar (yg ke-5)
    //tidak akan turut ditampilkan, karena pada if dibawahnya num kembali
    //bernilai satu ketika num mencapai 5 !!!
    if (num<6)
        {num++};
    if (num==6)
        {num=1};
}

```

```
document.mypic.src=eval("img"+num+".src");  
//memanggil function "autoslide()" setiap 1 detik  
    setTimeout(autoslide,1000)  
}  
  
//untuk membuat fungsi berjalan, kita harus memanggilnya!  
autoslide();
```

PERTEMUAN 18 Array level 2

metode push()

praktekan contoh berikut ini pd scratchpad kalian!

pahami outputnya!

memasukan data ke dlm array kosong:

```
var fruits = [];  
fruits.push('banana', 'apple', 'peach');  
fruits;
```

memasukan data ke dlm array yg berpenghuni:

```
var fruits = ['jengkol', 'pete'];  
fruits.push('banana', 'apple', 'peach');  
fruits;
```

memasukan data ke dlm sub array :

```
var fruits = ['jengkol', 'pete', ['sambel']];  
fruits[2].push('banana', 'apple', 'peach');  
coba panggil sub array nya!  
coba panggil salah satu item sub arraynya!
```

Metode pop()

praktekan pada scratchpad !

```
var fruits = ['jengkol', 'pete', ['sambel']];  
fruits.pop();  
fruits;
```

perhatikan output nya, pop() akan menghapus bagian terakhir array !

untuk menambahkan item pada array pd posisi pertama, gunakan:

```
fruits.unshift('nama item');
```

untuk menggeser item posisi pertama, gunakan:

```
fruits.shift();
```

Metode splice()

metode yg digunakan untuk memposisikan item baru dan menghapus suatu item pd obeejek tsb.

contoh:

```
arrayGue.splice(2, 2, 'pete', 'sambel');
```

Penjelasan:

-digit pertama berarti memposisikan item baru (pete & sambel) setelah index item ke-2 pada variable.

Maka, pete dan sambel berturut-turut akan berada pd index ke-3 dan ke-4.

-digit ke-2 artinya menghapus item setelah index ke-2 pd variable tersebut. Dengan demikian, item pd index

ke-3 dan seterusnya akan dihapus.

-Gunakan angka nol pd digit ke-2, apabila kita tidak memerlukan penghapusan

contoh : `arrayGue.splice(2, 0, 'pete', 'sambel');`

Chess Board of Array

Perhatikan struktur array dibawah ini! Ketik pada scratchpad! Tes keluarannya menggunakan display!

```
var board = [
```

```

['R','N','B','Q','K','B','N','R'],
['P','P','P','P','P','P','P','P'],
[' ',' ',' ',' ',' ',' ',' ',' '],
[' ',' ',' ',' ',' ',' ',' ',' '],
[' ',' ',' ',' ',' ',' ',' ',' '],
[' ',' ',' ',' ',' ',' ',' ',' '],
['p','p','p','p','p','p','p','p'],
['r','n','b','q','k','b','n','r'] ];

```

Coba kalian identifikasi,

- 1.Array tsb memiliki berapa anggota ?
- 2.Sebutkan berapa anggota yg dimiliki sub array !
- 3.Mengapa sub-sub array tsb harus berada dalam [dan] ?
- 4.Bisa ga cuy, kalian memindahkan salah satu item sub array di lokasi yg berbeda ?

```

// contoh menggerakan salah satu item ke tempat yg berbeda
//pahami strukturnya !
board[4][4] = board[6][4];
board[6][4] = ' ';
board;

```

Latihan !

- Gerakan tiga item yg berbeda, ke tiga lokasi yg berbeda !
- Jika sudah selesai maka, Ciken !

PERTEMUAN 19 Function + validation lv2

Sebelumnya kita telah mempraktekan validasi untuk isian angka. kita menggunakan `if(isNaN(nama variabel))` untuk memastikan angka yg dimasukan user adalah angka!

Selanjutnya kita akan bertemu kata kunci/struktur script lain untuk keperluan validasi. Pelajari dan pahami struktur dibawah!

1.The power of typeof

```
function konfirmasi() {  
  if (typeof nama = 'string')  
    {statement yg dieksekusi disini!}  
}
```

2.isLetter

```
if (isLetter(nama variabel)) {  
  statement yg dieksekusi disini!  
}
```

3.The power of typeof + switch

```
function checkType(n) {  
  console.log('.....');  
  var status;  
  var t = typeof n;  
  switch (t) {  
    case 'string' :  
      status = 'anda memasukan string';  
      break;  
    case 'number' :  
      status = 'anda memasukan angka';
```

```
        break;
    default :
        status = 'anda memasukan string';
    }
}
```

PERTEMUAN 20 self invoking function & OOP:inheritance (prototypal)

JS memiliki fasilitas pengaktifkan otomatis atas suatu fungsi, yg mana hal tsb tidak terdapat

pada bahasa pemrograman lain!

perhatikan contoh dibawah!

```
(function () {  
    console.log('aktif sendiri nih!');  
})();
```

Latihan!

tulis ulang code mirip seperti contoh diatas, sampai kalian hafal!

OOP:inheritance (prototypal)

JS berintikan objek. objek2 tsb sering kita gunakan dan dapat diwariskan sifat/strukturnya.

Hal ini berguna ketika aplikasi harus mempunyai kemampuan men-generate objek baru yg tidak tentu jumlahnya.

contoh:

```
var lalab = {  
    nama = 'jengkol',  
    efek = 'meraih ciken',  
    jumlah = 20  
};  
console.log(lalab);
```

contoh diatas adalah suatu objek yg sering disebut sbgai prototype object. Artinya objek tersebut memiliki sifat inheritance, untuk dipergunakan membuat objek baru. Objek baru tersebut bisa bersifat penambahan data, pengurangan data dan lain sebagainya.

Mati kita mulai memanfaatkan objek diatas:

```
//merubah properti objek
```

```
lalab.nama = 'pete';
```

```
lalab.efek = 'mempercepat ciken';
```

```
//menghapus properti
```

```
delete(lalab.efek);
```

```
//menambah properti baru
```

```
lalab.harga = 10000;
```

kemudian mari kita tes hasilnya pd web console:

```
console.log(lalab);
```

Latihan!

Dengan kekuatan bulan....eh dengan objek lalab diatas
coba tambahkan dua properti baru
dan hapus satu properti yg sudah ada!

PERTEMUAN 21 loop lv.2

Kita akan berlatih menggunakan switch, untuk melakukan pengecekan beberapa kategori atas dasar inputan/pilihan user.

pelajari contoh dibawah dgn seksama:

```
var kota = false;
for (var i = 0; i <= 4; i++) {
  if (checkKota === kota terbersih[i]) {
    kota = true;
    alert('Salah satu kota terbersih!');
  }
}
```

```
if (kota === false) {
  alert('Tidak ada dalam daftar!');
}
```

contoh berikutnya adalah bentuk nested loop, alias loop didalam loop:

```
var namaDepan    = ["Bapak", "akang"];
var namaBelakang = ["Suketi", "Katemi", "Otok"];
var namaLengkap = [];
for (var i = 0; i < namaDepan.lenght; i++) {
  for (var j = 0; j < namaBelakang; j++) {
    namaLengkap.push(namaDepan[i] + namaBelakang[j]);
  }
}
```

NOTE:

i adalah outer loop counter

j adalah inner loop counter

setiap item pd var namaDepan akan di pasangkn dgn setiap item pd var namaBelakang yang hasilnya akan disimpan pd var namaLengkap. Oleh karena itu, kita akan mendapat kombinasi nama sbb:

Bapak Suketi, Bapak Katemi dan Bapak Otok

Akang Suketi, Akang Katemi dan Akang Otok

Latihan!

Gunakan struktur script seperti diatas, namun gunakan data data variable yg berbeda!

PERTEMUAN 22 Data type conversion & manipulation

metode toLowerCase dan toUpperCase

contoh:

```
//merender menjadi huruf kecil
```

```
myData = myData.toLowerCase;
```

```
//merender menjadi huruf kapital
```

```
myData = myData.toUpperCase;
```

Latihan!

Praktekan ke dua metode diatas

buat sebuah variable berisi string

kemudian gunakan toLowerCase dan toUpperCase!

Manipulation string data

Disini kita berlatih mengatur masukan data dari user,

masukan user akan kita ubah menjadi berhuruf kapital pd huruf pertama, kemudian huruf kedua

dst menjadi huruf kecil. Kita membutuhkan tehnik konversi diatas dan tehnik slice().

contoh penggunaannya:

```
myData = userInput.slice(0, 1);
```

digit pertama artinya potong/ambil data pada index pertama [0]

digit kedua artinya berhenti ambil/potong sebelum data index kedua [1]

Latihan!

Buat output dgn huruf pertama kapital, huruf berikutnya huruf kecil semua.

Gunakan tehnik toLoweCase, toUpperCase dan slice() untuk mengerjakan latihan ini !

Kita dapat menggunakan prompt() pada variable, perhatikan contoh dibawah!

```
var month = prompt("Enter a month");
var charsInMonth = month.length;
if (charsInMonth > 3) {
  monthAbbrev = month.slice(0, 3);
  alert("sekarang bulan " + monthAbbrev);
}
```

Latihan!

Buka dan gunakan data latihan sebelumnya pada file JS pertemuan 1-16 yg berisi latihan menjemput data dari radio button, kemudian ditampilkan melalui alert. Gunakan tehnik pengisian form melalui prompt seperti contoh diatas!

PERTEMUAN 23 Controlling the length of decimals

pelajari contoh kasus berikut !

The price of the item is \$9.95. The sales tax is 6.5% of the price. You combine the two numbers to get the total.

```
var total = price + (price * taxRate);
```

The variable total now has a value of 10.59675.

But that isn't what you're going to charge the customer, is it? You're going to charge her an amount rounded off to two decimal places: \$10.60.

Here's how to do it.

```
var prettyTotal = total.toFixed(2);
```

The statement above rounds the number represented by total to 2 places and assigns the result to the variable prettyTotal . The number inside the parentheses tells JavaScript how many places to round the decimal to.

To shorten a number to no decimals, leave the parentheses empty.

```
var prettyTotal = total.toFixed();
```

Selanjutnya, kita bisa menambahkan tanda Rupiah seperti berikut:

```
var hasilAkhir = "Rp." + prettyTotal;
```

Unfortunately, the toFixed method comes with a surprise inside. If a decimal ends in 5, it usually rounds up, as you would expect. But, depending on the browser, sometimes it rounds

down! If, for example, you apply the method to 1.555, specifying 2 decimal places, it may give

you "1.56". Or it may produce "1.55".

There are sophisticated ways to address the problem. Here's an inelegant fix that uses

methods you already know and understand.

```
var str = num.toString();  
if (str.charAt(str.length - 1) === "5") {  
  str = str.slice(0, str.length - 1) + "6";  
}  
num = Number(str);  
prettyNum = num.toFixed(2);
```

If the decimal ends in a 5, the code changes it to a 6 so the method is forced to round up when it should. Here's what's happening, line-by-line:

1. Converts the number to a string and assigns it to the variable str .
2. Checks to see if the last character is "5".
3. If so, slices off the "5" and appends "6".
4. --
5. Converts it back to a number.
6. Rounds it to 2 places.

Note:

cara convert number ke string : `var num = nomor.toString();`

cara convert string ke number : `var str = Number(text);`

Latihan !

1. Harga sebuah jengkol Rp.10000
2. Setiap bulan Puasa di diskon 45%
3. Buatlah proses penghitungan harga sebuah jengkol setelah di diskon !

PERTEMUAN 24 UTS

1. Buat sebuah form pada HTML yg berisi tiga inputan tipe text!
2. Tambahkan tombol untuk merender inputan user!
3. Pada JS, buat sebuah function yg mengandung variable yg diperlukan untuk mengolah inputan dari form HTML!
4. Olah data inputan sampai menghasilkan output dgn huruf pertama kapital, huruf berikutnya huruf kecil semua.
Gunakan tehnik toLowerCase, toUpperCase dan slice().
5. Tampilkan hasil dgn alert("");

PERTEMUAN 25 Latihan menunjukan waktu dengan function

Materi dibawah juga ditujukan pada pengasahan lebih lanjut atas materi variable dan manipulasi variable.

Kita dapat memanggil bagian dari new Date() secara keseluruhan atau satu-persatu. Dengan menggunakan method yg terdapat pd fungsi new Date(), kita dapat memanggil setiap bagian satu persatu.

Contoh:

```
var now = new Date();  
var theHr = now.getHours();  
var theMin = now.getMinutes();
```

Note:

var now mengandung semua unsur waktu dan penanggalan.
var theHr hanya mengambil objek jam (hour) dalam new Date().
var theMin hanya mengambil objek menit dalam new Date().

Latihan!

Buat tiga variable,

- 1.var 1 untuk memanggil fungsi Date()
- 2.var 2 mengambil objek hour
- 3.var 3 mengambil objek menit
- 4.tampilkan hasil menggunakan alert("");
- 5.Masukan itu semua kedalam sebuah function()

Contoh:

```
function tellTime() {  
  var now = new Date();  
  var theHr = now.getHours();  
  var theMin = now.getMinutes();  
  alert("Current time: "+ theHr + ":" + theMin);  
}
```

PERTEMUAN 26 Object & Object Constructor

Contoh Objek:

```
var plan1 = {  
  name      : "Basic",  
  price     : 3.99,  
  space     : 100,  
  transfer  : 1000,  
  pages     : 10,  
  discountMonths: [6, 7]  
};
```

Note:

- The code begins like any variable definition, with the keyword `var` , followed by the object name,
and then an equal sign.
- But then, instead of a value, there's a curly bracket, whose twin comes at the end of the block.
- Each property begins with a name, followed by a colon, and then a value.
- Each property definition except the last ends with a comma.
- The closing curly bracket is followed by a semicolon, because of the equal sign in the first line.

untuk menggunakan objek diatas, cukup panggil nama objek plus sub-objeknya!

Contoh:

```
plan1.name;
```


Penggunaan objek sangat berguna apabila kita memiliki data yang kompleks atau ketika harus menyediakan fasilitas untuk men-generate objek baru. Oleh karena itu, dibutuhkan objek konstruktor untuk mempermudah hal tersebut.

Contoh objek konstruktor:

```
function Plan(name, price, space, transfer, pages) {  
  this.name = name;  
  this.price = price;  
  this.space = space;  
  this.transfer = transfer;  
  this.pages = pages;  
}
```

This is the calling code that creates the object for the Basic plan.

```
var plan1 = new Plan("Basic", 3.99, 100, 1000, 10);
```

This would be just a regular function call if it weren't for that new . It's the keyword that tells JavaScript to create a new object. The name of the new object is plan1 . Its properties are enumerated inside the parentheses.

Now it's easy to mass-produce as many objects as you want, using the same pattern.

```
1 var plan1 = new Plan("Basic", 3.99, 100, 1000, 10);  
2 var plan2 = new Plan("Premium", 5.99, 500, 5000, 50);  
3 var plan3 = new Plan("Ultimate", 9.99, 2000, 20000, 500);
```

It's common to use the same names for parameters and properties, but you don't have to. You could write:

```
function Plan(name, price, space, transfer, pages) {  
  this.doc = name;  
  this.grumpy = price;  
  this.sleepy = space;  
  this.bashful = transfer;  
  this.sneezy = pages;  
}
```

Latihan!

1. Buat sebuah objek, isi bebas!
2. Buat sebuah object constructor, isi bebas!

PERTEMUAN 27 try and catch

If you run the following code, nothing will happen.

```
function greetWorld() {  
  var greeting = "Hello world!";  
  aler(greeting);  
}
```

The keyword alert , misspelled aler , breaks the code. But JavaScript doesn't tell you what's wrong. It's like a car that won't start. Is it the battery? The fuel line? A spark plug? Of course, in the small piece of code above, it's easy to spot the problem. But what if the function runs 40 lines and calls three other functions? It would be nice if JavaScript reported the specific problem to you instead of just breaking. Well, that's what try...catch is for. Here's how it works.

```
function greetWorld() {  
  try {  
    var greeting = "Hello world!";  
    aler(greeting);  
  }  
  catch(err) {  
    alert(err);  
  }  
}
```

The original operational code is wrapped in the try block. The mis-formed aler causes a JavaScript error, which is passed to the catch block. An alert displays the error that caused

the problem.

When you wrap some operational code in a try block, you say to JavaScript, "Try to execute this code. If there's an error, stop executing, and go to the catch block to see what to

do." In the example above, the error thrown by JavaScript is assigned to the parameter `err`,

and that value is displayed in an alert:

"ReferenceError: Can't find variable: `aler`"

Okay, so in this case JavaScript doesn't give you exactly the information you want.

Instead of letting you know that you left a "t" off the `alert` keyword, it tells you it has encountered a variable that hasn't been defined. But close enough. JavaScript has pointed you

in the right direction.

Note that `try` and `catch` are always paired. Without `try`, an error won't trigger `catch`.

Without `catch`, JavaScript won't know what to do when it throws an error.

Some more things to keep in mind:

The code inside the `try` and the `catch` blocks is wrapped in curly brackets.

The functional code inside the `try` block is indented.

The error parameter, in this case `err`, can take any legal variable name.

In this example, an alert displays when an error occurs. But you could do something else.

The `try` and `catch` pair have limited usefulness. For example, in the example above, if you omit a parenthesis, bracket, or quotation mark, or if you have too many of them, no alert

will display. The code will just break, and nothing will happen. The `try` and `catch` approach is useful mainly for spotting variables that haven't been defined or, as in this case, errors that

JavaScript interprets as variables that haven't been defined.

Note:

Kita dapat menentukan jenis error dgn memanfaatkan throw.

Perhatikan contoh dibawah!

```
function checkPassword() {  
  try {  
    var pass = document.getElementById("f1").value;  
    if (pass.length < 8) {  
      throw "Please enter at least 8 characters.";  
    }  
    if (pass.indexOf(" ") !== -1) {  
      throw "No spaces in the password, please.";  
    }  
    var numberSomewhere = false;  
    for (var i = 0; i < pass.length; i++) {  
      if (isNaN(pass(i, i+1)) === false) {  
        numberSomewhere = true;  
        break;  
      }  
    }  
    if (numberSomewhere === false) {  
      throw "Include at least 1 number.";  
    }  
  }  
  catch(err) {  
    alert(err);  
  }  
}
```

PERTEMUAN 28 Prompt the name (inputs)

Pengisian data terhadap variable dapat dilakukan langsung oleh kita pada dokumen JS, atau mengambil melalui suatu input pada dokumen HTML, atau mengambil data melalui perintah prompt().

prompt akan menyajikan user jendela berisikan daerah inputan untuk user tsb. Nah, apa yg user isi disitu, dapat kita tangkap dan terapkan pada suatu objek baik pd JS maupun pd HTML.

Pada intinya, prompt() mirip dgn alert() !

contoh:

```
prompt("siapa eloeh ?");
```

coba praktekan pada webconsel/scracthpad !

Contoh dibawah ini memperagakan prompt pada variabelen sehingga, isi variable adalah apa yg diisikan pada jendela prompt. Lakukan hal yg sama sembari menyimak penjelasan!

//HTML

//menyiapkan objek untuk menampilkan hasil inputan user setelah di oleh JS

```
<p> Player 1 :</p>
```

//CSS

//tampilkan bordernya, tambahkan css lain agar <p> menyerupai button

//JS

//membuat variable yg diarahkan pd objek di HTML, dalam hal ini elemen <p>

```
var user = document.querySelector('p');
```

```
//menerapkan event listener pada variable berupa 'click'  
//dan mengatifkan function bernama userinput  
user.addEventListener('click', userinput);  
  
//membuat fungsi userinput()  
//berisi prompt kpd user, kemudian inputan dari user disimpan pd var name  
//pd akhirnya, isi var name digabungkan kedalam var user, yang mana var user  
//sudah diarahkan pada element <p> pada HTML  
function userinput() {  
    var name = prompt('Masukin Nama Loe!');  
    user.textContent = 'Player 1:' + name;  
}
```

Latihan!

- Gunakan salah satu latihan pada pertemuan 1-16, misal pengecekan pria, wanita atau banci
- copy paste menjadi latihan yg baru
- Gunakan tehnik inputan prompt untuk menjemput data dari user!

PERTEMUAN 29 Create element on Click !

Materi dibawah memanfaatkan beberapa tehnik:

1.document.createElement

Untuk membuat element pd HTML melalui JS

2.targetvariable.textContent = "isi disini;;

untuk menerapkan konten pada target variabel

3.document.body.appendChild(targetvariable);

Untuk menerapkan target variable pada bagian body di HTML

4.document.querySelectorAll('element/class/id');

memilih/menargetkan slh satu objek pd HTML

kata kunci ALL berarti memilih seluruh objek pd tipe yg sama dalam HTML

5.fungsi loop for

Mari kita langsung kedapur

Persiapkan document HTML dan JS, peragakan contoh dibawah ini, lalu pahami!

HTML:

```
<button id="aa"> Bukan Pabrik Paragraf</button>
```

JS:

```
function mesinParagraf() {  
    var para = document.createElement('p');  
    para.textContent = 'Kamu sudah Menyentuh Ku!';  
    document.body.appendChild(para);  
}
```

```
var tombol = document.querySelectorAll('div button');
```

```
for (var i = 0; i < tombol.length; i++) {
```



```
tombol[i].addEventListener('click', mesinParagraf);  
}
```

PERTEMUAN 30 kombinasi alert dengan prompt

HTML:

```
<button>Press me</button>
```

JS:

```
var tombol = document.querySelector('button');
```

```
tombol.onclick = function() {  
  var name = prompt('Nama loe siapa?');  
  alert('Yuhuu ' + name + ', kamu lagi, kamu lagi hadeuh');  
}
```

*/*perhatikan dimana spasi kosong di berikan pada line 6!!*/*

Materi Pemanasan Game 1:

HTML:

```
<title>Silly story generator</title>
```

```
<style>
```

```
body {
```

```
  font-family: helvetica, sans-serif;
```

```
  width: 350px;
```

```
}
```

```
label {
```

```

    font-weight: bold;
}
div {
    padding-bottom: 20px;
}
input[type="text"] {
    padding: 5px;
    width: 150px;
}
p {
    background: #FFC125;
    color: #5E2612;
    padding: 10px;
    visibility: hidden;
}
</style>
</head>

<body>
    <div>
        <label for="customname">Enter custom name:</label>
        <input id="customname" type="text" placeholder="">
    </div>
    <div>
        <label for="us">US</label><input id="us" type="radio" name="ukus" value="us"
checked>
        <label for="uk">UK</label><input id="uk" type="radio" name="ukus" value="uk">
    </div>
</div>
<button class="randomize">Generate random story</button>
</div>
<!-- Thanks a lot to Willy Aguirre for his help with the code for this assessment -->
<p class="story"></p>

```

JS:

```
var customName = document.getElementById('customname');  
var randomize = document.querySelector('.randomize');  
var story = document.querySelector('.story');
```

```
function randomValueFromArray(array){  
  var random = Math.floor(Math.random()*array.length);  
  return array[random];  
}
```

```
var storyText = 'It was 94 fahrenheit outside, so :insertx: went for a walk. When they got  
to :inserty:, they stared in horror for a few moments, then :insertz:. Bob saw the whole  
thing, but was not surprised — :insertx: weighs 300 pounds, and it was a hot day.';  
var insertX = ['Willy the Goblin','Big Daddy','Father Christmas'];  
var insertY = ['the soup kitchen','Disneyland','the White House'];  
var insertZ = ['spontaneously combusted','melted into a puddle on the sidewalk','turned  
into a slug and crawled away'];
```

```
randomize.addEventListener('click', result);
```

```
function result() {  
  var newStory = storyText;  
  
  var xItem = randomValueFromArray(insertX);  
  var yItem = randomValueFromArray(insertY);  
  var zItem = randomValueFromArray(insertZ);  
  
  newStory = newStory.replace(':insertx:',xItem);  
  newStory = newStory.replace(':insertx:',xItem);  
  newStory = newStory.replace(':inserty:',yItem);  
  newStory = newStory.replace(':insertz:',zItem);  
  
  if(customName.value !== "") {
```

```
var name = customName.value;
newStory = newStory.replace('Bob',name);
}

if(document.getElementById("uk").checked) {
    var weight = Math.round(300*0.0714286) + ' stone';
    var temperature = Math.round((94-32) * 5 / 9) + ' centigrade';
    newStory = newStory.replace('94 fahrenheit',temperature);
    newStory = newStory.replace('300 pounds',weight);
}

story.textContent = newStory;
story.style.visibility = 'visible';
}
```

PERTEMUAN 31 Final Lesson, Materi pemanasan Game tebak angka

Ketik dan coba pahami baris perbaris code javascript dibawah ini!

HTML:

```
<p class="guesses"></p>
```

```
<p class="lastResult"></p>
```

```
<p class="lowOrHi"></p>
```

```
<label for="guessField">Enter a guess: </label><input type="text" id="guessField"
class="guessField">
```

```
<input type="submit" value="Submit guess" class="guessSubmit">
```

JS:

```
var randomNumber = Math.floor(Math.random() * 100) + 1;
```

```
var guesses = document.querySelector('.guesses');
```

```
var lastResult = document.querySelector('.lastResult');
```

```
var lowOrHi = document.querySelector('.lowOrHi');
```

```
var guessSubmit = document.querySelector('.guessSubmit');
```

```
var guessField = document.querySelector('.guessField');
```

```
var guessCount = 1;
```

```
var resetButton;
```

```
/*automatically put the text cursor into the <input> text field as soon as the page loads,*/
guessField.focus();
```

```

function checkGuess() {
  var userGuess = Number(guessField.value);
  if (guessCount === 1) {
    guesses.textContent = 'Previous guesses: ';
  }
  guesses.textContent += userGuess + ' ';

  if (userGuess === randomNumber) {
    lastResult.textContent = 'Congratulations! You got it right!';
    lastResult.style.backgroundColor = 'green';
    lowOrHi.textContent = '';
    setGameOver();
  } else if (guessCount === 10) {
    lastResult.textContent = '!!!GAME OVER!!!';
    setGameOver();
  } else {
    lastResult.textContent = 'Wrong!';
    lastResult.style.backgroundColor = 'red';
    if (userGuess < randomNumber) {
      lowOrHi.textContent = 'Last guess was too low!';
    } else if (userGuess > randomNumber) {
      lowOrHi.textContent = 'Last guess was too high!';
    }
  }
}

guessCount++;
guessField.value = '';
guessField.focus();
}

guessSubmit.addEventListener('click', checkGuess);

```

```
function setGameOver() {  
    guessField.disabled = true;  
    guessSubmit.disabled = true;  
    resetButton = document.createElement('button');  
    resetButton.textContent = 'Start new game';  
    document.body.appendChild(resetButton);  
    resetButton.addEventListener('click', resetGame);  
}
```

```
function resetGame() {  
    guessCount = 1;  
  
    var resetParas = document.querySelectorAll('.resultParas p');  
    for (var i = 0 ; i < resetParas.length ; i++) {  
        resetParas[i].textContent = "";  
    }  
}
```

```
resetButton.parentNode.removeChild(resetButton);
```

```
guessField.disabled = false;  
guessSubmit.disabled = false;  
guessField.value = "";  
guessField.focus();
```

```
lastResult.style.backgroundColor = 'white';
```

```
randomNumber = Math.floor(Math.random() * 100) + 1;  
}
```

PERTEMUAN 32 UAS

Buat aplikasi penilaian akhir mahasiswa !

Nilai terdiri atas:

- absen (20%)
- Tugas (10%)
- UTS (35%)
- UAS (35%)

1. Gunakan teknik prompt untuk menjemput data
2. Atau gunakan teknik prompt+onclick untuk menjemput data
3. Tampilkan hasilnya pada HTML atau dgn alert()
4. Hasil akhir berupa angka dan huruf, contoh: 87 (A) atau A (87)

REFERENSI

Buku:

Darren Jones. JavaScript Novice to Ninja. 2014. Australia: Sitepoint Pty.Ltd.

Rex Van Der Spuy. 2012. Foundation Game Design With HTML5 & Javascript.
Fiendsoft:Appress.

Steve Suehring. 2013. JavaScript Step by Step. USA:MSPress.

Website:

<https://id.wikipedia.org/wiki/JavaScript>

<https://www.javascript.com/>

<https://www.w3schools.com/js/>