



Free Kelas WAPPSANDROID

Materi Pertemuan #06

Senin, 26 Maret 2018

Pukul 20.00 – 23.00 WIB

Pengenalan dan Dasar – Dasar JQuery

oleh :

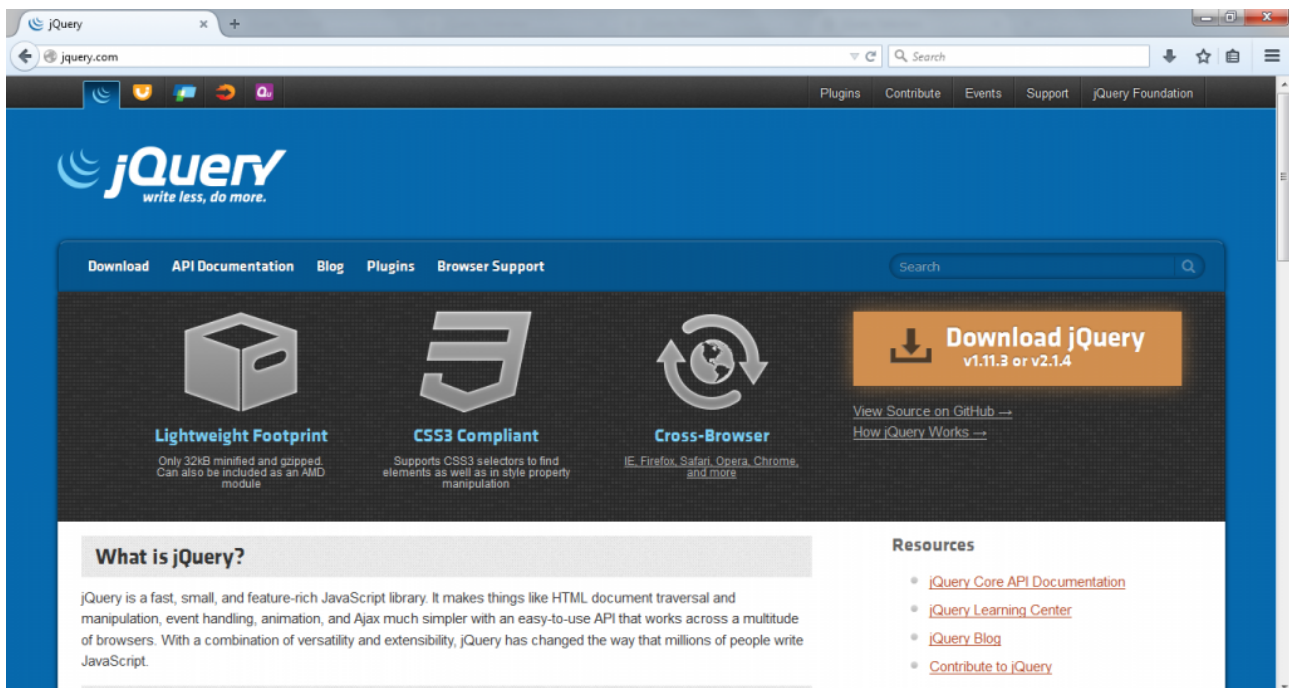
Agus Muhajir

<http://github.com/hajirodeon>

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](#) di tahun 2006. Sejak saat itu, jQuery berkembang menjadi proyek opensource dan menjadi library JavaScript paling populer di dunia. Situs resmi jQuery beralamat di jquery.com.



Gambar 01. Situs JQuery.com

Mengapa harus menggunakan jQuery?

Untuk dapat menjawab pertanyaan ini sebaiknya anda sudah pernah membuat 1 atau 2 kode program JavaScript. JavaScript sangat powerfull untuk memanipulasi element HTML, tapi penggunaannya juga tidak mudah dan ‘sedikit ribet’.

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 (sebagaimana CSS) 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. Beberapa diantaranya sangat mudah digunakan dan hanya membutuhkan 1 atau 2 baris kode program. Khusus 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.

Dengan berbagai fitur dalam jQuery, tidak heran jika jQuery sangat populer. Tidak jarang pula programmer pemula lebih mengenal jQuery daripada JavaScript.

Cara Menginput File jQuery ke HTML

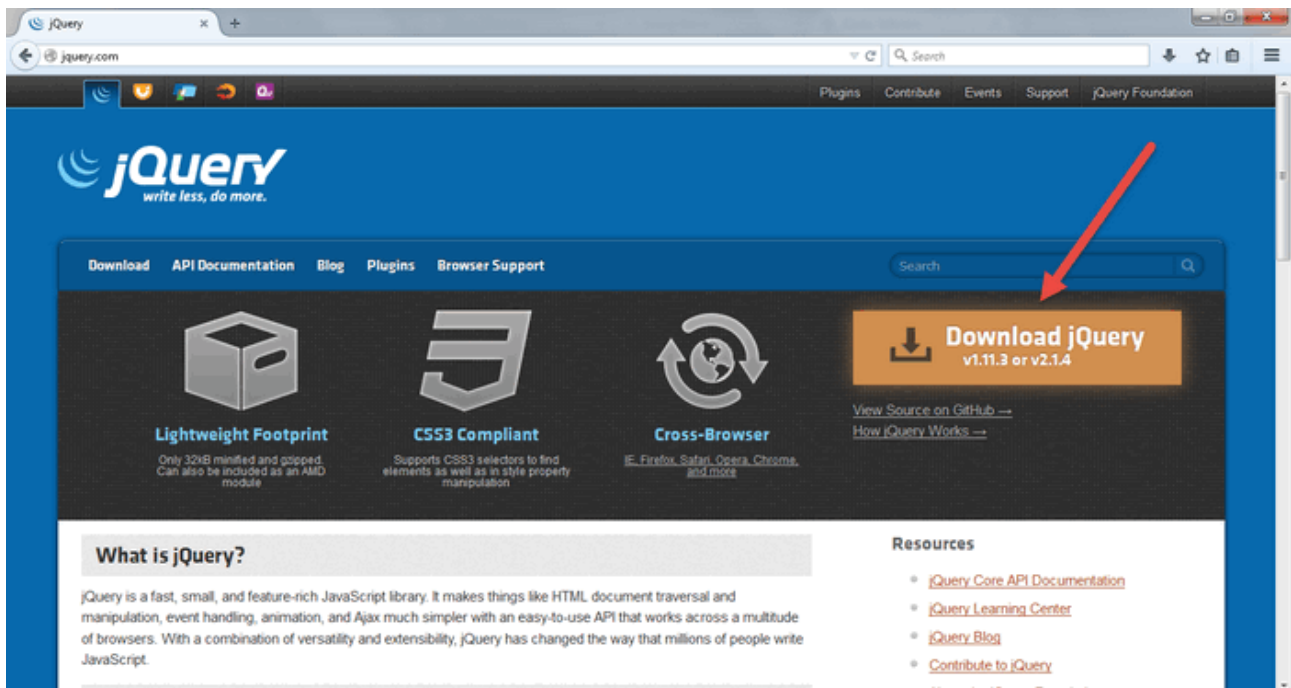
Pada dasarnya, jQuery hanyalah sebuah file **javascript external**. Agar bisa menggunakan jQuery, kita harus download file jQuery, kemudian menghubungkannya menggunakan tag **<script>**.

Terdapat 2 cara menginput file jQuery: mendownload dan mengakses jQuery secara lokal, atau menggunakan **CDN** (*Content Delivery Network*).

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. Pada bagian kanan tengah, terdapat tombol “Download jQuery”. Klik tombol ini.



Gambar 02. Tampilan Download JQuery

Kita akan dibawa ke halaman jquery.com/download. Pada bagian ini anda bisa melihat 2 versi dan 2 jenis file jQuery.

Pada saat tutorial ini ditulis, terdapat 2 versi jQuery: **jQuery 1.x** dan **jQuery 2.x**. Apa perbedaan keduanya?

Perbedaan jQuery 1.x dengan jQuery 2.x

Perbedaan paling mendasar adalah, **jQuery 2.x** tidak lagi mendukung Internet Explorer 6, 7 dan 8. Jika anda tidak peduli dengan IE, silahkan download versi 2.x. Namun jika anda akan menjalankan atau masih menggunakan IE 8, silahkan download versi 1.x

Perbedaan versi Compressed and Uncompressed

Setelah memilih salah satu versi, berikutnya kita juga harus memilih antara versi **compressed** and **uncompressed**.

Kedua file ini sebenarnya memiliki kode jQuery yang sama. Perbedaannya, 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.

Jika anda ingin mempelajari kode program yang ada di dalam jQuery (atau ingin mengembangkannya), pilih versi **uncompressed**. Anda bisa mendownload sendiri kedua versi ini untuk melihat perbedaannya.

Cara mendownload jQuery

Untuk tutorial ini saya akan menggunakan **jQuery versi 2.1.4 uncompressed**. Silahkan klik kanan link “**Download the uncompressed, development jQuery 2.1.4**”, lalu pilih **Save As**. Simpanlah file **jquery-2.1.4.js** di tempat yang mudah diakses.

jQuery 2.x

jQuery 2.x has the same API as jQuery 1.x, but *does not support Internet Explorer 6, 7, or 8*. All the notes in the [jQuery 1.9 Upgrade Guide](#) apply here as well. Since IE 8 is still relatively common, we recommend using the 1.x version unless you are certain no IE 6/7/8 users are visiting the site. Please read the [2.0 release notes](#) carefully.

[Download the compressed, production jQuery 2.1.4](#)

[Download the uncompressed, development jQuery 2.1.4](#)

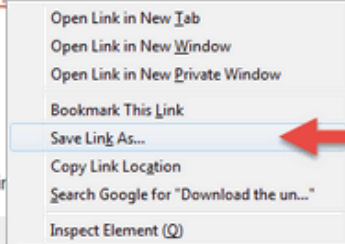
[Download the map file for jQuery 2.1.4](#)

[jQuery 2.1.4 release notes](#)

Downloading jQuery using Bower

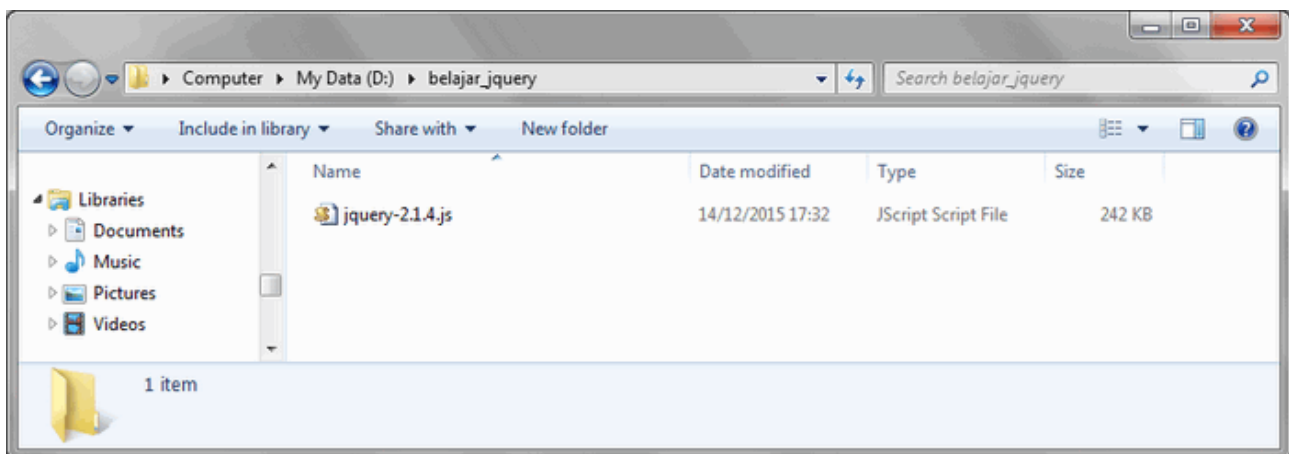
jQuery is registered as a package with [Bower](#). You can install it with the following command:

```
1 | bower install jquery
```



Gambar 03. Cara Download

Saya telah mempersiapkan sebuah folder **belajar_jquery** di drive **D**. Dengan demikian saya akan menyimpannya di **D:\belajar_jquery**.



Gambar 04. File JQuery

Menghubungkan File jQuery dengan HTML

Untuk menghubungkan file jQuery dengan HTML, kita cukup menambahkan tag **<script>** dengan alamat file **jquery-2.1.4.js**, seperti berikut ini:

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Belajar jQuery Duniailkom</title>
<script src="jquery-2.1.4.js"></script>
</head>
<body>
</body>
</html>
```

Kode HTML diatas harus ditempatkan di dalam 1 folder dengan file **jquery-2.1.4.js**. Apabila anda sudah mempelajari HTML dan CSS, tentu tidak asing dengan maksud penempatan file ini.

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 Duniailkom</title>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/2.1.4/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).

Cara Menjalankan Kode jQuery

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 Duniailkom</title>
<script src="jquery-2.1.4.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()**.

Pengertian 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:

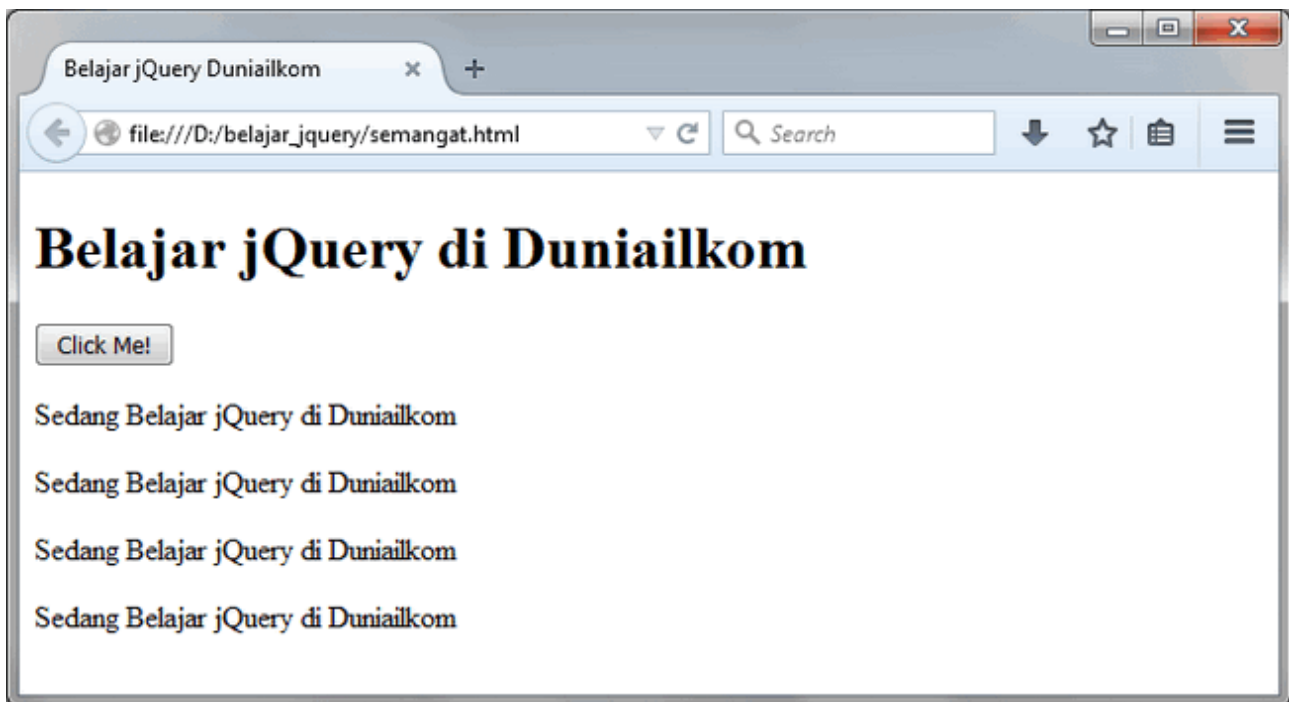
```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Belajar jQuery Duniailkom</title>
<script src="jquery-2.1.4.js"></script>
<script>
$(document).ready(function() {
// Kode jQuery dan JavaScript disini
// Kode jQuery dan JavaScript disini
// Kode jQuery dan JavaScript disini
});
</script>
</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 karena 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 Duniailkom</title>
<script src="jquery-2.1.4.js"></script>
<script>
    $(document).ready(function() {
        $("#tombol").click(function() {
            $(this).after('<p>Sedang Belajar jQuery di
Duniailkom</p>');
        });
    });
</script>
</head>
<body>
<h1>Belajar jQuery di Duniailkom</h1>
<button id="tombol">Click Me!</button>
</body>
</html>
```



Gambar 05. OnClick

Penjelasan maksud kode diatas, akan saya bahas dalam tutorial selanjutnya. Disini kita hanya fokus agar anda bisa menjalankan kode jQuery.

Agar bisa berjalan, simpanlah kode program diatas di folder yang sama dengan file **jquery-2.1.4.js**. Sebagai contoh, saya akan menyimpannya ke dalam file **semangat.html** di folder **belajar_jquery**,

Aturan Dasar Penulisan Kode jQuery

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")
```

Pengertian 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 , atau sesuatu yang lebih kompleks, seperti element <tr> pertama dari tabel kedua. Kita akan membahas jenis-jenis selector jQuery dalam tutorial 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 istilah **chaining**.

Berikut contoh penulisan *jQuery Object*, *selector* dan *action*:

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Belajar jQuery Duniaikom</title>
  <script src="jquery-2.1.4.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>
</html>
```

```
});  
</script>  
</head>  
<body>  
  <h1>Belajar jQuery di Duniaikom</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>
```



Gambar 06. Contoh penggunaan Object, Selector dan Action

Pengertian Fungsi/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.

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:

```
1var x = $("#tombol");
```

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

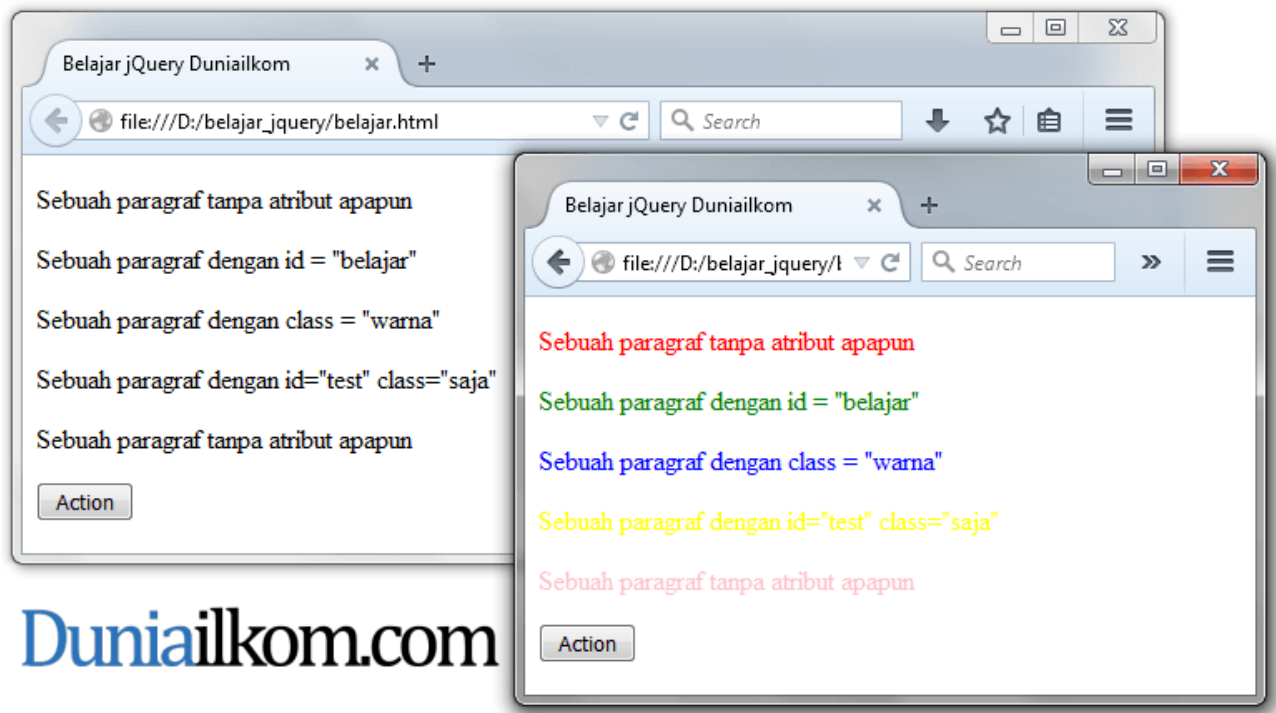
- Mencari seluruh element dengan tag tertentu. Misalkan kita ingin mencari seluruh tag `<p>`, penulisan jQuery Selectornya adalah: `$("p")`.
- Mencari element dengan id tertentu. Misalkan ingin mencari sebuah tag HTML yang memiliki **id="belajar"**, maka kita bisa menulis: `$("# belajar")`.
- Mencari 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 Duniailkom</title>
<script src="jquery-2.1.4.js"></script>
<script>
    $( document ).ready(function() {
        $("#tombol").click(function() {
            $("p").css( "color", "red" );
            $("#belajar").css( "color", "green" );
            $(".warna").css( "color", "blue" );
        });
    });
</script>
```

```
        $("#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>
```



Gambar 07. JQuery Selector

Pada contoh kode program diatas, terdapat 6 penulisan **jQuery Selector**. Dapatkah anda menemukannya? Ke enam selector tersebut adalah:

- \$("#tombol")
- \$("p")
- \$(" #belajar")
- \$(".warna")
- \$(" #test.saja")
- \$("div p.warna")

Selector pertama, `$("#tombol")` saya gunakan 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 tutorial terpisah.

Pengertian jQuery Event

Di dalam JavaScript, **Event** adalah suatu hal yang bisa dilakukan user ke sebuah element HTML, misalnya di **click**, **double click**, **mouseover** (cursor mouse berada diatas element), **mouseout** (cursor mouse sudah keluar dari atas element), dll.

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 Duniailkom</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 Duniailkom</title>
<script src="jquery-2.1.4.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()**.

Cara Penggunaan jQuery Event

Sebagai alternatif, jQuery menyediakan **jQuery Event** dengan penulisan yang lebih praktis, berikut contohnya:

```
<!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() {
```

```
    $("#paragraf").click(function() {  
        alert('Saya sudah di klik!');  
    });  
});  
</script>  
</head>  
<body>  
<p id="paragraf">  
    Silahkan klik saya  
</p>  
</body>  
</html>
```

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 javascript yang dibuat dengan jQuery:

```
<!DOCTYPE html>  
<html>  
<head>
```

```
<meta charset="UTF-8">
<title>Belajar jQuery Duniailkom</title>
<style>
#paragraf {
    width:600px;
    height:80px;
    background-color:pink;
    line-height:80px;
    text-align:center;
    font-size:30px;
}
</style>
<script src="jquery-2.1.4.js"></script>
<script>
    $( document ).ready(function() {

        $( "#paragraf" ).click(function() {
            $( "#paragraf" ).css( "color", "green" );
        });

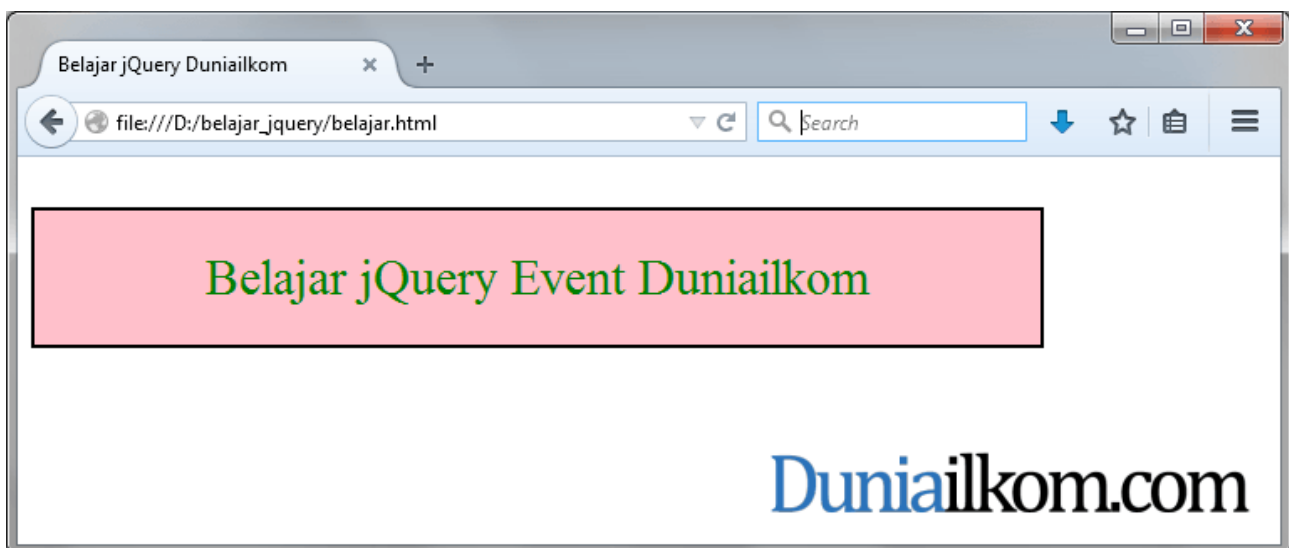
        $( "#paragraf" ).mouseover(function() {
            $( "#paragraf" ).css( "background-color", "silver" );
        });

        $( "#paragraf" ).mouseout(function() {
            $( "#paragraf" ).css( "background-color", "pink" );
        });

        $( "#paragraf" ).dblclick(function() {
            $( "#paragraf" ).css( "border", "solid 2px black" );
        });
    });
</script>
```



```
});  
  
});  
</script>  
</head>  
<body>  
<p id="paragraf">  
    Belajar jQuery Event Duniaikom  
</p>  
</body>  
</html>
```



Gambar 08. JQuery Event

Dalam contoh diatas, saya membuat 4 jQuery event, yakni **click**, **dblclick**, **mouseover**, dan **mouseleave**. Pada setiap event ini saya mengubah property CSS dari paragraf. Silahkan anda coba sendiri dengan men-klik, double klik, dan mouseover ke atas kotak pink:

Cara Mengambil nilai HTML dengan Method text() jQuery

Metode pertama untuk mengambil nilai sebuah elemen HTML dengan jQuery adalah menggunakan method **text()**. Caranya cukup menambahkan method/fungsi **text()** ke **jQuery Selector**.

Berikut contoh penggunaannya:

```
<!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() {

        $("#tombol").click(function() {
            var nilai = $("#paragraf").text();
            alert(nilai);
        })

    });
</script>
```

```
</head>
<body>
<p id="paragraf">
    Sedang belajar <em>jQuery</em> di <b>Duniaikom...</b>
</p>
<button id="tombol">Ambil Nilai</button>
</body>
</html>
```

Untuk mengambil nilai teks yang ada didalam tag **<p id="paragraf">**, saya bisa menggunakan perintah **\$("#paragraf").text()**. Selanjutnya nilai ini disimpan ke dalam variabel **nilai**, lalu ditampilkan dengan fungsi **alert(nilai)**.

Jika anda men-klik tombol “Ambil Nilai” , hasilnya adalah: **“Sedang belajar jQuery di Duniaikom...”**. Perhatikan bahwa method **text()** akan menghapus semua tag HTML yang terdapat di dalam tag **<p>**, seperti tag **** dan **** yang juga berada di dalam paragraf tersebut.

Bagaimana jika kita juga ingin mengambil nilai yang ada di dalam tag **<p>** lengkap beserta tag HTML-nya? Gunakan method **html()** jQuery

Cara Mengambil nilai HTML dengan method **html()** jQuery

Cara berikutnya untuk mengamnil nilai HTML dengan jQuery adalah menggunakan method **html()**. Penggunaannya sama persis dengan method **text()**. Bedanya, hasil dari method **html()** akan tetap mempertahankan tag HTML yang ada di dalam element HTML tersebut.

Menggunakan contoh yang sama, berikut hasil yang di dapat dengan method **html()**:

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Belajar jQuery Duniaikom</title>
<script src="jquery-2.1.4.js"></script>
```

```
<script>
    $(document).ready(function() {

        $("#tombol").click(function() {
            var nilai = $("#paragraf").html();
            alert(nilai);
        })

    });
</script>
</head>
<body>
<p id="paragraf">
    Sedang belajar <em>jQuery</em> di <b>Duniailkom...</b>
</p>
<button id="tombol">Ambil Nilai</button>
</body>
</html>
```

Cara Mengambil nilai form HTML dengan method `val()` jQuery

Baik method **`text()`** maupun **`html()`** sudah mencukupi untuk mengambil nilai yang ada disetiap element HTML. Tapi khusus untuk form, jQuery menyediakan method **`val()`**. Langsung saja kita lihat contoh penggunaannya:

```
<!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() {

        $("#tombol").click(function() {
            var nilai = $("#nama").val();
            alert(nilai);
        })

    });
</script>
</head>
<body>
Nama: <input type="text" id="nama" value="Duniailkom...">
<button id="tombol">Ambil Nilai</button>
</body>
</html>
```

Kali ini saya memiliki sebuah tag **<input type="text">**. Silahkan langsung anda klik tombol “Ambil Nilai”, atau ubah teks yang ada di dalam inputan form tersebut. Perintah **\$("#nama").val()** akan menampilkan nilai apapun yang anda input ke dalam isian form tersebut.

Bagaimana dengan objek form lain seperti <select>?

Caranya sama persis, berikut contohnya:

```
<!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() {

        $("#tombol").click(function() {
            var nilai = $("#kota").val();
            alert(nilai);
        })

    });
</script>
</head>
<body>
Pilih Kota:
<select id="kota" name="kota">
    <option value="jakarta">Jakarta</option>
```

```
<option value="bandung">Bandung</option>
<option value="bogor">Bogor</option>
</select>
<button id="tombol">Ambil Nilai</button>
</body>
</html>
```

Silahkan anda pilih salah satu nama kota, lalu klik tombol “Ambil Nilai”. Nilai yang ditampilkan berasal dari atribut value dari setiap tag **<option>**.

Bagaimana dengan checkbox?

Untuk **checkbox** perlu penyeleksian lebih lanjut, karena jika menggunakan method **val()** secara langsung, nilai form akan dikembalikan apapun kondisi checkbox (baik sudah dipilih maupun belum). Untuk mengatasi hal ini, kita bisa menggunakan selector khusus jQuery, yakni **:checked**.

Berikut contoh penggunaannya:

```
<!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() {

        $("#tombol").click(function() {
            var nilai1 = $("#keren:checked").val();
            var nilai2 = $("#ganteng:checked").val();
            var nilai3 = $("#cool:checked").val();
```

```
        alert(nilai1+' '+nilai2+' '+nilai3);
    })

});
</script>
</head>
<body>
Saya itu:
    <input type="checkbox" id="keren" value="Saya Keren"> Keren
    <input type="checkbox" id="ganteng" value="Saya Ganteng"> Ganteng
    <input type="checkbox" id="cool" value="Saya Cool"> Cool
<br><br>
    <button id="tombol">Ambil Nilai</button>
</body>
</html>
```

Disini saya memiliki 3 check box: **Keren**, **Ganteng** dan **Cool**. Ketika tombol **Ambil Nilai** di klik, akan dijalankan perintah berikut:

```
var nilai1 = $("#keren:checked").val();
var nilai2 = $("#ganteng:checked").val();
var nilai3 = $("#cool:checked").val();
alert(nilai1+' '+nilai2+' '+nilai3);
```

Artinya, variabel **nilai1**, **nilai2** dan **nilai3** akan berisi **value** dari checkbox hanya ketika kondisi checkbox tersebut dalam keadaan ter-checklist (perhatikan cara penulisan jQuery selector, **#keren:checked**).

Akan tetapi, sekarang muncul masalah baru. Ketika checkbox tidak terpilih, method **val()** akan mengembalikan nilai **undefined**. Ini terjadi karena selector **:checked** tidak menemukan element HTML yang kita tulis (karena belum di check).

Bagaimana cara mengatasinya? cukup dengan menambahkan beberapa kondisi IF, dimana jika isi variabel nilai adalah **undefined**, lakukan sesuatu, misalnya mengosongkan variabel.

Berikut revisinya:

```
<!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() {

        $("#tombol").click(function() {
            var nilai1 = $("#keren:checked").val();
            var nilai2 = $("#ganteng:checked").val();
            var nilai3 = $("#cool:checked").val();

            if (nilai1==undefined) {nilai1="";}
            if (nilai2==undefined) {nilai2="";}
            if (nilai3==undefined) {nilai3="";}

            alert(nilai1+' '+nilai2+' '+nilai3);
        })
    })
}
```

```
});  
</script>  
</head>  
<body>  
Saya itu:  
<input type="checkbox" id="keren" value="Saya Keren"> Keren  
<input type="checkbox" id="ganteng" value="Saya Ganteng"> Ganteng  
<input type="checkbox" id="cool" value="Saya Cool"> Cool  
<br><br>  
<button id="tombol">Ambil Nilai</button>  
</body>  
</html>
```

Kali ini ketika tombol **Ambil Nilai** di klik, yang akan tampil hanyalah nilai checkbox yang sudah dipilih. Dalam kasus seperti ini anda juga bisa menyaksikan bahwa jQuery saja tidaklah cukup. Kita tetap membutuhkan JavaScript dasar untuk menangani masalah yang muncul.

Cara Mengubah Nilai HTML dengan Method `text()` dan `html()`

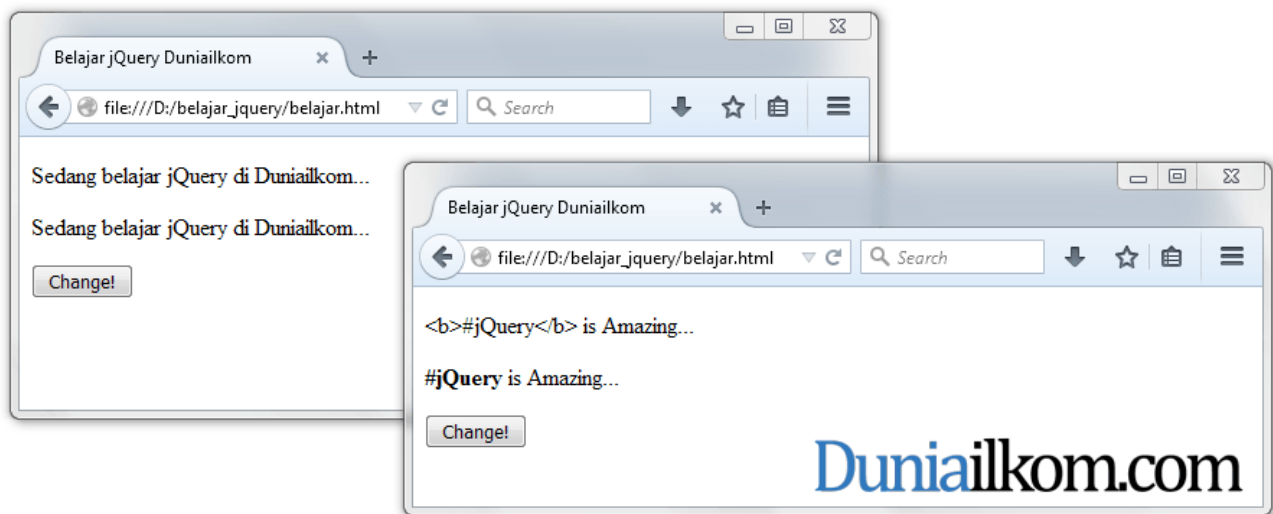
Untuk mengubah nilai elemen HTML, caranya sangat mirip dengan mengambil elemen. Bedanya, kita menyisipkan sebuah argumen ke dalam method **`text()`** dan **`html()`**. Langsung saja kita lihat contohnya:

```
<!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() {

        $("#tombol").click(function() {
            $("#paragraf1").text("<b>#jQuery</b> is Amazing...");
            $("#paragraf2").html("<b>#jQuery</b> is Amazing...");
        })

    });
</script>
</head>
<body>
<p id="paragraf1">
    Sedang belajar jQuery di Duniailkom...
</p>
<p id="paragraf2">
    Sedang belajar jQuery di Duniailkom...
</p>
<button id="tombol">Change!</button>
</body>
</html>
```



Gambar 09. Cara Ubah Nilai HTML

Dalam kode program diatas, saya memiliki 2 buah tag `<p>`, masing-masing dengan id **paragraf1** dan **paragraf2**. Ketika tombol “**Change!**” di-klik, saya mengubah isi setiap paragraf dengan method **text()** dan **html()**.

Perhatikan cara penggunaannya yang sangat mirip dengan tutorial kita sebelumnya (mengambil nilai HTML). Jika method **text()** dan **html()** diisi dengan argumen, ini artinya kita ingin mengubah isi element HTML, bukan mengambil nilainya. Konsep seperti ini umum digunakan di dalam jQuery.

Kode:

```
$("#paragraf1").text("<b>#jQuery</b> is Amazing...");
```

Artinya, cari sebuah tag HTML dengan **id="paragraf1"**, lalu ubah isi tag tersebut dengan string: **"#jQuery is Amazing..."**.

Begitu juga dengan perintah kedua:

```
$("#paragraf2").html("<b>#jQuery</b> is Amazing...");
```

Artinya, cari sebuah tag HTML dengan **id="paragraf2"**, lalu ubah isi tag tersebut dengan string: **"#jQuery is Amazing..."**.

Seperti yang terlihat, method **html()** akan memproses tag HTML yang ada di dalam teks. Dalam contoh diatas, teks **#jQuery** akan menjadi tebal karena efek tag ****. Sedangkan untuk method **text()**, teks akan ditampilkan apa adanya, dimana tag **** dianggap sebagai teks biasa, bukan sebuah tag HTML.

Cara Mengubah Nilai Form HTML dengan Method val()

Untuk form HTML, kita juga bisa mengubah nilai yang ada, atau tepatnya mengubah nilai atribut **value** dari objek form tersebut. Cara penggunaannya hampir sama dengan method **text()** dan **html()**, tapi kali ini kita menggunakan method **val()**. Berikut contohnya:

```
<!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() {

        $("#tombol").click(function() {
            $("#nama").val("Joko Sulisty");
        })

    });
</script>
</head>
<body>
Nama: <input type="text" id="nama">
<button id="tombol">Change!</button>
</body>
```

```
</html>
```

Ketika tombol **Change!** Di-klik, nilai dari tag `<input type="text">` akan diubah nilainya menjadi **“Joko Sulisty”**.

Dengan menggabungkan cara mengambil dan mengubah nilai elemen HTML, kita bisa membuat sebuah kode program yang secara dinamis akan bertukar tergantung inputan user. Berikut contohnya:

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Belajar jQuery Duniaikom</title>
<script src="jquery-2.1.4.js"></script>
<script>
    $(document).ready(function() {

        $("#tombol").click(function() {
            var nilai = $("#kotak_isian").val();
            $("#target").html(nilai);
        })

    });
</script>
</head>
<body>
Input Text: <input type="text" id="kotak_isian">
<button id="tombol">Salin!</button>
<p id="target">placeholder...</p>
</body>
```

```
</html>
```

Ketika tombol **Salin!** di klik, perintah `var nilai = $("#kotak_isian").val()` digunakan untuk mengambil isi form, dan disimpan ke dalam variabel **nilai**. Variabel **nilai** ini kemudian dipindahkan ke dalam paragraf melalui perintah `$("#target").html(nilai)`.

Silahkan anda ketik teks apa saja ke dalam form, lalu tekan tombol **Salin!**. Isian form tersebut akan tampil ke dalam paragraf dibawahnya.

Cara Penggunaan Method `append()` dan `prepend()` jQuery

Method **`append()`** dan **`prepend()`** bisa digunakan untuk menambah elemen HTML baru ke dalam kode HTML yang sudah ada. Method **`append()`** akan menambahkan elemen baru ke bagian akhir, sedangkan **`prepend()`** akan menambahkan elemen HTML baru ke bagian awal.

Yang perlu dicatat, kedua fungsi ini menambahkan element baru **ke dalam selector yang dicari**. Berikut contohnya:

```
<!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() {

        $("#tombol_app").click(function() {
            $("#box").append("<p>jQuery is Amazing...</p>");
        })
    })
}
```

```
$("#tombol_pre").click(function() {  
    $("#box").prepend("<p>Learning jQuery...</p>");  
})  
  
});  
</script>  
</head>  
<body>  
<div id="box">  
    <h2>Sedang belajar jQuery di Duniailkom...</h2>  
</div>  
<button id="tombol_app">Append</button>  
<button id="tombol_pre">Prepend</button>  
</body>  
</html>
```

Pada kode program diatas, saya memiliki sebuah tag **<div>** dengan **id="box"**. Di dalamnya terdapat sebuah tag **<h2>**.

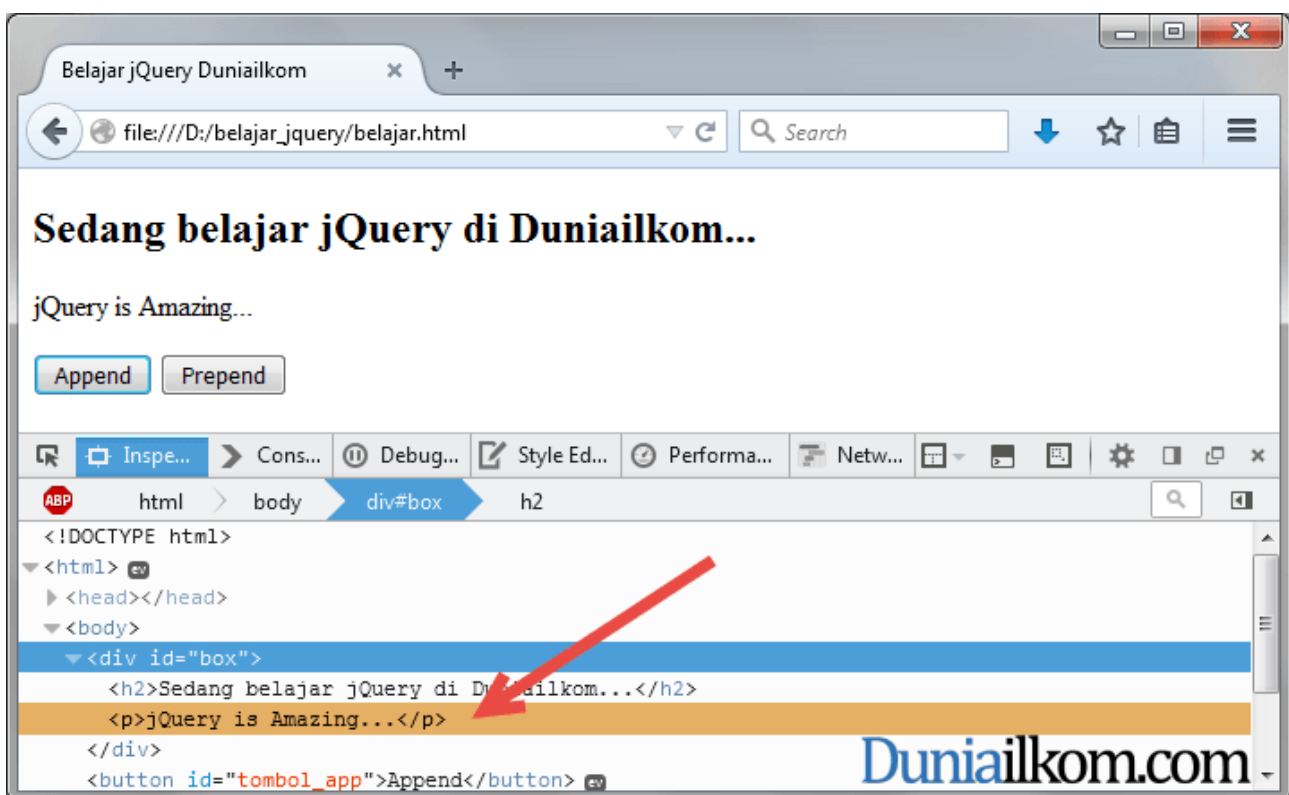
Ketika tombol **"Append"** di klik, akan di jalan perintah berikut:

```
$("#box").append("<p>jQuery is Amazing...</p>");
```

Ini maksudnya, tambahkan kode **"<p>jQuery is Amazing...</p>"** ke bagian **akhir** tag **<div id="box">**. Hasilnya adalah sebagai berikut:

```
<div id="box">  
    <h2>Sedang belajar jQuery di Duniailkom...</h2>  
    <p>jQuery is Amazing...</p>  
</div>
```


Dapat dilihat bahwa, element <p> akan ditambahkan ke bagian **akhir** di dalam tag <div id="box">. Hasil ini akan mudah dilihat jika anda mengaktifkan **Web Developer Tools** dari web browser, seperti gambar berikut:



Gambar 11. Penggunaan Append dan Prepend

Begitu juga ketika tombol “**Prepend**” di klik, perintah:

```
$("#box").prepend("<p>Learning jQuery...</p>");
```

Akan menambah sebuah paragraf baru ke dalam tag `<div id="box">`, pada posisi paling **awal**.

Berikut hasil yang di dapat:

```
<div id="box">
<p>Learning jQuery...</p>
<h2>Sedang belajar jQuery di Duniailkom...</h2>
</div>
```

Cara Penggunaan Method `after()` dan `before()` jQuery

Method **`after()`** dan **`before()`** jQuery sebenarnya sangat mirip dengan **`append()`** dan **`prepend()`**. Bedanya terletak di posisi di mana element baru akan ditambahkan. Method **`after()`** dan **`before()`** akan menambahkan elemen baru **di luar jQuery Selector**.

Agar lebih jelas, mari jalankan kode yang sama dengan sebelumnya, dan hanya merubah method **`append()`** menjadi **`after()`** dan **`prepend()`** menjadi **`before()`**:

```
<!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() {
```

```
$("#tombol_app").click(function() {
    $("#box").after("<p>jQuery is Amazing...</p>");
})

$("#tombol_pre").click(function() {
    $("#box").before("<p>Learning jQuery...</p>");
})

});
</script>
</head>
<body>
<div id="box">
    <h2>Sedang belajar jQuery di Duniailkom...</h2>
</div>
<button id="tombol_app">After</button>
<button id="tombol_pre">Before</button>
</body>
</html>
```

Jika anda menjalankan kode diatas, hasilnya tampak sama persis seperti kode program kita sebelumnya. Tapi method **after()** dan **before()** ini akan menambahkan elemen baru di **sisi luar**.

Ketika tombol “**After**” di klik, perintah berikut akan dijalankan:

```
$("#box").after("<p>jQuery is Amazing...</p>");
```

Hasilnya adalah sebagai berikut:

```
<div id="box">
<h2>Sedang belajar jQuery di Duniailkom...</h2>
</div>
```

```
<p>jQuery is Amazing...</p>
```

Perhatikan beda antara method **append()** dengan **after()**. Perintah **after()** akan menempatkan tag **<p>** disisi luar, yakni setelah tag **<div id="box">**, bukan di dalam tag **<div id="box">** sebagaimana method **append()**.

Begitu juga dengan method **before()**, hasilnya adalah sebagai berikut:

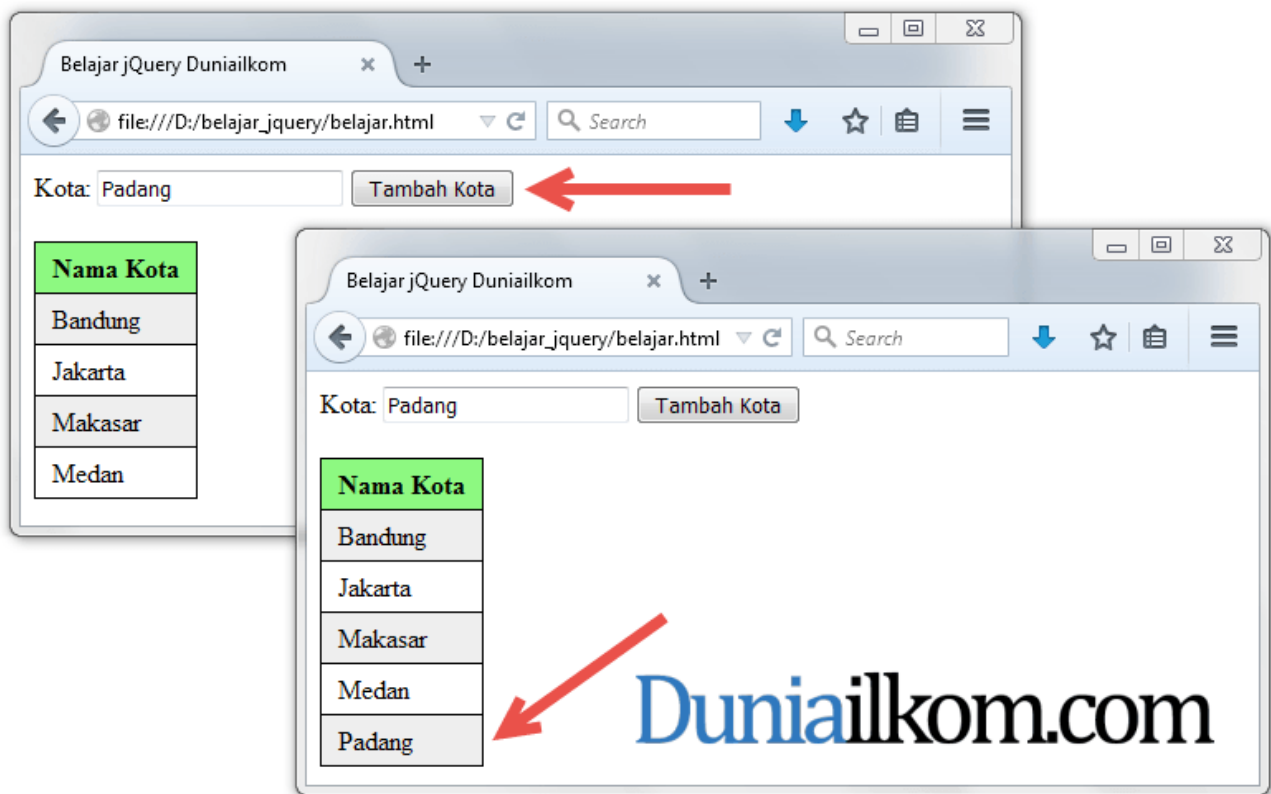
```
<p>Learning jQuery...</p>
<div id="box">
<h2>Sedang belajar jQuery di Duniailkom...</h2>
</div>
```

Method **before()** akan menempatkan tag **<p>** sebelum dan diluar **<div id="box">**.

Cara Menambah Baris Tabel HTML dengan jQuery

Mengambil Nilai HTML dan Menambahkannya di Tempat Lain

Kode program yang akan saya buat yaitu bagaimana cara mengisi tabel dengan element yang digenerate dari jQuery. Berikut tampilan akhir yang saya inginkan:



Gambar 12. Mengambil nilai HTML

Seperti yang terlihat, terdapat sebuah form input dari tag `<input type="text">`, sebuah tombol “**Tambah Kota**”, dan sebuah **Tabel** yang berisi nama-nama kota.

Saya ingin ketika kita mengisi nama kota ke dalam form dan men-klik tombol, nama kota tersebut akan tampil sebagai baris baru di dalam tabel.

Bagaimana cara membuatnya?

Pertama, ketika tombol di klik, **ambil nilai form**. Ini bisa dilakukan menggunakan method **val()** jQuery. Simpan nilai ini ke dalam sebuah variabel. Selanjutnya, nilai ini kita tambahkan ke dalam tabel menggunakan method **append()**. Tentu saja kita juga harus menambahkan tag-tag yang diperlukan seperti `<tr>` dan `<td>` yang merupakan tag HTML untuk membuat sebuah baris tabel.

```
<!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() {

        $("#tambah").click(function() {
            var nilai = $("#nama_kota").val();
            var baris_baru = "<tr><td>"+nilai+"</td></tr>";
            $("#tabel_kota").append(baris_baru);
        })

    });
</script>
<style>
table {
    border-collapse: collapse;
    border-spacing: 0;
    border: 1px black solid;
    margin-top: 20px;
}
th, td {
    padding: 5px 10px;
    border: 1px black solid;
}
th {
    background-color: #87F97B;
}
tr:nth-child(even) {
    background-color: #EDEDED;
```

```
    }
    tr:nth-child(odd) {
        background-color: #FFF;
    }
</style>
</head>
<body>
Kota: <input type="text" id="nama_kota" value="Surabaya">
<button id="tambah">Tambah Kota</button>
<table id="tabel_kota">
    <tr><th>Nama Kota</th></tr>
    <tr><td>Bandung</td></tr>
    <tr><td>Jakarta</td></tr>
    <tr><td>Makasar</td></tr>
    <tr><td>Medan</td></tr>
</table>
</body>
</html>
```

Kode diatas sedikit panjang karena saya menambahkan beberapa baris kode CSS untuk membuat tampilan tabel lebih menarik.

Proses pemindahan nilai form ke dalam tabel sendiri hanya membutuhkan 3 baris:

```
var nilai = $("#nama_kota").val();
var baris_baru = "<tr><td>" + nilai + "</td></tr>";
$("#tabel_kota").append(baris_baru);
•
```

Baris 1: Ambil nilai form, simpan ke variabel **nilai**.

- Baris 2: Tambahkan tag `<tr>` dan `<td>` yang diperlukan untuk membuat baris tabel. Saya menggunakan operator `' + '` yang di dalam JavaScript adalah operator penyambungan string. Simpan semuanya ke dalam variabel **baris_baru**.
- Baris 3: Tambahkan isi variabel **baris_baru** ke dalam tabel menggunakan method **append()**.

Yup, hanya dengan sekitar 5 baris kode program jQuery, kita sudah bisa membuat aplikasi HTML dinamis. Inilah keuntungan menggunakan jQuery daripada JavaScript murni.

Silahkan anda utak-atik kode diatas, misalnya dengan menambahkan tag input kedua atau menambahkan 1 lagi kolom tabel.

Menghapus Elemen HTML dengan method `remove()` jQuery

Kita bisa menggunakan method **remove()** jQuery untuk menghapus element HTML yang sudah tampil di halaman saat ini. Method **remove()** akan menghapus tag saat ini, **beserta seluruh isinya**.

Langsung saja kita lihat contoh cara menghapus element HTML dengan method **remove()** jQuery:

```
<!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() {

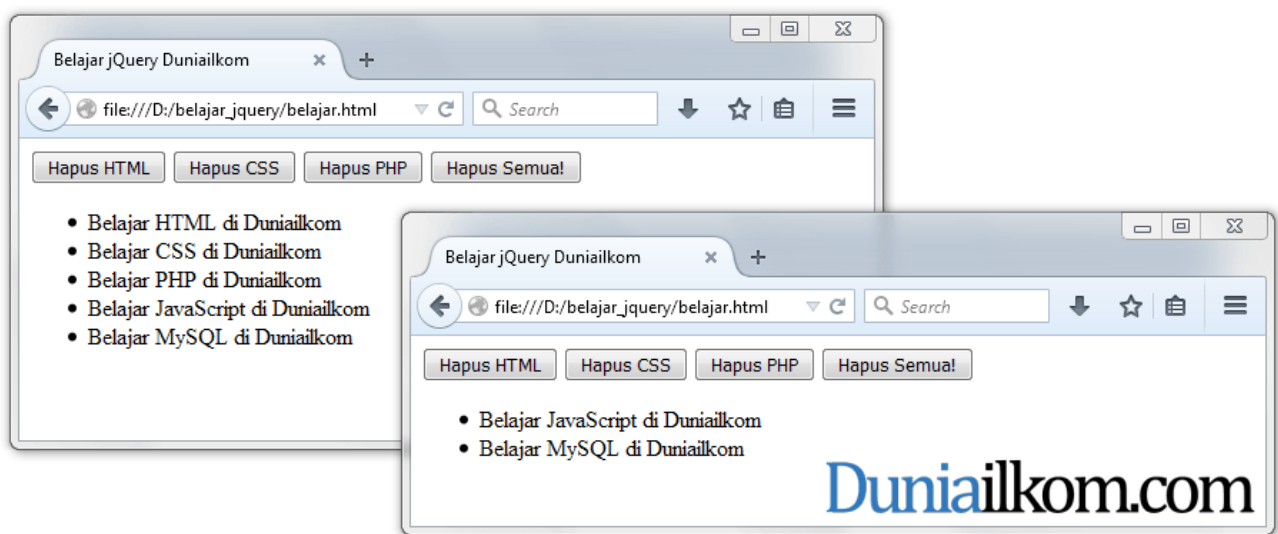
        $("#hapus_html").click(function() {
            $("#list_html").remove();
        })

        $("#hapus_css").click(function() {
            $("#list_css").remove();
        })

        $("#hapus_php").click(function() {
            $("#list_php").remove();
        })

        $("#hapus_all").click(function() {
            $("#belajar").remove();
        })
    });
</script>
</head>
<body>
<button id="hapus_html">Hapus HTML</button>
<button id="hapus_css">Hapus CSS</button>
<button id="hapus_php">Hapus PHP</button>
<button id="hapus_all">Hapus Semua!</button>
<ul id="belajar">
    <li id="list_html">Belajar HTML di Duniailkom</li>
    <li id="list_css">Belajar CSS di Duniailkom</li>
    <li id="list_php">Belajar PHP di Duniailkom</li>
```

```
<li id="list_js">Belajar JavaScript di Duniaikom</li>
<li id="list_mysql">Belajar MySQL di Duniaikom</li>
</ul>
</body>
</html>
```



Gambar 13. Menghapus Elemen HTML

Disini saya memiliki 4 tombol dan sebuah list dengan 5 tag ``. Baik tag `` maupun tag `` dari list memiliki atribut `id` masing-masing.

Ketika tombol **Hapus HTML** di klik, perintah: `$("#list_html").remove()` akan dijalankan. Akibatnya, list `<li id="list_html">Belajar HTML di Duniaikom` akan terhapus. List yang tersisa akan mengisi element yang sudah hilang ini (naik ke atas). Demikian juga yang terjadi ketika tombol **Hapus CSS** dan **Hapus PHP** di klik.

Bagaimana dengan tombol **Hapus Semua!**? Tombol ini akan menjalankan perintah `$("#belajar").remove()`. Elemen HTML yang memiliki `id="belajar"` adalah tag ``. Element ini merupakan *'parent element'* dari seluruh tag ``. Dengan demikian, semua list akan ikut terhapus.

Menghapus Elemen HTML dengan Method `empty()` jQuery

Selain menggunakan method **`remove()`**, jQuery juga menyediakan method **`empty()`** untuk menghapus sebuah element HTML. Apa perbedaannya?

Method **`empty()`** hanya menghapus isi dari sebuah element, sedangkan element itu sendiri **tidak ikut dihapus**. Hal ini berbeda dengan method **`remove()`** yang menghapus isi element beserta element itu sendiri.

Untuk melihat contoh penggunaan method **`empty()`**, perhatikan kode program dibawah ini:

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Belajar jQuery Duniaikom</title>
<script src="jquery-2.1.4.js"></script>
<script>
    $(document).ready(function() {

        $("#tambah").click(function() {
            $("#belajar").append("<li>Belajar jQuery di
Duniaikom</li>");
        })

        $("#kosong").click(function() {
            $("#belajar").empty();
        })

        $("#hapus").click(function() {
            $("#belajar").remove();
        })
    })
}
```

```
});  
</script>  
</head>  
<body>  
<button id="tambah">Tambah jQuery</button>  
<button id="kosong">Kosongkan</button>  
<button id="hapus">Hapus Semua</button>  
<ul id="belajar">  
  <li id="list_html">Belajar HTML di Duniaikom</li>  
  <li id="list_css">Belajar CSS di Duniaikom</li>  
  <li id="list_php">Belajar PHP di Duniaikom</li>  
  <li id="list_js">Belajar JavaScript di Duniaikom</li>  
  <li id="list_mysql">Belajar MySQL di Duniaikom</li>  
</ul>  
</body>  
</html>
```

Kali ini saya memodifikasi sedikit kode program sebelumnya. Sekarang terdapat 3 tombol.

Tombol pertama: **Tambah jQuery** akan menjalankan perintah:

```
$("#belajar").append("<li>Belajar jQuery di Duniaikom</li>");
```

Artinya, ketika tombol ini di-klik, akan menambah sebuah elemen baru di bagian bawah list.

Method **append()** sudah kita pelajari di dalam Tutorial Belajar jQuery: Cara Menambah Elemen HTML baru dengan jQuery.

Tombol kedua: **Kosongkan** akan menjalankan perintah:

```
$("#belajar").empty();
```

Artinya, seluruh isi list akan dihapus. Tetapi tag **** tersebut sebenarnya masih ada. Untuk membuktikannya, silahkan anda klik kembali tombol **Tambah jQuery**, list baru akan kembali muncul.

Akan tetapi, ketika tombol **Hapus Semua** di klik, perintah yang dijalankan adalah:

```
$("#belajar").remove();
```

Perintah ini akan menghapus seluruh list, beserta tag `` nya. Dengan demikian, tag `` tidak bisa lagi diakses. Ketika kita coba menambahkan element baru dengan men-klik tombol **Tambah jQuery**, tidak akan terjadi apa-apa. Ini karena tag `` tempat dimana list berada sudah hilang.

Cara Mengambil Atribut HTML dengan jQuery

Untuk “berurusan” dengan atribut elemen HTML, jQuery menyediakan method **attr()**. Method ini bisa diisi dengan 1 atau 2 argumen. Jika diisi dengan 1 argumen, berarti kita ingin mengambil atribut dari element tersebut.

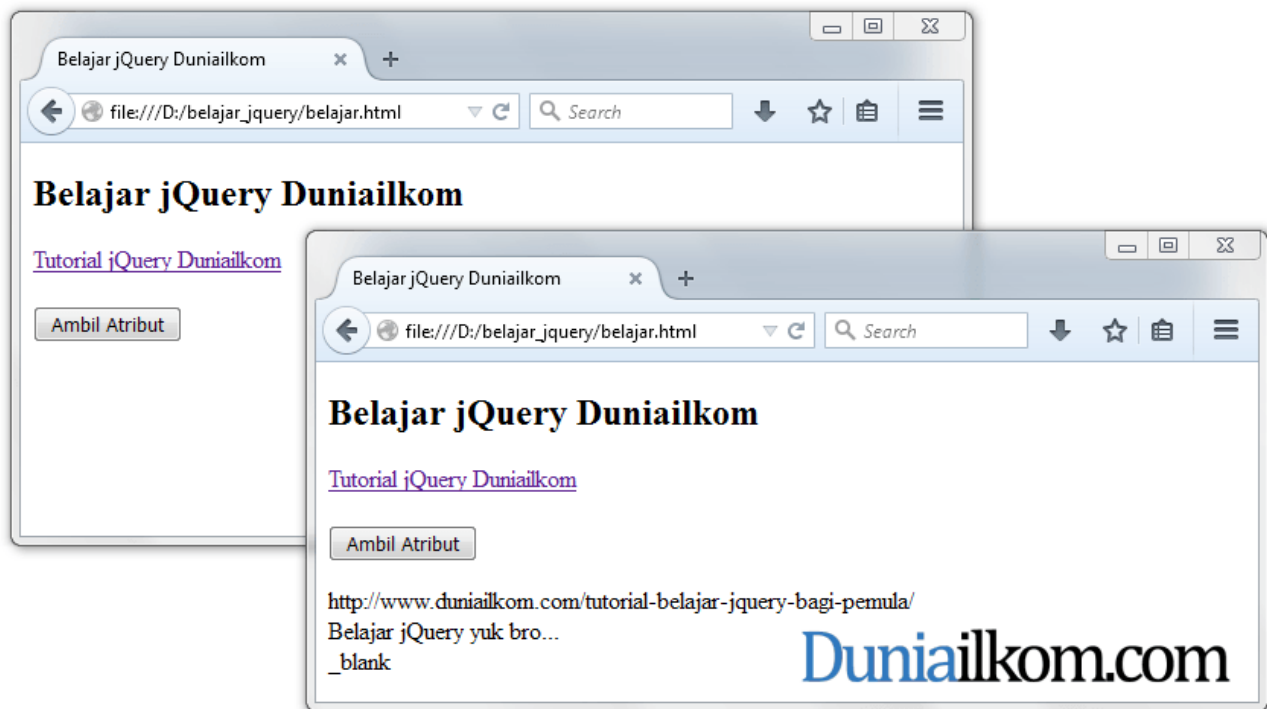
Sebagai contoh, untuk mengambil nilai atribut **title** dari sebuah tag HTML yang memiliki **id="paragraf"**, saya bisa menulis kode jQuery berikut:

```
var nilai = $("# paragraf ").attr("title");  
Sekarang, variabel nilai akan berisi nilai atribut title.
```

```
<!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() {  
  
        $("#ambil").click(function() {  
            var nilai1 = $("a[title]").attr("href");  
            var nilai2 = $("a[title]").attr("title");  
            var nilai3 = $("a[title]").attr("target");
```

```
        $("#tekape").html(nilai1+"<br>" + nilai2 + "<br>" + nilai3);
    })

});
</script>
</head>
<body>
<h2>Belajar jQuery Duniaikom</h2>
<a href="https://www.duniaikom.com/tutorial-belajar-jquery-bagi-
pemula/"
    title="Belajar jQuery yuk bro..."
    target="_blank">
    Tutorial jQuery Duniaikom
</a>
<br><br>
<button id="ambil">Ambil Atribut</button>
<p id="tekape"></p>
</body>
</html>
```



Gambar 14. Cara Mengambil Atribut HTML

Dalam halaman ini saya memiliki sebuah link yang dibuat dari tag `<a>`. Perhatikan link ini memiliki 3 atribut: **href**, **title**, dan **target**. Dibawahnya terdapat sebuah tombol **Ambil Atribut** yang dibuat dengan tag `<button id="ambil">`. Element terakhir adalah sebuah paragraf kosong yang memiliki **id="tekape"**.

Disini saya ingin mengambil ketiga atribut dari tag `<a>`, lalu menampilkan hasilnya ke `<p id="tekape">`. Ini saya lakukan dengan 4 baris berikut:

```
var nilai1 = $("a[title]").attr("href");
var nilai2 = $("a[title]").attr("title");
var nilai3 = $("a[title]").attr("target");
$("#tekape").html(nilai1+"<br>" + nilai2 + "<br>" + nilai3);
```

Perhatikan bagaimana cara mengambil nilai setiap atribut. Variabel **nilai1**, **nilai2** dan **nilai3** akan berisi nilai atribut **href**, **title** dan **target** dari tag `<a>`.

Untuk *jQuery selector* tag `<a>`, saya mencarinya dengan menggunakan **“a[title]”**. Di dalam CSS, selector seperti ini dikenal sebagai **atribut selector**. Artinya, saya mencari sebuah tag `<a>` yang memiliki atribut **title**. Karena di halaman saat ini hanya terdapat 1 tag `<a>`, tag inilah yang akan diambil. Ini hanya sekedar alternatif dari atribut **id** yang selalu kita gunakan hingga saat ini. Terakhir saya menggunakan method **html()** untuk memindahkan ketiga nilai atribut kedalam tag `<p id="tekape"></p>`.

Cara Mengubah dan Menambahkan Atribut HTML dengan jQuery

Baik, kita sudah mempelajari cara mengambil nilai element HTML. Sekarang, bagaimana cara mengubahnya?

Kita masih tetap menggunakan method **attr()**. Tapi kali ini dengan menulis 2 argumen. Argumen pertama berfungsi sebagai **nama atribut**, sedangkan argumen kedua adalah **nilai dari atribut** tersebut.

Sebagai contoh, untuk mengubah atau menambahkan atribut **title= “Sedang belajar jQuery”** ke dalam elemen HTML dengan **id=“paragraf”**, saya bisa menulis kode berikut:

```
$("#paragraf").attr("title", " Sedang belajar jQuery");
```

Mari kita lihat contoh prakteknya:

```
<!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() {
```



```
$("#tambah").click(function() {
    $("#link_jquery").attr("href",
"https://www.duniailkom.com/tutorial-belajar-jquery-bagi-
pemula/");
    $("#link_jquery").attr("title", "Belajar jQuery yuk bro...");
})

});
</script>
</head>
<body>
<h2>Belajar jQuery Duniailkom</h2>
<p>
    <a id="link_jquery">Tutorial jQuery Duniailkom</a>
</p>
<button id="tambah">Tambah Atribut</button>
</body>
</html>
```

Pada saat halaman HTML pertama kali dijalankan, tag **<a>** tidak memiliki atribut apa-apa selain **id="link_jquery"**. Tag **<a>** ini belum berupa link, karena tidak memiliki atribut **href**.

Ketika tombol **Tambah Atribut** di klik, saya akan menambah 2 buah atribut baru dengan perintah berikut:

```
$("#link_jquery").attr("href",
"https://www.duniailkom.com/tutorial-belajar-jquery-bagi-
pemula/");
$("#link_jquery").attr("title", "Belajar jQuery yuk bro...");
```

Sekarang, tag `` telah mendapat atribut **href** dan **title**. Artinya, tag `<a>` tersebut berubah menjadi sebuah link ke duniaiikom. Anda bisa melihat efeknya dengan perubahan warna font menjadi biru dan memiliki garis bawah, persis seperti link normal.

Pengertian Variabel `$(this)` dalam jQuery

Di dalam bahasa pemrograman objek (seperti JavaScript), variabel **this** berfungsi sebagai referensi untuk objek saat ini. Sedangkan di dalam jQuery, selector `$(this)` berfungsi sebagai referensi untuk element HTML saat ini.

Sebagai contoh, misalkan saya memiliki sebuah box HTML dengan kode berikut:

```
<div id="box">Click Me!</div>
```

Saya ingin menambahkan event **click** pada tag tersebut. Di dalam jQuery, saya bisa membuatnya dengan kode program:

```
$("#box").click(function() {  
// kode program disini...  
// kode program disini...  
// kode program disini...  
})
```

Kemudian bagaimana jika saya ingin agar warna background box tersebut berubah menjadi abu-abu (*silver*) saat di-klik? Saya bisa menambahkan kode berikut ke dalam event **click** :

```
$("#box").click(function() {  
$("#box").css("background-color", "silver");  
})
```

Tidak ada hal yang baru disini, saya menggunakan method `css()` untuk menambahkan property CSS **background-color: silver.**

Kode diatas bisa juga bisa ditulis menjadi:

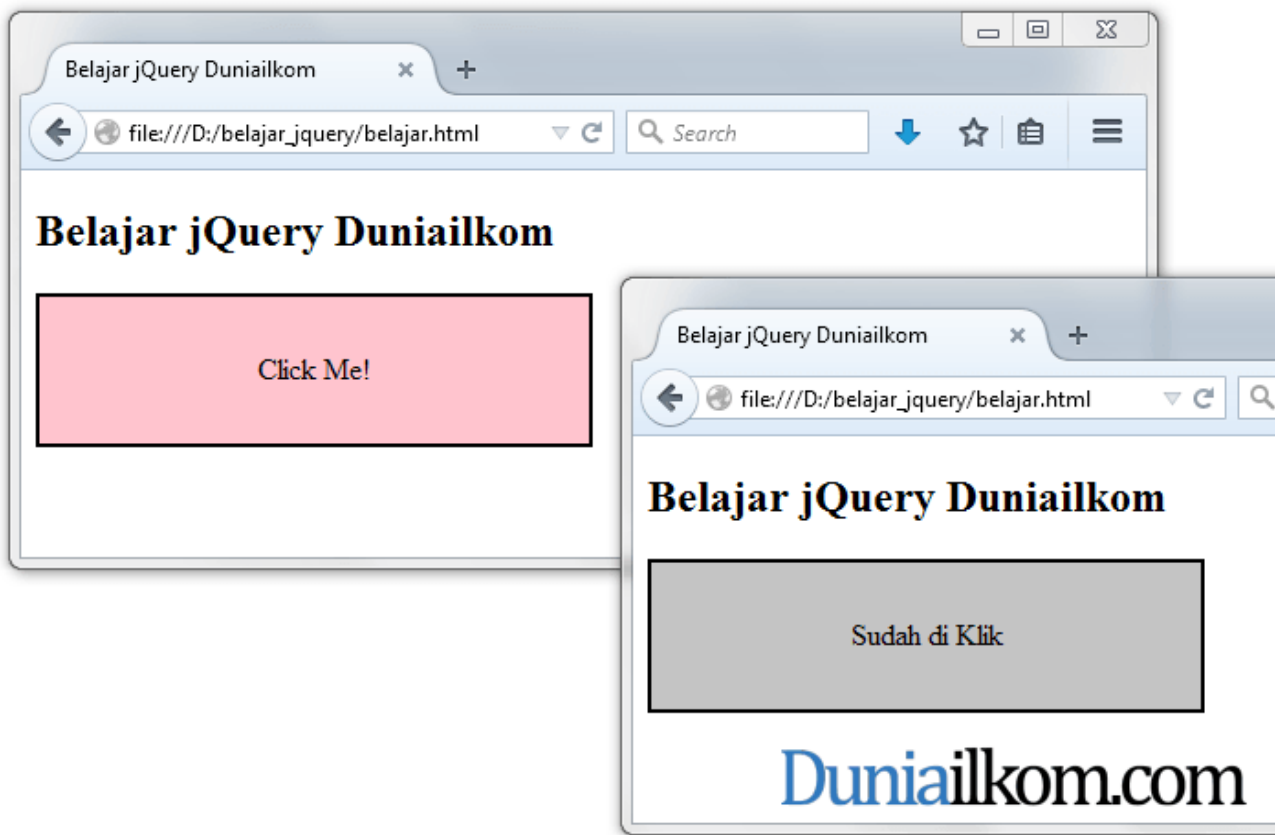
```
$("#box").click(function() {  
$(this).css("background-color", "silver");  
})
```

Perhatikan bagaimana saya mengubah selector jQuery `$("#box").css()` menjadi `$(this).css()`. Ini berarti saya ingin mengubah warna background-color untuk elemen HTML **yang saat ini sedang aktif**. Karena kita berada di dalam event click `$("#box")`, artinya `$(this)` sama dengan `$("#box")`.

Berikut contoh lengkap dari penjelasan ini:

```
<!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() {  
  
$("#box").click(function() {  
$(this).html("Sudah di Klik");  
$(this).css("background-color", "silver");  
})  
})
```

```
});  
</script>  
<style>  
#box {  
width: 300px;  
height: 80px;  
background-color: pink;  
border: 2px solid black;  
line-height: 80px;  
text-align: center;  
}  
</style>  
</head>  
<body>  
<h2>Belajar jQuery Duniailkom</h2>  
<div id="box">Click Me!</div>  
</body>  
</html>
```



Gambar 15. Pengertian Variable \$(this)

Kalau hasilnya sama seperti selector biasa, untuk apa repot-repot pakai \$this?

Dengan menggunakan selector **\$(this)**, kita bisa membuat kode program yang lebih efisien untuk berbagai selector sekaligus. Mari masuk ke contoh praktek selanjutnya:

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Belajar jQuery Duniaikom</title>
<script src="jquery-2.1.4.js"></script>
<script>
    $(document).ready(function() {
```

```
$("#box1").click(function() {  
    var lebar_box = $("#box1").width();  
    var tinggi_box = $("#box1").height();  
    $("#box1").width(lebar_box + 10);  
    $("#box1").height(tinggi_box + 10);  
})  
  
$("#box2").click(function() {  
    var lebar_box = $("#box2").width();  
    var tinggi_box = $("#box2").height();  
    $("#box2").width(lebar_box + 10);  
    $("#box2").height(tinggi_box + 10);  
})  
  
$("#box3").click(function() {  
    var lebar_box = $("#box3").width();  
    var tinggi_box = $("#box3").height();  
    $("#box3").width(lebar_box + 10);  
    $("#box3").height(tinggi_box + 10);  
})  
  
});  
</script>  
<style>  
div {  
    width: 150px;  
    height: 30px;  
    background-color: pink;  
    border: 2px solid black;
```

```
        margin-bottom: 10px;
        cursor: pointer;
    }
</style>
</head>
<body>
<h2>Belajar jQuery Duniailkom</h2>
<div id="box1"></div>
<div id="box2"></div>
<div id="box3"></div>
</body>
</html>
```

Kali ini saya memiliki 3 buah box dari tag **<div>**. Untuk setiap box, ketika di klik, lebar dan tingginya bertambah 10 pixel. Untuk membuat efek ini, saya menggunakan method **width()** dan **height()** yang telah dipelajari dari tutorial sebelumnya.

Disini saya bisa membuat 3 buah event untuk masing-masing box, yakni **\$("#box1").click()**, **\$("#box2").click()** dan **\$("#box3").click()**. Di dalamnya, terdapat kode jQuery untuk memperbesar ukuran box.

Kode diatas berjalan seperti rencana dan tidak salah. Namun akan jauh lebih efisien jika saya bisa membuat 1 event untuk ketiga box. Berikut perubahannya:

```
<!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() {  
    $("div").click(function() {  
        var lebar_box = $(this).width();  
        var tinggi_box = $(this).height();  
        $(this).width(lebar_box + 10);  
        $(this).height(tinggi_box + 10);  
    })  
  
    });  
</script>  
<style>  
div {  
    width: 150px;  
    height: 30px;  
    background-color: pink;  
    border: 2px solid black;  
    margin-bottom: 10px;  
    cursor: pointer;  
}  
</style>  
</head>  
<body>  
    <h2>Belajar jQuery Duniaikom</h2>  
    <div id="box1"></div>  
    <div id="box2"></div>  
    <div id="box3"></div>  
</body>  
</html>
```


Bisakah anda membedakannya dengan kode program saya sebelumnya? Daripada membuat 3 event untuk setiap box, saya bisa membuat 1 event untuk seluruh box. Variabel **\$(this)** di dalam event ini akan mengurus semuanya. Sebagai nilai bonus, jika saya menambah 1 lagi tag `<div>`, box tersebut juga langsung memiliki event click yang sama.

Jika anda belum begitu paham dengan variabel **\$(this)** ini silahkan mempelajari sejenak maksud kode diatas, lalu buat kode program lain sebagai sarana latihan.

DAFTAR PUSTAKA :

<https://www.duniailkom.com/tutorial-belajar-jquery-bagi-pemula/>