

|  |
| --- |
| **Web Programming II** |
|  |
| **Modul 10** |
| **Dasar jQuery** |

**Disusun oleh:**

**Dwi Intan Af’idah, S.T., M.Kom**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**POLITEKNIK HARAPAN BERSAMA**

**TAHUN AJARAN 2020/2021**

Daftar Isi

[Daftar Isi ii](#_Toc57798204)

[1 Pengertian jQuery 1](#_Toc57798205)

[2 Cara Menginput File jQuery ke HTML 4](#_Toc57798206)

[2.1 Cara Mendownload dan Mengakses jQuery di Komputer Lokal 4](#_Toc57798207)

[2.2 Menghubungkan File jQuery dengan HTML 6](#_Toc57798208)

[2.3 Cara Menginput File jQuery menggunakan CDN 7](#_Toc57798209)

[3 Cara Menjalankan Kode jQuery (event ready) 8](#_Toc57798210)

[3.1 Event ready() atau The Document Ready Event 9](#_Toc57798211)

[4 Aturan Dasar Penulisan Kode jQuery 12](#_Toc57798212)

[4.1 Pengertian jQuery Object ($) 12](#_Toc57798213)

[4.2 jQuery Action / jQuery Event 12](#_Toc57798214)

[5 Cara Mencari Elemen HTML dengan jQuery Selector 15](#_Toc57798215)

[5.1 Pengertin Method getElementById() JavaScript 15](#_Toc57798216)

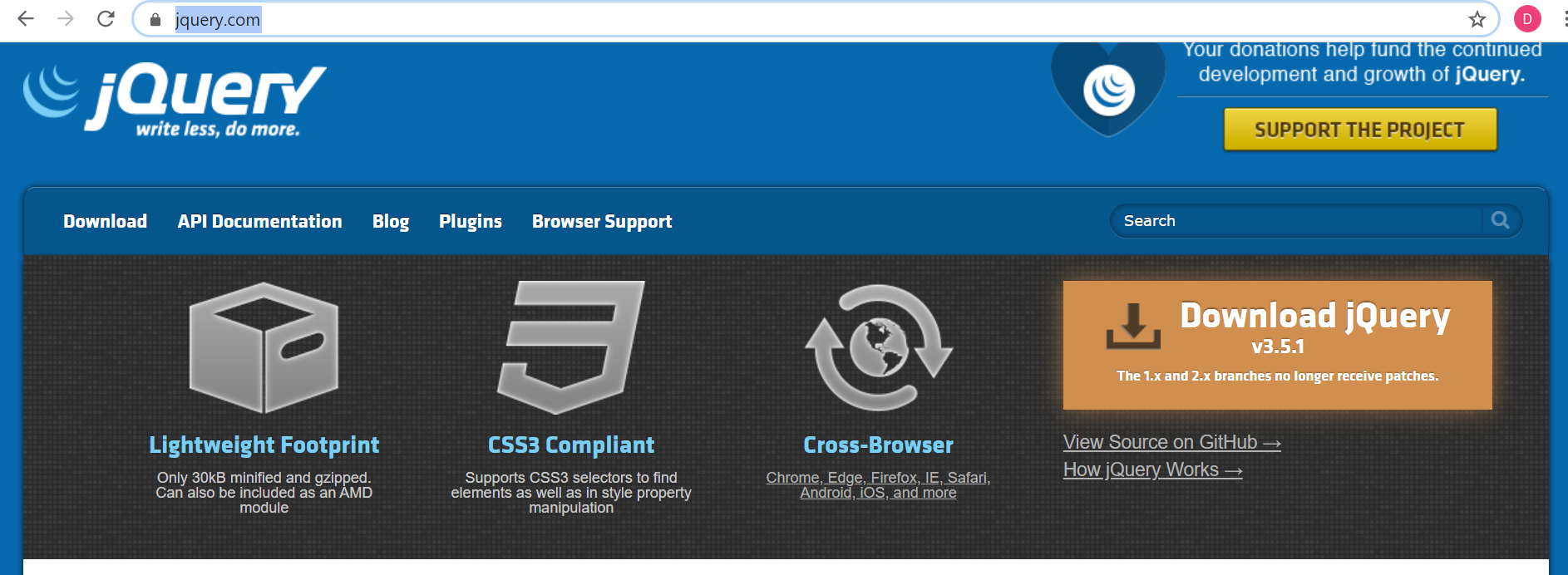
[5.2 Pengertian jQuery Selector 16](#_Toc57798217)

[6 Pengertian jQuery Event dan Cara Penggunaanya 19](#_Toc57798218)

[6.1 Cara Penggunaan jQuery Event 20](#_Toc57798219)

1. Pengertian jQuery

* **jQuery** adalah **sebuah library JavaScript**.
* Dalam dunia pemrograman, **library** adalah kumpulan dari berbagai fungsi ‘siap pakai’ untuk memudahkan pembuatan sebuah aplikasi.
* Dengan demikian, **jQuery adalah kumpulan fungsi-fungsi JavaScript yang memudahkan penulisan kode JavaScript**.
* **jQuery** dikembangkan pertama kali oleh [John Resig](https://en.wikipedia.org/wiki/John_Resig) di tahun 2006.
* Sejak saat itu, jQuery berkembang menjadi proyek opensource dan menjadi libary JavaScript paling populer di dunia.
* Situs resmi jQuery beralamat di [jquery.com](http://jquery.com/).



* **Mengapa menggunakan jQuery?**
* JavaScript sangat powerfull untuk memanipulasi element HTML, tapi penggunaannya juga tidak mudah dan ‘sedikit rumit’.
* Sebagai contoh, untuk mencari sebuah element HTML, dengan JavaScript kita menggunakan fungsi **getElementById()**:

|  |
| --- |
| var x = document.getElementById("tombol"); |
|  |

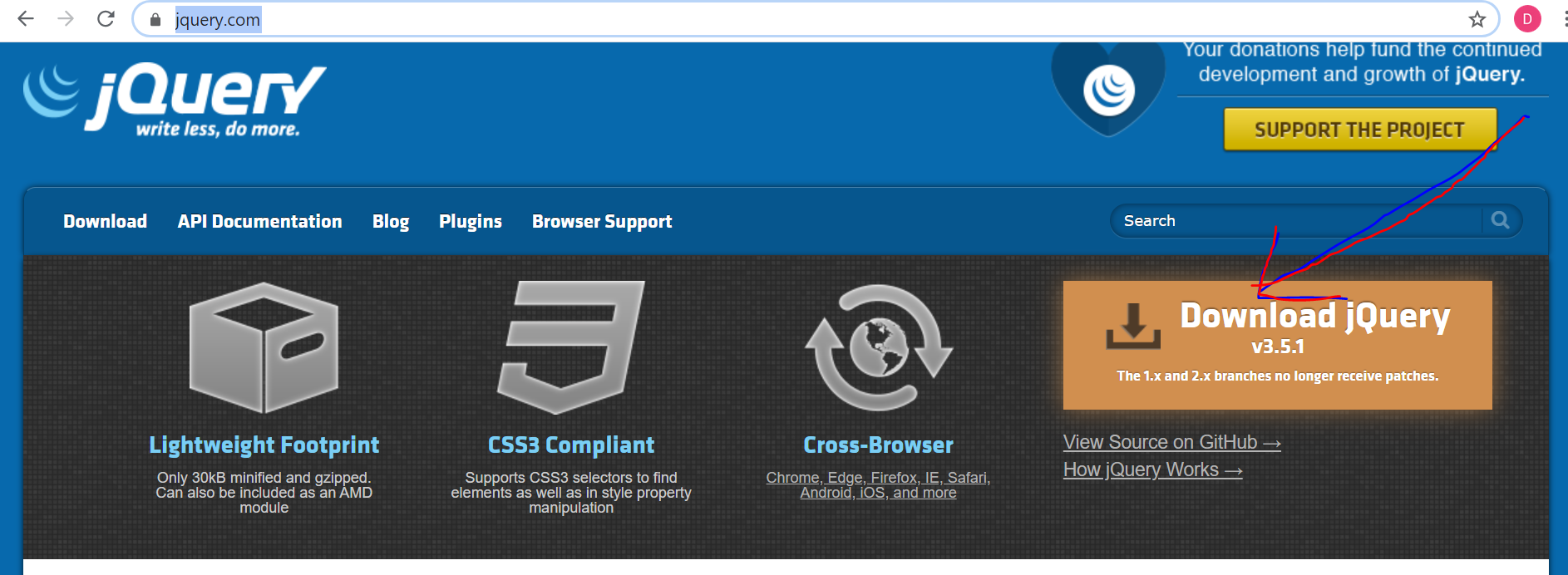
* Dengan jQuery, penulisannya jauh lebih singkat:

|  |
| --- |
| var x = $("#tombol"); |
|  |
|  |

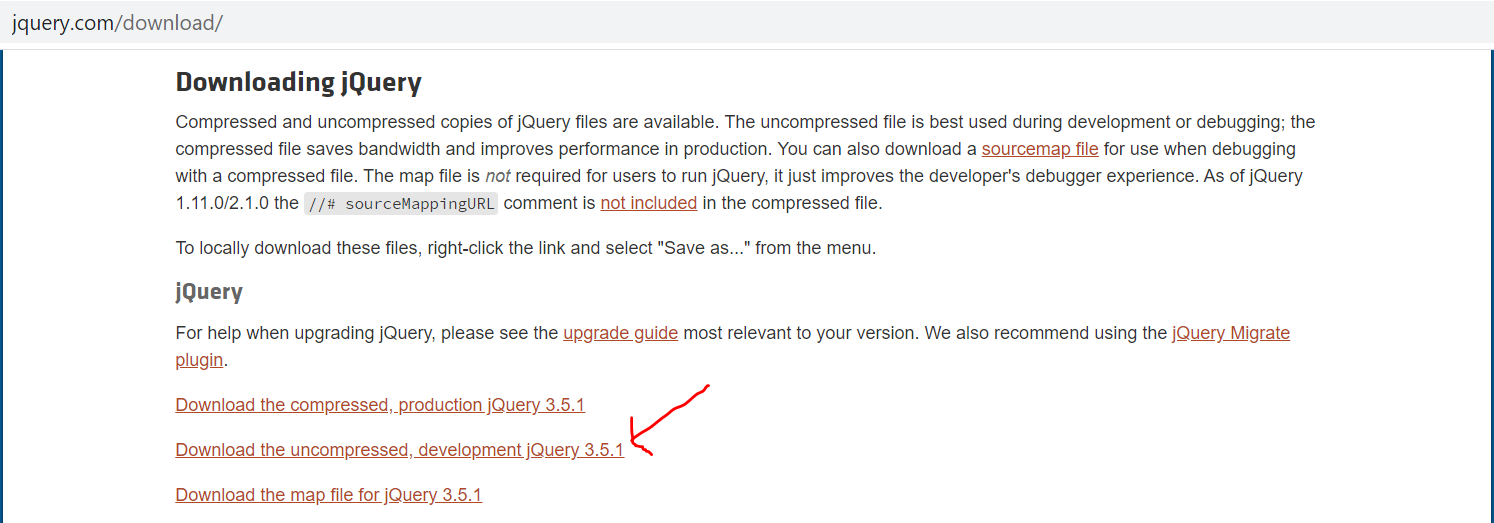
* Selain itu terdapat beberapa perbedaan implementasi JavaScript dari 1 web browser dengan web browser lain.
* Kode JavaScript yang berjalan mulus di Mozilla Firefox, belum tentu bisa berjalan di Internet Explorer, begitu juga sebaliknya.
* Hal ini diperparah dengan perbedaan versi dari masing-masing web browser.
* Untuk mengatasi hal ini, jQuery memiliki fungsi bawaan yang mendeteksi web browser dan menyediakan fungsi pengganti.
* Dengan demikian, kita tidak perlu pusing memikirkan perbedaan implementasi JavaScript dari setiap web browser.
* Fungsi JavaScript lanjutan seperti **AJAX** (Asynchronous JavaScript and XML), juga disederhanakan oleh jQuery. Untuk menggunakan AJAX, kita hanya butuh beberapa baris perintah .
* Fitur lain dari jQuery (dan mungkin menjadi alasan paling penting), jQuery menyediakan berbagai efek animasi menarik. sangat mudah digunakan dan hanya membutuhkan 1 atau 2 baris kode program. Khusus Beberapa diantaranya untuk membuat efek animasi ini, jQuery juga memiliki **jQuery UI** yang fokus ke animasi dan interaksi user.
* jQuery juga memiliki berbagai plugin atau kode tambahan untuk fungsi-fungsi lanjutan, seperti membuat efek slideshow, image carousel, hingga jQuery mobile. Efek-efek seperti ini dibuat berdasarkan kode dasar jQuery.

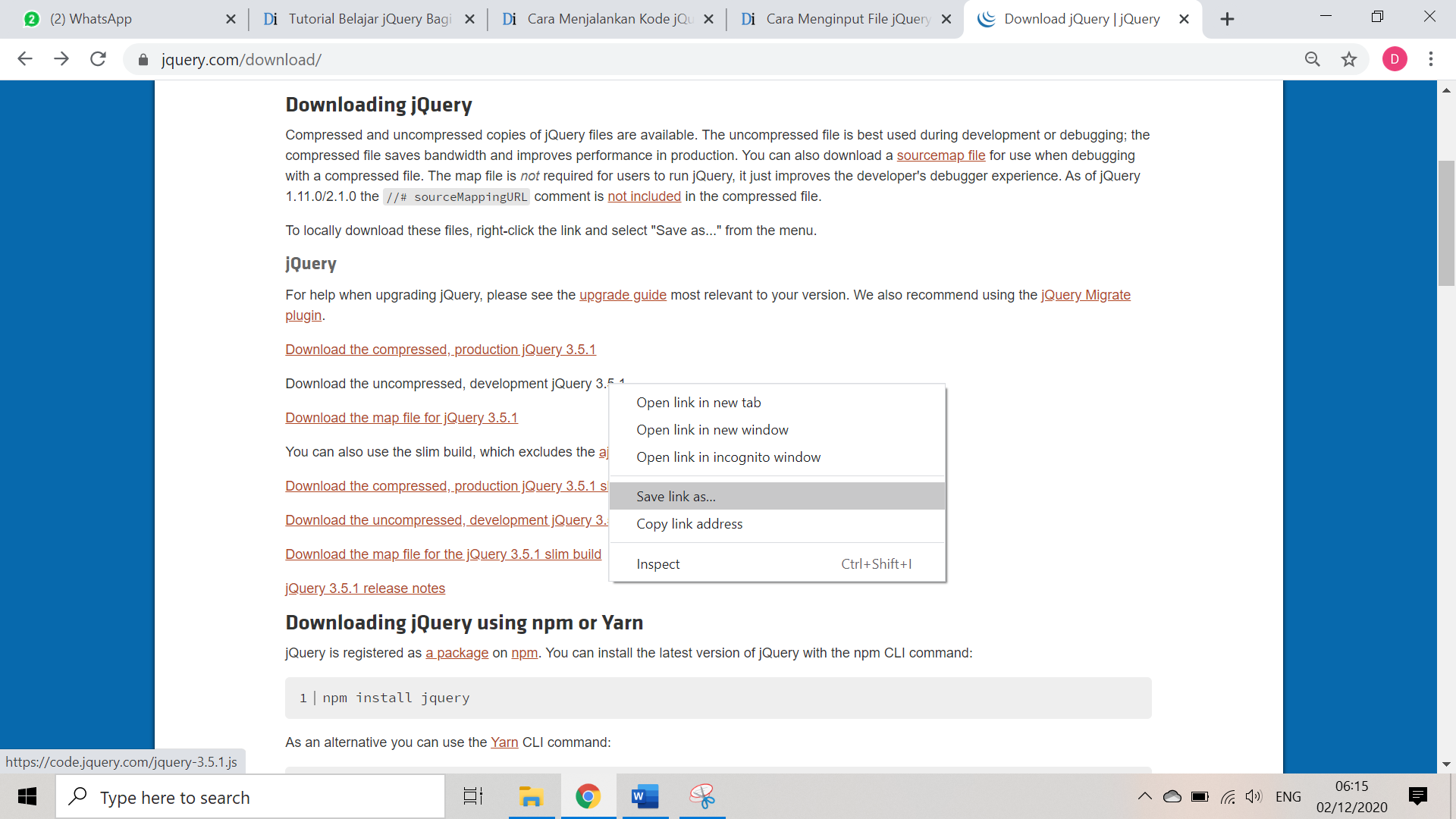
1. Cara Menginput File jQuery ke HTML

* Pada dasarnya, jQuery hanyalah sebuah file **javascript external**. Agar bisa menggunakan jQuery, kita harus download file jQuery, kemudian menghubungkannya menggunakan tag **<script>**.
* Terdapat 2 cara menginput file jQuery: mendownload dan mengakses jQuery secara lokal, atau menggunakan **CDN** (Content Delivery Network).
  1. Cara Mendownload dan Mengakses jQuery di Komputer Lokal
* Untuk menggunakan jQuery secara lokal (offline), kita harus mendownload file jQuery, kemudian mengaksesnya melalui tag **<script>**.
* Silahkan buka situs resmi jQuery di [jquery.com](http://jquery.com/). Pada bagian kanan tengah, terdapat tombol “**Download jQuery**”.



* **Compressed vs Uncompressed**
* Versi **compressed** berukuran lebih kecil karena seluruh karakter yang tidak perlu, telah di hapus (seperti spasi dan baris komentar). Hasilnya, ukuran file jQuery versi **compressed** menjadi lebih kecil, tapi tidak bisa dibaca.
* Disarankan memimilih versi **uncompressed.**





* mempersiapkan sebuah folder **belajar\_jquery** di drive **D**. Dengan demikian saya akan menyimpannya di **D:\belajar\_jquery**.
  1. Menghubungkan File jQuery dengan HTML
* Untuk menghubungkan file jQuery dengan HTML, kita cukup menambahkan tag **<script>** dengan alamat file **jquery-3.5.1.js**, seperti berikut ini:

<!DOCTYPE html>

<html>

<head>

  <meta charset="UTF-8">

  <title>Belajar jQuery</title>

  <script src="jquery-3.5.1.js"></script>

</head>

<body>

</body>

</html>

* Kode HTML diatas harus ditempatkan di dalam 1 folder dengan file **jquery-3.5.1.js**. Apabila anda sudah mempelajari HTML dan CSS, tentu tidak asing dengan maksud penempatan file ini.
  1. Cara Menginput File jQuery menggunakan CDN
* Cara lain untuk bisa menggunakan jQuery adalah mengakses file jQuery yang ditempatkan di **CDN** (Content Delivery Network).
* Secara sederhana, **CDN** adalah sejenis harddisk global yang bisa diakses oleh siapa saja. Terdapat banyak pilihan CDN, kali ini saya akan menggunakan **Google CDN,** dengan demikian kode HTMLnya menjadi seperti berikut:

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

  <title>Belajar jQuery</title>

  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.4/jquery.min.js">

  </script>

</head>

<body>

</body>

</html>

* Dengan menggunakan CDN, kita tidak perlu repot-repot mendownload file jQuery. Tapi agar bisa berjalan, harus selalu online (terhubung ke internet).

1. Cara Menjalankan Kode jQuery (event ready)

* Seperti yang sudah kita bahas, **jQuery** sebenarnya adalah **JavaScript**. Dengan demikian, kode yang ditulis pun adalah kode JavaScript. jQuery hanya ‘memudahkan’ penulisan JavaScript dengan berbagai fungsi-fungsi bawaan.
* Untuk menginput kode JavaScript dengan ‘rasa’ jQuery, cara paling umum adalah menempatkannya setelah tag **<script>** yang digunakan untuk memanggil file jQuery, seperti contoh berikut:

<!DOCTYPE html>

<html>

<head>

  <meta charset="UTF-8">

  <title>Belajar jQuery</title>

  <script src="jquery-3.5.1.js"></script>

  <script>

    // kode JavaScript dan jQuery disini

    // kode JavaScript dan jQuery disini

    // kode JavaScript dan jQuery disini

    // kode JavaScript dan jQuery disini

  </script>

</head>

<body>

</body>

</html>

* Dengan ‘template’ diatas, sebenarnya kita sudah bisa langsung menjalankan fungsi-fungsi jQuery.
* Namun terdapat 1 fungsi yang selalu disertakan dalam setiap kode jQuery, yakni event **ready()**.
  1. Event ready() atau The Document Ready Event
* Untuk memahami fungsi **ready() jQuery** atau secara formal disebut sebagai **The Document Ready Event**, kita harus paham bagaimana JavaScript bekerja.
* Secara sederhana, JavaScript digunakan untuk memanipulasi objek HTML seperti paragraf, tombol, form, gambar, dll.
* Jika kita menempatkan kode JavaScript di awal (pada bagian **<head>**), besar kemungkinan objek HTML ini belum tersedia. Ini terjadi karena bagian **<body>** baru diproses setelah bagian **<head>**.
* Salah satu solusi untuk hal ini adalah dengan meletakkan kode javaScript di bagian penutup halaman, yakni sebelum tag penutup **</body>**. Namun jQuery memiliki solusi yang lebih elegant, yakni event**ready().**
* Event **ready() jQuery**men-simulasi efek dari event **Onload** **JavaScript**, tapi lebih efisien. Dengan menggunakan fungsi **ready() jQuery**, kode program JavaScript baru akan dijalankan setelah web browser selesai memproses tag HTML, tapi sebelum file external seperti gambar.
* Dengan demikian, kode JavaScript diproses dengan lebih awal daripada event **Onload** bawaan JavaScript.
* Penulisan fungsi **ready() jQuery**adalah sebagai berikut:

<!DOCTYPE html>

<html>

<head>

  <meta charset="UTF-8">

  <title>Belajar jQuery Duniailkom</title>

  <script src="jquery-2.1.4.js"></script>

  <script>

    $(document).ready(function() {

      // Kode jQuery dan JavaScript disini

      // Kode jQuery dan JavaScript disini

      // Kode jQuery dan JavaScript disini

    });

  </script>

</head>

<body>

</body>

</html>

* Kode JavaScript yang berada didalam fungsi **$(document).ready(function())** akan dijalankan setelah seluruh tag HTML diproses oleh web browser. Dengan demikian, kita bisa menghindari error yang terjadi kerena objek yang ingin dimanipulasi belum tersedia.
* Jika anda melihat tutorial-tutorial tentang jQuery, hampir dipastikan seluruh kode akan berada di dalam event **ready()** ini.
* Sebagai contoh, saya akan membuat sebuah kode program sederhana, dimana ketika tombol HTML di klik, akan tampil sebuah kalimat. Berikut kode programnya:

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Belajar jQuery</title>

<script src="jquery-3.5.1.js"></script>

<script>

$(document).ready(function() {

$("#tombol").click(function() {

$(this).after('<p>Sedang Belajar jQuery di kelas Web Programming 2</p>');

});

});

</script>

</head>

<body>

<h1>Belajar jQuery di Web Programming 2</h1>

<button id="tombol">Klik Disini!</button>

</body>

</html>

* Agar bisa berjalan, simpanlah kode program diatas di folder yang sama dengan file **jquery-3.5.1.js**.

1. Aturan Dasar Penulisan Kode jQuery
   1. Pengertian jQuery Object ($)

* Sepanjang penggunaan jQuery, hampir selalu didahului tanda $.
* Tanda dollar ‘$’ (*dollar sign*) adalah sebuah shortcut untuk mengakses **jQuery Object**.
* Di dalam*jQuery Object* inilah seluruh fungsi-fungsi jQuery berada.
* Sebagai contoh, untuk mencari element HTML yang memiliki atribut **id=”tombol”**, kita bisa menulis:

|  |
| --- |
| $("#tombol") |

* Jika karena sesuatu hal kita tidak bisa menggunakan tanda $ (biasanya karena bentrok dengan library lain), bisa menggunakan **jQuery()**, seperti berikut:

|  |
| --- |
| jQuery("#tombol") |

* 1. jQuery Action / jQuery Event
* Untuk bisa melakukan ‘sesuatu’ terhadap element HTML, kita tinggal menyambung penulisan **jQuery Object** dengan **jQuery Action**, atau dikenal juga dengan jQuery Event.
* Format dasarnya adalah sebagai berikut:

|  |
| --- |
| $(selector).action() |

* **Selector** digunakan untuk mencari element mana dari HTML yang akan dimanipulasi. Bisa berupa sebuah paragraf <p>, sebuah gambar <img>, atau sesuatu yang lebih kompleks, seperti element <tr> pertama dari tabel kedua. Kita akan membahas jenis-jenis selector jQuery dalam modul selanjutnya.
* **Action** adalah sesuatu yang bisa kita lakukan dengan element HTML tersebut, apakah menyembunyikannya, menampilkannya, mengubah warna, menambahkan element baru, dll. Action mirip seperti **Event** dalam JavaScript, tetapi dengan penambahan berbagai fitur lain. Di dalam jQuery, sebuah action bisa disambung dengan action lain, atau dikenal dengan isitilah **chaining**.
* Berikut contoh penulisan*jQuery Object*, *selector* dan *action*:

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Belajar jQuery</title>

<script src="jquery-3.5.1.js"></script>

<script>

$( document ).ready(function() {

$( "#tombol\_hide" ).click(function() {

$("h1").hide();

});

$( "#tombol\_show" ).click(function() {

$("h1").show();

});

$( "#tombol\_small" ).click(function() {

$("h1").css("font-size","20px");

});

$( "#tombol\_red" ).click(function() {

$("h1").css( "color", "red" );

});

});

</script>

</head>

<body>

<h1>Belajar jQuery di Web Programming 2</h1>

<button id="tombol\_hide">Sembunyikan</button>

<button id="tombol\_show">Tampilkan</button>

<button id="tombol\_small">Perkecil</button>

<button id="tombol\_red">Merahkan</button>

</body>

</html>

* Tampilan diatas masih sangat sederhana, dimana ketika tombol ditekan, akan terjadi “sesuatu” terhadap tag <h1>. Menggunakan jQuery, kita bisa membuat berbagai efek menarik lainnya.
* Disini hanya memperlihatkan bagaimana sebuah kode jQuery ditulis, terutama bagian seperti  **$(“h1”).hide()**, atau **$(“h1”).css(“font-size”,”20px”)**. Ini semua adalah cara penulusan gabungan dari **jQuery Object**, **jQuery Selector**, dan **jQuery Action**.

1. Cara Mencari Elemen HTML dengan jQuery Selector
   1. Pengertin Method getElementById() JavaScript

* Sebagaimana fungsi utama JavaScript, kita memerlukan JavaScript untuk memanipulasi element HTML.
* Langkah pertama adalah mencari element mana yang akan diubah, apakah itu tag **<p>**, tag **<p>** dengan atribut **id=”penting”**, tag **<p>** dengan atribut **class=”warning”**, atau sebuah tombol **<button>**.
* Di dalam JavaScript, untuk mencari element HTML ini kita biasa menggunakan method **getElementById().**
* Method atau fungsi ini berfungsi untuk mencari element HTML berdasarkan atribut id-nya. Sebagai contoh, jika saya memiliki sebuah kode HTML berikut:

|  |
| --- |
| <button id="tombol">Click Me!</button> |

* Maka untuk mengakses element ini kita bisa menggunakan:

|  |
| --- |
| var x = document.getElementById("tombol"); |

* Dengan menggunakan**jQuery Selector**, kode programnya jauh lebih singkat dan lebih powerfull.
  1. Pengertian jQuery Selector
* jQuery memudahkan proses pencarian element HTML dengan menyediakan jQuery Selector.
* **jQuery Selector** adalah sebuah cara yang disediakan jQuery untuk mencari element HTML. Tidak hanya dengan atribut id saja, kita juga bisa menggunakan class, atribut lain, dan kombinasi diantaranya.
* Dengan jQuery, untuk mencari sebuah element HTML yang memiliki **id=”tombol”**, penulisannya jauh lebih singkat:

var x = $("#tombol");

* Karena sama seperti selector CSS, jQuery selector mendukung hampir semua penulisan selector dari CSS, berikut contohnya:

1. Mencari seluruh element dengan tag tertentu. Misalkan kita ingin mencari seluruh tag <p>, penulisan jQuery Selectornya adalah: **$(“p”)**.
2. Mencari element dengan id tertentu. Misalkan ingin mencari sebuah tag HTML yang memiliki **id=”belajar”**, maka kita bisa menulis: **$(“# belajar”)**.
3. Mencari element dengan class tertentu. Misalkan kita ingin mencari seluruh tag HTML yang menggunakan atribut **class=”warna”**, maka bisa ditulis dengan: **$(“.warna”)**.

* Sama seperti CSS, kita juga bisa mengkombinasikan selector untuk mencari element yang lebih spesifik, misalnya untuk mencari tag **<p>** yang memiliki **class=”warna”** dan berada di dalam tag **<div>**, selectornya adalah: **$(“div p.warna”)**.
* Berikut contoh penggunaan dari jQuery Selector:

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Belajar jQuery</title>

<script src="jquery-3.5.1.js"></script>

<script>

$( document ).ready(function() {

$("#tombol").click(function() {

$("p").css( "color", "red" );

$("#belajar").css( "color", "green" );

$(".warna").css( "color", "blue" );

$("#test.saja").css( "color", "yellow" );

$("div p.warna").css( "color", "pink" );

});

});

</script>

</head>

<body>

<p>

Sebuah paragraf tanpa atribut apapun

</p>

<p id="belajar">

Sebuah paragraf dengan id = "belajar"

</p>

<p class="warna">

Sebuah paragraf dengan class = "warna"

</p>

<p id="test" class="saja">

Sebuah paragraf dengan id="test" class="saja"

</p>

<div>

<p class="warna">

Sebuah paragraf tanpa atribut apapun

</p>

</div>

<button id="tombol">Action</button>

</body>

</html>

* Pada contoh kode program diatas, terdapat 6 penulisan **jQuery Selector**.

$( “#tombol” )

$(“p”)

$(“#belajar”)

$(“.warna”)

$(“#test.saja”)

$(“div p.warna”)

* Selector pertama, **$(“#tombol”)** digunakan untuk membuat event **click**.
* Maksudnya, ketika sebuah tag HTML yang memiliki **id=”tombol**” di klik, jalankan perintah yang dibuat.
* Di dalam kode diatas, tag HTML tersebut adalah sebuah tag **<button id=”tombol”>**.
* Setelah tag **<button id=”tombol”>** di-klik, saya membuat 5 baris perintah yang masing-masingnya digunakan untuk mengubah warna paragraf:

|  |  |
| --- | --- |
|  | $("p").css( "color", "red" );  $("#belajar").css( "color", "green" );  $(".warna").css( "color", "blue" );  $("#test.saja").css( "color", "yellow" );  $("div p.warna").css( "color", "pink" ); |

* Sebagai contoh, kode **$(“#belajar”).css( “color”, “green” )** artinya: temukan sebuah tag HTML dengan id=”belajar”, kemudian ubah warnanya (color) menjadi hijau (green).
* Sekali lagi, disini kita hanya fokus ke cara penulisan selector.
* Event **css()** akan kita pelajari dalam modul terpisah.

1. Pengertian jQuery Event dan Cara Penggunaanya

* **jQuery Selector** dan **jQuery Event** merupakan inti dari jQuery.
* Dengan menggabungkan keduanya, kita bisa membuat berbagai interaksi menarik.
* Dalam tutorial belajar jQuery modul ini akan membahas lebih jauh [Pengertian dan Cara Penggunaan jQuery Event](https://www.duniailkom.com/tutorial-belajar-jquery-pengertian-jquery-event-dan-cara-penggunaannya/).
* Misalkan saya ingin membuat pesan **alert()** ketika sebuah paragraf di klik, dengan JavaScript ini bisa dilakukan dengan kode program berikut:

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Belajar jQuery</title>

</head>

<body>

<p id="paragraf" onclick="alert('Saya sudah di klik!')">

Silahkan klik saya

</p>

</body>

</html>

* Atribut **onclick** dalam tag <p> diatas adalah sebuah **Event**. Ketika di klik, akan tampil pesan popup: ‘Saya sudah di klik!’. Hasil ini didapat dengan fungsi **alert()** JavaScript.
* Selain **onclick**, di dalam JavaScript juga banyak event-event lain, seperti ondblclick, onmouseover, onmouseout, dll.
* Penulisan seperti ini sangat praktis namun tidak “rapi”. Karena kita mencampurkan kode HTML dengan JavaScript. Bagaimana cara memisahkannya?
* Di dalam JavaScript, ini bisa dilakukan menggunakan fungsi **addEventListener()**. Hasil yang sama bisa dihasilkan dengan kode program berikut:

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Belajar jQuery</title>

<script src="jquery-3.5.1.js"></script>

<script>

window.onload = function()

{

document.getElementById("paragraf").

addEventListener("click", function(){alert ('Saya sudah di klik!')});

}

</script>

</head>

<body>

<p id="paragraf">

Silahkan klik saya

</p>

</body>

</html>

* Sekarang, di dalam bagian **<body>** tidak ada kode JavaScript apapun. Event **click** saya tambahkan dengan fungsi **addEventListener()**.
  1. Cara Penggunaan jQuery Event
* Perhatikan bagaimana jQuery memudahkan penulisan event ini, sangat sederhana. Format penulisannya adalah:

|  |
| --- |
| $("jQuery\_Selector").jQuery\_Event(function() { ...isi event disini...   }) |

* Selain event **click**, juga tersedia berbagai event lain, seperti **dblclick**, **mouseenter**, **mouseover**, **mouseleave**, dll.
* Berikut contoh “aksi” dari berbagai event javascipt yang dibuat dengan jQuery:

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Belajar jQuery</title>

<script src="jquery-3.5.1.js"></script>

<script>

$( document ).ready(function() {

$("#tombol").click(function() {

$("p").css( "color", "red" );

$("#belajar").css( "color", "green" );

$(".warna").css( "color", "blue" );

$("#test.saja").css( "color", "yellow" );

$("div p.warna").css( "color", "pink" );

});

});

</script>

</head>

<body>

<p>

Sebuah paragraf tanpa atribut apapun

</p>

<p id="belajar">

Sebuah paragraf dengan id = "belajar"

</p>

<p class="warna">

Sebuah paragraf dengan class = "warna"

</p>

<p id="test" class="saja">

Sebuah paragraf dengan id="test" class="saja"

</p>

<div>

<p class="warna">

Sebuah paragraf tanpa atribut apapun

</p>

</div>

<button id="tombol">Action</button>

</body>

</html>