**Javascript**

**Pemrograman sisi client dan pemrogrman sisi server**

Pada pemrograman web, dikenal dengan 2 istilah yaitu :

1. **Pemrograman sisi client (*client – side programming*)**

*client – side programming* adalah proses pembuatan kode – kode program yang dikenal dengan istilah script yang nantinya akan dieksekusi oleh web browser. Script – script yang dibuat di *client – side programming* pada umumnya disisipkan di dalam kode HTML. Para pengembang web membuat script – script di *client – side programming* menggunakan Bahasa javascript. Karena script – script dalam *client – side programming* akan dieksekusi oleh web browser, sehingga peranan web browser sangatlah penting. Maka web browser yang di pilih harus mendukung Bahasa pemrograman yang digunakan dalam menulis script.

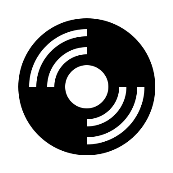
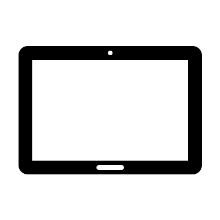
1. **Pemrograman sisi server (*server – side programming*).**

Berbeda dengan *client – side programming* kode – kodenya di eksekusi oelh web browser, dalam *server – side programming* kode – kode program akan dieksekusi oleh *interpreter* yang terpasang di web bowser. Script di *server – side programming* akan dieksekusi jika ada permintaan dokumen dari *user* ke *web server*. Selanjutnya script diterjemahkan ke dalam format Bahasa yang dimengerti oleh web browser yaitu Bahasa HTML dan dikirim lagi ke user. Sehingga dokumen yang diterima oleh user bukan lagi berupa script, melainkan dokumen yan sudah berupa HTML. Ditampilkan di layar *web browser* sebagai halaman web. Script pada *server – side programming* dapat dikembangkan menggunakan Bahasa seperti PHP, perl, ASP dan lain – lain.

Walaupun *client – side programming* dan  *server – side programming* memiliki fungsi yang berbeda, pada kenyataannya dalam pengembangan suatu sistem berbasis web keduanya tetap dipakai untuk dapat saling bekerja sama. Contoh : script yang berada di *client – side programming*digunakan untuk melakukan validasi input dari user sedangkan *server – side programming* digunakan untuk memproses data tersebut dan menyimpannya ke dalam flat file atau RDBMS tertentu. Berikut ini adalah gambar hubungan antara *client – side programming* dan *server – side programming* :

Client (web browser)

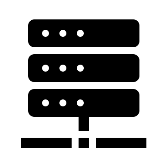
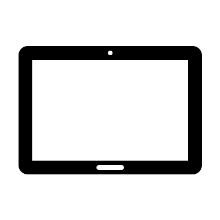
PHP engine

Shooping\_chart.php

Client (web browser)

server

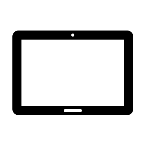
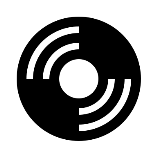
 

Shooping\_chart.asp

Homepage.html

Client (web browser)

ASP engine



Gambar 1 hubungan antara *client – side programming* dan *server – side programming*

Sumber (raharjo, 2013)

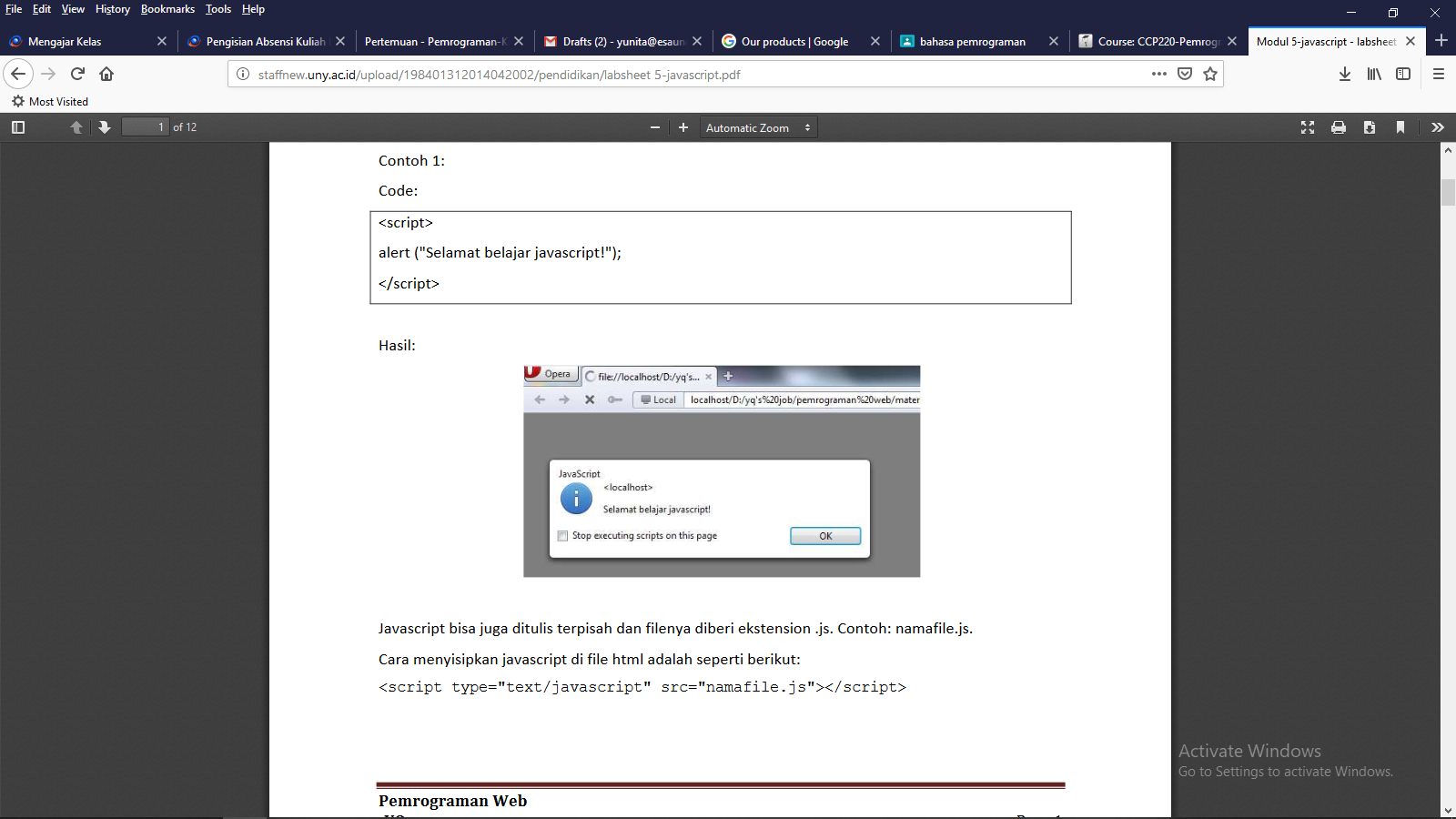
**Apa itu Javascript ?**

Javascript adalah bahasa yang berfungsi untuk membuat script – script program yang dapat dikenal dan dieksekusi oleh web browser dengan tujuan untuk menjadikan halaman web lebih bersifat interaktif. Javascript dan java merupakan dua hal yang konsepnya sangat berbeda, meskipun ada kesamaan dalam penulisan sintaksnya. Javascript dikembangkan oleh Netscape dan merupakan Bahasa yang bersifat terbuka (*open*) sehingga setiap orang dapat menggunakannya tanpa harus membeli lisensi.

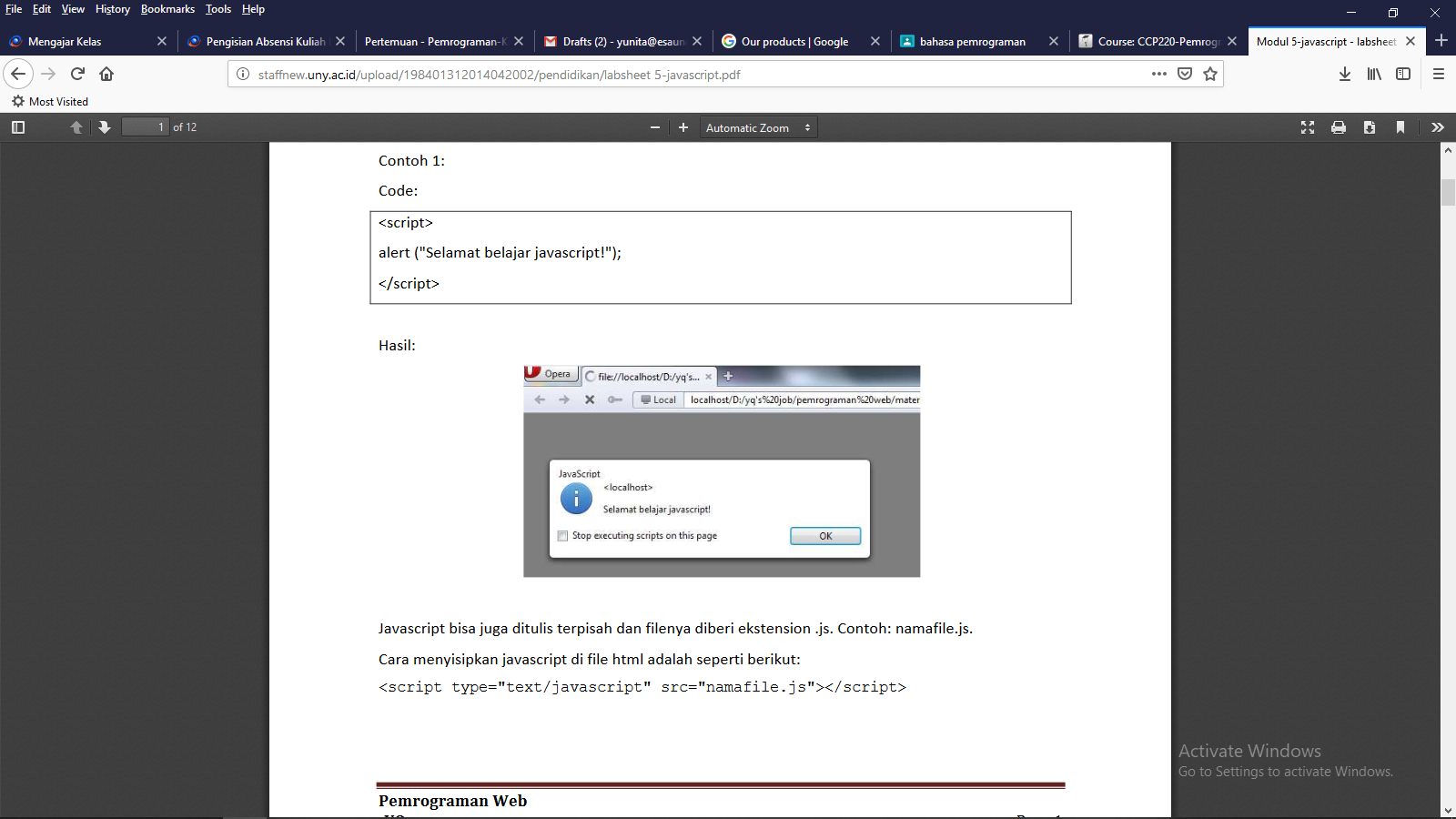
Contoh – contoh dalam penerapan Javascript, adalah :

* Untuk menampilkan pesan peringatan atau konfirmasi ke *user*
* Untuk menampilkan *popup window*
* Untuk membuat menu dropdown
* Untuk melakukan validasi pada saat user memasukkan data dalam suatu form
* Untuk menampilkan tanggal dan waktu

Kode javascrip ditulis di antara tag <script> dan </script> dan bias disisipkan di kode – kode HTML, berikut bentuk penerapannya :



Tampilan pada halaman web :



**JavaScript sebagai bahasa berorientasi pada obyek**

* **Properti**

Properti adalah atribut dari sebuah objek. Contoh, objek mobil punya properti warna mobil.

Penulisan :

Nama\_objek.nama\_properti = nilai window.default

Status = ”Selamat Belajar JavaScript”;

* **Metode**

Metode adalah suatu kumpulan kode yang digunakan untuk melakukan sesuatu tindakan terhadap objek.

Penulisan :

Nama\_objek.nama\_metode(parameter) document.write (”Hallo”)

**Letak JavaScript dalam HTML**

Skrip Javascript dalam dokumen HTML dapat diletakkan pada :

1. Bagian Head
2. Bagian Body (jarang digunakan).

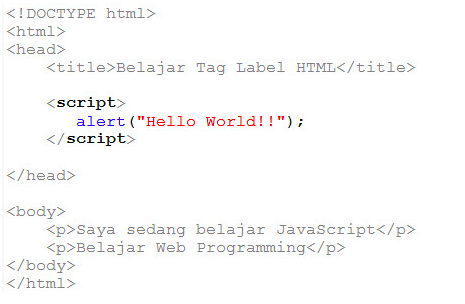
**Cara penulisan kode Javascript di HTML**

Terdapat 3 cara dalam menulis kode Javascript di HTML, yaitu :

1. **embedded** yaitu dengan menempelkan kode Javascript ke dalam file HTML dengan cara menambahkan tag <script> di dalam tag <head>

contoh :

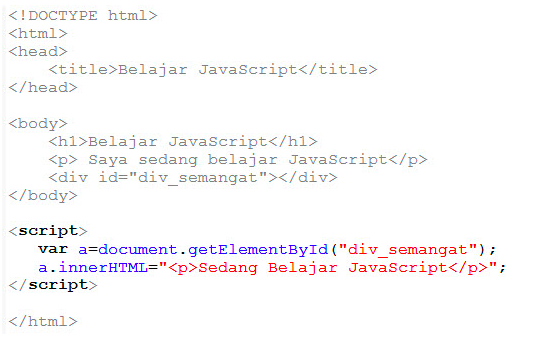
**file .html**



1. **inline** yaitu dengan menuliskan kode Javascript ke dalam atribut HTML. Cara ini biasanya digunakan untuk memanggil suatu fungsi pada event tertentu.

Contoh :

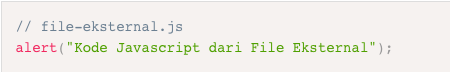
**File .html**

****

1. **eksternal** yaitu dengan menuliskan kode Javascript secara terpisah dengan file HTML. Biasanya digunakan dalam proyek – proyek besar, dimana penggunaan ini diyakin memiliki keunggulan yaitu mudah untuk di modifikasi / berubah. File yang dibuat dua yaitu ekstensi .html dan ekstensi .js (penyimpanan file javascript).

Contoh :

**File .js**

****

**File .html**

****

**Dasar – dasar pemrograman javascript**

Berikut ini adalah dasar – dasar pemrograman javascript meliputi:

* **pembuatan komentar**

komentar adalah bagian kode yang tidak ikut dieksekusi oleh interpreter. Sehingga kehadiran dari komentar di dalam kode javascript tidak mempengaruhi proses logic dari kode yang ditulis. Komentar biasa digunakan untuk membantu mendeskripsikan proses logic dari kode yang ditulis dengan tujuan mudah untuk dipahami.

Javascript memiliki du acara untuk menulis komentar, yaitu :

* menggunakan tanda // digunakan untuk komentar yang terdiri dari satu baris.
* menggunakan tanda /\* …. \*/ digunakan untuk komentar yang terdiri dari beberapa baris.
* **variabel**

variabel adalah tempat menyimpan data, variabel memiliki nama. Sebuah nama yang mewakili sebuah nilai. Variabel bisa diisi dengan berbagai macam nilai seperti string (teks), number (angka), objek array dan sebagainya .

contoh : jumlah\_bilangan, nilai;

**cara penulisan variabel di javascript :**

* penulisan nama variabel bisa menggunakan huruf, angka, tanda underscore dan tanda dollar
* penulisan nama variabel harus diawali dengan huruf
* penulisan nama variabel javascript harus memperhatikan huruf besar kecil (case sensitive)
* nama variabel tidka boleh mengandung spasi
* penulisan variabel javascript diawali dengan sintak “**var**”.

bentuk penulisan umum variabel di javascript, yaitu :

Var title = “belajar pemrograman javascript”

* **operator**

operator digunakan untuk melakukan operasi – operasi yang dibutuhkan di dlaam kode, seperti operator matematika, pengisian nilai ke dalam variabel, membandingkan nilai dan sebagainya.

Berikut adalah daftar operator di dalam javascript yang dikelompokkan berdasarkan kategori:

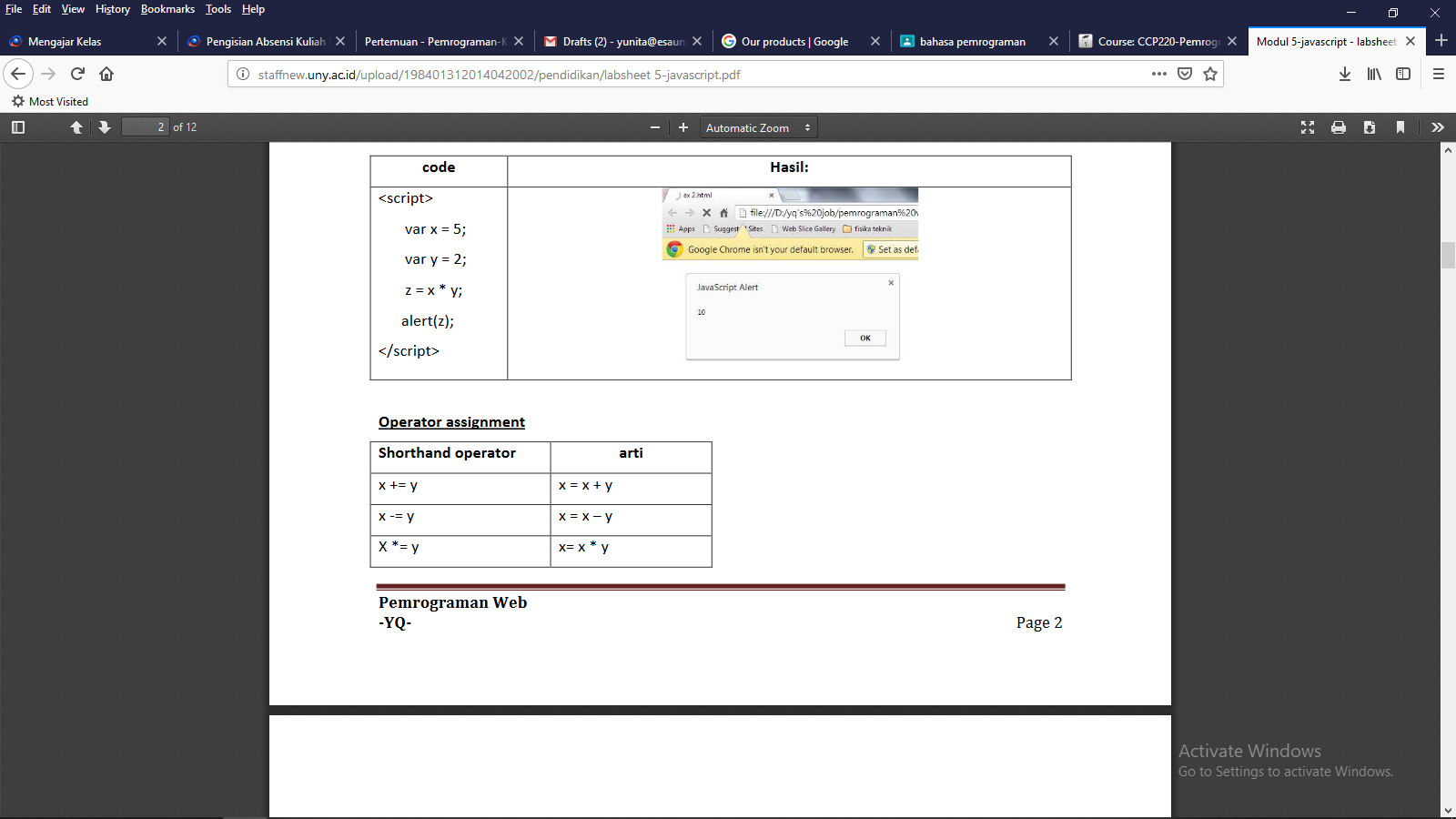
* **operator aritmatika**

berikut ini adalah daftar operator aritmatika, diantaranya :

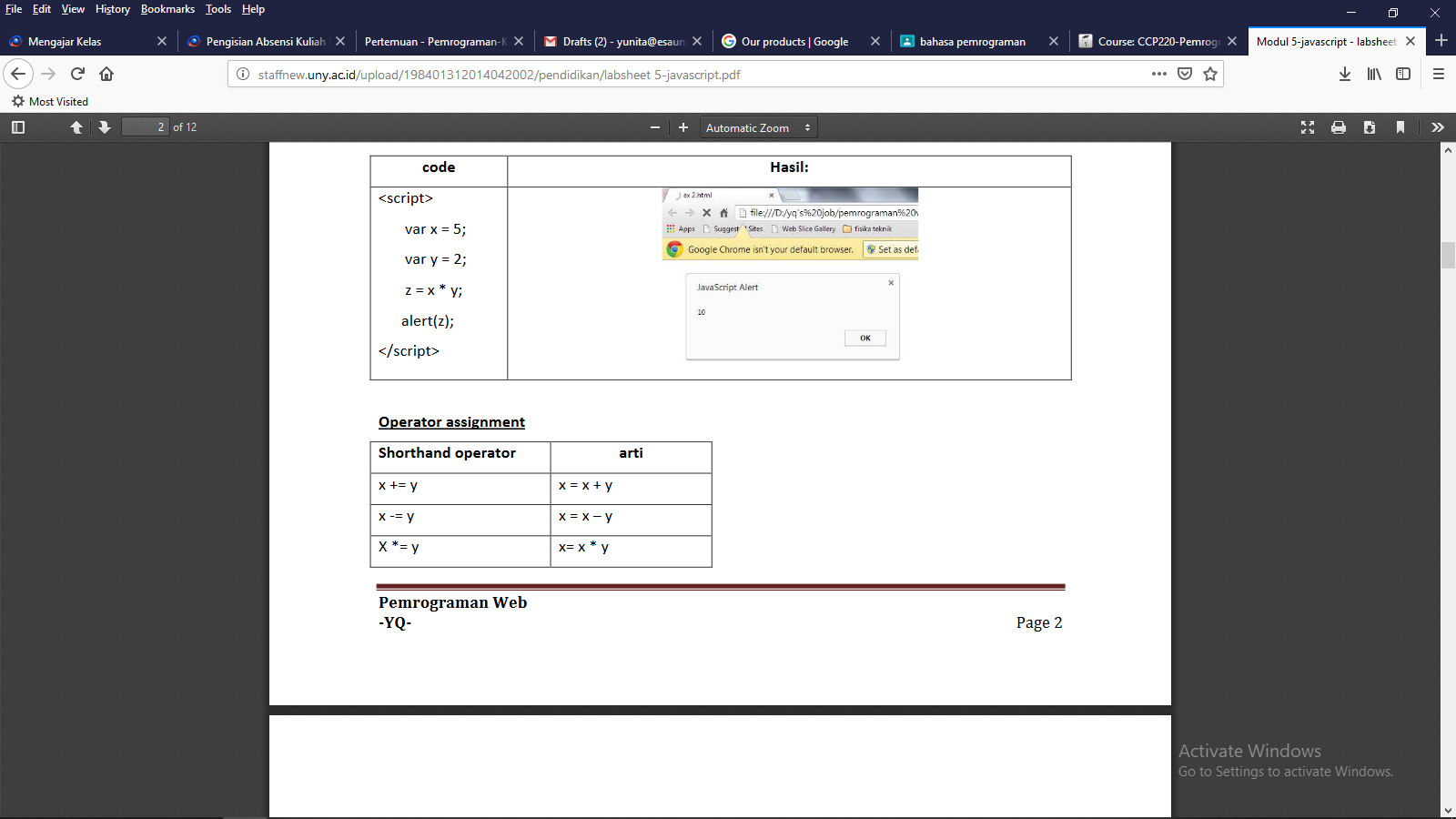
**Tabel Operator Aritmatika**

|  |  |
| --- | --- |
| **Operator** | **Keterangan** |
| + | Penjumlahan |
| - | Pengurangan |
| \* | Perkalian |
| / | Pembagian |
| % | Sisa bagi (modulus) |
| ++ | *Increment* (menambah operand dengan nilai 1) |
| -- | *Decrement* (mengurangi operand dengan nilai 1) |

Contoh penerapan operator aritmatika dalam javascript dan HTML :



Tampilan pada halaman web :



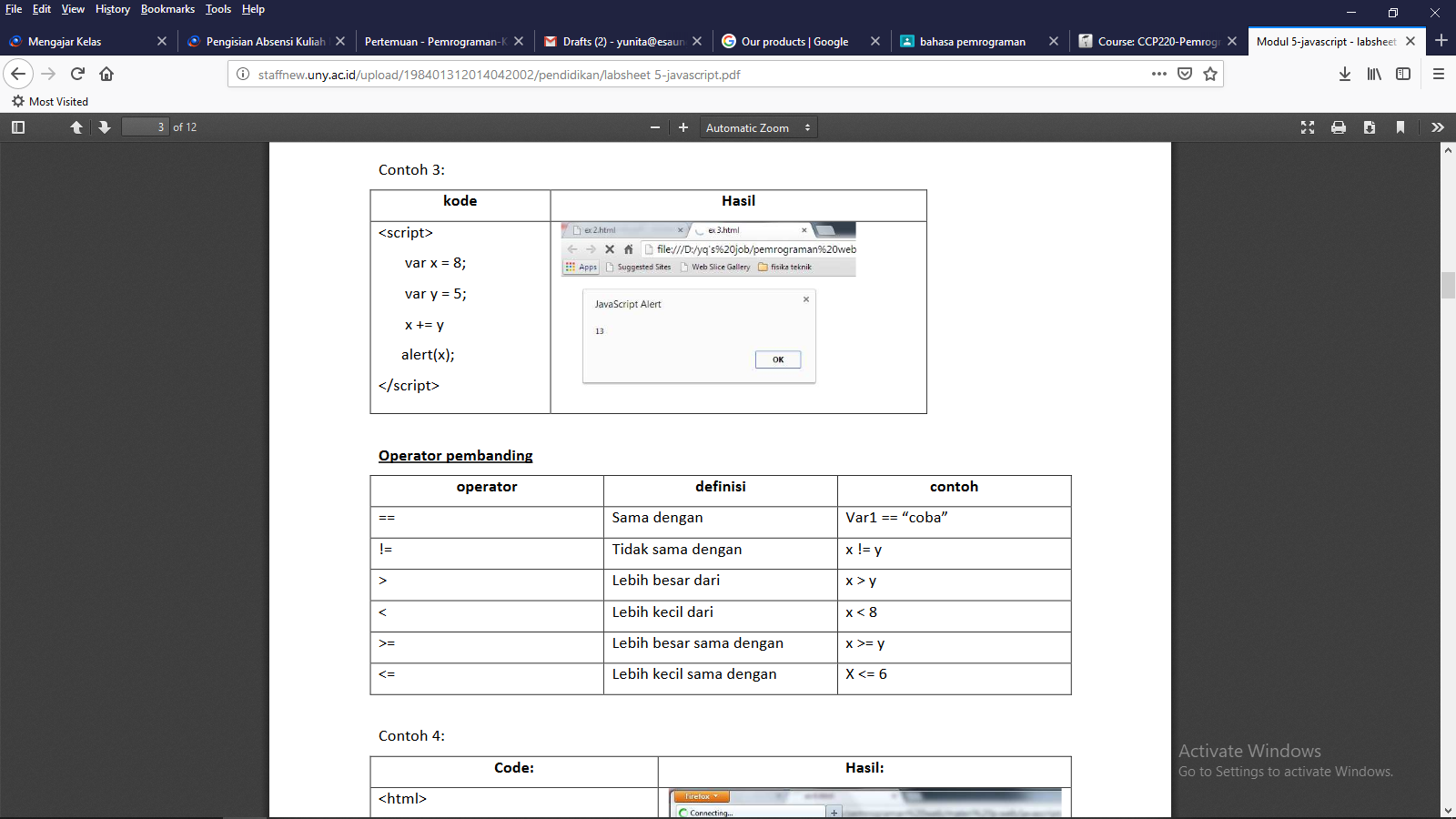
* **operator pengisian (*assignment*)**

berikut ini adalah daftar operator aritmatika, yaitu :

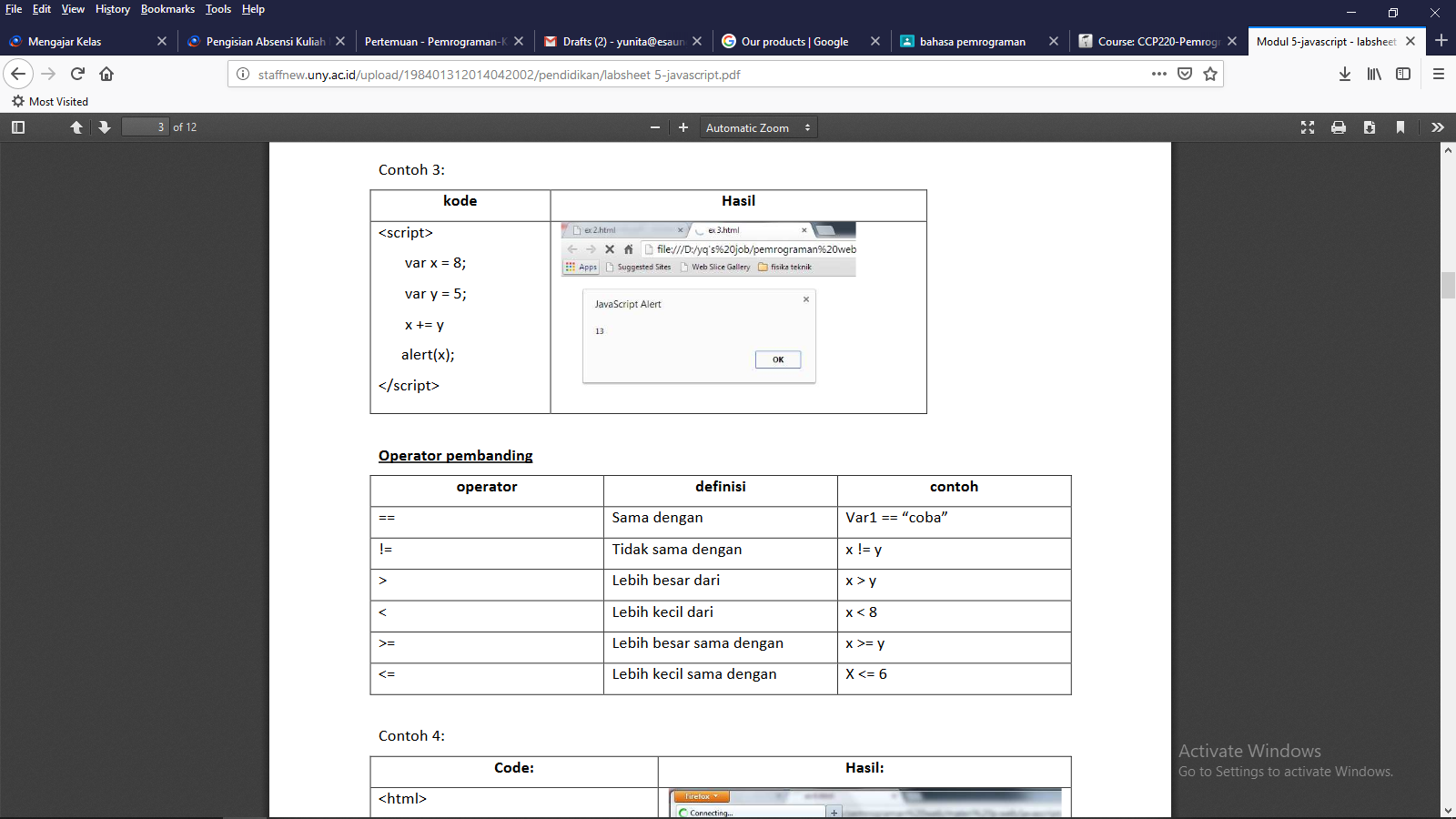
**Tabel Operator Pengisian (*assignment*)**

|  |  |
| --- | --- |
| **Operator** | **Keterangan** |
| = | Isi nilai |
| += | Tambah dan isi nilai. Contoh x += y sama dengan x = x + y |
| -= | Kurang dan isi nilai. Contoh x -= y sama dengan x = x - y |
| \*= | Kali dan isi nilai. Contoh x \*= y sama dengan x = x \* y |
| /= | Bagi dan isi nilai. Contoh x /= y sama dengan x = x / y |
| %= | *Modulus* dan isi nilai. Contoh x %= y sama dengan x = x % y |

Contoh penerapan operator pengisian (*assignment*) dalam javascript dan HTML :



Tampilan pada halaman browser :



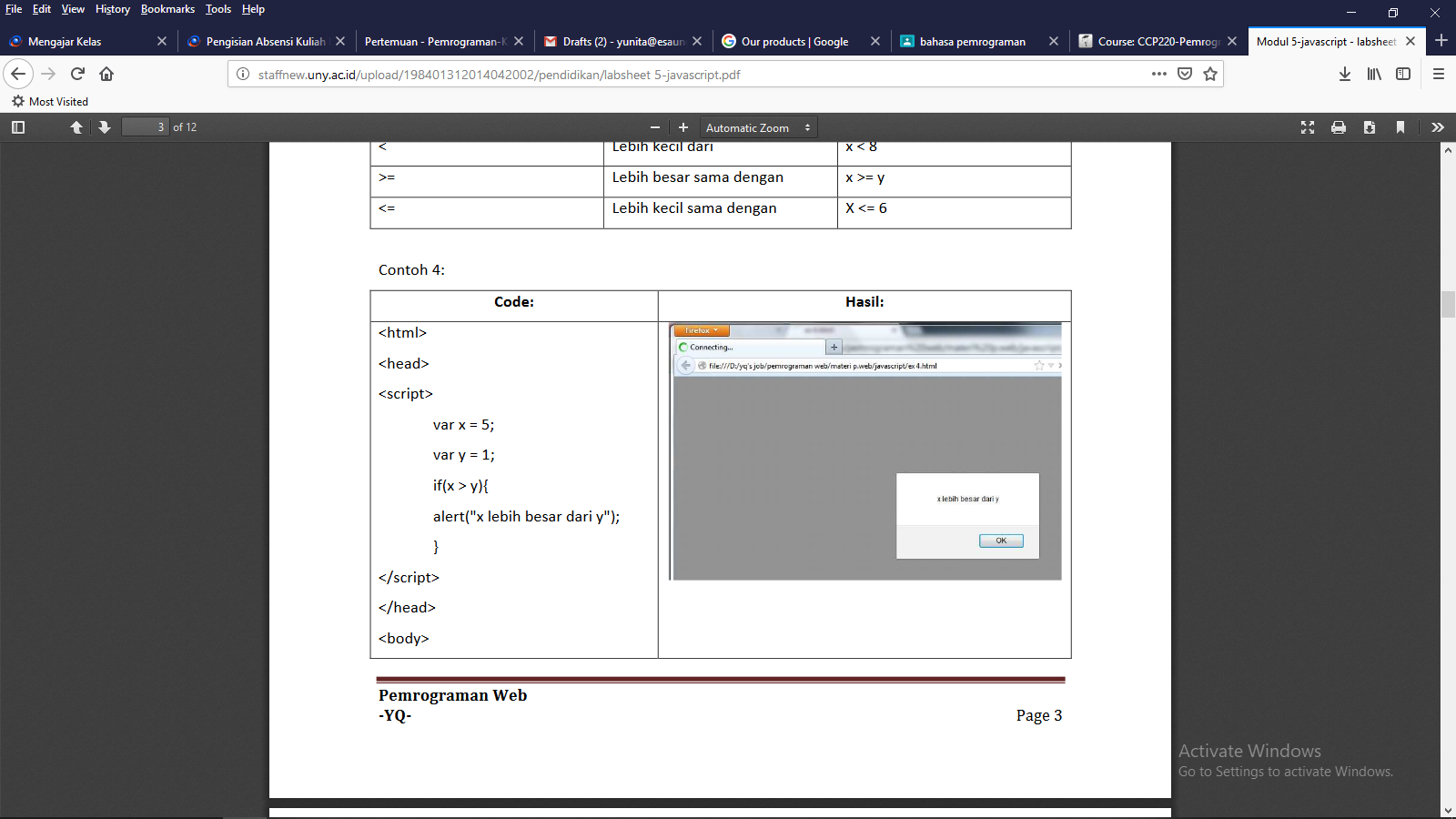
* **operator pembanding**

berikut ini adalah daftar operator pembanding:

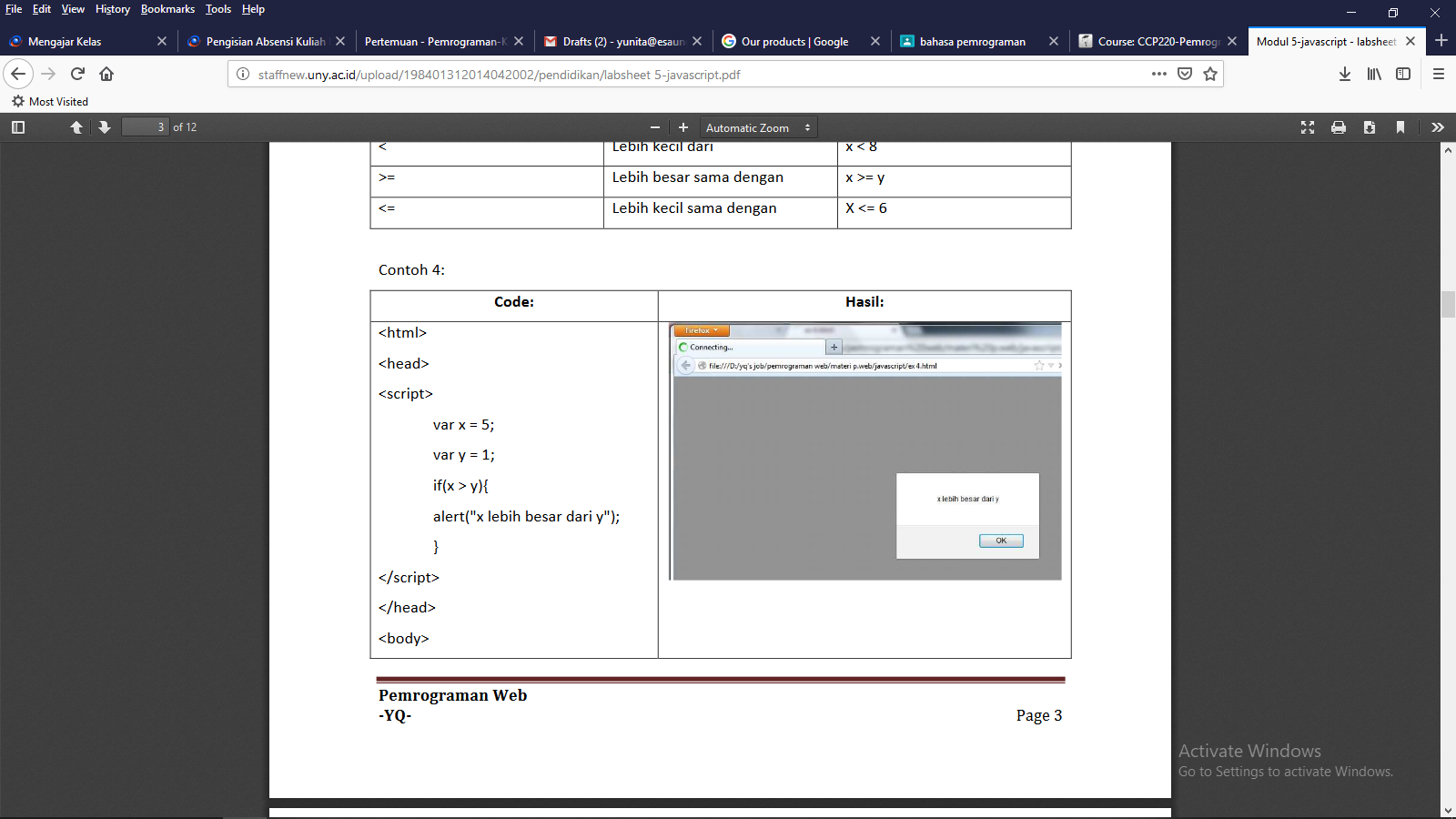
table operator pembanding

|  |  |
| --- | --- |
| **Operator** | **Keterangan** |
| == | Sama dengan |
| != | Tidak sama dengan |
| > | Lebih besar dari |
| < | Lebih kecil dari |
| >= | Lebih besar sama dengan |
| <= | Lebih kecil sama dengan |

Contoh penerapan operator pembanding dalam javascript dan HTML:



Tampilan pada halaman browser :



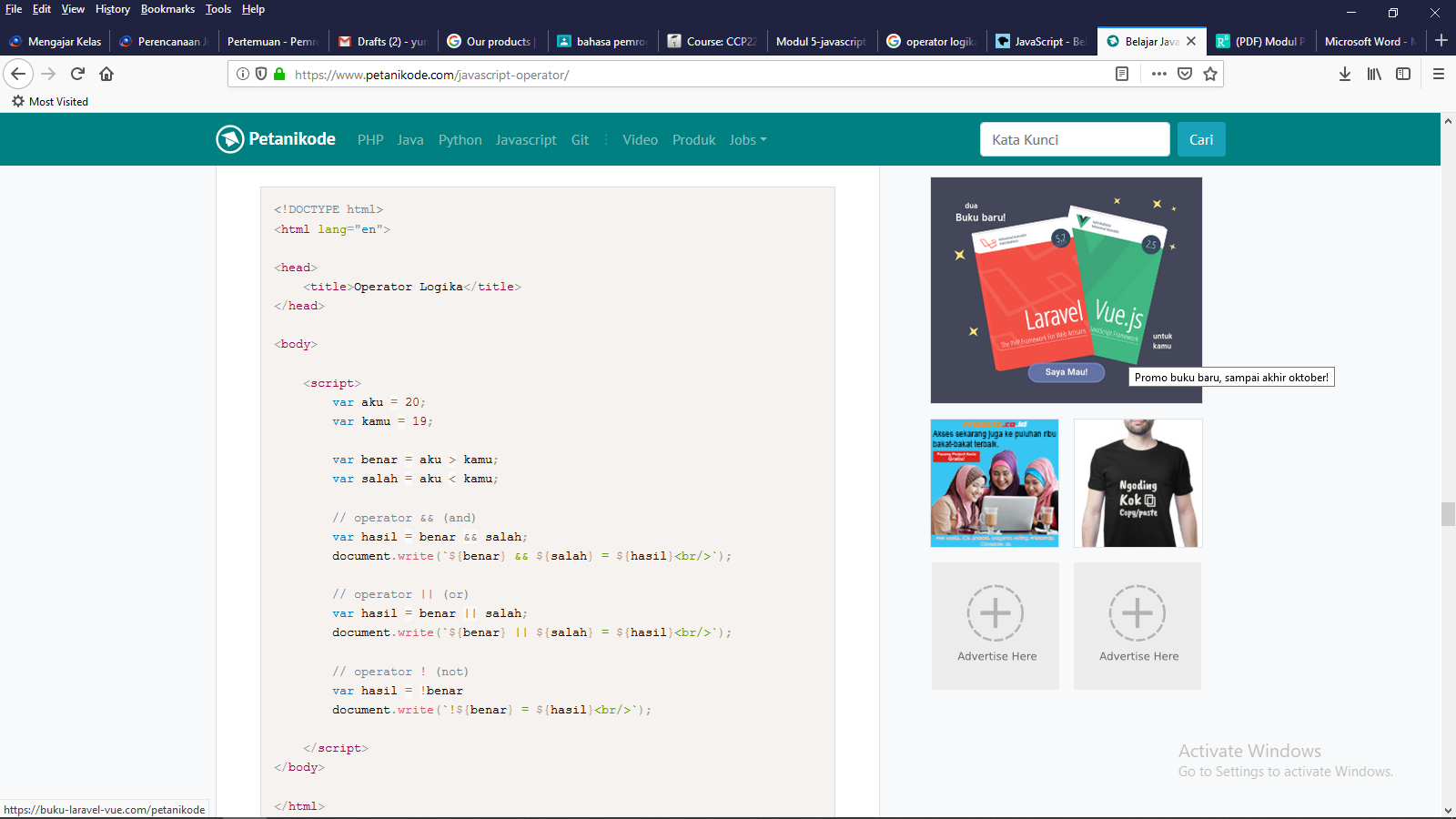
* **operator logika**

berikut ini adalah daftar operator logika, yaitu :

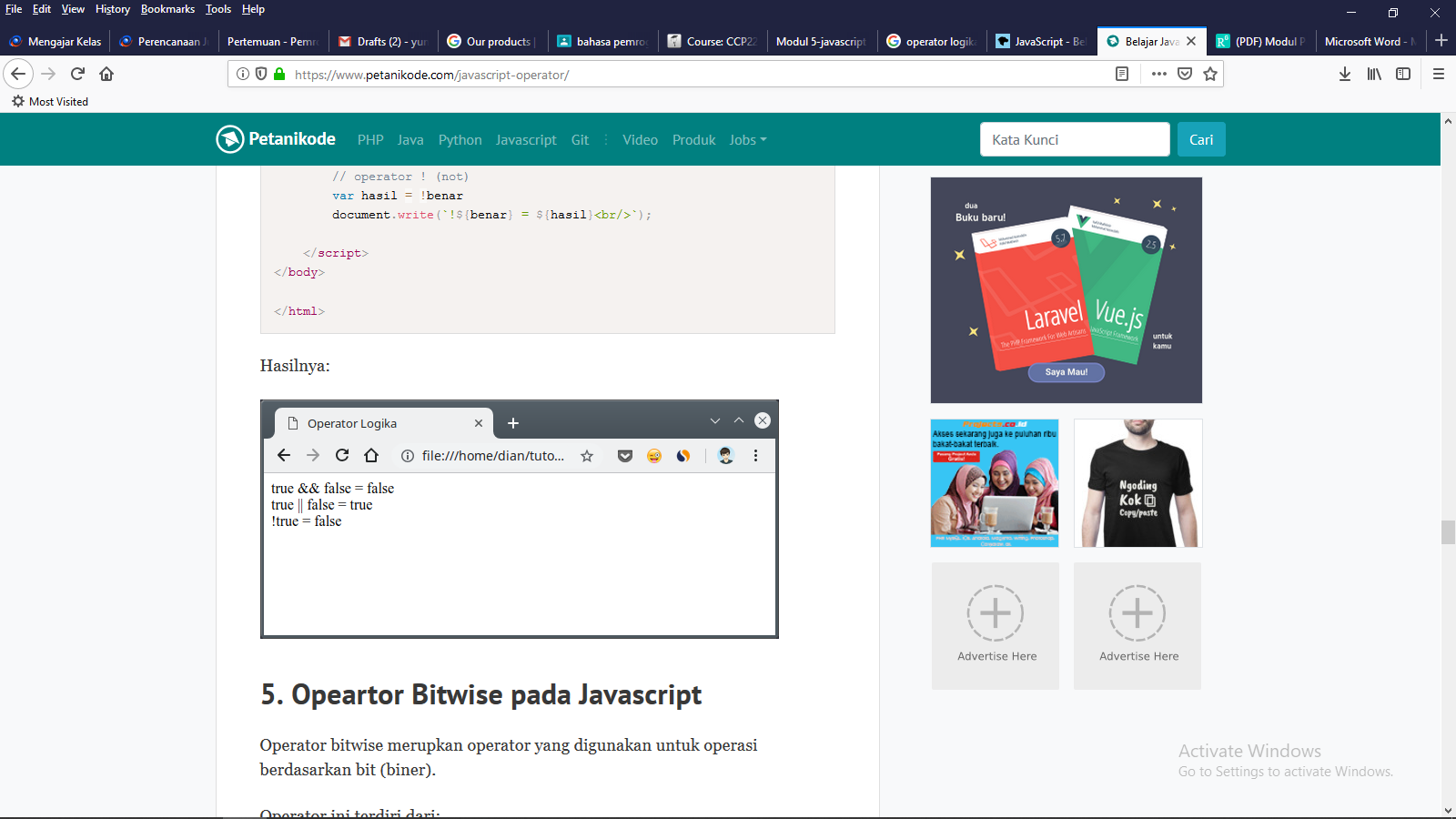
**Tabel Operator Logika**

|  |  |
| --- | --- |
| **Operator** | **Keterangan** |
| && | AND |
| || | OR |
| ! | NOT |

Contoh penerapan operator logika dalam javascript dan HTML:



Tampilan pada halaman browser :



* **Pemilihan Statemen**

Pemilihan statemen (sering juga disebut dengan percabangan atau *branching*) adalah proses eksekusi statemen yang dilakukan oleh program berdasarkan kondisi – kondisi tertentu yang didefinisikan.

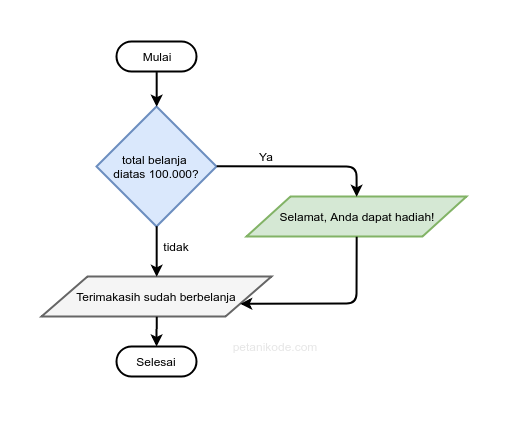
Pada javascript terdapat dua acara untuk melakukan pemilihan statemen, yaitu :

1. Menggunakan statemen if

Pada statemen if, terdapat 3 bentuk, diantaranya :

* **IF Satu kondisi**

Percabangan if merupakan percabangan yang hanya memiliki **satu blok pilihan** saat kondisi bernilai benar. Berikut agoritma dalam penggunaan statemen if



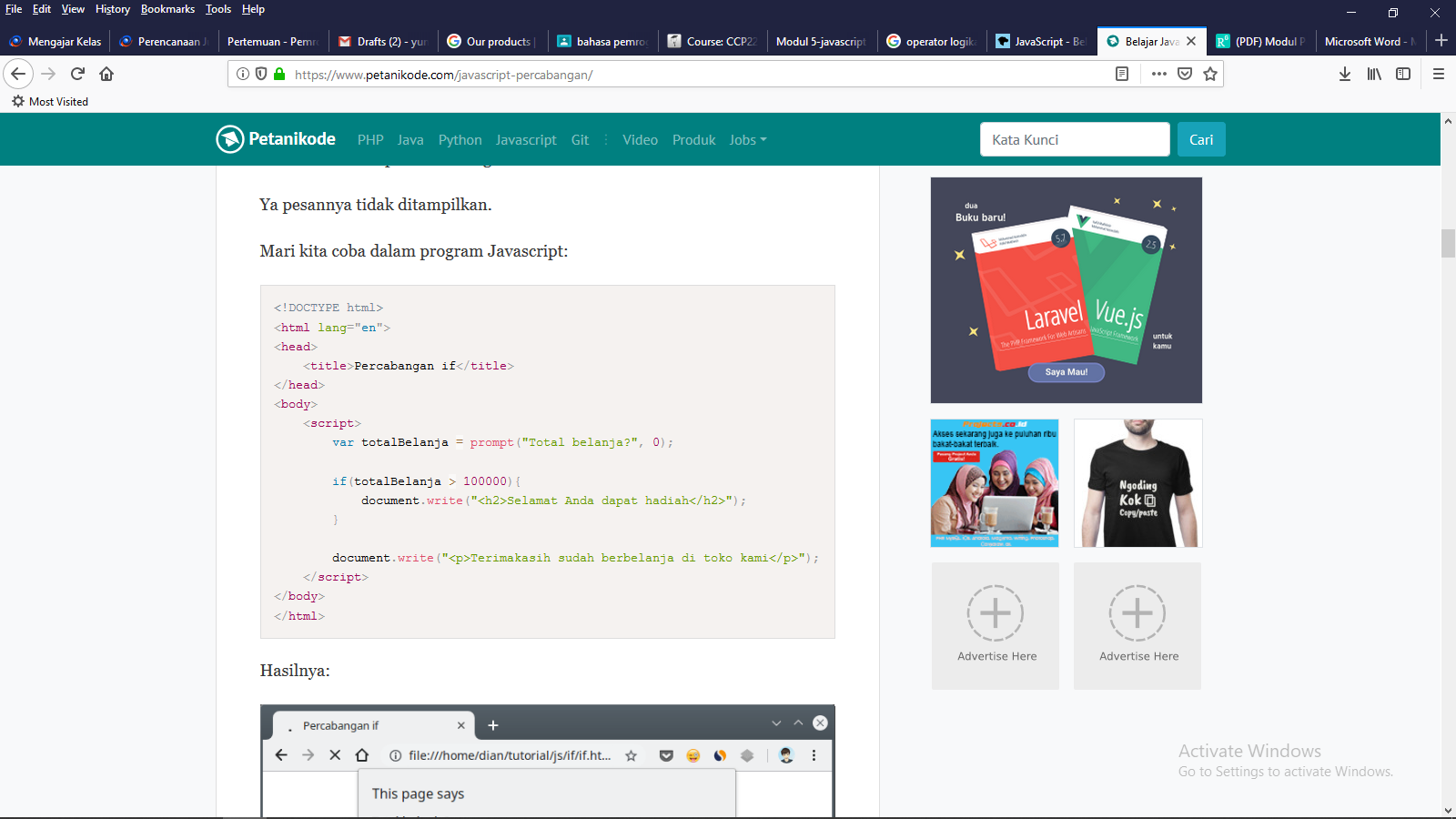
Berikut bentuk penulisan statemen if :

If (kondisi ) {

Kode yang dijalankan jika benar

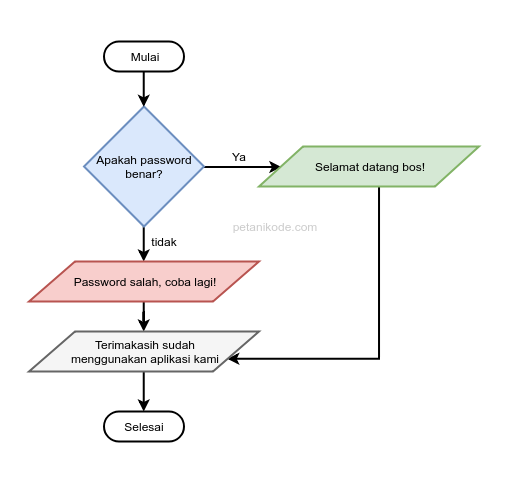
}

Contoh penerapan statemen if pada javascript :



* **IF - else**

Pemilihan if – else merupakan pemilihan yang memiliki **dua blok pilihan**. Pilihan pertama untuk kondisi **benar**, dan pilihan kedua untuk kondisi **salah** (else).



Bentuk penulisan if-else :

If (kondisi ) {

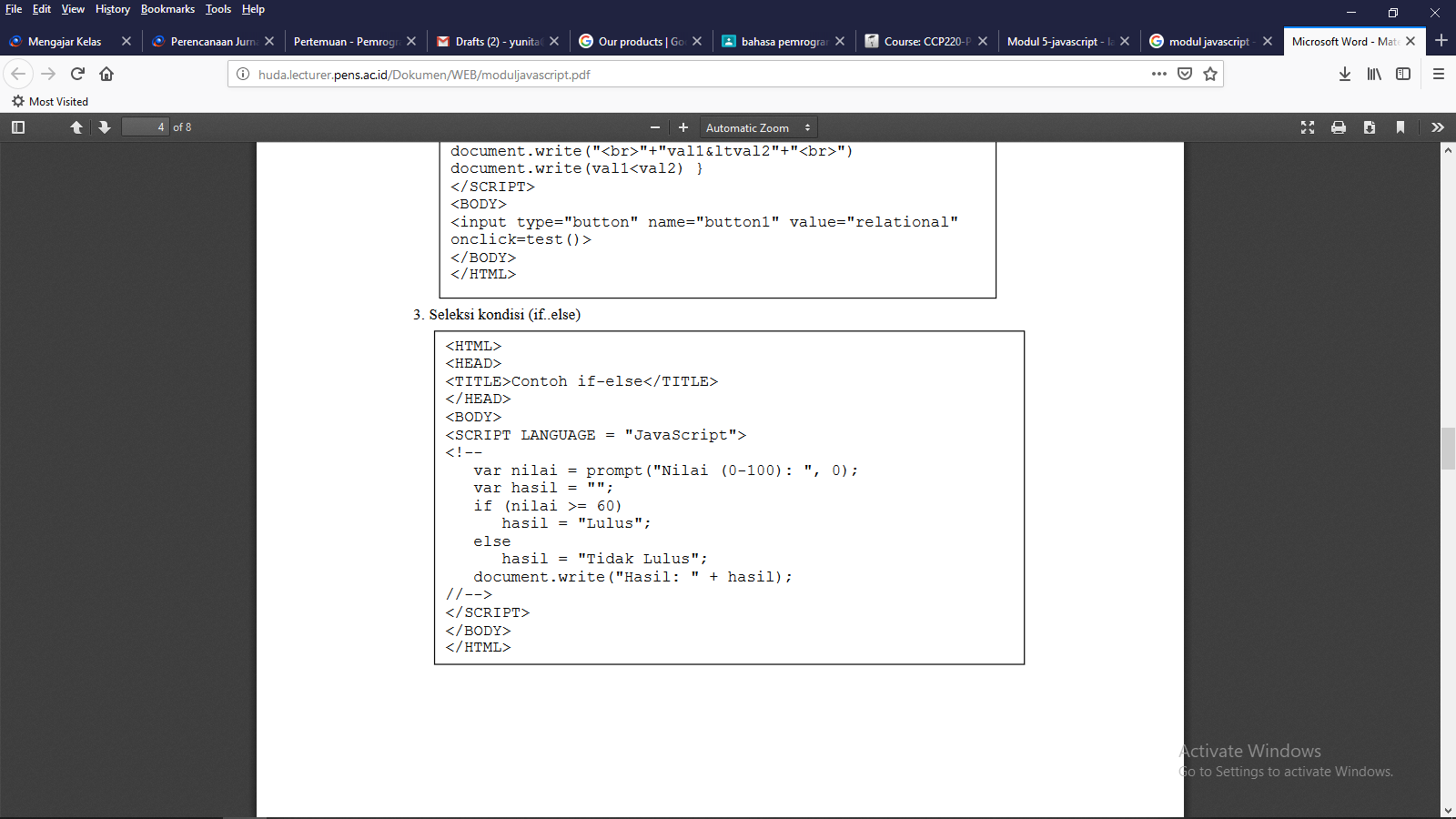
Kode yang dijalankan jika benar

} else {

Kode yang dijalankan jika salah

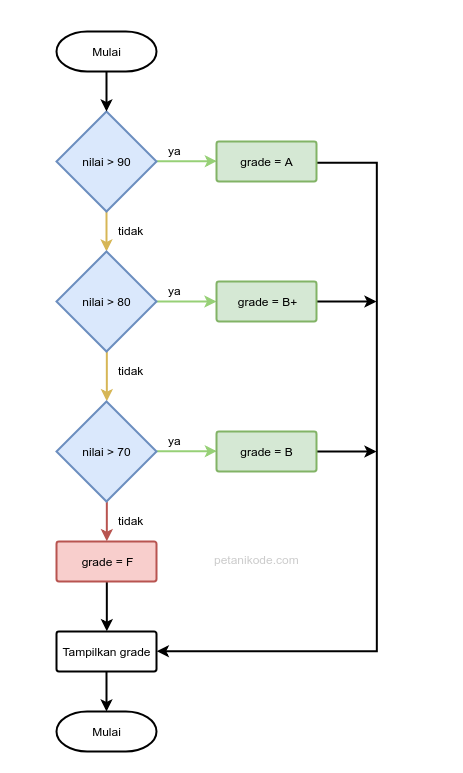
}

Contoh penerapan statemen if – else pada javascript :



* If bertingkat (banyak kondisi )

Pemilihan if – else if - else yang mempunyai kondisi lebih dari satu dengan memiliki **lebih dari dua blok pilihan**.



Berikut bentuk penulisan if – else if – else:

If (kondisi 1) {

Kode yang dijalankan jika kondisi 1 benar

} else if (kondisi 2){

Kode yang dijalankan jika kondisi 2 benar

} else if (kondisi 3){

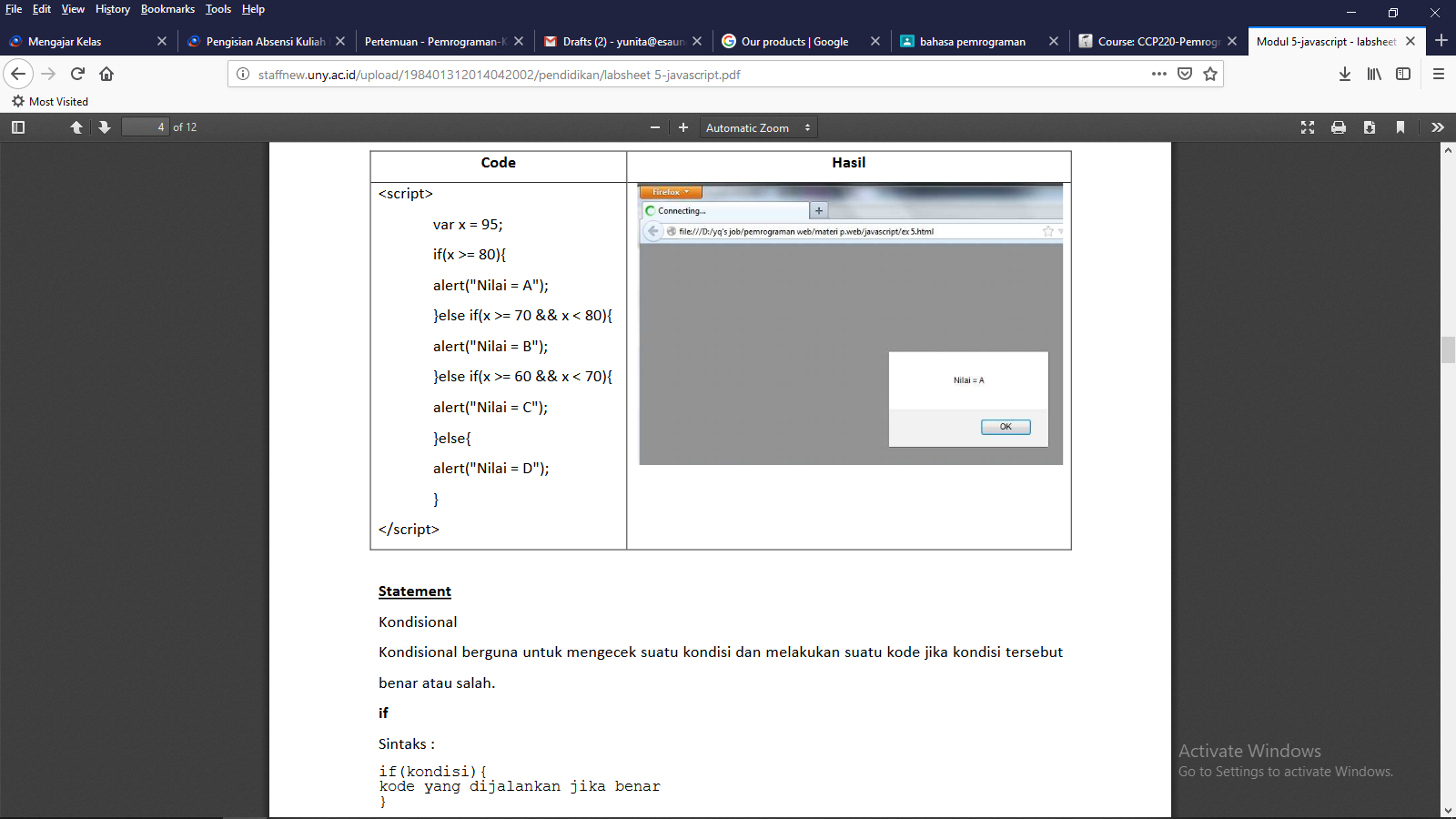
Kode yang dijalankan jika kondisi 3 benar

} else{

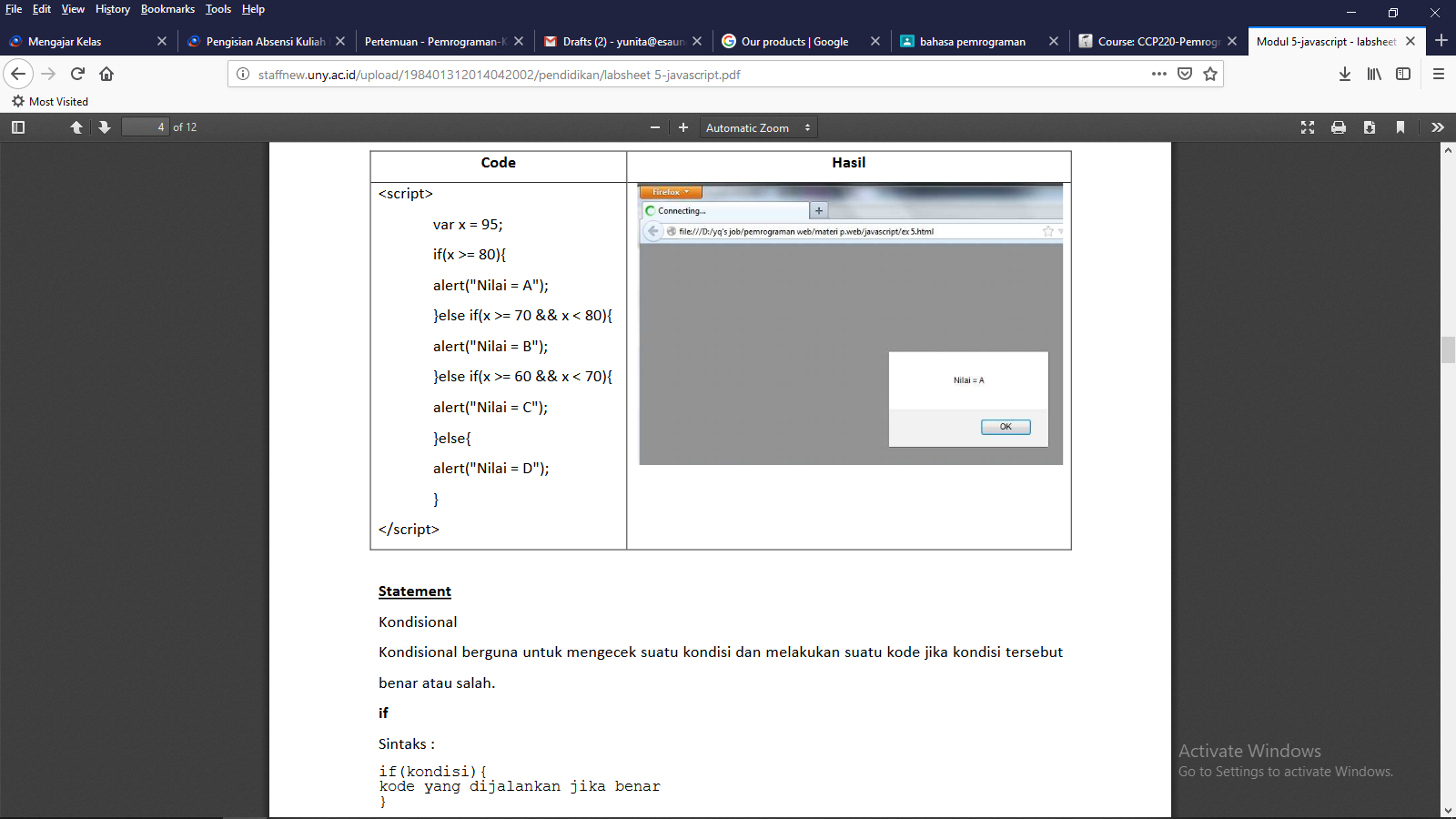
Kode jika salah satu kondisi di atas tidak ada yang benar

}

contoh penerapan if – else if – else :



Tampilan dalam halaman browser :



1. **Menggunakan statemen switch**

Statemen switch dapat digunakan untuk menggantikan struktur if – else yang banyak mengandung kondisi yang harus diperiksa. Sehingga statemen switch digunakan untuk menyederhanakan proses pemilihan yang sebenarnya dapat dilakukan menggunakan struktur if – else.

Berikut bentuk penulisan statemen switch :

Switch (ekspresi) {

Case nilai\_konstan1: {

//statemen;

Break;

}

Case nilai\_konstan2: {

//statemen;

Break;

}

………..

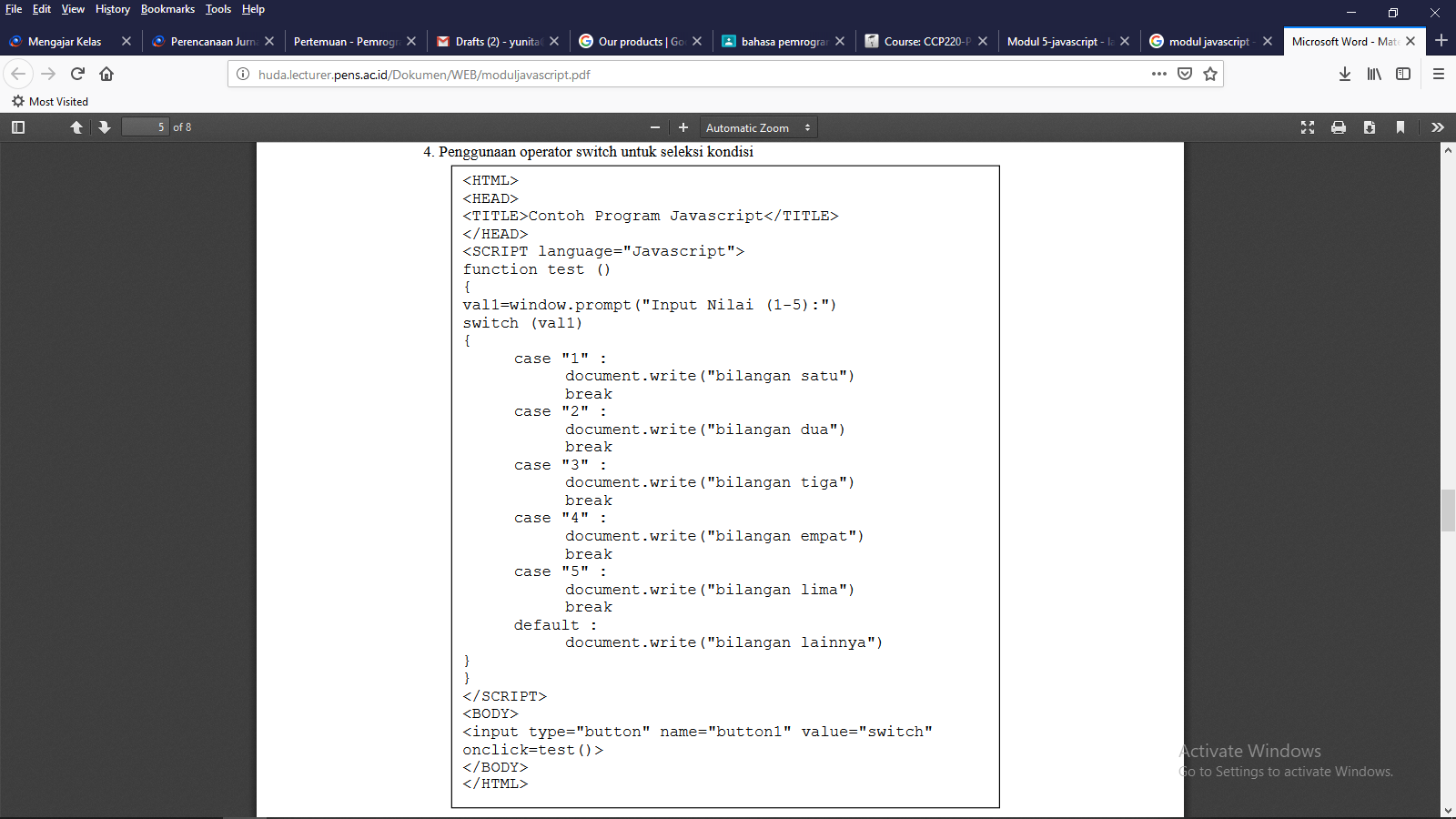
Default:

//statemen;

}

}

Contoh penerapan statemen switch pada javascript:



* **Pengulangan Statemen**

Konsep dasar dari pengulangan diantaranya :

* Berapa jumlah perulangan yang terjadi harus diketahui dengan baik
* Kekurangan atau kelebihan langkah akan mengakibatkan tidak validnya suatu perhitungan dalam sebuah pengulangan
* Dalam teknik pengulangan, yang harus ditentukan pertamakali adalah kapan sebuah pengulangan dimulai dan kapan suatu pengulangan harus berakhir / berhenti berproses
* Untuk pergerakan dari titik awal sampai pada titik akhir, diperlukan suatu langkah (*step)* untuk mengontrol nilai pergerakan perulangan.

Dalam Bahasa pemrograman javascript terdapat tiga jenis pengulangan, yaitu :

1. **Statemen for**

Statemen for digunakan untuk melakukan pengulangan statemen yang jumlah sudah pasti. Misalkan terdapat pengulangan 5 kali, 10 kali dan seterusnya.

Bentuk penulisan statemen for :

For (inisialisasi; kondisi; iterasi) {

// statemen yang akan dilakukan perulangan

}

Contoh penerapan statemen for pada javascript :



1. **Statemen while**

pengulangan yang dibuat menggunakan statemen while menempatkan kondisi pada bagian awal perulangan. Jika kondisi bernilai **true** maka statemen yang ada di dalam perulangan akan dieksekusi. Tetapi jika sebaliknya, program akan langsung keluar dari perulangan.

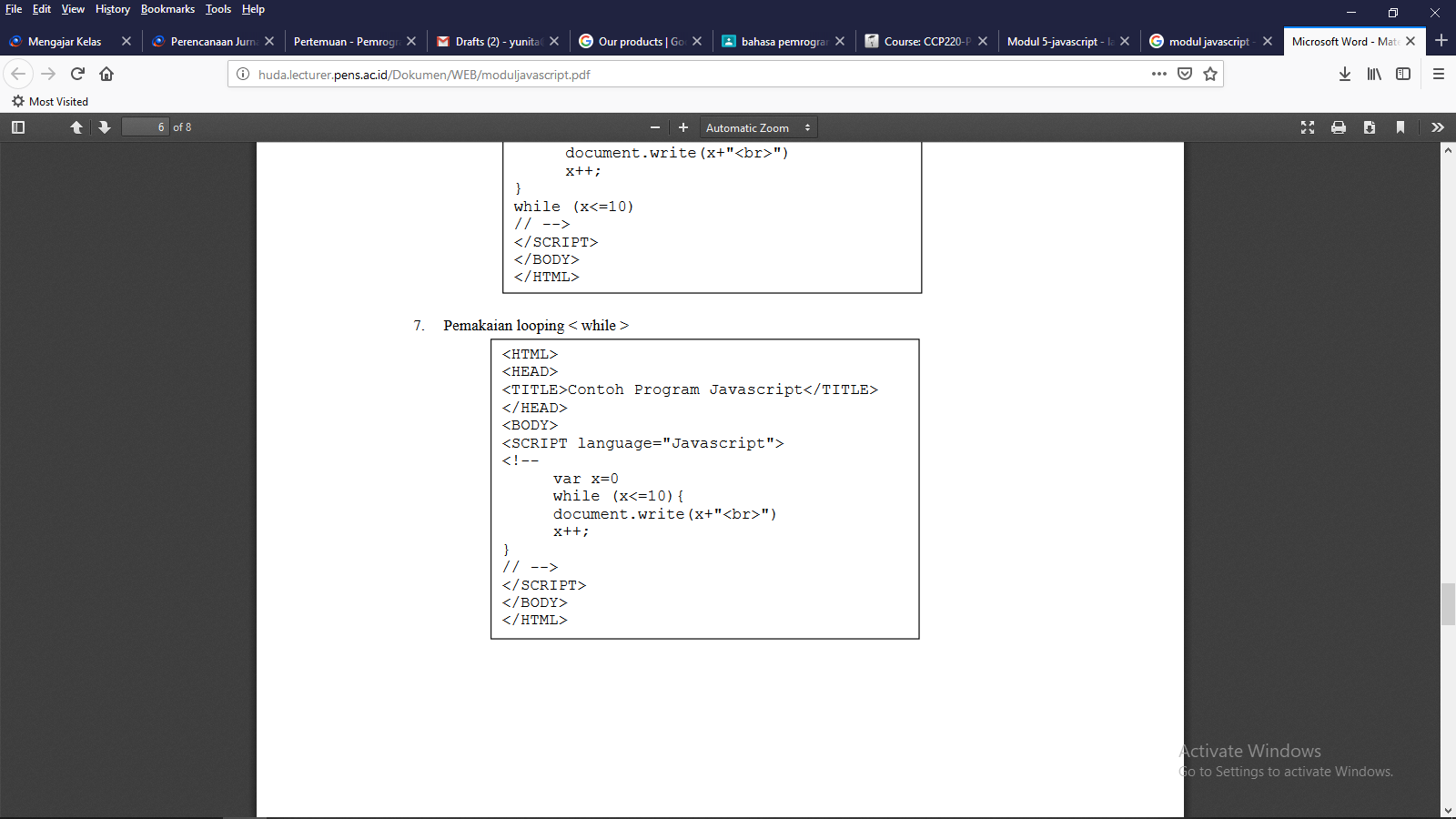
Berikut bentuk penulisan statemen while pada javascript :

While ( kondisi ) {

// statemen yang akan dilakukan perulangan

}

Contoh penerapan statemen while pada javascript :



1. **Statemen do-while**

Bentuk penulisan statemen do – while pada javascript :

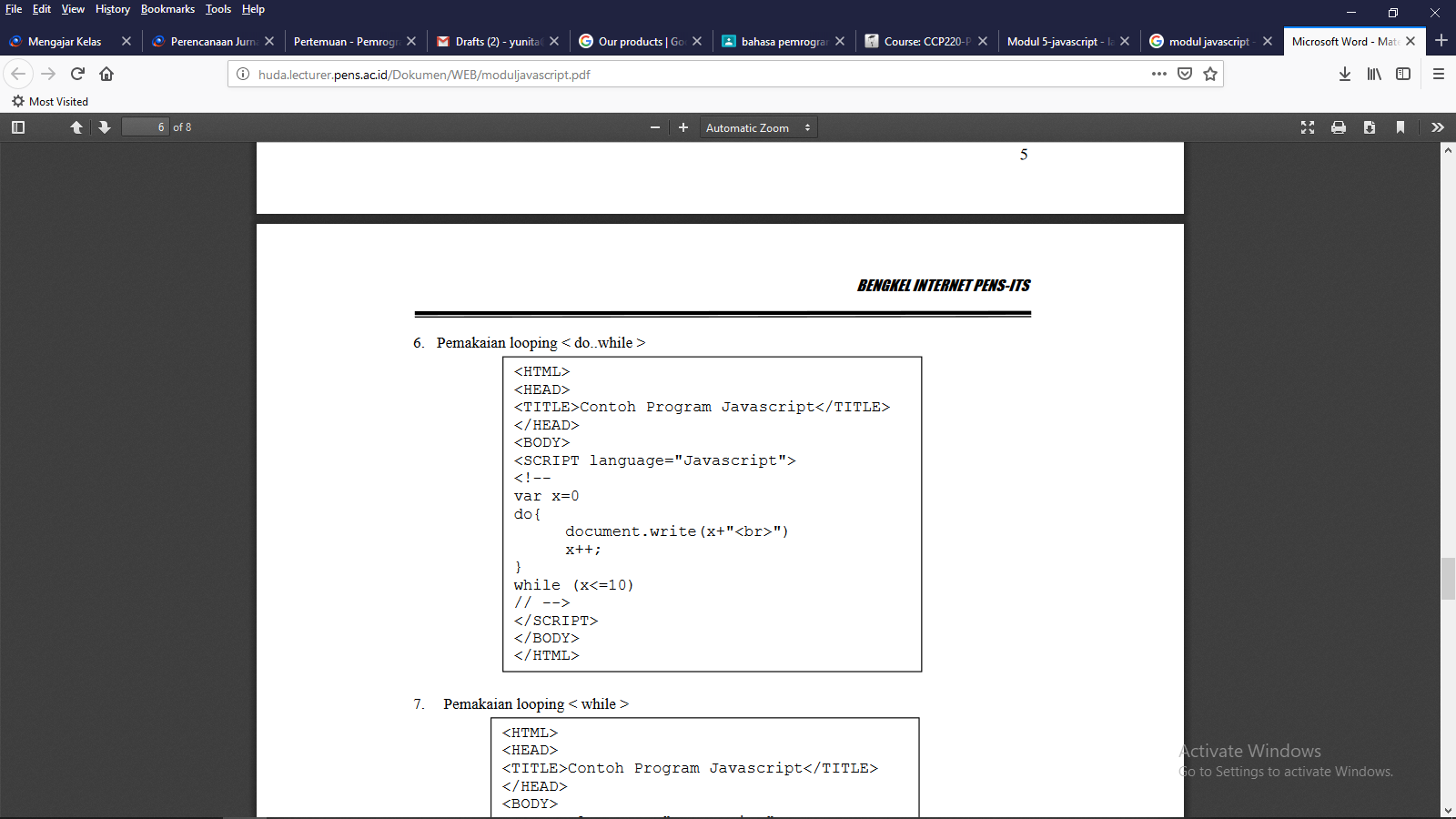
Do {

// statemen yang akan dilakukan perulangan;

}

While ( kondisi ) ;

Contoh penerapan statemen do - while pada javascript :



**Membuat fungsi dalam javascript**

Fungsi adalah sub-program yang bisa digunakan untuk kembali baik di dalam program itu sendiri, atau di program yang lain. Fungsi di dalam javascript adalah sebuah objek. Karena memiliki atribut dan juga *method*.

**Cara membuat fungsi di javascript**

Pada javascript terdapat 4 cara yang bisa dilakukan dalam membuat fungsi di javascript, yaitu:

1. Menggunakan cara biasa

Berikut cara yang sering digunakan untuk membuat fungsi di javascript :

Function namafungsi ( ) {

console.log ( “Hello world” ) ;

}

1. Menggunakan ekspresi

Berikut cara yang digunakan untuk membuat fungsi dengan ekspresi :

Var namafungsi = function ( ) {

Console.log ( “hello world” );

}

1. Menggunakan tanda panah (=>)

Berikut cara yang digunakan untuk membuat fungsi dengan tanda panah (=>) :

Var namaFungsi = ( ) = > {

Console.log( “hello world” );

}

//atau dapat dibuat seperti ini (jika ingin mengisi fungsi hanya satu baris);

Var namaFungsi = ( ) = > Console.log( “hello world” );

Penggunaan tanda panah (=>) digunakan sebagai pengganti function dan pembuatan fungsi dengan cara seperti ini dapat disebut dengan ***arrow function*.**

1. Menggunakan *constructor*

Berikut cara yang digunakan untuk membuat fungsi dengan menggunakan constructor:

Var namaFungsi = new Function ( ‘console.log (“hello world” );’);

**Pemanggilan / eksekusi fungsi**

Berikut contoh penulisan dalam memanggil fungsi pada javascript

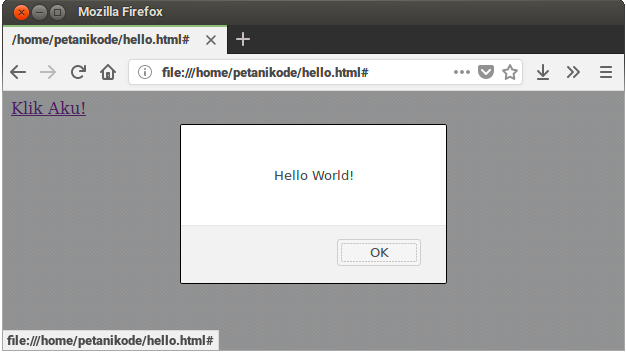
**File .js**



**File .html**



Tampilan pada halaman browser:



**Fungsi dengan parameter**

Parameter adalah variabel yang menyimpan nilai untuk diproses di dalam fungsi

Berikut Penulisan membuat fungsi dengan menggunakan parameter :

Function kali (a, b) {

Hasilkali = a \* b ;

Console.log ( “hasil kali a \* b = “ + Hasilkali);

}

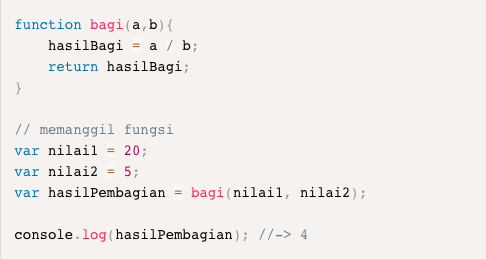
Variabel a dan b adalah sebuah parameter dan untuk pemanggilan fungsinya adalah dengan cara seperti dibawah ini :

kali (5, 6); // hasil perkalian dari a \* b = 30

Isi dari parameter a adalah 5 dan isi / nilai dari parameter b adalah 6.

**Fungsi dengan mengembalikan nilai**

Pengembalian nilai pada fungsi menggunakan kata kunci **return**, lalu diikuti dengan nilai atau variabel yang akan dikembalikan. Berikut contoh penulisan dalam membuat fungsi dengn mengembalikan nilai :



**Tipe dialog dalam javascript**

Javascript memiliki tiga tipe dialog, yaitu :

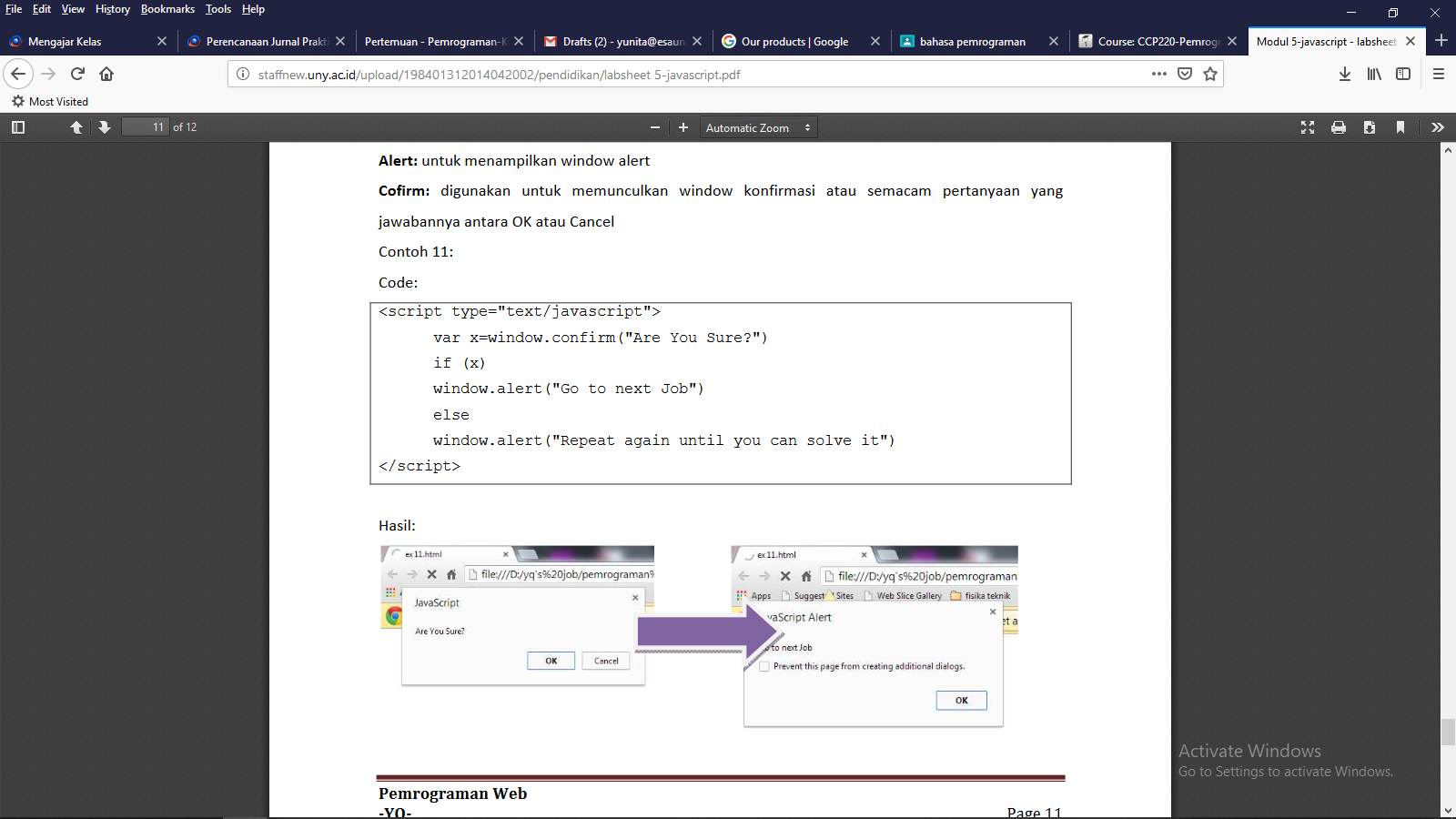
1. **Tipe Alert**

Tipe ini digunakan untuk menampilkan pesan informasi dan peringatan.

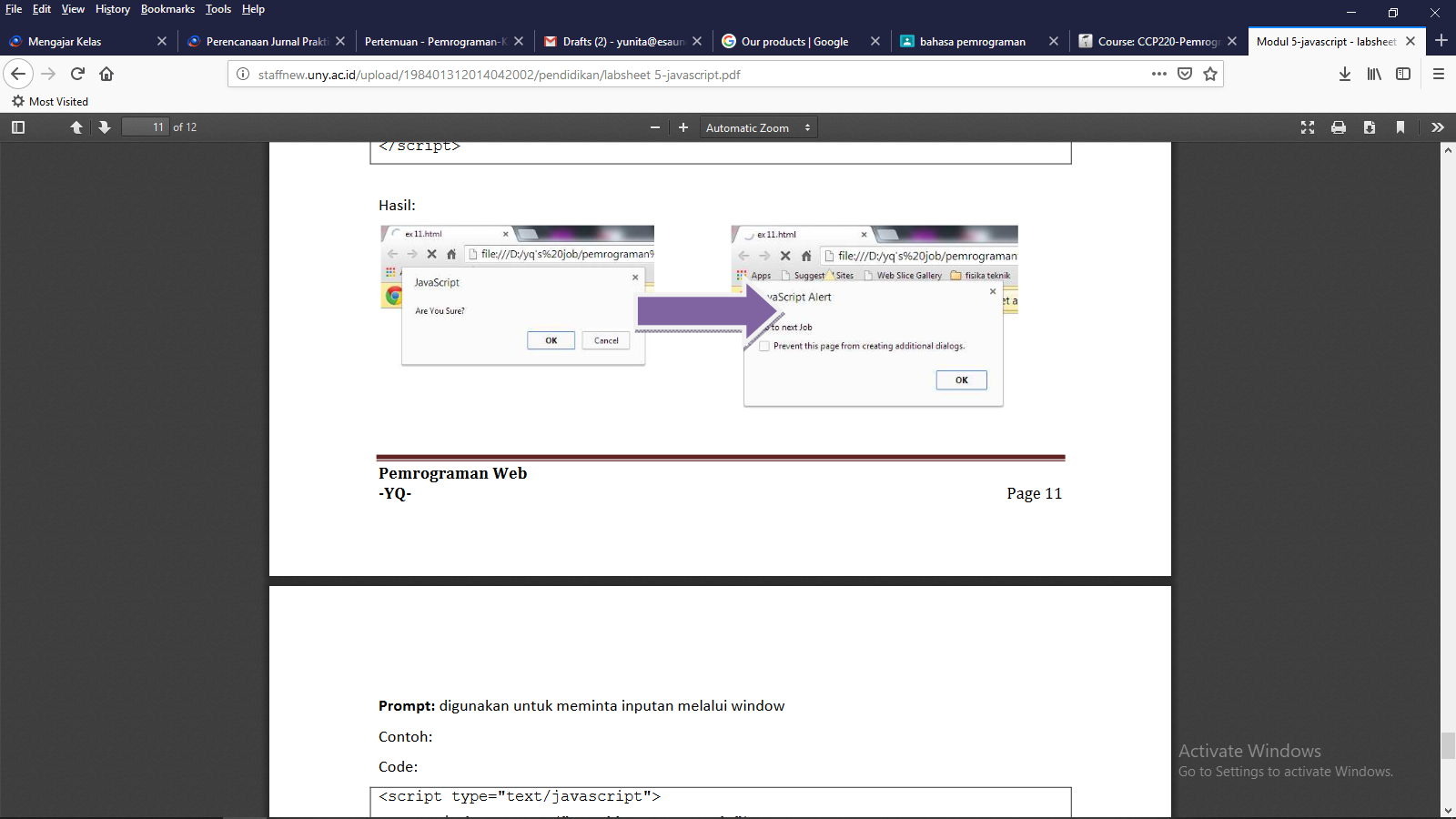
1. **Tipe Confirm**

Tipe ini digunakan utnuk menampilkan pesan konfirmasi.

Berikut contoh penulisan tipe confirm dan alert dalam javascript :

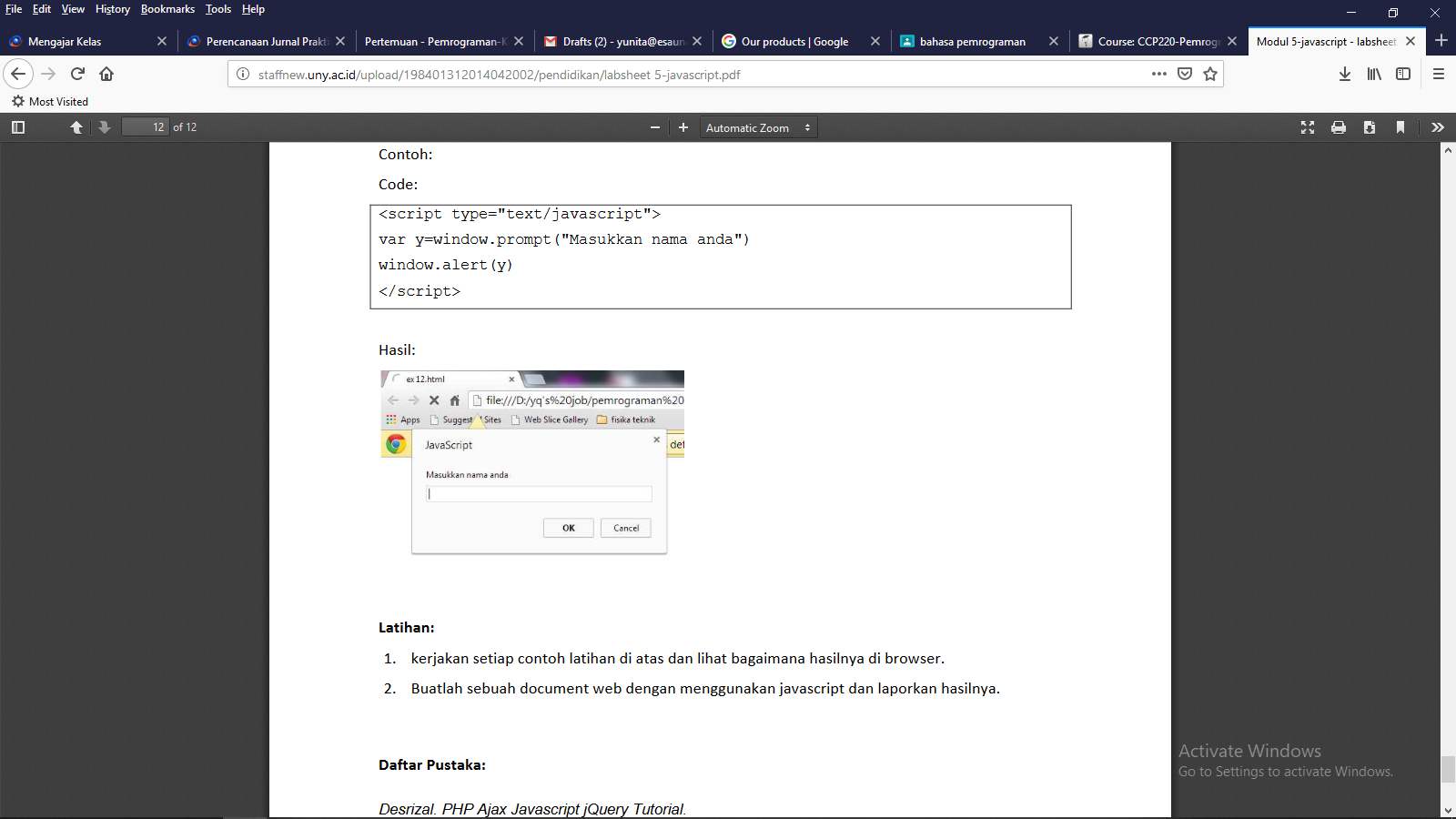


Tampilan dalam halam browser:

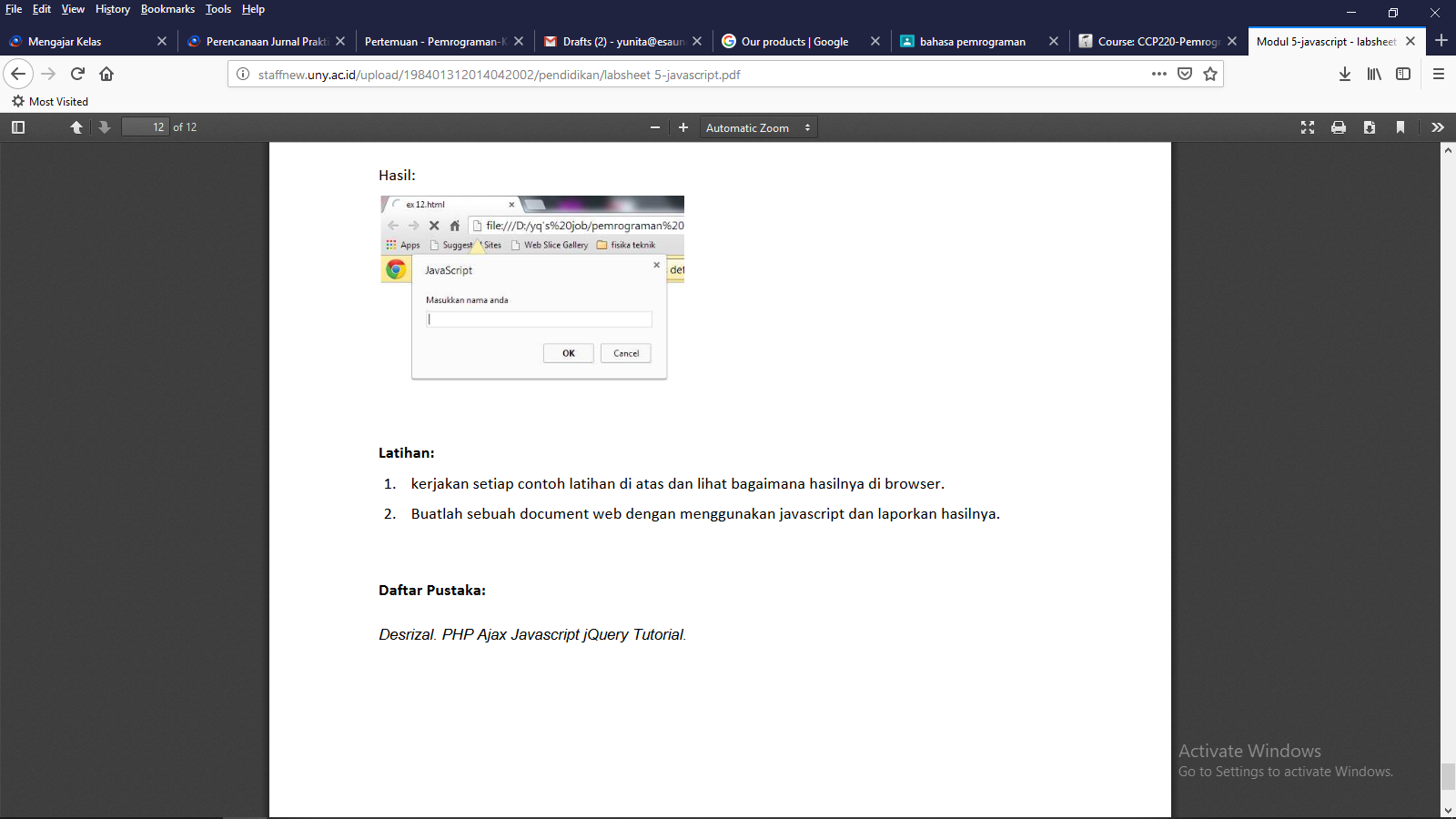


1. **Tipe Prompt**

Tipe ini digunakan untuk meminta user memasukkan suatu nilai tertentu. Berikut contoh penulisan tipe prompt dalam javascript :



Tampilan dalam halaman browser :



**Event handler**

Even handler adalah kemampuan javascript untuk mendeteksi event atau kejadian – kejadian yang terjadi di halaman web, kemudian menangani atau melakukan suatu proses jika terdeteksi suatu event. Terdapat beberapa macam event diantaranya klik, double klik, menggerakan mouse, bila pointer mouse berada di atas suatu objek HTML, dan sebagainya.

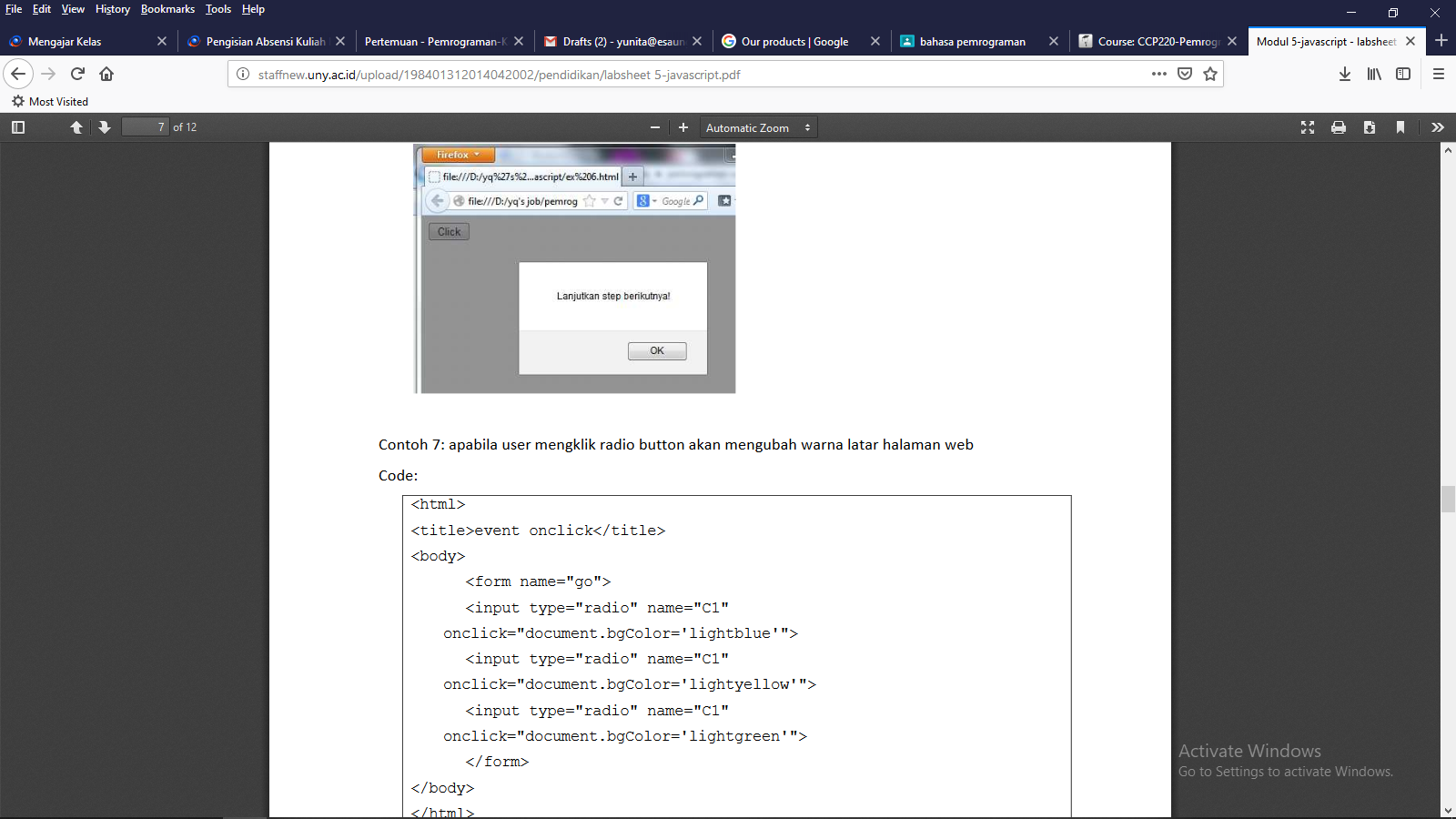
Bentuk penulisan even handler :

Nama\_event = “kode javascript”

Penerapan even handler : onclick (diklik)



Tampilan dalam halaman browser:



**Memformat teks dengan javascript**

Untuk membantu melakukan formating terhadap teks secara programming, javascript menyediakan beberapa metode formating menggunakan javascript. Berikut daftar property dan metode memformat teks dengan javascript :

**Tabel Daftar Properti rmating menggunakan Javascript**

|  |  |
| --- | --- |
| **Properti** | **Deskripsi** |
| length | Menghasilkan jumlah karakter dari suatu string atau teks |

**Tabel Daftar Metode Formating menggunakan Javascript**

|  |  |
| --- | --- |
| **metode** | **deskripsi** |
| Anchor(nama) | Menghasilkan string dengan diapit tag <A name = “nama”> |
| Big() | Menghasilkan string dengan diapit tag <BIG> |
| Blink() | Menghasilkan string dengan diapit tag <BLINK> |
| Bold() | Menghasilkan string dengan diapit tag <B> |
| Fixed() | Menghasilkan string dengan diapit tag <TT> |
| Fontcolor(warna) | Menghasilkan string dengan diapit tag <FONT color = “warna”> |
| Fontsize(size) | Menghasilkan string dengan diapit tag <FONT size = “size”> |
| Italics() | Menghasilkan string dengan diapit tag <I> |
| Link(url) | Menghasilkan string dengan diapit tag <A href=”url”> |
| Small() | Menghasilkan string dengan diapit tag <SMALL> |
| Strike() | Menghasilkan string dengan diapit tag <STRIKE> |
| Sub() | Menghasilkan string dengan diapit tag <SUB> |
| Sup() | Menghasilkan string dengan diapit tag <SUP> |
| toLowerCase() | Mengubah string menjadi huruf kecil semua |
| toUpperCase() | Mengubah string menjadi huruf besar semua |

Penerapan dari salah satu metode formatting teks :

<script>

Var message = “welcome to my site”

Document.write(message.toUpperCase().bold())

</script>

**Daftar pustaka**

Desrizal. PHP Ajax Javascript jQuery Tutorial.

Raharjo, Budi. 2013. Belajar pemrograman web. Modula.

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

<http://huda.lecturer.pens.ac.id/Dokumen/WEB/moduljavascript.pdf>

<http://staffnew.uny.ac.id/upload/198401312014042002/pendidikan/labsheet%205-javascript.pdf>