



ASP.NET & jQuery



**Mudah menggunakan jQuery dengan
Visual Web Developer 2008 & Glimmer
M. Reza Faisal**



Daftar Isi

<i>Daftar Isi</i>	<i>I</i>
<i>1 . Pendahuluan.....</i>	<i>1-1</i>
<i>JavaScript.....</i>	<i>1-1</i>
Apa yang dapat JavaScript lakukan ?	1-1
Tata Cara menulis JavaScript	1-2
Memisahkan HTML, CSS dan JavaScript	1-5
<i>jQuery.....</i>	<i>1-7</i>
Memulai jQuery	1-7
Cara Kerja jQuery.....	1-10
Tanda dollar pada jQuery.....	1-12
Kenapa memilih jQuery ?	1-13
Kemampuan jQuery	1-13
Siapa pengguna jQuery ?	1-14
<i>Visual Web Developer 2008, jQuery & ASP.NET.....</i>	<i>1-14</i>
JavaScript Intellisense.....	1-14
jQuery Intellisense	1-17
Kerjasama jQuery & ASP.NET	1-18
HTML & ASP.NET.....	1-19
<i>2 . Bekerja dengan jQuery</i>	<i>2-1</i>
<i>jQuery Core</i>	<i>2-1</i>
Fungsi \$ (...)	2-1
jQuery Object Accessors.....	2-6
Plugin.....	2-13
Interoperability.....	2-15
<i>Selector.....</i>	<i>2-16</i>
Basic	2-16
Hierarchy	2-18
Basic Filter.....	2-22
Content Filter.....	2-28
Visibility Filter.....	2-32
Attribute Filter.....	2-33

Child Filter	2-36
Form.....	2-39
Form Filter.....	2-41
Attribute.....	2-43
Attr	2-43
Class.....	2-45
HTML	2-46
Text.....	2-48
Value	2-49
Traversing.....	2-49
Filtering	2-49
Finding	2-59
Chaining	2-69
Manipulation	2-71
Changing Content.....	2-71
Inserting Inside.....	2-71
Inserting Outside.....	2-74
Inserting Around.....	2-76
Replacing.....	2-81
Removing.....	2-82
Copying	2-83
Mengelola CSS	2-84
CSS	2-84
Positioning	2-86
Height & Width.....	2-87
Event	2-88
Page Load.....	2-88
Event Handling	2-88
Interaction Helper	2-92
Event Helper.....	2-94
Effect.....	2-100
Basic	2-100
Sliding.....	2-101
Fading.....	2-102
Custom	2-102

3 . <i>jQuery UI & Plug-in</i>	3-105
jQuery UI	3-105
Interaksi.....	3-105
Widget	3-133
Effect	3-136
jQuery Plug-in	3-136
4 . <i>AJAX dengan jQuery</i>	4-137
Test 1	4-137
5 . <i>jQuery & ASP.NET</i>	5-138
6 . <i>Glimmer</i>	6-139
Mengenal Glimmer.	6-139
Wizard	6-139
Image Sequencer	6-139
Dropdown Menu	6-139
Custom	6-139
Tooltip Wizard	6-139
Plug-in	6-139
7 . <i>Membangun Aplikasi Web dengan ASP.NET & jQuery</i>	7-140

1.

Pendahuluan

JavaScript

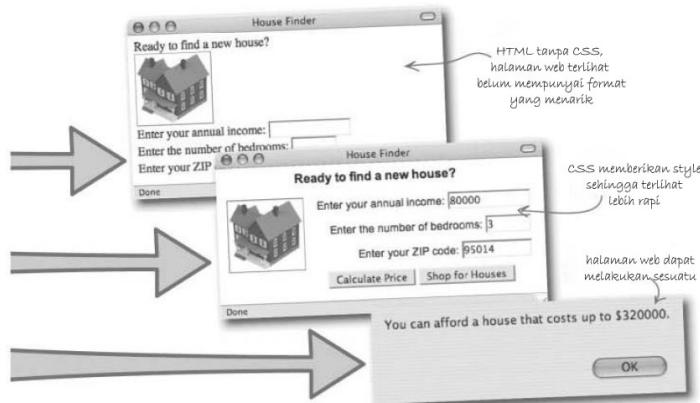
JavaScript, web developer mana yang tidak pernah mendengar kata JavaScript. JavaScript merupakan bahasa pemrograman web yang berjalan pada sisi client. Artinya JavaScript sangat tergantung dengan web browser sebagai aplikasi client untuk melihat sebuah halaman web. JavaScript tidak akan berfungsi jika web browser yang digunakan tidak mendukungnya. Kebergantungan JavaScript pada web browser akan membuat beberapa perbedaan cara penulisan kode program JavaScript agar dapat berjalan dengan benar pada masing-masing web browser.

Apa yang dapat JavaScript lakukan ?

Kelebihan JavaScript yang dapat berjalan di sisi client dapat membuat halaman web menjadi lebih interaktif dan responsif. JavaScript membuat halaman web tidak seperti tembok yang hanya diam ketika diajak bicara. JavaScript membuat halaman web dapat berbicara, merespon yang user lakukan pada halaman web tersebut.

Untuk membuat halaman web yang interaktif, terdapat tiga pengetahuan yang mesti diketahui, yaitu :

- HyperText Markup Language (HTML) adalah struktur halaman web.
- Cascading Style Sheets (CSS) memberikan style pada halaman web.
- JavaScript akan melengkapi sebagai pemberi reaksi ketika user melakukan aksi.



Gambar 1-1 : HTML, CSS & JavaScript

Ilustrasi di atas dapat dipaparkan sebagai berikut.

HTML mempunyai elemen-elemen seperti image, textbox, button, table, text dan lain-lain. Pada window pertama pada gambar di atas, dapat dilihat image, text, text, textbox dan button ditampilkan pada halaman web. Elemen-elemen tersebut tersusun kurang rapi. Susunan dan style elemen-elemen tersebut dapat diubah dengan menggunakan CSS. CSS dapat mengubah style text seperti ukuran font, warna dan ketebalan, hal lain yang dapat diubah

adalah posisi text, textbox dan button. Dapat dilihat pada window kedua pada gambar di atas, implementasi CSS membuat elemen-elemen HTML pada halaman web tersusun lebih rapi. Selanjutnya akan dibuat aplikasi sederhana untuk menghitung harga rumah berdasarkan jumlah kamar yang dimasukkan pada form tersebut. Perhitungan akan dilakukan ketika tombol “Calculate Price” ditekan. Bagaimana JavaScript dapat digunakan untuk membuat aplikasi ini? Ketika tombol diklik maka akan dipanggil sebuah fungsi JavaScript. Pada fungsi tersebut, hal pertama yang dilakukan adalah membaca nilai-nilai pada textbox yang ada pada halaman web kemudian menampilkan hasil perhitungan dengan memanfaatkan fungsi alert milik JavaScript. Fungsi alert akan menampilkan message box seperti yang dilihat pada window ketiga pada gambar di atas.

Dari paparan di atas dapat disimpulkan, JavaScript dapat memanipulasi elemen-elemen HTML, dalam kasus ini adalah memberikan kemampuan pada tombol agar bereaksi ketika diklik, mengambil nilai pada textbox dan menampilkan message box. Selain kemampuan tersebut, JavaScript juga dapat memanipulasi style pada elemen-elemen HTML. Sehingga JavaScript dapat mengubah warna, posisi elemen dan atribut elemen HTML lainnya.

Dengan kemampuan tersebut, maka JavaScript dapat dimanfaatkan untuk membuat hal-hal berikut :

- JavaScript membuat validasi form pada halaman web dapat dilakukan dengan cepat.
- JavaScript dapat membuat animasi yang menarik pada halaman web.
- JavaScript dapat membuat suatu area pada halaman web dipindah-pindahkan dengan cara drag-n-drop.
- JavaScript dapat berkomunikasi dengan server secara asynchronous, metode ini dikenal dengan AJAX.
- Dan lain-lain.

Tata Cara menulis JavaScript

Baris-baris kode JavaScript tidak dapat begitu saja ditulis pada halaman web. Ada aturan yang harus diikuti agar baris-baris kode JavaScript dapat berjalan dan digunakan pada halaman web. Setiap baris kode JavaScript harus berada di antar tag `<script>` dan `</script>`, tag ini disebut *script block*.

Script block ini dapat diletakkan diantara tag HTML berikut :

- Di antara tag `<head>` dan `</head>`.
- Di antara tag `<body>` dan `</body>`.

Script block tidak dapat diletakkan sebelum tag `<html>` dan setelah tag `</html>`.

Seperti tag-tag HTML lainnya, tag `<script>` juga mempunyai atribut, atribut penting yang digunakan untuk memberikan identitas bahwa baris-baris kode yang ditulis pada *script block* adalah baris kode JavaScript adalah sebagai berikut ini :

```
<script type="text/JavaScript"> . . . </script>
```

Berikut ini contoh penulisan program singkat dengan JavaScript.

```
<html>
<body>
<script type="text/JavaScript">
    alert('Hello World');
</script>
</body>
</html>
```

Hasil program di atas akan menampilkan message box seperti pada gambar berikut ini.



Gambar 1-2 : Message Box

Contoh berikutnya adalah aplikasi “House Finder” yang gambarnya telah dapat dilihat pada Gambar 1-1.

```
<html>
<head>
<title>House Finder</title>
<style type="text/css">
body
{
    font-family:Tahoma;
    font-size:0.8em;
}

h1
{
    font-size:1.1em;
    margin:0 0 13px 0;
    text-transform:uppercase;
}

.label
{
    float:left;
    width:200px;
    padding:3px 3px 0 0;
}

.input
{
    float:left;
    width:200px;
    padding:3px 3px 0 0;
}

.separator
{
    clear:both;
}
</style>

<script type="text/JavaScript">
function calculate() {
    var incomeTxt = document.getElementById('income').value;
    var roomTxt = document.getElementById('room').value;

    if (incomeTxt != "" && roomTxt != "") {
        incomeYear = 12 * incomeTxt;
        housePrice = roomTxt * 5000;
    }
}
</script>
```

```

        if (incomeYear > housePrice) {
            alert('Congrats, for your new house.');
        }
        else {
            alert('Please decrease your room.');
        }
    }
    else {
        alert('Please complete your data');
    }
}
</script>
</head>
<body>
    <h1>Ready to find a new house ?</h1>

    <div class="label">Enter your annual income :</div>
    <div class="input"><input type="text" id="income" /></div>

    <div class="separator"></div>

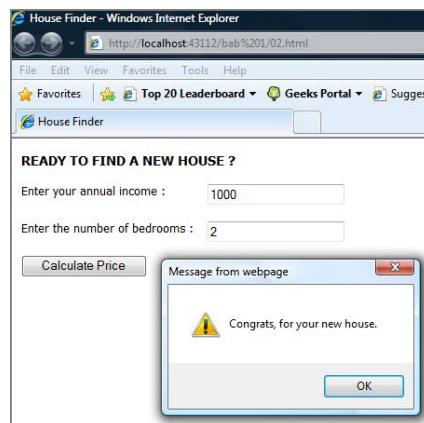
    <div class="label">Enter the number of bedrooms :</div>
    <div class="input"><input type="text" id="room" /></div>

    <div class="separator"></div>

    <div>
        <input type="button" id="btn" value="Calculate Price" onclick="calculate()" />
    </div>
</body>
</html>

```

Dari baris-baris kode di atas akan didapatkan antarmuka seperti berikut ini.



Gambar 1.3 : Aplikasi JavaScript Sederhana

Aplikasi JavaScript sederhana ini terdiri atas baris-baris kode HTML, CSS dan JavaScript. Baris-baris style dari CSS dapat dilihat di antara tag `<style type="text/css">` dan `</style>`. CSS disini digunakan untuk mengatur tata letak elemen-elemen HTML agar terlihat rapi.

Pada halaman web di atas terdapat dua elemen HTML penting, yaitu :

- Button (`<input type="button"/>`), pada button ini terdapat event `onclick` yang memanggil fungsi `calculate`.
- Textbox (`<input type="text"/>`), pada baris kode di atas terdapat dua textbox dengan identitas (`id`) `income` dan `room`.

Setelah user memasukkan nilai pada kedua textbox tersebut dan button Calculate Price diklik maka fungsi `calculate()` akan dijalankan. Pada fungsi ini akan dilihat baris berikut.

```

var incomeTxt = document.getElementById('income').value;
var roomTxt = document.getElementById('room').value;

```

Dan berikut baris pada elemen HTML :

```
...
<div class="input"><input type="text" id="income" /></div>
...
<div class="input"><input type="text" id="room" /></div>
...
```

Dari baris-baris kode di atas, dapat dilihat cara untuk mengambil nilai pada textbox income dan room. Baris `document.getElementById` berfungsi untuk mengambil object elemen HTML, sedangkan isi dari parameter fungsi ini adalah identitas (`id`) dari element HTML yang diinginkan. Bila objek sudah ditentukan, maka bila ingin mengambil nilai dari salah satu textbox tersebut dapat dipanggil property `value`, seperti pada baris kode di atas.

Kemudian nilai yang didapat pada masing-masing textbox akan diolah seperti baris berikut ini :

```
if (incomeTxt != "" && roomTxt != "") {
    incomeYear = 12 * incomeTxt;
    housePrice = roomTxt * 5000;
    ...
}
```

Perhitungan di atas adalah hitungan sederhana untuk menghitung gaji selama satu tahun dan harga rumah berdasarkan jumlah kamarnya.

Dan yang terakhir adalah menentukan apakah income yang dimasukkan cukup untuk mendapatkan rumah dengan jumlah kamar yang diinginkan, berikut baris programnya :

```
if (incomeTxt != "" && roomTxt != "") {
    ...
    if (incomeYear > housePrice) {
        alert('Congrats, for your new house.');
    } else {
        alert('Please decrease your room.');
    }
    ...
}
```

Pada baris kode di atas terdapat fungsi `alert`, fungsi ini bertujuan untuk menampilkan message box dengan pesan sesuai yang kita tuliskan.

Memisahkan HTML, CSS dan JavaScript

Pada contoh di atas CSS dan JavaScript berada dalam satu file HTML. Seperti yang telah banyak diketahui style dapat disimpan dalam satu file tersendiri, begitu juga baris-baris program JavaScript.

Untuk memisahkan style pada satu file tersendiri digunakan sintaks seperti berikut ini :

```
<link rel="stylesheet" type="text/css" href="style.css" />
```

Nilai pada atribut `href` adalah nama file yang menyimpan style seperti berikut ini. Pada file `style.css` tidak diperlukan lagi tag `<style>` dan `</style>`.

```
body
{
    font-family: Tahoma;
    font-size: 0.8em;
}

h1
{
```

```

        font-size: 1.1em;
        margin: 0 0 13px 0;
        text-transform: uppercase;
    }

    .label
    {
        float: left;
        width: 200px;
        padding: 3px 3px 0 0;
    }

    .input
    {
        float: left;
        width: 200px;
        padding: 3px 3px 0 0;
    }

    .separator
    {
        clear: both;
    }

```

Sedangkan untuk memisahkan JavaScript pada file tersendiri digunakan sintaks seperti berikut ini :

```
<script type="text/JavaScript" src="script/script.js"></script>
```

Atribut `scr` pada tag di atas berisi nilai nama file beserta lokasi file tersebut berada. Penulisan baris-baris JavaScript pada file `script.js` tidak memerlukan tagl `<script>` dan `</script>` lagi, sehingga isi dari file `script.js` dapat ditulis seperti berikut ini.

```

function calculate() {
    var incomeTxt = document.getElementById('income').value;
    var roomTxt = document.getElementById('room').value;

    if (incomeTxt != "" && roomTxt != "") {
        incomeYear = 12 * incomeTxt;
        housePrice = roomTxt * 5000;

        if (incomeYear > housePrice) {
            alert('Congrats, for your new house.');
        }
        else {
            alert('Please decrease your room.');
        }
    }
    else {
        alert('Please complete your data');
    }
}

```

Setelah memisahkan style dan JavaScript pada file tersendiri maka halaman web akan hanya terlihat seperti berikut ini.

```

<html>
<head>
<title>House Finder</title>
<link rel="stylesheet" type="text/css" href="style.css" />
<script type="text/JavaScript" src="script.js"></script>
</head>
<body>
    <h1>Ready to find a new house ?</h1>

    <div class="label">Enter your annual income :</div>
    <div class="input"><input type="text" id="income" /></div>

    <div class="separator"></div>

    <div class="label">Enter the number of bedrooms :</div>

```

```
<div class="input"><input type="text" id="room" /></div>
<div class="separator"></div>
<div>
<input type="button" id="btn" value="Calculate Price" onclick="calculate()" />
</div>
</body>
</html>
```

Sebagai informasi, penggunaan tag `<link>` dan `<script>` dapat lebih dari satu pada setiap filenya, artinya pada satu halaman web dapat ditambahkan lebih dari satu file CSS dan JavaScript.

jQuery

jQuery adalah JavaScript library atau framework JavaScript. jQuery mempunyai semboyan singkat yaitu “*write less, do more*”. Semboyan tersebut bukan hanya isapan jempol saja, jQuery telah membuktikan bahwa baris-baris program JavaScript yang ditulis dengan jQuery akan lebih singkat bila dibandingkan menulis sendiri baris-baris program JavaScript dari awal.

Pada bagian sebelumnya telah disebutkan ada tiga hal yang penting dalam membangun halaman web yaitu :

- a. HTML sebagai struktur, sebagai *content* yang terdiri atas elemen-elemen HTML seperti text, image, input, table dan lain-lain.
- b. CSS sebagai style yang berfungsi memisahkan karakteristik tampilan dari struktur HTML. Tampilan di setiap elemen HTML yang ada pada halaman web diatur oleh CSS. Tampilan disini meliputi ukuran, warna, posisi dan lain-lain.
- c. JavaScript sebagai aksi, pada bagian ini JavaScript akan diganti dengan jQuery. jQuery akan berfungsi untuk memisahkan karakteristik *behavior* dari struktur HTML.

Memulai jQuery

Untuk menggunakan jQuery, langkah pertama yang harus dilakukan adalah mengunduh file library jQuery yang ada pada site <http://jquery.com> untuk mendapatkan versi jQuery terbaru.

Ada dua versi file library jQuery yang dapat digunakan yaitu :

- Production, untuk produksi biasanya library jQuery telah ‘di-compile’ sehingga ukurannya akan lebih kecil, biasanya ukurannya 50% lebih kecil dibandingkan dengan library jQuery yang tidak ‘di-compile’. File library jQuery untuk produksi ini biasanya tidak bersahabat untuk dibaca karena susunan baris-baris kode yang telah dimanpatkan menjadi satu baris.
- Development, untuk fase pengembangan halaman web maka dapat digunakan file library jQuery ini. Isi file ini dapat dengan mudah dibaca, sehingga developer dapat dengan mudah mempelajari fungsi-fungsi yang dimiliki oleh jQuery.

Saat ebook ini ditulis, versi terbaru dari jQuery adalah 1.3.2. Nama file library jQuery untuk production adalah `jquery-1.3.2.min.js` sedangkan nama file untuk development adalah `jquery-1.3.2.js`.

Untuk menggunakan library jQuery pada halaman web maka cukup tambahkan seperti baris berikut ini.

```
<script type="text/javascript" src="jquery-1.3.2.js"></script>
```

Berikut ini adalah program singkat dengan menggunakan jQuery untuk menampilkan message box ketika halaman web dibuka.

```
<html>
<head>
<title>Hello jQuery</title>

<script language="javascript" src="jquery-1.3.2.js" type="text/javascript">
</script>

<script language="javascript">
    $(document).ready(function() {
        alert('Hello jQuery');
    });
</script>
</head>
<body>

</body>
</html>
```

Program di bawah ini adalah program House Finder yang ditulis ulang dengan menggunakan jQuery.

```
<html>
<head>
<title>House Finder</title>
<link rel="stylesheet" type="text/css" href="style.css" />

<script language="javascript" type="text/javascript" src="jquery-1.3.2.js">
</script>

<script language="javascript" type="text/javascript">
    function calculate() {
        var incomeTxt = $("#income").val();
        var roomTxt = $("#room").val();

        if (incomeTxt != "" && roomTxt != "") {
            incomeYear = 12 * incomeTxt;
            housePrice = roomTxt * 5000;

            if (incomeYear > housePrice) {
                alert('Congrats, for your new house.');
            }
            else {
                alert('Please decrease your room.');
            }
        }
        else {
            alert('Please complete your data');
        }
    }

    $(document).ready(function() {
        $("#btn").click(function() {
            calculate();
        });
    });
</script>
</head>
<body>
    <h1>Ready to find a new house ?</h1>

    <div class="label">Enter your annual income :</div>
    <div class="input"><input type="text" id="income" /></div>

```

```

<div class="separator"></div>

<div class="label">Enter the number of bedrooms :</div>
<div class="input"><input type="text" id="room" /></div>

<div class="separator"></div>

<div><input type="button" id="btn" value="Calculate Price" /></div>
</body>
</html>

```

Bila dibandingkan dengan program “House Finder” sebelumnya ada beberapa perbedaan yang dapat dilihat.

Pada program sebelumnya (tanpa menggunakan jQuery), untuk mengambil nilai pada textbox income (`<input type="text" id="income" />`) dilakukan dengan cara berikut ini :

```

. . .
var incomeTxt = document.getElementById('income').value;
. . .

```

Dengan jQuery, cukup dilakukan dengan cara seperti berikut ini.

```

. . .
var incomeTxt = $("#income").val();
. . .

```

Berbeda dengan cara sebelumnya, untuk mengambil elemen dengan `id` tertentu, pada jQuery perlu ditambahkan tanda `#`, sehingga bila ingin mengakses elemen dengan nilai `id=income`, maka akan ditulis seperti ini `#income`.

Perbedaan yang lain adalah pada bagian button, pada program sebelumnya pada button ditambahkan event `onclick` yang memanggil fungsi `calculate`, seperti baris berikut ini :

```

. . .
<input type="button" id="btn" value="Calculate Price" onclick="calculate()" />
. . .

```

Sedangkan dengan menggunakan jQuery pada bagian button akan terlihat seperti berikut ini :

```

. . .
<input type="button" id="btn" value="Calculate Price" />
. . .

```

Pada elemen button di atas tidak terdapat event `onclick`. Event `onclik` diberikan pada button ini dengan cara menulis baris seperti berikut ini.

```

<script language="javascript" type="text/javascript">
. . .
$(document).ready(function() {
    $("#btn").click(function() {
        calculate();
    });
});
</script>

```

Pada kode di atas ada dua hal penting yang harus diketahui. Yang pertama adalah baris `$(document).ready(function() {});`, method `ready()` ini berfungsi untuk memeriksa kesiapan dokumen untuk dimanipulasi. Kesiapan yang dimaksudkan diantaranya adalah semua elemen pada halaman web sudah dimuat dengan sempurna. Bila status telah terpenuhi maka akan dilakukan baris kode di dalamnya, yaitu :

```

$("#btn").click(function() {
    calculate();
});

```

Baris kode di atas berfungsi untuk mendaftarkan event click pada button dengan `id="btn"`, fungsi yang akan dijalankan ketika tombol diklik adalah `calculate()`.

Contoh di atas hanya untuk menunjukkan perbedaan antara penulisan baris program tanpa jQuery dengan baris program yang menggunakan jQuery. Paparan detail mengenai cara penulisan jQuery dan fungsi-fungsi yang telah disediakan oleh jQuery akan dibahas pada bagian selanjutnya.

Cara Kerja jQuery

Untuk mengetahui cara kerja jQuery, akan digunakan hidangan yang ada di meja sebagai analoginya. Ada tiga tahapan yang harus dilakukan untuk menikmati hidangan, yaitu :

- a. Menunggu hidangan disajikan.
- b. Memilih hidangan.
- c. Menyantap hidangan.

Untuk melihat lebih jelas tiga langkah di atas dilakukan maka terlebih dahulu perhatikan baris-baris HTML & CSS di bawah ini.

```
<html>
<head>
<title>Sajian Web</title>
<style type="text/css">
.ContentText
{
    font-family:Tahoma;
    font-size:0.8em;
    font-weight:bold;
    color:Red;
}

.ContentLink
{
    font-family:Tahoma;
    font-size:0.7em;
    color:Blue;
    text-decoration:underline;
    text-transform:uppercase;
}

.ContentLink:hover
{
    font-family:Tahoma;
    font-size:0.7em;
    color:Blue;
    text-decoration:none;
    text-transform:uppercase;
}
</style>
<script type="text/javascript" src="jquery-1.3.2.js"></script>
<script language="javascript" type="text/javascript">
$(document).ready(function() {
    $("#btn").click(function() {
        $("p").toggleClass("ContentText");
        jQuery("p a").toggleClass("ContentLink");
    });
});
</script>
</head>
<body>
<p>
    Page content. Page content. Page content. Page content.
    Page content. Page content. Page content. Page content.
    Page content. Page content. Page content. Page content.
    <br />
    <a href="#">Link in Page Content</a>
</p>
<input type="button" id="btn" value="Change Us" />
</body>
```

```
</html>
```

Baris-baris kode di atas adalah sajian yang akan dihidangkan. Sebelum memilih sajian untuk disantap tentu saja sajian yang diinginkan harus sudah terhidang. Artinya untuk memilih elemen yang ingin dilakukan operasi, maka elemen tersebut harus sudah ditampilkan pada halaman web.

jQuery mempunyai fungsi berikut ini untuk mengetahui apakah semua elemen atau elemen yang diinginkan sudah ditampilkan pada halaman web.

```
$(document).ready(function() {  
    // baris program  
});
```

Setelah hidangan telah disajikan, langkah selanjutkan adalah memilih hidangan yang diinginkan. Untuk proses pemilihan eleman, jQuery mempunyai fungsi selector yang sangat mudah dan singkat untuk digunakan. Berikut ini fungsi selector yang dapat digunakan :

```
$(selector)
```

Atau

```
jQuery(selector)
```

Contoh penggunaanya dapat dilihat seperti di bawah ini.

```
$("#btn") . . .  
$("p") . . .  
jQuery("p a") . . .
```

Pada baris pertama, elemen yang dipilih adalah elemen yang mempunyai `id=btn`. Sedangkan pada baris kedua, elemen yang dipilih adalah elemen tag `p`. Dan pada baris ketiga, elemen yang dipilih adalah elemen tag `a` yang berada di dalam elemen tag `p`.

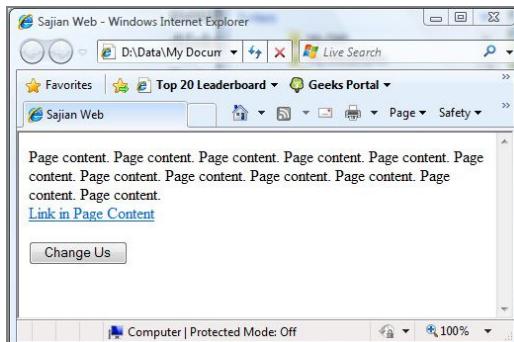
Setelah hidangan-hidangan yang diinginkan telah dipilih maka langkah selanjutnya adalah menikmatinya, artinya elemen-elemen pada halaman web yang telah dipilih dapat dilakukan operasi terhadapnya.

Pada contoh di bawah ini, dapat dilihat operasi yang dilakukan pada elemen yang dipilih.

```
$("#btn").click(function() {  
    $("p").toggleClass("ContentText");  
    jQuery("p a").toggleClass("ContentLink");  
});
```

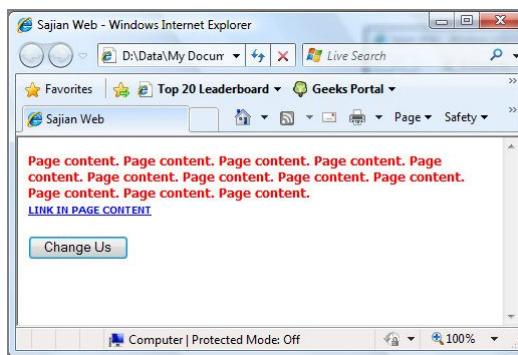
Pada baris pertama dapat dilihat elemen dengan `id=btn` akan diberikan event `click`. Sedangkan pada baris kedua dan ketiga, akan ditambahkan dijalankan fungsi `toggleClass` untuk semua elemen tag `p` dan tag `a` yang berada di dalam elemen tag `p`.

Bila halaman web ini dilihat pada web browser maka akan dilihat tampilan seperti berikut ini.



Gambar 1-4 : Halaman web dengan jQuery.

Ketika tombol Change Us diklik maka style dari text dan link yang berada dalam tag p akan berubah seperti pada gambar berikut ini.



Gambar 1-5 : Tampilan setelah tombol diklik.

Tombol Change Us bersifat toggle, yang berfungsi untuk menambahkan style pada text dan link ketika diklik dan menghilangkan style pada text dan link ketika tombol diklik lagi. Jadi bila tombol Change Us pada gambar 1-5 diklik, tampilannya akan kembali seperti pada gambar 1-4.

Tanda dollar pada jQuery

Pada contoh-contoh di atas sering ditemui tanda `$()`. Tanda ini adalah alias dari fungsi jQuery yang berfungsi sebagai shortcut menuju ke suatu elemen pada struktur html yang dipilih. Tanda ini juga dapat juga disebut fungsi selector.

Selain berfungsi sebagai selector untuk elemen atau item yang sudah ada pada halaman web, fungsi ini juga bisa berisi string (dapat berisi elemen HTML) yang dapat ditambahkan pada halaman web.

```
$( "<div><p>Hello</p></div>" ).appendTo( "body" )
```

Pada contoh di atas dapat dilihat penggunaan fungsi ini untuk menambahkan elemen berikut ini :

Hello

Ke bagian body pada halaman web..

Penggunaan fungsi `$()` harus dilakukan dengan hati-hati bila library jQuery digabungkan dengan library JavaScript lainnya. Karena tanda `$` sangat lajim digunakan juga pada library JavaScript lainnya dengan fungsi yang berbeda.

Apabila halaman web yang dibangun mempunyai kemungkinan menggunakan library JavaScript selain jQuery, ada baiknya menggunakan fungsi `jQuery()` untuk menggantikan `$()` ini.

Kenapa memilih jQuery ?

Dari paparan di atas, ada satu alasan kuat yang membuat jQuery bisa dijadikan salah satu pilihan dari banyak pilihan library JavaScript yang telah tersedia, yaitu jQuery mampu mempersingkat waktu dan baris program untuk menambahkan fungsi-fungsi dinamik pada halaman web mengambil objek dari suatu elemen, menambahkan class CSS kepada elemen tersebut, membuat animasi bahkan melakukan modifikasi atribut dari elemen tersebut. Apabila hal-hal tersebut dilakukan dengan menulis baris-baris program JavaScript dari awal, sudah dapat dipastikan baris program yang akan ditulis akan lebih panjang dan akan memakan banyak waktu untuk membuat sesuatu seperti yang telah dimiliki oleh jQuery.

Library jQuery tergolong berukuran kecil, ukuran library jQuery terkecil adalah 15kb sedangkan jQuery untuk development berukuran 118kb, masih tergolong kecil bila dibandingkan kemampuan yang dapat diberikan oleh jQuery.

jQuery adalah library jQuery yang dapat berjalan pada banyak browser, seperti Internet Explorer 6, Firefox, Chrome, Safari, Opera dan lain-lain.

Alasan yang terakhir adalah kerena jQuery telah compliant CSS3.

Kemampuan jQuery

Dari contoh yang telah ditulis di atas, sudah dapat dilihat beberapa kemampuan jQuery. Berikut ini adalah uraian lengkap kemampuan yang dimiliki jQuery, yaitu :

- a. jQuery mampu mengakses elemen pada dokumen. Tanpa menggunakan JavaScript library, diantaranya jQuery, harus dibuat banyak baris program untuk mengakses dan mengelola Document Object Model (DOM) dan struktur dokumen HTML. jQuery mempunyai mekanisme selector yang bagus dan efisien untuk memilih elemen yang diinginkan untuk nantinya dikelola atau dimanipulasi.
- b. jQuery mempunyai kemampuan untuk memodifikasi atau mengubah antarmuka halaman web. Implementasi CSS pada halaman web adalah metode yang sering digunakan untuk memberikan memberikan style pada elemen-elemen yang akan ditampilkan pada halaman web tersebut. Tetapi permasalahan yang sering muncul adalah CSS yang ditulis sangat berpengaruh dengan web browser yang digunakan. Sering terjadi perbedaan antarmuka antara masing-masing web browser. Sebagai contoh, halaman web yang terlihat baik pada browser Internet Explorer mungkin akan terlihat berbeda bila dibuka dengan browser Firefox. Permasalahan ini dapat diselesaikan dengan menggunakan jQuery, sebagai library JavaScript yang dapat bekerja pada semua browser yang mempunyai kemampuan untuk mengubah class-class dan style suatu elemen pada dokumen sebelum atau setelah elemen tersebut ditampilkan.
- c. jQuery dapat menambah content pada dokumen, seluruh struktur pada HTML dapat ditambah dan diperluas dengan menggunakan jQuery. Teks dapat diubah, gambar dapat ditambahkan, list dapat disusun atau disorting. Fungsi-fungsi untuk melakukan itu semua dapat ditemui pada Application Programming Interface (API) milik jQuery.

- d. jQuery digunakan untuk merespon interaksi user.
- e. jQuery menyediakan fasitas untuk membuat animasi dengan mudah.
- f. jQuery mampu mengambil informasi dari server tanpa melakukan refresh halaman web, pattern kode berikut ini dikenal dengan nama Asynchronous Javascript and XML (AJAX).
- g. jQuery juga menyediakan fungsi-fungsi hasil pengembangan dari fungsi-fungsi yang dimiliki oleh JavaScript, seperti iterasi dan manipulasi array.

Siapa pengguna jQuery ?

Dari kemampuan jQuery yang telah dipaparkan di atas ada dua hal yang dapat digarisbawahi, yaitu :

- a. jQuery dapat mengambil informasi dari server, tanpa melakukan refresh halaman web.
- b. jQuery mempunyai kemampuan untuk memodifikasi atau mengubah antarmuka halaman web.

Dari kedua fitur tersebut, maka jelas sudah jQuery dapat digunakan oleh Web Developer dan Web Designer.

Visual Web Developer 2008, jQuery & ASP.NET

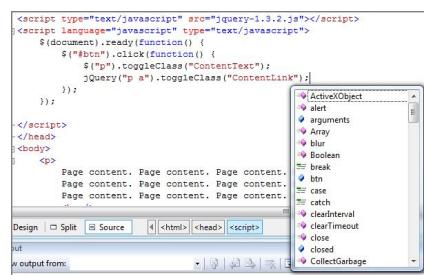
Pada bagian ini akan mengenalkan Visual Web Developer sebagai tool gratis yang dapat digunakan oleh web developer, bahkan oleh web designer juga. Visual Web Developer 2008 (VWD 2008) adalah tool untuk membangun aplikasi web dengan ASP.NET.

Bagi pembaca yang telah menggunakan Visual Studio 2008, apa yang ditulis ebook ini juga dapat diimplementasikan pada Visual Studio 2008, walaupun contoh-contoh yang digunakan pada ebook ini menggunakan Visual Web Developer 2008.

JavaScript Intellisense

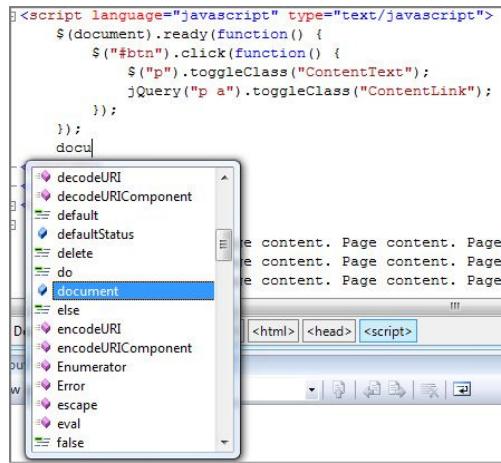
Visual Web Developer 2008 mempunyai satu kelebihan tambahan yaitu JavaScript Intellisense. Fitur ini memberikan bantuan bagi web developer untuk menulis program JavaScript dengan lebih mudah, dikarenakan dengan fitur ini web developer akan mengetahui keyword dan fungsi yang dimiliki JavaScript.

Dengan menekan tombol **ctrl+spasi**, maka Visual Web Developer akan menampilkan keyword dan fungsi yang dimiliki oleh JavaScript. Tentu saja fitur ini hanya akan keluar ketika cursor atau pointer berada pada script block (`<script language="javascript" type="text/javascript">`).



Gambar 1-6 : JavaScript Intellisense pada Visual Web Developer.

Selain menekan `ctrl+spasi`, intellisense ini akan muncul dengan sendirinya jika ketika kode JavaScript diketik, intellisense akan muncul sesuai dengan keyword yang diketikkan.



Gambar 1-7 : Bantuan sesuai dengan keyword yang diketik.

Fitur intellisense ini tidak hanya menampilkan keyword dan fungsi yang dimiliki oleh JavaScript saja, tapi juga dapat menampilkan fungsi-fungsi yang dibuat oleh developer. Sebagai contoh terdapat file `math.js` yang berisi fungsi-fungsi matematika sederhana seperti pada contoh di bawah ini.

```
function tambah(bil1, bil2) {
    /// <summary>Menjumlahkan dua bilangan</summary>
    /// <param name="bil1" type="Number">Bilangan pertama</param>
    /// <param name="bil2" type="Number">Bilangan kedua</param>
    hasil = bil1 + bil2;

    return hasil;
}

function kurang(bil1, bil2) {
    /// <summary>Mengurangkan dua bilangan</summary>
    /// <param name="bil1" type="Number">Bilangan pertama</param>
    /// <param name="bil2" type="Number">Bilangan kedua</param>
    hasil = bil1 - bil2;

    return hasil;
}
```

Pada contoh di atas, pada fungsi tambah dan kurang terdapat keterangan dalam format XML. Berikut ini adalah contoh halaman web yang akan menggunakan file `math.js`.

```
<html>
<head>
<title>Matematika</title>
<script language="javascript" type="text/javascript" src="math.js"></script>
<script language="javascript" type="text/javascript">

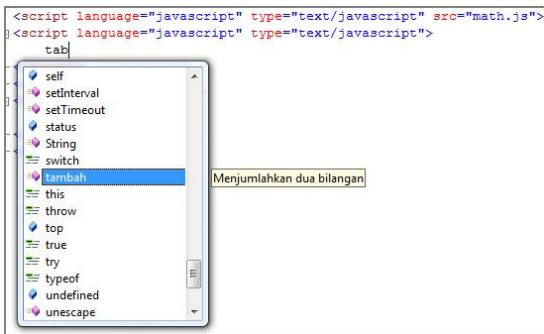
</script>
</head>
<body>

</body>
</html>
```

pada contoh ini dapat dilihat file `math.js` sudah dikaitkan dengan halaman web ini dengan cara berikut :

```
<script language="javascript" type="text/javascript" src="math.js"></script>
```

Fungsi tambah dan kurang yang ada pada `math.js` dapat langsung dikenali pada Visual Web Developer, seperti yang dilihat pada gambar berikut ini.



Gambar 1-8 : VWD menampilkan keterangan fungsi tambah.



Gambar 1-9 : VWD menampilkan parameter pada fungsi tambah.

Dari gambar di atas dapat dilihat Visual Web Developer 2008 mempunyai kemampuan untuk membaca keterangan pada suatu fungsi.

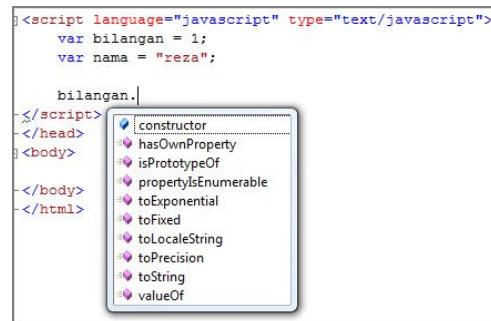
Visual Web Developer juga mampu mengetahui tipe data dari nilai suatu variable yang ditulis dengan JavaScript. Dengan mengetahui tipe data tersebut, maka Visual Web Developer akan memberikan fungsi-fungsi atau operasi-operasi yang sesuai dengan variable tersebut.

Sebagai contoh dapat dilihat pada baris berikut ini :

```
<script language="javascript" type="text/javascript">
    var bilangan = 1;
    var nama = "reza";

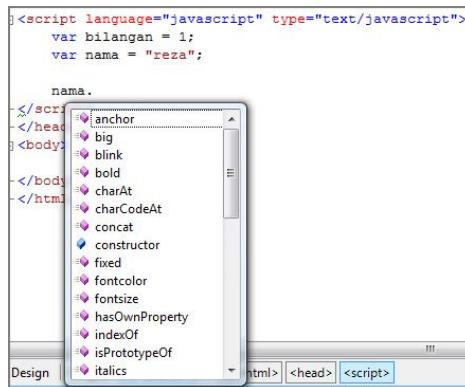
</script>
```

Pada gambar di bawah ini dapat dilihat fungsi yang bisa digunakan oleh variable bilangan, dimana variable bilangan ini mempunyai tipe data integer/number.



Gambar 1-10 : Fungsi yang dapat digunakan oleh variable dengan tipe data integer/number.

Sedangkan gambar di bawah ini adalah fungsi yang dapat digunakan oleh variable nama.



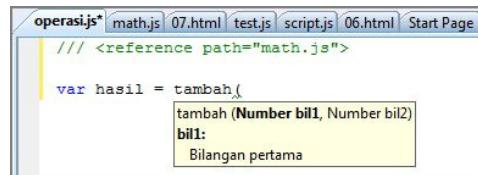
Gambar 1-11 : Fungsi yang dapat digunakan oleh variable dengan tipe data string.

Dari gambar 1.10 dan gambar 1-11, dapat dilihat bahwa Visual Web Developer dapat mengenali tipe data yang digunakan pada variable yang ditulis dengan JavaScript dan akan memfilter fungsi-fungsi yang dapat digunakan oleh masing-masing variable tersebut.

Fitur lain yang diberikan oleh Visual Web Developer adalah mengaitkan satu file *.js dengan file *.js lainnya. Misalnya telah dimiliki file dengan nama `math.js`, kemudian dibuat dibuat file `operasi.js` yang akan menggunakan fungsi-fungsi yang ada pada file `math.js`. Untuk mengaitkan file `math.js` pada file `operasi.js`, pada file `operasi.js` perlu ditambahkan baris berikut ini.

```
//> <reference path="math.js">
```

Dengan menuliskan baris tersebut, maka ketika file `operasi.js` diedit dengan menggunakan Visual Web Developer maka fungsi-fungsi pada `math.js` dapat ditampilkan oleh fitur intellisense.



Gambar 1-12 : Contoh penggunaan reference pada VWD 2008.

jQuery Intellisense

Pada bagian sebelumnya telah diterangkan fitur JavaScript Intellisense. Pada bagian ini akan diterangkan bagaimana cara untuk membangkitkan jQuery Intellisense pada Visual Web Developer.

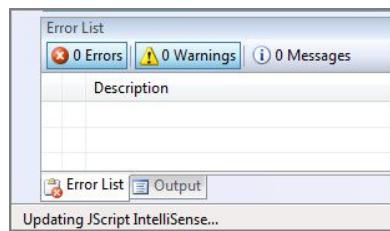
Untuk membangkitkan fitur jQuery Intellisense pada Visual Web Developer ada beberapa langkah yang harus dilakukan.

Sebelumnya pastikan terlebih dahulu Visual Web Developer 2008 yang digunakan sudah menggunakan SP1. Kemudian download patch KB958502 untuk Visual Web Developer disini

<http://code.msdn.microsoft.com/KB958502/Release/ProjectReleases.aspx?ReleaseId=1736>.
Dan install.

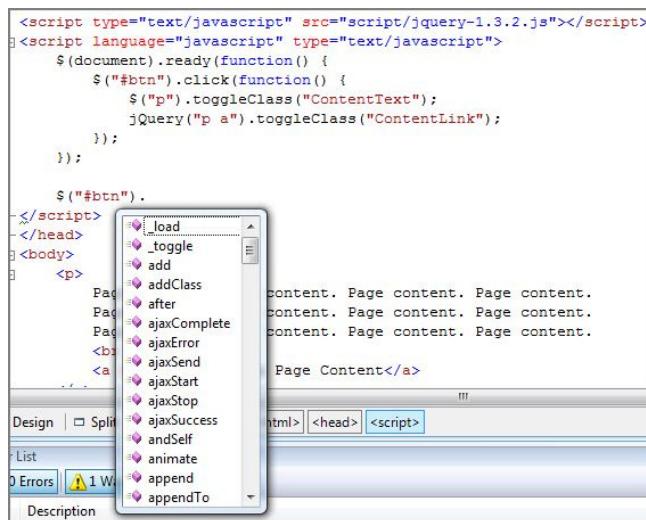
Selanjutnya Untuk membangkitkan fitur jQuery Intellisense pada Visual Web Developer, ada file yang perlu diunduh yaitu `jquery-1.3.2-vsdoc2.js` yang dapat diunduh disini http://docs.jquery.com/Downloading_jQuery#Download_jQuery, untuk mendapatkan versi terbaru. File ini berukuran kecil, sekitar 188KB. Pastikan file ini direname menjadi `jquery-1.3.2-vsdoc.js` dan diletakkan pada lokasi yang sama dengan lokasi file `jquery-1.3.2.js`.

Setelah semua langkah diatas dilakukan maka pastikan lakukan update intellisense dengan memilih menu `Edit > IntelliSense > Update JScript Intellisense` atau dapat dengan menekan tombol `Ctrl+Shift+J`. Bila proses berhasil maka tidak akan ada pesan Error atau Warning.



Gambar 1-13 : Proses update JScript IntelliSense.

Setelah langkah tersebut selesai maka sudah dapat dipastikan jQuery Intellisense sudah dapat digunakan pada Visual Web Developer seperti pada gambar berikut ini.



Gambar 1-14 : jQuery Intellisense.

Kerjasama jQuery & ASP.NET

Sesuai kemampuan jQuery yang telah disebutkan di atas, maka penggunaan jQuery pada ASP.NET dapat dilakukan untuk beberapa keperluan, yaitu :

- a. jQuery dapat bekerja untuk memberikan style dengan lebih cepat, lebih mudah dan lebih baik pada server control yang dimiliki oleh ASP.NET.
- b. jQuery dapat melakukan operasi-operasi di sisi client seperti validasi nilai-nilai pada input yang ada pada form, menampilkan dialog konfirmasi saat user akan menghapus data dan lain-lain.
- c. jQuery dapat membuat animasi dan hal-hal dinamis lainnya sehingga halaman ASP.NET menjadi lebih menarik dan responsif.

- d. Komunikasi data pada server dengan jQuery. Sebagai contoh penggunaan jQuery untuk berkomunikasi dengan ASP.NET web service. Komunikasi data ini juga didukung jQuery dengan memberi kemudahan implementasi AJAX.

HTML & ASP.NET

jQuery adalah library yang berjalan di sisi client, sedangkan ASP.NET adalah framework yang lebih banyak berjalan disisi server. Peran jQuery pada ASP.NET adalah saat control-control server dari ASP.NET sudah dirender menjadi HTML.

Pengetahuan akan elemen-elemen HTML yang dihasilkan oleh control server ASP.NET akan sangat membantu web developer maupun designer untuk menggunakan jQuery pada ASP.NET. jQuery mempunyai fungsi selector untuk mengoleksi elemen-elemen HTML yang akan dimodifikasi, di berikan event, dilakukan operasi terhadapnya dan lain-lain. Cara jQuery mengoleksi elemen tersebut dengan memanggil nama `tag`, `nama class` atau nilai `id` dari elemen yang diinginkan.

Pada bagian ini hanya akan menunjukkan perbedaan-perbedaan yang terjadi antara apa yang ditulis pada server control ASP.NET dengan output HTML akhir.

Berikut ini contoh dua server control ASP.NET.

```
<asp:Label ID="Label_Name" runat="server" Text="Name" CssClass="TxtLabel">  
</asp:Label>  
  
<asp:TextBox ID="TextBox_Name" runat="server" CssClass="InputForm"></asp:TextBox>
```

Dan berikut ini adalah kode dari HTML setelah file `ASPX` di atas dijalankan pada browser.

```
<span id="Label_Name" class="TxtLabel">Name</span>  
  
<input name="TextBox_Name" type="text" id="TextBox_Name" class="InputForm" />
```

Server control `asp:Label` akan menjadi tag `span`, atribut `ID` pada server control akan menjadi `id` dengan nilai yang sama yaitu `Label_Name`. sedangkan atribut `CssClass` pada server control `asp:Label` akan menjadi atribut `class` pada tag `span` dengan nilai yang sama yaitu `TxtLabel`.

server control di atas dibuat di dalam file `ASPX` yang tidak menggunakan `Master Page`. Ada perbedaan yang dapat kita lihat ketika menggunakan file `ASPX` yang menggunakan `Master Page`.

Berikut contoh isi file `ASPX` dengan `Master Page`.

```
<%@ Page Title="" Language="C#" MasterPageFile="~/MasterPage.master"  
AutoEventWireup="true" CodeFile="10.aspx.cs" Inherits="_10" %>  
  
<asp:Content ID="Content1" ContentPlaceHolderID="head" Runat="Server">  
</asp:Content>  
  
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1"  
Runat="Server">  
  
<asp:Label ID="Label_Name" runat="server" Text="Name" CssClass="TxtLabel">  
</asp:Label>  
  
<asp:TextBox ID="TextBox_Name" runat="server" CssClass="InputForm"></asp:TextBox>  
  
</asp:Content>
```

Isinya tidak berbeda dengan contoh sebelumnya, tetapi perhatikan hasil kode HTML setelah dijalankan pada browser.

```
<span id="ctl00_ContentPlaceHolder1_Label_Name" class="TxtLabel">Name</span>
```

```
<input name="ctl00$ContentPlaceHolder1$TextBox_Name" type="text"
id="ctl00_ContentPlaceHolder1_TextBox_Name" class="InputForm" />
```

Perbedaan yang terlihat adalah nilai id dari masing-masing elemen.

Nilai id yang ditampilkan pada halaman HTML tersebut adalah sama dengan nilai property ClientID yang dimiliki oleh setiap server control. Jadi bila ingin mengambil nilai id dari elemen tersebut pada pemrograman client side, sebagai contoh JavaScript dapat digunakan cara seperti di bawah ini.

```
<asp:Content ID="Content1" ContentPlaceholderID="head" Runat="Server">
</asp:Content>

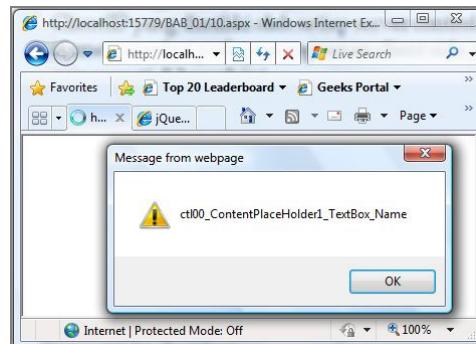
<asp:Content ID="Content2" ContentPlaceholderID="ContentPlaceholder1"
Runat="Server">

    <script language="javascript" type="text/javascript">
        alert('<%= TextBox_Name.ClientID %>');
    </script>

    <asp:Label ID="Label_Name" runat="server" Text="Name" CssClass="TxtLabel">
    </asp:Label>

    <asp:TextBox ID="TextBox_Name" runat="server" CssClass="InputForm">
    </asp:TextBox>
</asp:Content>
```

Contoh program di atas akan menampilkan nilai id dari input.



Paparan pada bagian ini diharapkan dapat menjadi informasi awal yang nanti dapat digunakan untuk memadukan ASP.NET dengan jQuery.

2.

Bekerja dengan jQuery

Pada bab sebelumnya telah diterangkan sedikit dasar-dasar jQuery yang mesti diketahui oleh pada web developer atau web designer. Selain itu juga juga diterangkan penggunaan Visual Web Developer 2008 sebagai tool gratis yang dapat digunakan untuk memudahkan penulisan program dengan JavaScript dan jQuery.

Pada bab ini akan memperkenalkan jQuery secara lebih mendalam. Akan dijelaskan secara lebih mendetail tentang fungsi-fungsi inti yang dimiliki oleh jQuery, mengenal fungsi selector, manipulasi elemen HTML atau DOM, membuat animasi dan efek. Contoh-contoh implementasi ini akan dilakukan pada halaman ASP.NET dan akan bekerja pada control-control yang dimiliki oleh ASP.NET. Semuanya akan ditulis dengan menggunakan Visual Web Developer 2008 sehingga diharapkan pembaca akan terbiasa menggunakan jQuery dengan tool ini.

jQuery Core

Pada jQuery terdapat fungsi-fungsi utama yang akan selalu digunakan pada setiap program jQuery yang ditulis. Selain itu juga terdapat fungsi-fungsi yang dapat digunakan bagi developer yang ingin melakukan extend jQuery menjadi sebuah plugin (plugin akan dijelaskan lebih detail pada bab selanjutnya).

Fungsi `\$ (...)`

Seperti yang telah disebutkan pada bab sebelumnya, fungsi fungsi `\$ (...)` dapat digunakan sebagai pengganti `\$(...)`.

Fungsi ini adalah bagian dari jQuery Core, fungsi yang akan sering digunakan pada setiap program jQuery. Fungsi ini mempunyai banyak kegunaan. Selain sebagai selector elemen pada halaman HTML seperti yang telah dicontohkan pada bab pertama, fungsi ini juga dapat digunakan untuk keperluan hal-hal berikut ini :

- a. Menambahkan elemen *on-the-fly* pada halaman web.
- b. Pengganti fungsi `\$(document).ready()`.

Berikut ini adalah contoh-contoh penggunaan fungsi `\$ (...)` berdasarkan fungsinya.

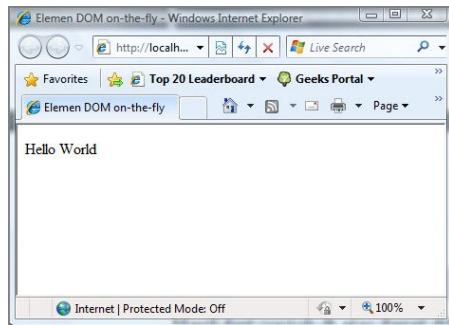
Menambahkan elemen secara *on-the-fly* pada halaman web

Berikut ini adalah contoh penggunaan fungsi `\$ (...)` untuk menambahkan elemen *on-the-fly*.

```
<html>
<head>
    <title>Elemen DOM on-the-fly</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        \$(document).ready(function() {
            $("<p>Hello World</p>").appendTo("body");
        });
    </script>
</head>
<body>
```

```
</body>  
</html>
```

Hasil dari contoh di atas dapat dilihat pada gambar di bawah ini.



Gambar 2-1 : Menambahkan elemen on-the-fly.

Pada contoh kode di atas dapat dilihat tidak terdapat elemen apapun di antara tag `<body>` dan `</body>`. Tulisan "Hello World" yang dilihat pada browser ada karena baris berikut ini.

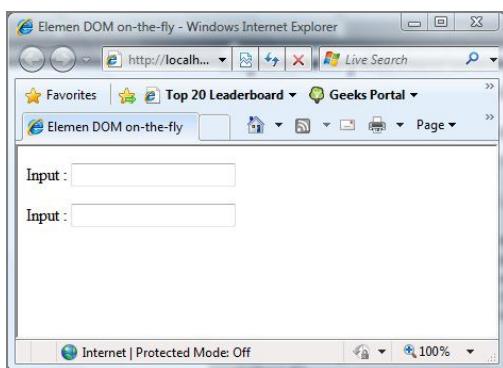
```
$( "<p>Hello World</p>" ).appendTo("body");
```

Dapat dilihat pada baris di atas bahwa fungsi `$(...)` diisi dengan elemen, yaitu `<p>Hello World</p>`, elemen ini akan ditambahkan diantara tag `body`.

Masih membahas penggunaan fungsi `$(...)` untuk menambahkan elemen *on-the-fly*, maka berikut ini contoh lainnya.

```
<html xmlns="http://www.w3.org/1999/xhtml">  
<head>  
    <title>Elemen DOM on-the-fly</title>  
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>  
    <script language="javascript" type="text/javascript">  
        $(document).ready(function() {  
            $("<label>Input : </label><input type='text' />").appendTo("p");  
        });  
    </script>  
</head>  
<body>  
  
    <p></p>  
  
    <p></p>  
  
</body>  
</html>
```

Hasil dari kode ini dapat dilihat pada gambar berikut ini.



Gambar 2-2 : Menambahkan elemen pada elemen lain.

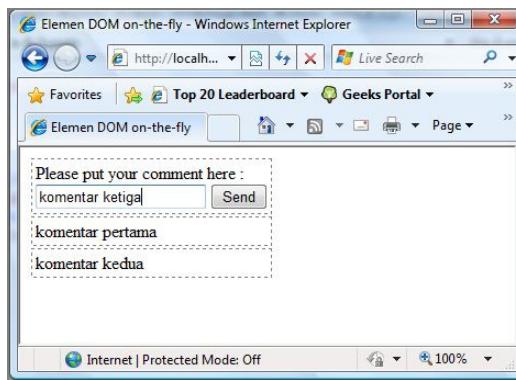
Pada contoh ini terdapat baris kode yang mirip seperti pada contoh sebelumnya.

```
$(“<label>Input : </label><input type='text' />”).appendTo(“p”);
```

Pada fungsi `$(...)` terdapat elemen yang akan ditambahkan kedalam tag `p`. Pada halaman web terdapat dua tag `p`, sehingga elemen yang ada pada fungsi `$(...)` akan ditambahkan pada kedua tag `p` tersebut.

Pada kedua contoh di atas, proses penambahan elemen pada halaman web dilakukan secara otomatis. Tetapi bukan hanya itu saja yang bisa dilakukan, sebagai contoh, proses penambahan elemen ini juga dapat dilakukan oleh user dengan mentrigger event click yang ada pada suatu tombol atau link.

Contoh berikut ini adalah akan mempunyai antarmuka seperti pada gambar ini.



Gambar 2-3 : Menambahkan komentar.

Pada halaman web ini terdapat form untuk menulis komentar, komentar akan dituliskan pada textbox dan ketika tombol Send diklik maka isi textbox akan ditambahkan secara on-the-fly.

Berikut ini kode program dari contoh ini.

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Elemen DOM on-the-fly</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        function SendComment() {
            $(document).ready(function() {
                var comment = $("#txtComment").val();
                $("<tr><td>" + comment + "</td></tr>").appendTo("#TblComment");
                $("#txtComment").val("");
            });
        }
    </script>

    <style type="text/css">
        td
        {
            border:dashed 1px Gray;
        }
    </style>
</head>
<body>
    <table id="TblComment" cellpadding="3">
        <tr>
            <td>
                Please put your comment here : <br />
                <input type="text" id="txtComment" />
            </td>
        </tr>
    </table>
</body>
</html>
```

```

        <input type="button" value="Send" onclick="SendComment ()" />
    </td>
</tr>
</table>
</body>
</html>

```

Berikut ini adalah baris-baris kode form untuk mengisi komentar.

```

<table id="TblComment" cellpadding="3">
<tr>
<td>
Please put your comment here : <br />
<input type="text" id="txtComment" />
<input type="button" value="Send" onclick="SendComment ()" />
</td>
</tr>
</table>

```

Pada baris-baris program di atas dapat dilihat teks ditulis miring dengan warna merah, terdapat tiga hal penting yang dapat kita lihat, yaitu :

- a. `TblComment`, pada dua contoh sebelumnya elemen ditambahkan pada diantara tag `body` dan tag `p`. Pada contoh ini akan ditambahkan elemen pada tabel dengan `id` yang bernilai `TblComment`.
- b. `txtComment`, nilai ini adalah nilai `id` dari textbox. Pemberian nilai `id` pada textbox ini dimaksudkan untuk mempermudah mengambil nilai dari textbox ini.
- c. `SendComment` adalah nama fungsi yang akan dipanggil ketika tombol `Send` diklik.

Di dalam fungsi `SendComment` terdapat tiga baris penting yang dapat dilihat di bawah ini.

```

var comment = $("#txtComment").val();
$("<tr><td>" + comment + "</td></tr>").appendTo("#TblComment");
$("#txtComment").val("");

```

Baris pertama berfungsi untuk mengambil nilai pada textbox dengan `id=txtComment`, nilai dari textbox tersebut akan ditampung oleh variable `comment`.

Pada baris kedua akan dilakukan proses penambahan elemen berubah 1 baris dan 1 kolom pada tabel dengan `id=TblComment`.

Dan yang terakhir dilakukan penghapusan nilai pada textbox dengan `id=txtComment`.

Pengganti fungsi `$(document).ready()`

Seperti yang telah disebutkan pada bab sebelumnya, fungsi `$(document).ready()` mempunyai peran untuk menunggu status selesai loading seluruh elemen pada halaman web.

Penulisan sintaks berikut :

```

$(document).ready(function() {
    // baris kode yang dieksekusi ketika dokumen siap
});

```

Dapat diganti dengan sintaks berikut ini :

```

$(function() {
    // baris kode yang dieksekusi ketika dokumen siap
});

```

Atau sintak ini :

```

jQuery(function($) {
    // baris kode yang dieksekusi ketika dokumen siap
});

```

```
});
```

Fungsi selector elemen DOM

Fungsi ini sudah sering digunakan pada bagian sebelumnya dari ebook ini. Fungsi selector yang sering digunakan mempunyai sintaks seperti berikut ini.

```
$(elemen) atau jQuery(elemen)
```

Fungsi selektor tidak hanya akan mengambil satu elemen saja, tetapi dapat memilih lebih dari satu elemen sesuai dengan “*kata kunci*”. Kata kunci dari elemen yang bisa ditulis dapat berupa :

- a. id dari elemen.
- b. class dari elemen.
- c. tag dari elemen.

Sintaks lain yang dapat digunakan adalah seperti berikut.

```
$(expression, context) atau jQuery(expression, context)
```

Sintaks di atas memungkinkan untuk menggunakan ekspresi untuk memilih elemen DOM. Ekspresi akan ditampilkan pada parameter pertama, expression. Sedangkan pada parameter kedua (context), parameter ini dapat diisi dengan “area” yang akan tempat elemen yang akan dipilih berada.

Contoh-contoh penggunaan fungsi ini secara detail akan diterangkan pada bagian selanjutnya. Tetapi berikut ini akan diperlihatkan dua contoh penggunaan sintaks di bagian ini.

Berikut ini adalah contoh struktur elemen pada halaman web.

```
<p>one</p> <div><p>two</p></div> <p>three</p>
```

Bila ingin mengambil elemen p yang berada di dalam tag div, maka dapat digunakan baris seperti berikut ini.

```
$("#div > p") . . .
```

Dari contoh di atas dapat dilihat ekspresi yang digunakan adalah :

```
div > p
```

Contoh yang lain dapat dilihat dari baris program berikut ini.

```
 $("input:radio", document.forms[0]);
```

Pada contoh di atas parameter expression diisi dengan nilai berikut ini.

```
input:radio
```

Ekspresi ini bertujuan untuk menemukan input dengan tipe radio. Sedangkan parameter context diisi dengan nilai berikut ini.

```
document.forms[0]
```

Nilai pada parameter ini menunjukkan area yang akan diperiksa yaitu form pertama pada halaman web.

Fungsi `$(...)` sebagai selector akan dibahas lebih detail lagi pada bab ini pada bagian Selector.

jQuery Object Accessors

Pada bagian ini terdapat fungsi dan property yang berhubungan dengan operasi pengolahan elemen yang telah dipilih.

`size()` & `length`

Fungsi selector `$(...)` dapat digunakan untuk memilih tidak hanya satu elemen saja, tetapi juga dapat memilih lebih dari satu elemen. Sebagai contoh dapat dilihat dari baris program di bawah ini.

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>jQuery Object Accessors</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        $(document).ready(function() {
            $("div") . . .
        });
    </script>
</head>
<body>
    <div>Div #1</div>
    <div>Div #2</div>
    <div>Div #3</div>
</body>
</html>
```

Fungsi `$("div")` pada contoh di atas akan mengoleksi semua elemen `div` yang ada pada halaman web. Terdapat dua fungsi yang dapat digunakan untuk mengetahui jumlah elemen yang telah dipilih, yaitu `size()` dan `length`.

Berikut ini contoh penggunaan kedua fungsi ini.

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>jQuery Object Accessors</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        $(document).ready(function() {
            var divSize = $("div").size();
            var divLength = $("div").length;

            $("#BySize").text("Size : " + divSize);
            $("#ByLength").text("Length : " + divLength);
        });
    </script>
</head>
<body>
    <span id="BySize"></span> <span id="ByLength"></span>
    <br />
    <div>Div #1</div>
    <div>Div #2</div>
    <div>Div #3</div>
</body>
</html>
```

`get()` & `get(index)`

Elemen yang telah dipilih dengan fungsi selector dapat dikoleksi ke dalam suatu array. Fungsi yang dapat digunakan untuk mengoleksi elemen-elemen tersebut dan mengambil objek pada setiap elemen adalah :

- `get()`, fungsi ini berfungsi untuk mengakses seluruh elemen yang dipilih untuk disimpan dalam array.

- `get(index)`, fungsi ini berfungsi untuk mengakses satu elemen DOM tertentu dari koleksi hasil fungsi selector, sesuai dengan index yang diberikan.

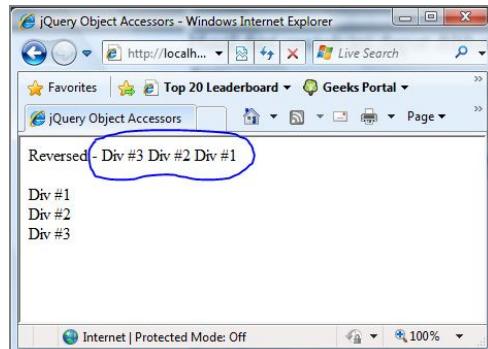
Berikut ini contoh penggunaan fungsi `get()` ini.

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>jQuery Object Accessors</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        $(document).ready(function() {
            disp($(".div").get().reverse());
        });

        function disp(divs) {
            var a = [];
            for (var i = 0; i < divs.length; i++) {
                a.push(divs[i].innerHTML);
            }
            $("span").text(a.join(" "));
        }
    </script>
</head>
<body>
    Reversed - <span></span>
    <br />
    <div>Div #1</div>
    <div>Div #2</div>
    <div>Div #3</div>
</body>
</html>
```

Baris dengan teks berwarna merah adalah contoh bagaimana fungsi `get()` digunakan, karena fungsi ini akan menyimpan hasilnya ke dalam array, maka array tersebut dapat diolah dengan fungsi `reverse()` untuk mengurutkan elemen yang tersimpan di dalam array secara terbalik.

Hasil dari contoh ini dapat dilihat pada gambar berikut ini.



Gambar 2-4 : Hasil reverse array.

Hasil dari program di atas dapat dilihat pada bagian yang dilingkari biru.

Contoh lain penggunaan fungsi `get()` dapat dilihat di bawah ini.

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>jQuery Object Accessors</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        $(document).ready(function() {
            $("#btn").click(function() {
                Hitung();
            });
        });
    </script>
</head>
<body>
    <input type="button" value="Hitung" id="btn" />
    <div>Hasil:</div>
</body>
</html>
```

```

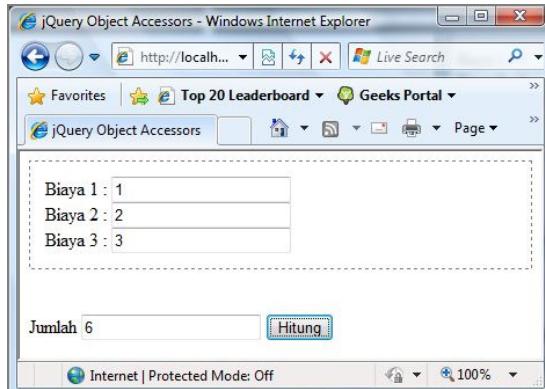
        function Hitung() {
            var inputs = $("div > input").get();
            var jumlah = parseInt(0);

            for (var i = 0; i < $("div > input").size(); i++) {
                jumlah = parseInt(jumlah) + parseInt(inputs[i].value);
            }

            $("#jumlah").val(jumlah);
        }
    </script>
</head>
<body>
    <div style="border:dashed 1px Gray;padding:13px">
        Biaya 1 : <input type="text" /><br />
        Biaya 2 : <input type="text" /><br />
        Biaya 3 : <input type="text" /><br />
    </div>
    <br /><br />
    Jumlah <input type="text" id="jumlah" /> <input id="btn" type="button" value="Hitung" />
</body>
</html>

```

Antarmuka halaman web di atas berbentuk seperti pada gambar di bawah ini.



Gambar 2-5 : Menjumlahkan nilai dari semua input.

Pada contoh ini akan dijumlahkan nilai-nilai dari textbox yang berada pada kotak abu-abu, ketika tombol `Hitung` diklik dan hasilnya akan ditampilkan pada textbox jumlah.

Textbox yang berada pada kotak abu-abu, artinya akan digunakan fungsi selector untuk mengambil seluruh nilai `input` yang ada di dalam tag `div`.

```

<div style="border:dashed 1px Gray;padding:13px">
    Biaya 1 : <input type="text" /><br />
    Biaya 2 : <input type="text" /><br />
    Biaya 3 : <input type="text" /><br />
</div>

```

Untuk mengambil elemen-elemen ini dan memasukkan ke dalam array, digunakan baris berikut ini :

```
var inputs = $("div > input").get();
```

Dan berikut ini adalah cara menjumlahkan nilai dari elemen `input` yang telah dikoleksi di dalam array.

```

for (var i = 0; i < $("div > input").size(); i++) {
    jumlah = parseInt(jumlah) + parseInt(inputs[i].value);
}

```

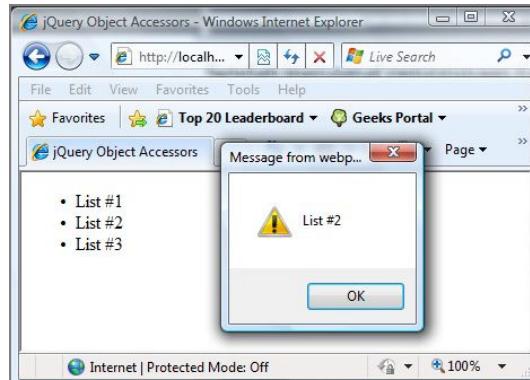
Jumlah seluruh nilai pada input yg berada pada tag `div` akan disimpan ke dalam variabel `jumlah`. Untuk menampilkan nilai pada variabel jumlah digunakan baris berikut ini.

```
$("#jumlah").val(jumlah);
```

Setelah mengenal penggunaan fungsi `get()`, selanjutkan akan diberikan contoh dan penjelasan penggunaan fungsi `get(index)`. Untuk lebih jelasnya cara penggunaan fungsi ini dengan melihat contoh di bawah ini.

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>jQuery Object Accessors</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        $(document).ready(function() {
            alert($("#ul > li").get(1).innerHTML);
        });
    </script>
</head>
<body>
    <ul>
        <li>List #1</li>
        <li>List #2</li>
        <li>List #3</li>
    </ul>
</body>
</html>
```

Contoh di atas akan menghasilkan program seperti pada gambar di bawah ini.



Gambar 2-6 : Fungsi `get(index)`.

Pada teks merah yang bisa dilihat dari baris program di atas, maka dapat diketahui fungsi selector melakukan pemilihan elemen `li` yang berada di dalam tag `ul`. Terdapat tiga elemen `li` di dalam tag `ul`. Seperti diketahui bahwa index yang digunakan pada array dimulai dari nol (0), sehingga ketika index kesatu yang dipanggil artinya sama saja dengan memanggil elemen `li` kedua, sehingga hasil dari program di atas dapat dilihat seperti pada gambar di atas.

index(subject)

Untuk mengetahui nilai index dari elemen yang telah dipilih dengan fungsi selector dapat digunakan fungsi dengan sintaks seperti berikut.

```
index(subject)
```

Output dari fungsi ini dimulai dari 0 (nol) dan apabila elemen tidak ditemukan output dari fungsi ini adalah -1. Di bawah ini contoh penggunaan fungsi `index(subject)`.

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>jQuery Object Accessors</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        $(document).ready(function() {
            $("div").click(function() {
                var index = $("div").index(this);
                $("span").text("That was div index #" + index);
            });
        });
    </script>
</head>
<body>
    <span>Click a div!</span>

    <div>First div</div>
    <div>Second div</div>
    <div>Third div</div>
</body>
</html>
```

Pada contoh di atas, akan dipilih semua elemen `div` dan diberikan event `click` pada setiap `div`.

```
$("div").click(function() {
    . . .
});
```

Ketika salah satu `div` diklik maka akan dilakukan mengambilan index dari elemen yang diklik.

```
var index = $("div").index(this);
```

Nilai index dari elemen akan disimpan di dalam variabel `index`, dan akan ditampilkan pada elemen `span`.

```
$("span").text("That was div index #" + index);
```

each(callback)

Fungsi `each(callback)` ini berfungsi untuk mengeksekusi fungsi untuk setiap elemen DOM yang sudah dipilih pada fungsi selector. Fungsi `each(callback)` ini seperti pengulangan (loop) yang akan berjalan sesuai dengan jumlah elemen DOM yang sudah terpilih. Seperti pengulangan (loop) pada umumnya, maka dapat diberhentikan.

Fungsi ini banyak gunanya, sebagai contoh, fungsi ini dapat digunakan untuk mengubah ukuran seluruh `image` yang ada pada halaman web. Atau mengubah warna `text` yang berada dalam tag `div`.

Berikut ini contoh mengubah warna seluruh teks yang berada pada tag `div`.

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>jQuery Object Accessors</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        $(document.body).click(function() {
            $("div").each(function(i) {
                if (this.style.color != "blue") {
                    this.style.color = "blue";
                }
            });
        });
    </script>
</head>
<body>
    <div>Blue text</div>
    <div>Red text</div>
    <div>Green text</div>
</body>
</html>
```

```

        } else {
            this.style.color = "";
        }
    });
});
</script>
</head>
<body>
    <div>Click here</div>

    <div>to iterate through</div>
    <div>these divs.</div>
</body>
</html>

```

Hasil dari contoh di atas akan mengubah teks di dalam `div`, ketika halaman web diklik. Ketika halaman web diklik pertama kali, warna teks akan menjadi berwarna biru, ketika halaman web diklik lagi maka teks akan kembali ke warna awal.

Baris berikut ini membuat halaman web mempunyai fungsi `click`.

```

$(document.body).click(function() {
    . . .
});

```

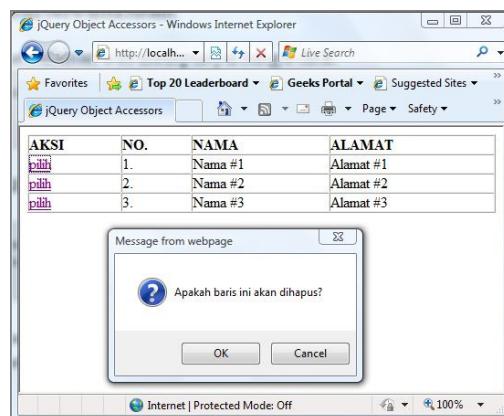
Selanjutkan akan dipilih elemen yang akan digunakan dan diberikan fungsi pada setiap elemen tersebut.

```

$("div").each(function(i) {
    // isi fungsi ditulis disini
});

```

Contoh lainnya adalah menggunakan fungsi `each(callback)` untuk memberikan event `click` pada semua tag `a` pada halaman web.



Gambar 2-7 : Event click pada setiap tag a.

Dan berikut ini adalah kode programnya.

```

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>jQuery Object Accessors</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        $(document).ready(function() {
            $("a").each(function(i) {

```

```

        this.onclick = Konfirmasi;
    });
});

function Konfirmasi() {
    confirm('Apakah baris ini akan dihapus?');
}
</script>

<style type="text/css">
.header
{
    text-transform:uppercase;
    font-weight:bold;
}
</style>
</head>
<body>
<table width="100%" cellpadding="0" cellspacing="0" border="1">
<tr>
<td class="header">Aksi</td>
<td class="header">No.</td>
<td class="header">Nama</td>
<td class="header">Alamat</td>
</tr>
<tr>
<td><a href="#">pilih</a></td>
<td>1.</td>
<td>Nama #1</td>
<td>Alamat #1</td>
</tr>
<tr>
<td><a href="#">pilih</a></td>
<td>2.</td>
<td>Nama #2</td>
<td>Alamat #2</td>
</tr>
<tr>
<td><a href="#">pilih</a></td>
<td>3.</td>
<td>Nama #3</td>
<td>Alamat #3</td>
</tr>
</table>
</body>
</html>

```

Untuk memberikan event `click` pada setiap tag `a`, digunakan baris kode berikut ini :

```

$("a").each(function(i) {
    this.onclick = Konfirmasi;
});

```

`Konfirmasi` adalah nama fungsi, yang berguna untuk menampilkan dialog konfirmasi seperti yang dapat dilihat pada gambar di atas.

eq(position)

Fungsi ini berguna untuk mengambil objek dari elemen yang telah dipilih oleh fungsi selector, sesuai dengan nilai position yang diisi. Dengan menangkap objek tersebut, maka dapat dilakukan operasi dan modifikasi terhadap objek tersebut. Operasi atau modifikasi yang bisa dilakukan di antaranya menambahkan style dengan fungsi `addClass`, menambah nilai attribut dengan fungsi `attr` dan lain-lain.

Fungsi ini sangat berbeda bila dibandingkan dengan fungsi `get(index)`, fungsi `get(index)` hanya dapat digunakan untuk mengakses nilai-nilai yang dimiliki oleh elemen yang dipilih, dan tidak dapat melakukan operasi dan modifikasi seperti halnya yang dapat dilakukan oleh fungsi `eq(position)`.

Berikut ini contoh penggunaan fungsi ini.

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>jQuery Object Accessors</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        $(document).ready(function() {
            var o = $("p").eq(1);
            o.addClass("text-tebal");
        });
    </script>
    <style type="text/css">
        .text-tebal
        {
            font-weight:bold;
        }
    </style>
</head>
<body>
    <p>This is just a test.</p><p>So is this</p>
</body>
</html>
```

Dapat dilihat cara menggunakan fungsi eq(position) dari baris berikut ini :

```
var o = $("p").eq(1);
```

Variable o mewakili dari objek kesatu (1) yang ada di dalam koleksi fungsi selector.

Plugin

jQuery sangat mudah untuk *di-extend*. Untuk keperluan itu terdapat fungsi yang dapat digunakan sesuai dengan keperluan yang diinginkan.

jQuery.extend(object)

Fungsi ini dapat digunakan untuk menambahkan fungsi pada *namespace* jQuery. Berikut ini contoh pembuatan dan penggunaannya.

Berikut ini adalah contoh penggunaan fungsi `jQuery.extend(object)`, baris-baris kode ini disimpan dengan nama `matematika.js`.

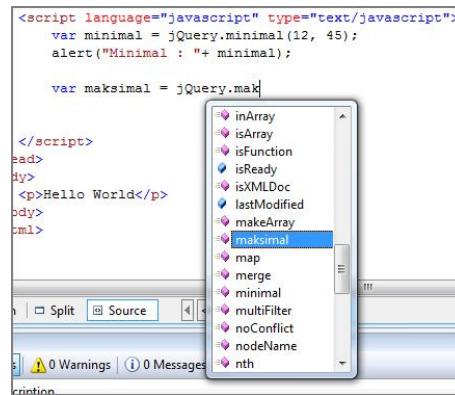
```
/// <reference path="jquery-1.3.2.js">

jQuery.extend({
    minimal: function(a, b) { return a < b ? a : b; },
    maksimal: function(a, b) { return a > b ? a : b; }
});
```

Pada contoh di atas ditambahkan dua fungsi ke dalam *namespace* jQuery, yaitu `minimal` dan `maksimal`. Dari contoh di atas dapat ditulis sintaks seperti berikut ini.

```
jQuery.extend({
    fungsi_1:function() { // isi fungsi },
    fungsi_2:function() { // isi fungsi },
    . . ,
    fungsi_n:function() { // isi fungsi }
});
```

Setelah fungsi ditambahkan, secara otomatis fungsi-fungsi tersebut terlihat dapat dikenali oleh *intellisense* pada Visual Web Developer.



Gambar 2-8 : Fungsi baru di dalam namespace jQuery.

Dan berikut ini contoh penggunaannya pada halaman web.

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Plugin</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script src="script/matematika.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        var minimal = jQuery.minimal(12, 45);
        alert("Minimal : " + minimal);

        var maksimal = jQuery.maksimal(12, 45);
        alert("Maksimal : " + maksimal);
    </script>
</head>
<body>

</body>
</html>
```

jQuery.fn.extend(object)

Berbeda dengan fungsi `jQuery.extend(object)`, fungsi `jQuery.fn.extend(object)` berguna untuk menambahkan fungsi atau method yang dapat digunakan oleh suatu elemen.

Berikut ini contoh cara menggunakan fungsi `jQuery.fn.extend(object)` untuk menambahkan fungsi yang nanti dapat digunakan oleh elemen. Nama file yang menyimpan baris-baris kode di bawah adalah `elemen.js`.

```
/// <reference path="jquery-1.3.2.js">

jQuery.fn.extend({
    cek: function() {
        return this.each(function() { this.checked = true; });
    },
    tidakcek: function() {
        return this.each(function() { this.checked = false; });
    }
});
```

Sintaks dari fungsi ini adalah seperti berikut :

```
jQuery.fn.extend({
    fungsi_1: function() {
        // isi fungsi
    },
});
```

```

        fungsi_2: function() {
            // isi fungsi
        },
        . . .
        fungsi_n: function() {
            // isi fungsi
        }
    );
}

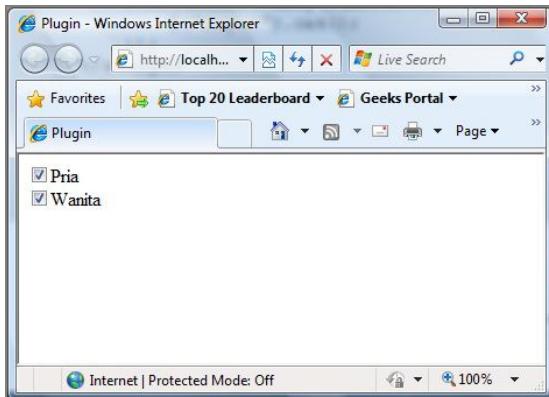
```

Dan berikut ini contoh penggunaan pada halaman web.

```

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Plugin</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script src="script/elemen.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        $(document).ready(function() {
            $("input").cek();
        });
    </script>
</head>
<body>
    <input type="checkbox" value="1" />Pria<br />
    <input type="checkbox" value="2" />Wanita<br />
</body>
</html>

```



Gambar 2-9 : Cek, Fungsi baru yang ditambahkan pada jQuery.

Fungsi yang dibuat ini hanya dapat digunakan pada elemen, sebagai contoh adalah elemen yang telah dipilih dengan menggunakan fungsi selector.

Interoperability

jQuery adalah library yang mampu bekerja sama dengan library JavaScript lainnya. Pada library-library JavaScript yang lain juga sangat sering menggunakan tanda \$. Agar tidak terjadi konflik dengan penggunaan tanda \$ pada jQuery, maka jQuery menyediakan fungsi untuk meminimalisir konflik yang terjadi.

Ada dua fungsi yang dapat digunakan, yaitu :

- a. `jQuery.noConflict()`.
- b. `jQuery.noConflict(extreme)`, parameter `extreme` bertipe `boolean`.

Apabila kemungkinan menggunakan jQuery dengan library JavaScript yang lain besar, maka sebaiknya gunakan fungsi `jQuery(...)` untuk menggantikan fungsi `$(...)`.

Selector

Fungsi selector adalah fungsi yang penting untuk mengambil elemen-elemen yang ingin dilakukan operasi kepadanya atau modifikasi terhadap elemen-elemen tersebut. fungsi selector yang ada pada jQuery termasuk yang sangat lengkap dan mudah untuk digunakan.

Berikut ini akan dijelaskan penggunaan fungsi selector secara lebih detail.

Basic

Elemen dapat dipilih atau dikoleksi oleh fungsi selector berdasarkan :

- Nama tag.
- Nama class yang terdapat pada elemen.
- Nama id yang digunakan oleh elemen.

Berikut ini adalah gambar dan penjelasan bagaimana fungsi selector bekerja untuk mengambil elemen dengan nama tag tertentu.



```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Selector</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        $("div")
    </script>
    <style type="text/css">
        .TxtLabel { font-weight:bold; }
    </style>
</head>
<body>
    <h1>Selector</h1>
    <form action="15.html" method="post">
        <div>
            <label class="TxtLabel">Username : </label> <input type="text" id="username" />
        </div>
        <div>
            <label class="TxtLabel">Password : </label> <input type="text" id="password" />
        </div>
        <input type="button" id="btn" />
    </form>
</body>
```

Gambar 2-10 : Selector seluruh elemen dengan tag div.

Gambar di atas adalah fungsi selector untuk memilih seluruh elemen dengan tag div. Bila ingin memilih seluruh elemen dengan nama tag input maka ganti parameter fungsi selector ini menjadi seperti berikut.



```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Selector</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        $("label")
    </script>
    <style type="text/css">
        .TxtLabel { font-weight:bold; }
    </style>
</head>
<body>
    <h1>Selector</h1>
    <form action="15.html" method="post">
        <div>
            <label class="TxtLabel">Username : </label> <input type="text" id="username" />
        </div>
        <div>
            <label class="TxtLabel">Password : </label> <input type="text" id="password" />
        </div>
        <input type="button" id="btn" />
    </form>
</body>
```

Gambar 2-11 : Selector seluruh elemen dengan tag label.

Berikut ini adalah contoh fungsi selector untuk mengambil elemen dengan nilai id tertentu.

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Selector</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        $("#btn")
    </script>
    <style type="text/css">
        .TxtLabel { font-weight:bold; }
    </style>
</head>
<body>
    <h1>Selector</h1>
    <form action="15.html" method="post">
        <div>
            <label class="TxtLabel">Username : </label> <input type="text" id="username" />
        </div>
        <div>
            <label class="TxtLabel">Password : </label> <input type="text" id="password" />
        </div>
        <input type="button" id="btn" />
    </form>
</body>
```

The code highlights the selector `$("#btn")` with a red box and a circled `#`.

Gambar 2-12 : Selector untuk memilih elemen dengan id=btn.

Untuk mengambil elemen dengan `id=btn`, maka pada fungsi selector tambahkan tanda `#` dengan nilai dari `id` tersebut, sehingga fungsi selector akan menjadi `$("#btn")`. Untuk mengambil elemen dengan `id=username` maka fungsi selector menjadi `$("#username")`.

Berikut ini contoh penggunaan fungsi selector untuk mengambil elemen dengan nilai `class` tertentu.

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Selector</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        $(".TxtLabel")
    </script>
    <style type="text/css">
        .TxtLabel { font-weight:bold; }
    </style>
</head>
<body>
    <h1>Selector</h1>
    <form action="15.html" method="post">
        <div>
            <label class="TxtLabel">Username : </label> <input type="text" id="username" />
        </div>
        <div>
            <label class="TxtLabel">Password : </label> <input type="text" id="password" />
        </div>
        <input type="button" id="btn" />
    </form>
</body>
```

The code highlights the selector `$(".TxtLabel")` with a red box and a circled `.`.

Gambar 2-13 : Selector untuk memilih elemen yang dengan atribut class=TxtLabel.

Bila elemen yang ingin dipilih dengan adalah elemen yang mempunyai atribut `class` dengan nilai tertentu maka nilai yang digunakan untuk mengisi parameter pada fungsi selector adalah tanda `.` (titik) ditambah nama `class`. Sehingga bila nama `class=TxtLabel` maka fungsi selector menjadi `$(".TxtLabel")`.

Fungsi selector dapat digunakan juga untuk memilih seluruh elemen yang ada dengan menggunakan cara seperti berikut ini :

```
$ ("")
```

Fungsi selector tidak hanya dapat menerima satu nilai parameter saja dapat dapat digunakan untuk lebih dari satu parameter. Berikut ini contoh fungsi selector yang menggunakan lebih dari satu parameter.

```
 $("div,#username,#password,label,.TxtLabel")
```

Pada contoh di atas, elemen yang akan dipilih adalah elemen dengan tag `div` dan tag `label`, elemen dengan `id` yang bernilai `username` dan `password` serta elemen dengan nama `class` `TxtLabel`.

Hierarchy

HTML atau pada perkembangannya menjadi XHTML, merupakan dokumen yang mempunyai hirarki seperti layaknya dokumen XML. Artinya setiap elemen akan dapat berada di dalam elemen lainnya sehingga membentuk suatu hirarki.

Berdasarkan hal tersebut maka fungsi selector yang dimiliki jQuery juga dapat mencari dan memilih elemen-elemen berdasarkan hirarki yang diinginkan.

Contoh berikut ini menunjukan cara memilih elemen yang berada di dalam elemen tertentu.

Misalkan dimiliki baris-baris kode HTML seperti berikut ini.

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Selector - Hierarchy</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        $(document).ready(function() {
            $("#Main > label").addClass("TxtLabel");
        });
    </script>
    <style type="text/css">
        .TxtLabel
        {
            text-transform:uppercase;
            font-weight:bold;
        }
    </style>
</head>
<body>
    <form action="15.html" method="post">
        <div id="Main">
            <label>No : </label> <input type="text" name="No" id="No" /><br />
            <label>Nama : </label> <input type="text" name="Nama" id="Nama" /><br />
            <label>Alamat : </label> <input type="text" name="Alamat" id="Alamat" /><br />
            <div>
                <label>Kode Post :</label>
                <input type="text" name="KodePost" id="KodePost" /><br />

                <label>Telp :</label><input type="text" name="Telp" id="Telp" /><br />
            </div>
            <input type="button" value="Submit" id="btn" />
        </div>
    </form>
</body>
</html>
```

Pada contoh di atas dapat dilihat akan dipilih semua elemen `label` yang berada tepat di bawah elemen dengan `id=Main`, maka cara penulisannya adalah seperti berikut ini.

```
 $("#Main > label")
```

Elemen dengan `id=Main` adalah `induk (parent)`, sedangkan elemen dengan tag `label` adalah `anak (child)`. Sehingga dapat dituliskan sintaks umum seperti berikut ini.

```
 $("parent > child")
```

Nilai untuk parent ataupun child dapat berupa nama tag, class atau nilai id dari suatu elemen.

Hasil dari contoh di atas akan dapat dilihat dari gambar berikut ini.



Gambar 2-14 : Hirarki parent-child.

Seperti yang dilihat pada gambar di atas, elemen `label` yang terpengaruh hanyalah elemen yang tepat berada di bawah elemen dengan nilai `id=Main`. Sedangkan elemen `label` yang tidak tepat di bawahnya tidak terpengaruh, hal itu bisa dilihat pada label untuk `Kode Post` dan `Telp`.

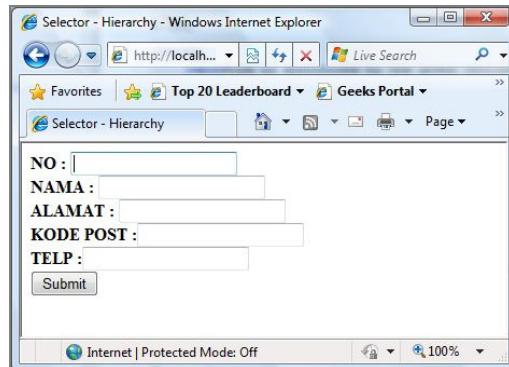
```
<script language="javascript" type="text/javascript">
$(document).ready(function() {
    $("#Main > label").addClass("TxtLabel");
})
</script>
<style type="text/css">
.TxtLabel
{
    text-transform:uppercase;
    font-weight:bold;
}
</style>
</head>
</body>
<form action="15.html" method="post">
<div id="Main">
<label>No : </label> <input type="text" name="No" id="No" /><br />
<label>Nama : </label> <input type="text" name="Nama" id="Nama" /><br />
<label>Alamat : </label> <input type="text" name="Alamat" id="Alamat" /><br />
<div>
    <label>Kode Post :</label><input type="text" name="KodePost" id="KodePost" /><br />
    <label>Telp :</label><input type="text" name="Telp" id="Telp" /><br />
</div>
<input type="button" value="Submit" id="btn" />
</div>
</form>
```

Gambar 2-15 : Label pada lingkaran biru tidak terpengaruh oleh fungsi selector.

Fungsi selector lain yang mempunyai fungsi mirip seperti contoh di atas mempunyai sintaks seperti berikut ini.

```
$(“ancestor descendant”)
```

Nilai parameter untuk `ancestor` dan `descendant` dapat berupa nama `tag`, `id` atau nama `class`. Dengan mengubah nilai `$("#Main > label")` menjadi `$("#Main label")`, maka dapat dilihat hasilnya seperti pada gambar berikut ini.



Gambar 2-16 : Berpengaruh pada semua label di bawah elemen dengan `id=Main`.

Dapat dilihat semua `label` yang berada di dalam elemen dengan `id=Main` terpilih oleh fungsi selector.

Berikut ini adalah contoh selector yang berfungsi untuk memilih elemen yang berada setelah elemen lain. Sintaks yang dapat digunakan adalah seperti berikut ini.

```
#("prev + next")
```

Paremeter `prev` berisi elemen sebelum elemen `next` yang akan dipilih.

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Selector - Hierarchy</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        $(document).ready(function() {
            $("label + input").addClass("InputForm");
        });
    </script>
    <style type="text/css">
        .TxtLabel
        {
            text-transform:uppercase;
            font-weight:bold;
        }

        .InputForm
        {
            border:solid 1px #000;
        }
    </style>
</head>
<body>
    <form action="15.html" method="post">
        <div id="Main">
            <label>No : </label> <input type="text" name="No" id="No" /><br />
            <label>Nama : </label> <input type="text" name="Nama" id="Nama" />
            <br />
            <label>Alamat : </label> <input type="text" name="Alamat" id="Alamat" />
            <br />
            <div>
                <label>Kode Post :</label>
                <input type="text" name="KodePost" id="KodePost" /><br />
                <label>Telp :</label><input type="text" name="Telp" id="Telp" />
                <br />
            </div>
            <input type="button" value="Submit" id="btn" />
        </div>
    </form>
```

```
</body>  
</html>
```

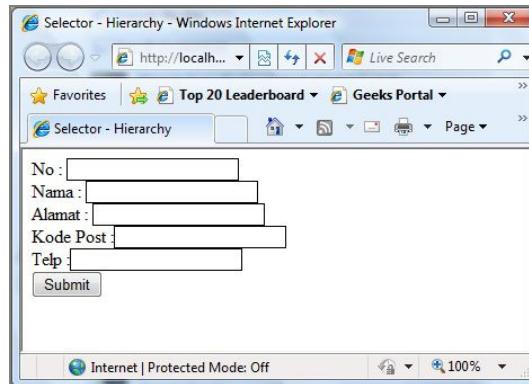
Dengan menggunakan fungsi selector `$(".label + input")`, maka hanya akan dipilih elemen `input` yang berada **tepat** setelah elemen `label`, sehingga hanya elemen `input` pada lingkaran merah saja yang terpilih. Sedangkan elemen `input` yang berada pada tidak tepat setelah elemen `label` tidak dipilih, seperti yang terlihat pada lingkaran biru.

seperti yang dapat dilihat pada gambar di bawah ini.

```
<script language="javascript" type="text/javascript">  
$(document).ready(function() {  
    $(".label + input").addClass("InputForm");  
});  
</script>  
<style type="text/css">  
.InputForm  
{  
    border:solid 1px #000;  
}  
</style>  
</head>  
<body>  
    <form action="15.html" method="post">  
        <div id="Main">  
            <label>No : </label> <input type="text" name="No" id="No" /><br />  
            <label>Nama : </label> <input type="text" name="Nama" id="Nama" /><br />  
            <label>Alamat : </label> <input type="text" name="Alamat" id="Alamat" /><br />  
            <div>  
                <label>Kode Post : </label> <input type="text" name="KodePost" id="KodePost" /><br />  
                <label>Telp : </label> <input type="text" name="Telp" id="Telp" /><br />  
            </div>  
            <input type="button" value="Submit" id="btn" />  
        </div>  
    </form>  
</body>
```

Gambar 2-17 : Elemen `input` pada lingkaran merah dipilih, elemen `input` pada lingkaran biru tidak dipilih.

Dan berikut ini adalah gambar dari output baris-baris kode di atas.



Gambar 2-18 : Elemen `input` setelah elemen `label`.

Contoh terakhir pada bagian ini adalah fungsi selector yang berfungsi mirip seperti `("prev + next")`, yaitu `("prev ~ siblings")`. Selector `("prev ~ siblings")` berfungsi untuk memilih seluruh elemen `siblings` setelah elemen `prev`, dan elemen `siblings` yang dipilih tidak mesti tepat setelah elemen `prev`.

Apabila contoh fungsi `("prev + next")` dimodifikasi seperti berikut ini.

```
$(".label ~ input")
```

Maka pengaruhnya dapat dilihat pada gambar di bawah ini.

```
<script language="javascript" type="text/javascript">
$(document).ready(function() {
    $("label ~ input").addClass("InputForm");
});
</script>
<style type="text/css">
.InputForm
{
    border:solid 1px #000;
}
</style>
</head>
<body>
<form action="15.html" method="post">
<div id="Main">
<label>No : </label><input type="text" name="No" id="No" /><br />
<label>>Nama : </label><input type="text" name="Nama" id="Nama" /><br />
<label>Alamat : </label><input type="text" name="Alamat" id="Alamat" /><br />
<div>
<label>Kode Pos :</label><input type="text" name="KodePost" id="KodePost" /><br />
<label>Telp :</label><input type="text" name="Telp" id="Telp" /><br />
</div>
<input type="button" value="Submit" id="btn" />
</div>
</form>
</body>
```

Gambar 2-19 : Fungsi selector `$("prev ~ siblings")`.

Dapat dilihat pada gambar, fungsi ini tidak hanya berpengaruh pada elemen `input` yang **tepat** setelah elemen `label` (nomor 1), tapi juga berpengaruh pada elemen `input` setelah elemen `label`, walaupun **tidak tepat** setelahnya (nomor 2).

Basic Filter

Pada bagian ini akan memperlihatkan fitur filter pada fungsi selector. Berikut ini adalah sintaks fitur filter pada fungsi selector.

```
$(“element:filter”)
```

Parameter `element` dapat berisi nama `tag`, `class` atau nilai `id`. Parameter `filter` berisi keyword yang berfungsi sebagai aturan filter. Sedangkan bila ingin memilih seluruh elemen yang ada dapat digunakan sintaks sebagai berikut.

```
$(“:filter”)
```

Selanjutnya akan diperkenalkan keyword-keyword filter yang dapat digunakan.

:animated

Keyword `:animated` berfungsi untuk memfilter elemen yang sedang *beranimasi*. Berikut ini contoh penggunaan filter ini.

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Selector - Basic Filter</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        $(document).ready(function() {
            $("#run").click(function() {
                $("div:animated").toggleClass("colored");
            });
        });

        function animateIt() {
            $("#mover").slideToggle("slow", animateIt);
        }

        animateIt();
    });
</script>
<style type="text/css">
#mover
{
    width:100px;
    height:100px;
}
```

```

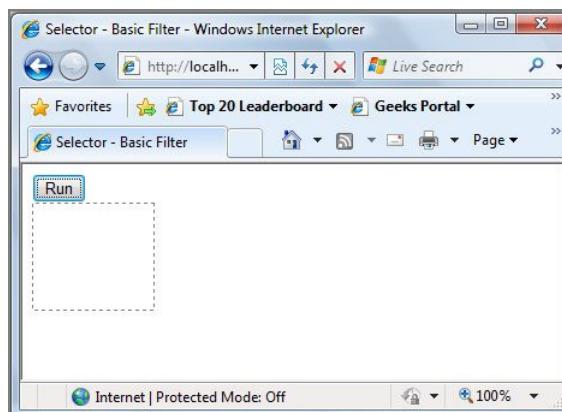
        border:dashed 1px Gray;
    }

.colored
{
    background-color:Lime;
}
</style>
</head>
<body>
    <button id="run">Run</button>

    <div></div>
    <div id="mover"></div>
    <div></div>
</body>
</html>

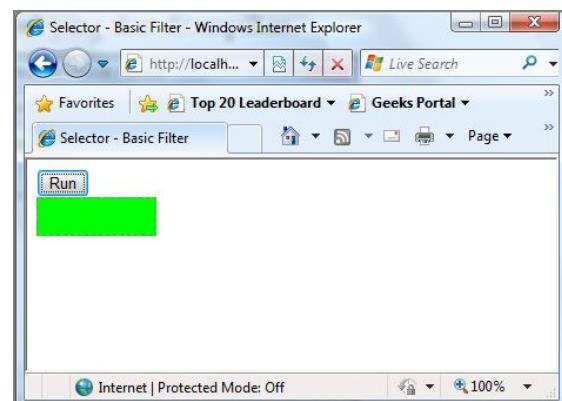
```

Dan berikut adalah gambar halaman web yang akan ditampilkan.



Gambar 2-20 : Animasi kotak dengan fungsi slideToggle.

Animasi kotak di atas berupa gerak membuka dan menutup. Ketika tombol Run ditekan maka background dari kotak tersebut menjadi warna hijau seperti pada gambar di bawah ini.



Gambar 2-21 : Background kotak menjadi hijau setelah tombol Run diklik.

Kotak tersebut dibuat dengan menggunakan tag `div`, dan berikut ini bagaimana cara memfilter elemen `div` yang sedang beranimasi.

```
$( "div:animated" )
```

:header

Filter ini berfungsi untuk memilih semua tag heading seperti h1, h2, h3 dan seterusnya.

Berikut ini adalah contoh penggunaan filter ini.

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Selector - Basic Filter</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        $(document).ready(function() {
            $(":header").css({ background: '#CCC', color: 'blue' });
        });
    </script>
</head>
<body>
    <h1>Header 1</h1>

    <p>Contents 1</p>
    <h2>Header 2</h2>
    <p>Contents 2</p>
</body>
</html>
```

Gambar berikut ini adalah hasil dari contoh di atas.



Gambar 2-22 : Filter header.

:not

Sintaks dari filter ini adalah sebagai berikut.

```
$(":not(selector)")
```

Parameter selector di atas akan berisi elemen yang tidak akan dipilih. Contoh penggunaanya adalah sebagai berikut.

```
$(":not(p)")
```

Dan berikut ini contoh lain penggunaan filter ini.

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Selector - Basic Filter</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        $(document).ready(function() {

            $("body > :not(p)").css({ background: '#CCC', color: 'blue' });
        });
    </script>
</head>
<body>
```

```

</script>
</head>
<body>
    <h1>Header 1</h1>

    <p>Contents 1</p>
    <h2>Header 2</h2>
    <p>Contents 2</p>
</body>
</html>

```

Dari contoh di atas dapat dilihat fungsi selector `$("body > :not(p)")` akan memilih semua elemen di dalam tag `body` selain tag `p`. Sehingga nanti yang dipilih oleh fungsi selector ini adalah elemen dengan nama tag `h1` dan `h2`.

:first

Filter ini berfungsi untuk memilih elemen pertama dari koleksi elemen yang ada. Sintaks dari filter ini adalah sebagai berikut.

```
$("element:first")
```

Parameter element berisi nama `tag`, `class` atau nilai suatu `id` pada elemen.

The screenshot shows a Microsoft Internet Explorer window with the title "Selector - Basic Filter - Windows Internet Explorer". The address bar shows "http://localhost...". The main content area displays a table with the following data:

No.	Nama	Alamat
1.	Nama #1	Alamat #1
2.	Nama #2	Alamat #2
3.	Nama #3	Alamat #3
4.	Nama #4	Alamat #4
5.	Nama #5	Alamat #5

Below the table, there are navigation links labeled "1 2 3".

Gambar 2-23 : Tabel tanpa style.

Apabila ingin menambahkan style pada header tabel tersebut maka filter `:first` dapat digunakan. Filter `:first` digunakan untuk mengambil elemen `tr` (baris) pertama pada tabel, kemudian dapat dilakukan operasi tertentu yang diinginkan, termasuk memberikan style sehingga akan dapat dilihat hasil seperti berikut ini.

The screenshot shows the same Microsoft Internet Explorer window as before, but now the header row of the table has a different style applied. The header row is bolded and has a black background. The data rows remain the same as in the previous screenshot.

NO.	NAMA	ALAMAT
1.	Nama #1	Alamat #1
2.	Nama #2	Alamat #2
3.	Nama #3	Alamat #3
4.	Nama #4	Alamat #4
5.	Nama #5	Alamat #5

Below the table, there are navigation links labeled "1 2 3".

Gambar 2-24 : Style pada header tabel.

Untuk mengambil elemen pertama dari elemen `tr` yang ada maka digunakan baris berikut ini.

```
$( "tr:first" )
```

Berikut ini adalah baris kode secara lengkap dari kasus di atas.

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Selector - Basic Filter</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        $(document).ready(function() {
            $("tr:first").addClass("Header");
        });
    </script>
    <style type="text/css">
        body
        {
            font-family:Verdana;
            font-size:12px;
        }

        .Header
        {
            background-color:Black;
            color:White;
            font-weight:bold;
            text-align:center;
            text-transform:uppercase;
        }
    </style>
</head>
<body>
    <table border="1" cellpadding="0" cellspacing="0" width="100%">
        <tr>
            <td>No.</td>
            <td>Nama</td>
            <td>Alamat</td>
        </tr>
        <tr>
            <td>1.</td>
            <td>Nama #1</td>
            <td>Alamat #1</td>
        </tr>
        <tr>
            <td>2.</td>
            <td>Nama #2</td>
            <td>Alamat #2</td>
        </tr>
        <tr>
            <td>3.</td>
            <td>Nama #3</td>
            <td>Alamat #3</td>
        </tr>
        <tr>
            <td>4.</td>
            <td>Nama #4</td>
            <td>Alamat #4</td>
        </tr>
        <tr>
            <td>5.</td>
            <td>Nama #5</td>
            <td>Alamat #5</td>
        </tr>
        <tr>
            <td colspan="3">1 2 3</td>
        </tr>
    </table>
</body>
</html>
```

:last

Filter ini akan memilih elemen terakhir, dan berikut ini adalah sintaks penggunaan filter ini.

```
$(“element:last”)
```

Untuk memilih elemen `tr` terakhir dan menambahkan class `Footer` ke dalamnya maka digunakan baris berikut ini.

```
<script language="javascript" type="text/javascript">
$(document).ready(function() {
    .
    .
    $("tr:last").addClass("Footer");
});
</script>
```

Sedangkan style dari class `Footer` adalah sebagai berikut.

```
<style type="text/css">
.Footer
{
    text-align:center;
    font-weight:bold;
    background-color:Gray;
}
</style>
```

Dan hasilnya dapat dilihat pada gambar di bawah ini.

NO.	NAMA	ALAMAT
1.	Nama #1	Alamat #1
2.	Nama #2	Alamat #2
3.	Nama #3	Alamat #3
4.	Nama #4	Alamat #4
5.	Nama #5	Alamat #5

Gambar 2-25 : Style dengan jQuery pada elemen `tr` terakhir.

:even & :odd

Filter `:even` akan memilih elemen dengan nilai index yang genap, sedangkan `:odd` akan memilih elemen dengan nilai index yang ganjil.

Bila dimiliki style sebagai berikut.

```
<style type="text/css">
.EvenRow
{
    background-color:Red;
    color:White;
}

.OddRow
{
    background-color:Yellow;
    color:Black;
}
</style>
```

Dan ingin diberikan style class EvenRow untuk setiap baris yang genap, dan style class OddRow untuk setiap baris yang ganjil, maka perlu ditambahkan baris berikut ini.

```
$(“tr:even”).addClass(“EvenRow”);  
$(“tr:odd”).addClass(“OddRow”);
```

Dan hasilnya dapat dilihat pada gambar berikut ini.

A screenshot of a Windows Internet Explorer window titled "Selector - Basic Filter". The address bar shows "http://localhost...". The page content displays a table with three columns: NO., NAMA, and ALAMAT. The table has 5 rows, indexed 1 to 5. Rows 1, 3, and 5 are colored yellow (EvenRow), while rows 2 and 4 are colored red (OddRow). The table is styled with a border and centered text. Below the table, there is a navigation bar with links labeled 1, 2, and 3.

NO.	NAMA	ALAMAT
1.	Nama #1	Alamat #1
2.	Nama #2	Alamat #2
3.	Nama #3	Alamat #3
4.	Nama #4	Alamat #4
5.	Nama #5	Alamat #5

Gambar 2-26 : Style pada baris genap dan ganjil.

:lt & :gt

Sintaks pemakaian kedua filter ini adalah sebagai berikut.

```
$(“element:lt(index)”)  
$(“element:gt(index)”)
```

Filter `:lt` berfungsi untuk memilih elemen dengan index lebih kecil dari nilai `index` yang diisikan. Sedangkan filter `:gt` berfungsi untuk memilih elemen dengan index lebih besar dari nilai `index` yang diisikan.

```
$(“element:lt(3)”)  
$(“element:gt(5)”)
```

Filter `$(“element:lt(3)”)` akan memilih elemen dengan `index` di bawah 3, yaitu 2, 1 dan 0. Sedangkan `$(“element:gt(5)”)` akan memilih elemen dengan `index` lebih besar dari 5, yaitu 6,7 dan seterusnya.

:eq(index)

Filter ini berfungsi untuk mengambil elemen dengan nilai index tertentu. Berikut adalah sintaks penggunaan filter ini.

```
$(“element:eq(index)”)
```

Fungsi filter ini sama dengan fungsi `eq(position)` yang dapat dilihat pada bagian jQuery Object Accessors.

Content Filter

Pada bagian ini akan dipaparkan filter yang dapat digunakan pencarian text atau elemen di dalam suatu elemen.

:contain(text)

Filter ini berfungsi untuk mencari text pada elemen yang diinginkan.

```
$( "div:contains('jQuery')")
```

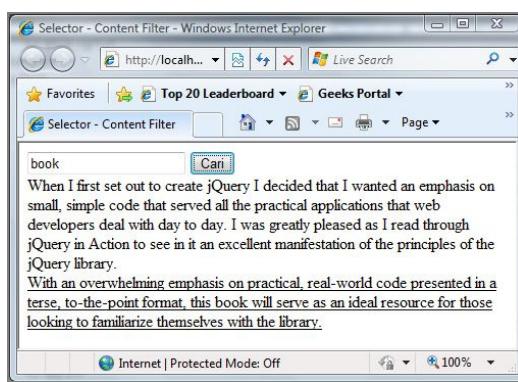
Contoh di atas akan mencari kata “jQuery” di dalam setiap elemen `div` yang ada pada halaman web. Berikut ini contoh lengkap penggunaan filter ini.

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Selector - Content Filter</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        function cari() {
            var katakunci = $("#katakunci").val();
            $("div").css("text-decoration", "none");
            $("div:contains('" + katakunci + "')").css("text-decoration", "underline");
        }
    </script>
</head>
<body>
    <input type="text" id="katakunci" name="katakunci" />
    <input type="button" value="Cari" onclick="cari()" />

    <div>
        When I first set out to create jQuery I decided that I wanted an emphasis on
        small, simple code that served all the practical applications that web
        developers deal with day to day. I was greatly pleased as I read through
        jQuery in Action to
        see in it an excellent manifestation of the principles of the jQuery library.
    </div>

    <div>
        With an overwhelming emphasis on practical, real-world code presented in
        a terse, to-the-point format, this book will serve as an ideal resource for
        those looking to familiarize themselves with the library.
    </div>
</body>
</html>
```

Program di atas akan mencari text pada elemen `div` sesuai dengan nilai yang diisikan pada textbox. Apabila text yang dicari ditemukan di dalam elemen `div`, maka elemen tersebut akan diberi garis bawah seperti yang dilihat pada gambar berikut ini.



Gambar 2-27 : Elemen `div` dengan text `book` didalamnya digarisbawahi.

:empty

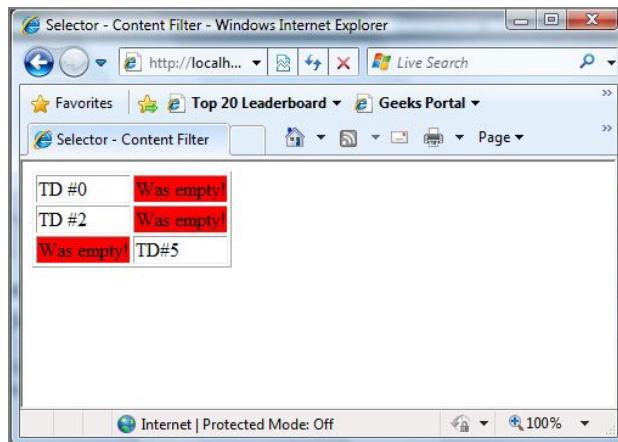
Filter ini berfungsi untuk mencari elemen yang tidak memiliki elemen anak atau text. Filter ini mempunyai sintaks sebagai berikut.

```
$(“element:empty”)
```

Paramenter elemen akan berisi elemen yang ingin yang akan diperiksa. Berikut ini contoh lengkap penggunaan filter ini.

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Selector - Content Filter</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        $(document).ready(function() {
            $("td:empty").text("Was empty!").css('background', 'red');
        });
    </script>
</head>
<body>
    <table border="1">
        <tr>
            <td>
                TD #0
            </td>
            <td>
            </td>
            <td>
            </td>
        </tr>
        <tr>
            <td>
                TD #2
            </td>
            <td>
            </td>
            <td>
            </td>
        </tr>
        <tr>
            <td>
            </td>
            <td>
            </td>
            <td>
                TD#5
            </td>
        </tr>
    </table>
</body>
</html>
```

Hasil contoh di atas dapat dilihat pada gambar di bawah ini.



Gambar 2-28 : Elemen kosong diisi dengan text Was empty!.

:has(selector)

Sintaks filter ini adalah sebagai berikut.

```
$(“element1:has(element2)”)
```

Filter ini berfungsi untuk memilih setiap `element1` yang di dalamnya terdapat `element2`.

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Selector - Content Filter</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        $(document).ready(function() {
            $("div:has(p)").css("text-decoration", "underline");
        });
    </script>
</head>
<body>
    <div><p>Hello in a paragraph</p></div>

    <div>Hello again! (with no paragraph)</div>
</body>
</html>
```

Pada contoh di atas akan dipilih semua elemen `div` yang di dalamnya terdapat elemen `p`.

:parent

Filter ini mempunyai fungsi yang sebaliknya bila dibandingkan dengan filter `:empty`.
Sintaks penggunaan filter ini adalah sebagai berikut.

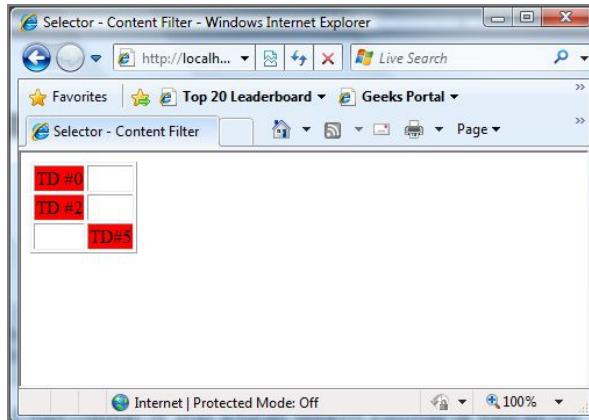
```
$(“element:parent”)
```

Dengan filter ini dapat dipilih semua elemen yang didalamnya terdapat elemen lain atau text.

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Selector - Content Filter</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        $(document).ready(function() {
            $("td:parent").css('background', 'red');
        });
    </script>
</head>
<body>
    <table border="1">
        <tr>
            <td>
                TD #0
            </td>
            <td>
            </td>
        </tr>
        <tr>
            <td>
                TD #2
            </td>
            <td>
            </td>
        </tr>
        <tr>
            <td>
                TD#5
            </td>
            <td>
            </td>
        </tr>
    </table>
</body>
</html>
```

```
</table>
</body>
</html>
```

Hasil dari contoh di atas adalah seperti gambar di bawah ini.



Gambar 2-29 : Elemen yang mempunyai isi elemen atau text akan diwarnai merah.

Visibility Filter

Status visibility terdiri atas dua yaitu hidden dan visible. Fungsi selector dapat memilih elemen-elemen berdasarkan status visibility.

:hidden

Sintaks pemakaian dari filter ini adalah sebagai berikut.

```
$(“element:hidden”)
```

:visible

Sintaks pemakaian dari filter ini adalah sebagai berikut.

```
$(“element:visible”)
```

Contoh berikut ini adalah contoh penggunaan filter :hidden dan :visible.

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Selector - Visibility Filter</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        $(document).ready(function() {
            });

            function Show() {
                $("div:hidden").show(3000);
            }

            function Hide() {
                $("div:visible").hide(3000);
            }
        </script>
        <style type="text/css">
        .box
        {
            border:dashed 1px #000;
            width:113px;
        }
    </style>
</head>
<body>
    <table border="1">
        <tr>
            <td>1</td>
            <td>2</td>
            <td>3</td>
        </tr>
        <tr>
            <td>4</td>
            <td>5</td>
            <td>6</td>
        </tr>
        <tr>
            <td>7</td>
            <td>8</td>
            <td>9</td>
        </tr>
    </table>
</body>
</html>
```

```

        height:113px;
        margin-right:23px;
        float:left;
    }

```

```
</head>
<body>
```

```
<div class="box" style="display:none">
```

```
</div>
```

```
<div class="box">
```

```
</div>
```

```
<br style="clear:both;" />
```

```
<input type="button" value="Show" onclick="Show()" />
```

```
<input type="button" value="Hide" onclick="Hide()" />
```

```
</body>
```

```
</html>
```

Pada fungsi `Show` dapat dilihat fungsi selector yang akan memilih elemen `div` dengan status `hide`, sedangkan pada fungsi `Hide` terdapat fungsi selector yang akan memilih elemen `div` dengan status `visible`.

Attribute Filter

Fungsi selector dapat mencari suatu elemen berdasarkan nilai atribut yang dimiliki oleh elemen tersebut.

[attribute*=value]

Filter ini dapat digunakan dengan sintaks berikut.

```
$(“element[attribute*='value']”)
```

Sebagai contoh adalah seperti berikut ini.

```
$(“input[name*='man']”)
```

Dari contoh di atas maka fungsi selector akan memilih semua elemen `input` dengan nilai atribut `name` berisi (contain) kata *man*. Bila dimiliki elemen-elemen berikut ini pada halaman web.

```

<input name="man-news" />
<input name="milkman" />
<input name="letterman2" />
<input name="newmilk" />

```

Fungsi selector `$(“input[name*='man']”)` akan memilih elemen pertama, kedua dan ketiga. Elemen `input` keempat tidak terpilih karena nilai atribut `name` tidak terdapat kata *man*.

[attribute=value]

Filter ini dapat digunakan dengan sintaks berikut.

```
$(“element[attribute='value']”)
```

Sebagai contoh adalah seperti berikut ini.

```
$(“input[name='man']”)
```

Dari contoh di atas maka fungsi selector akan memilih semua elemen `input` dengan nilai atribut `name` mempunyai nilai tepat sama dengan kata *man*. Bila dimiliki elemen-elemen berikut ini pada halaman web.

```
<input name="man-news" />
<input name="milkman" />
<input name="letterman2" />
<input name="man" />
```

Fungsi selector hanya akan memilih elemen input yang keempat, walaupun elemen pertama sampai dengan ketiga mempunyai kata man sebagai nilai dari atribut name tetapi tidak tepat sama.

[attribute[^]=value]

Filter ini untuk memilih elemen yang mempunyai nilai atribut dengan awalan kata sesuai dengan nilai value yang diisikan.

Filter ini dapat digunakan dengan sintaks berikut.

```
$(“element [attribute^=‘value’]”)
```

Sebagai contoh adalah seperti berikut ini.

```
$(“input [name^=‘man’]”)
```

Dari contoh di atas maka fungsi selector akan memilih semua elemen `input` dengan nilai atribut `name` mempunyai nilai awal sama dengan kata *man*. Bila dimiliki elemen-elemen berikut ini pada halaman web.

```
<input name="man-news" />
<input name="milkman" />
<input name="letterman2" />
<input name="man" />
```

Maka elemen yang akan dipilih dengan menggunakan filter ini adalah elemen pertama dan terakhir.

[attribute\$=value]

Filter ini untuk memilih elemen yang mempunyai nilai atribut dengan akhiran kata sesuai dengan nilai value yang diisikan.

Filter ini dapat digunakan dengan sintaks berikut.

```
$(“element [attribute$=‘value’]”)
```

Sebagai contoh adalah seperti berikut ini.

```
$(“input [name$=‘man’]”)
```

Dari contoh di atas maka fungsi selector akan memilih semua elemen `input` dengan nilai atribut `name` mempunyai nilai akhir sama dengan kata *man*. Bila dimiliki elemen-elemen berikut ini pada halaman web.

```
<input name="man-news" />
<input name="milkman" />
<input name="letterman2" />
<input name="man" />
```

Maka elemen yang akan dipilih dengan menggunakan filter ini adalah elemen kedua.

[attribute]

Sintaks penggunaan filter ini pada fungsi selector adalah sebagai berikut.

```
$(“element[attribute]”)
```

Fungsi selector yang menggunakan filter ini akan memilih semua elemen yang mempunyai atribut tertentu sesuai nilai `attribute` yang diberikan pada sintaks di atas. Berikut ini contoh penggunaannya.

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Selector - Attribute Filter</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        $(document).ready(function() {
            $("div[id]").one("click", function() {
                var idString = $(this).text() + " = " + $(this).attr("id");
                $(this).text(idString);
            });
        });
    </script>
</head>
<body>
    <div>no id</div>
    <div id="hey">with id</div>
    <div id="there">has an id</div>
    <div>nope</div>
</body>
</html>
```

Fungsi selector pada contoh di atas akan memilih elemen `div` yang mempunyai attribut `id`.

[selector1][selector2][selectorN]

Fungsi selector dapat memilih elemen dengan filter beberapa attribut sekaligus. Berikut ini sintaks yang dapat digunakan untuk keperluan ini.

```
$(“element[attribute1][attribute2][attribute3]...[attributeN]”)
```

Berikut ini penggunaannya.

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Selector - Attribute Filter</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        $(document).ready(function() {
            $("input[id][name$='man']").val("only this one");
        });
    </script>
</head>
<body>
    <input id="man-news" name="man-news" />

    <input name="milkman" />
    <input id="letterman" name="new-letterman" />
    <input name="newmilk" />
</body>
</html>
```

Dapat dilihat fungsi selector seperti berikut ini.

```
$(“input[id][name$='man']”)
```

Fungsi selector seperti di atas akan memilih semua elemen `input` yang mempunyai atribut `id` dan `name` dengan nilai berakhiran `man`.

[attribute!=value]

Fungsi filter ini adalah mencari elemen yang tidak mempunyai nilai pada atribut tertentu. Sebagai contoh dapat dilihat pada kode di bawah ini.

```
$( "input[name!='newsletter']" )
```

Contoh di atas akan memilih semua elemen input yang memiliki atribut `name` dengan nilai bukan `newsletter`.

Child Filter

Seperti diketahui halaman html atau xhtml adalah susunan elemen-elemen yang berbentuk hirarki seperti layaknya dokumen XML.

Filter-filter di bawah ini digunakan untuk memilih elemen-elemen yang menjadi anak dari suatu elemen.

:first-child

Filter ini digunakan untuk memilih elemen anak yang pertama dari suatu elemen induk yang dipilih. Berikut adalah sintaks dari penggunaan filter ini.

```
$( "parent-elemen child-element:first-child" )
```

Berikut ini adalah kumpulan elemen yang membentuk hirarki.

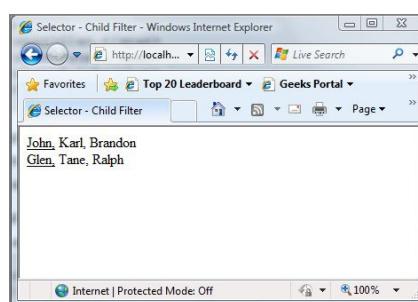
```
<div>
  <span>John,</span>
  <span>Karl,</span>
  <span>Brandon</span>
</div>
<div>
  <span>Glen,</span>
  <span>Tane,</span>
  <span>Ralph</span>
</div>
```

Dengan filter `:first-child` dapat dipilih elemen pertama yang berada di dalam suatu elemen induk. Pada contoh di atas, elemen `div` merupakan induk, dan elemen-elemen `span` merupakan elemen anak.

Berikut ini contoh fungsi selector yang akan memilih elemen `span` pertama yang berada di dalam elemen `div`.

```
$(document).ready(function() {
  $("div span:first-child").css("text-decoration", "underline");
});
```

Hasil dari baris di atas adalah seperti yang dilihat pada gambar.



Gambar 2-30 : setiap elemen span pertama diberi garis bawah.

:last-child

Filter `:last-child` berfungsi hampir sama dengan filter `:first-child` kecuali elemen yang dipilih adalah elemen anak terakhir. Bila contoh pada penggunaan filter `:first-child` di atas diganti menjadi seperti ini.

```
$("parent-elemen child-element:last-child")
```

Maka hasil dari contoh ini dapat dilihat pada gambar berikut.



Gambar 2-31 : setiap elemen span terakhir diberi garis bawah.

:nth-child

Filter ini berfungsi untuk memilih elemen anak dengan index tertentu. Sebagai contoh dapat dilihat dibawah ini.

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Selector - Child Filter</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        $(document).ready(function() {
            $("div span:nth-child(3)").css("text-decoration", "underline");
        });
    </script>
</head>
<body>
    <div>
        <span>John,</span>
        <span>Karl,</span>
        <span>Brandon,</span>
        <span>Brad,</span>
        <span>Jimmy</span>
    </div>
    <div>
        <span>Glen,</span>
        <span>Tane,</span>
        <span>Ralph</span>
        <span>Tine,</span>
        <span>Laura</span>
    </div>
</body>
</html>
```

Pada contoh di atas dapat dilihat baris berikut ini.

```
$("div span:nth-child(3)").css("text-decoration", "underline");
```

Baris ini akan memilih setiap elemen `span` ketiga yang berada di dalam elemen `div`. dengan filter ini dapat dipilih setiap elemen anak dengan index tertentu. Parameter pada filter `:nth-child` selain berisi angka, juga dapat berisi nilai *odd* atau *even*. Bila nilai parameter ini berisi

odd maka akan dipilih semua elemen anak yang ganjil, sedangkan bila parameter berisi *even* maka akan dipilih semua elemen anak yang genap.

Berikut contoh fungsi selector yang akan memilih semua elemen anak yang genap.

```
$( "div span:nth-child(even)" )
```

Dan hasilnya dapat dilihat pada gambar berikut ini.



Gambar 2-32 : garis bawah pada elemen anak yang genap.

:only-child

Untuk mengetahui fungsi dari filter ini dapat dengan melihat kode dan hasil dari kode berikut ini.

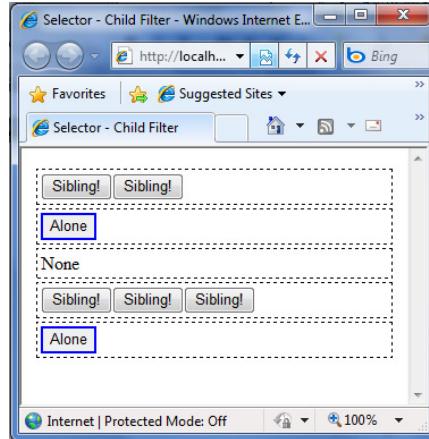
```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Selector - Child Filter</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        $(document).ready(function() {
            $("div button:only-child").text("Alone").css("border", "2px blue solid");
        });
    </script>
    <style type="text/css">
        div
        {
            clear:both;
            padding:3px;
            margin:3px;
            border:dashed 1px #000;
        }
    </style>
</head>
<body>
    <div>
        <button>Sibling!</button>
        <button>Sibling!</button>
    </div>

    <div>
        <button>Sibling!</button>
    </div>
    <div>
        None
    </div>

    <div>
        <button>Sibling!</button>
        <button>Sibling!</button>
        <button>Sibling!</button>
    </div>
    <div>
        <button>Sibling!</button>
    </div>
</body>
```

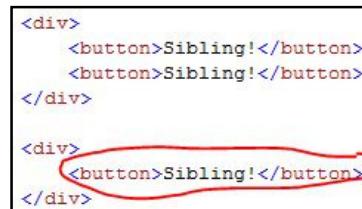
```
</div>
</body>
</html>
```

Hasilnya dapat dilihat pada gambar berikut ini.



Gambar 2-33 : Filter untuk memilih elemen yang hanya mempunyai satu elemen di dalamnya.

Dari contoh di atas, dapat dilihat bahwa filter ini hanya akan memilih elemen anak yang sendirian. Artinya elemen anak yang mempunyai teman tidak akan dipilih.



Gambar 2-34 : contoh elemen div yang hanya mempunyai satu elemen anak.

Dari contoh kode di atas, elemen yang akan dipilih hanya elemen `button` yang sendirian, yaitu elemen `button` yang di dalam elemen `div` kedua.

Form

Filter pada kelompok ini berfungsi untuk memilih elemen-elemen form, seperti `button`, `radiobutton`, `textarea` dan lain-lain.

Berikut ini filter-filter yang dapat digunakan untuk memilih elemen-elemen form :

- a. `:input`, filter ini untuk memilih semua elemen `input` (`<input />`).
- b. `:button`, filter ini untuk memilih semua elemen `input` tipe `button` (`<input type="button" />` atau `<button>tombol</button>`).
- c. `:checkbox`, filter ini untuk memilih semua elemen `input` tipe `checkbox` (`<input type="checkbox" />`).
- d. `:file`, filter ini untuk memilih semua elemen `input` tipe `file` (`<input type="file" />`).
- e. `:hidden`, filter ini untuk memilih semua elemen `input` tipe `hidden` (`<input type="hidden" />`).
- f. `:image`, filter ini untuk memilih semua elemen `input` tipe `image` (`<input type="image" />`).

- g. : password, filter ini untuk memilih semua elemen input tipe password (<input type="password" />).
- h. : radio, filter ini untuk memilih semua elemen input tipe radio (<input type="radio" />).
- i. : reset, filter ini untuk memilih semua elemen input tipe reset (<input type="reset" />).
- j. : submit, filter ini untuk memilih semua elemen input tipe submit (<input type="submit" />).
- k. : text, filter ini untuk memilih semua elemen input tipe text (<input type="text" />).

Berikut ini contoh penggunaan filter-filter yang telah disebutkan di atas.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Selector - Form Filter</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        $(document).ready(function() {
            // filter input
            var allInputs = $(":input");
            var formChildren = $("form > *");
            $("#jumlahInput").text("Found " + allInputs.length +
                " inputs and the form has " +
                formChildren.length + " children.");
            .css("color", "red");

            // filter button
            $(":button").css({ background: "yellow", border: "3px red solid" });

            // filter checkbox
            ":checkbox").css({ background: "yellow", border: "3px red solid" });

            // filter file
            ":file").css({ background: "yellow", border: "3px red solid" });

            // filter hidden
            $("#jumlahHidden").text("Found " + $("input:hidden").length +
                " hidden inputs.").css("color", "red");

            // filter image
            ":image").css({ background: "yellow", border: "3px red solid" });

            // filter password
            ":password").css({ background: "yellow", border: "3px red solid" });

            // filter radio
            ":radio").css({ background: "yellow", border: "3px red solid" });

            // filter reset
            ":reset").css({ background: "yellow", border: "3px red solid" });

            //filter submit
            ":submit").css({ background: "yellow", border: "3px red solid" });

            // filter text
            ":text").css({ background: "yellow", border: "3px red solid" });
        });
    </script>
</head>
<body>
<form>
    <input type="button" value="Input Button"/>
    <input type="checkbox" />

    <input type="file" />
    <input type="hidden" />

```

```

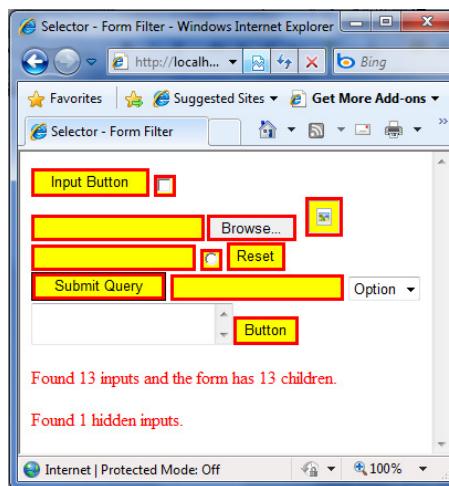
<input type="image" />
<input type="password" />
<input type="radio" />
<input type="reset" />

<input type="submit" />
<input type="text" />
<select><option>Option</option></select>

<textarea></textarea>
<button>Button</button>
</form>
<div id="jumlahInput"></div>
<br />
<div id="jumlahHidden"></div>
</body>
</html>

```

Dan berikut adalah hasil dari contoh kode di atas.



Gambar 2-35 : Memilih elemen form dengan filter form.

Form Filter

Elemen-elemen form mempunyai attribut yang menjadi status akses dari elemen-elemen tersebut seperti :

- a. `checked`, status ini dapat terjadi pada elemen `checkbox` atau `radiobutton` ketika dicentang.
- b. `selected`, status ini dapat terjadi pada elemen `select` ketika salah satu opsinya dipilih.
- c. `disabled`, status elemen tidak aktif dan tidak dapat dioperasikan.
- d. `enabled`, status elemen aktif dan dapat digunakan.

Fungsi selector mempunyai filter yang dapat digunakan untuk memilih elemen berdasarkan status akses dari elemen-elemen tersebut. Berikut ini adalah filter yang dapat digunakan untuk keperluan ini, yaitu :

- a. `:checked`, filter untuk memilih elemen `form` yang dicentang.
- b. `:selected`, filter untuk memilih elemen `form` yang dipilih.
- c. `:disabled`, filter untuk memilih elemen `form` yang mempunyai status `disabled`.
- d. `:enabled`, filter untuk memilih elemen `form` yang mempunyai status `enabled`.

Berikut ini contoh penggunaan form filter.

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Selector - Form Filter</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        $(document).ready(function() {
            // filter disabled
            $("input:disabled").val("Disabled");

            // filter enabled
            $("input:enabled").val("Enabled");

            // filter checked
            var n = $("input:checked").length;
            $("div").text(n + (n == 1 ? " is" : " are") + " checked!");

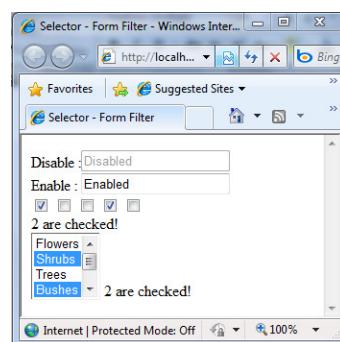
            // filter selected
            var s = $("select option:selected").length;
            $("span").text(s + (s == 1 ? " is" : " are") + " selected!");
        });
    </script>
</head>
<body>
<form>
    Disable :<input type="text" name="email" disabled="disabled" />
    <br />
    Enable :<input type="text" name="id" />
    <br />
    <input type="checkbox" name="newsletter" checked="checked" value="Hourly" />
    <input type="checkbox" name="newsletter" value="Daily" />
    <input type="checkbox" name="newsletter" value="Weekly" />
    <input type="checkbox" name="newsletter" checked="checked" value="Monthly" />
    <input type="checkbox" name="newsletter" value="Yearly" />
    <div></div>

    <select name="garden" multiple="multiple">
        <option>Flowers</option>
        <option selected="selected">Shrubs</option>
        <option>Trees</option>
        <option selected="selected">Bushes</option>
        <option>Grass</option>
        <option>Dirt</option>
    </select>
    <span></span>
</form>

</body>
</html>

```

Berikut ini hasil dari kode di atas.



Gambar 2-36 : Form filter untuk memilih elemen form berdasarkan status akses.

Attribute

Fungsi-fungsi pada kelompok ini berfungsi untuk mengakses atribut yang dimiliki oleh suatu elemen yang telah dipilih oleh fungsi selector. Dengan fungsi-fungsi ini nilai atribut suatu elemen dapat diambil, diubah atau ditambahkan.

Attr

Fungsi-fungsi pada kelompok ini berfungsi untuk mengambil nilai atribut pada suatu elemen, selain itu juga terdapat fungsi untuk menambahkan atribut pada suatu elemen beserta nilainya dan fungsi untuk menghapus atribut.

Untuk mengambil nilai atribut pada suatu elemen dapat digunakan fungsi `attr(nama_atribut)` dengan contoh sebagai berikut ini.

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Attributes - Attr</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        $(document).ready(function() {
            });

            // fungsi untuk menampilkan nilai dari atribut value
            // pada elemen input dengan id=username
            function ShowUsername() {
                alert($("#username").attr("value"));
            }

            // fungsi untuk menampilkan nilai dari atribut alt
            // pada elemen img dengan id=logo
            function ShowAlt() {
                alert($("#logo").attr("alt"));
            }
        </script>
    </head>
    <body>
        <form>
            Username : <input type="text" id="username" value="indc" />
            <input type="button" value="Show value" onclick="ShowUsername()" />

            <br />

            <img src="" id="logo" alt="Logo" />
            <input type="button" value="Show value" onclick="ShowAlt()" />
        </form>
    </body>
</html>
```

Sedangkan untuk menambahkan atribut beserta nilainya pada suatu elemen dapat digunakan dengan beberapa cara, yang paling sederhana adalah seperti berikut ini.

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Attributes - Attr</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        $(document).ready(function() {
            $("div").attr("style", "border:dashed 1px #000;margin-bottom:13px");
            $("input").attr("value", "Tombol");
        });
    </script>
</head>
```

```
<body>
    <div>Hello jQuery</div>
    <input type="button" />
</body>
</html>
```

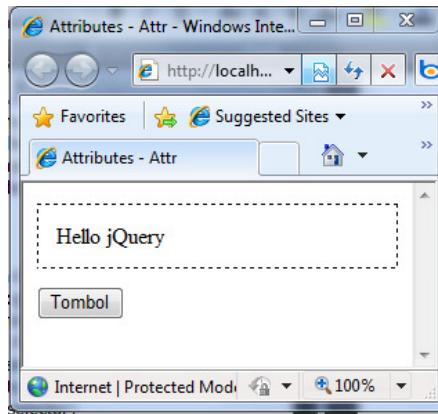
Pada contoh di sintaks untuk menambahkan atribut beserta nilainya adalah sebagai berikut.

```
attr("nama_atribut", "nilai_atribut")
```

Sebagai contoh adalah sebagai berikut.

```
$( "input" ).attr( "value", "Tombol" )
```

Artinya, pada elemen `input` akan ditambahkan atribut `value` dengan nilai `Tombol`. Dan gambar berikut adalah hasil dari contoh kode di atas.



Gambar 2-37 : Menambahkan atribut on-the-fly dengan fungsi attr.

Pemberian nilai atribut dengan sintaks di atas dapat dilakukan dengan menggunakan nilai output dari suatu fungsi. Sintaks dari cara ini adalah sebagai berikut.

```
attr("nama_atribut", function() { . . . })
```

Berikut ini contoh penggunaannya.

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Attributes - Attr</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        $(document).ready(function() {
            $("div").attr("id", function(arr) {
                return "div-id" + arr[0];
            })
            .each(function() {
                $("span", this).html("(ID = '<b>" + this.id + "</b>')");
            });

            $("img").attr("src", function() {
                return "images/" + this.title;
            });

        });
    </script>
</head>
<body>
    <div>Zero-th <span></span></div>
    <div>First <span></span></div>
    <div>Second <span></span></div>
```

```
<img title="logo.gif"/>
</body>
</html>
```

Fungsi attr juga dapat digunakan untuk menambahkan atribut lebih dari satu dengan sintaks seperti berikut.

```
attr({attribut1: "value1", attribut2: "value2", . . . attributN: "valueN"})
```

Berikut ini contoh penggunaannya.

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Attributes - Attr</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        $(document).ready(function() {
            $("img").attr({
                alt:"Logo",
                src: "images/logo.gif",
                width: "113"
            });
        });
    </script>
</head>
<body>
    <img />
</body>
</html>
```

Class

Fungsi-fungsi pada kelompok ini berfungsi untuk menambahkan dan menghilangkan attribut class beserta nilainya pada suatu elemen. Ada tiga fungsi yang termasuk dalam kelompok ini, yaitu :

- a. `addClass("nama_class")`, menambahkan `class` pada elemen yang dipilih.
- b. `removeClass("nama_class")`, menghapus `class` pada elemen yang dipilih.
- c. `toggleClass("nama_class")`, menambahkan `class` pada suatu elemen bila pada elemen tersebut belum digunakan `class` yang ingin ditambahkan, bila pada elemen telah menggunakan `class` yang dimaksud maka `class` tersebut akan dihapus pada elemen tersebut.

Berikut ini contoh penggunaan fungsi-fungsi di atas.

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Attributes - Class</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        function AddClass() {
            $("div").addClass("box");
        }

        function RemoveClass() {
            $("div").removeClass("box");
        }

        function ToggleClass() {
            $("div").toggleClass("box");
        }
    </script>
    <style type="text/css">
```

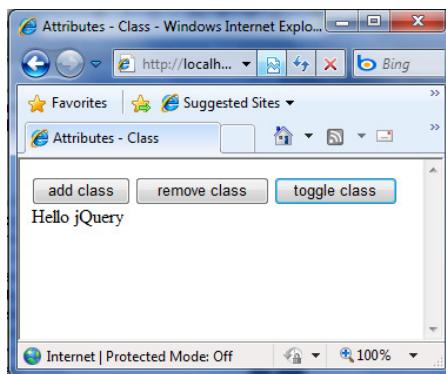
```

.box
{
    border:dashed 1px #000;
    padding:13px;
    margin-top:13px;
}
</style>
</head>
<body>
    <input type="button" value="add class" onclick="AddClass()" />
    <input type="button" value="remove class" onclick="RemoveClass()" />
    <input type="button" value="toggle class" onclick="ToggleClass()" />

    <div>
        Hello jQuery
    </div>
</body>
</html>

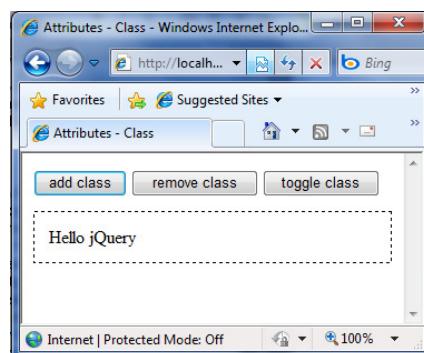
```

Berikut ini gambar ketika kode di atas dibuka pada web browser.



Gambar 2-38 : Antarmuka saat dibuka pertama kali.

Ketika tombol *add class* diklik, maka dapat dilihat hasil seperti ini.



Gambar 2-39 : Antarmuka setelah tombol add class diklik.

Dapat dilihat saat tombol *add class* diklik akan dieksekusi fungsi `addClass("box")`, dimana `box` adalah nama `class` yang memberikan style `border` pada elemen `div`, seperti yang dilihat pada gambar di atas. Bila tombol *remove class* diklik maka antarmuka akan kembali seperti pada gambar 2-38.

HTML

Fungsi-fungsi pada kelompok ini dapat digunakan untuk mengambil nilai atau memberikan nilai berupa tag-tag HTML secara on-the-fly.

Untuk mengambil nilai dari suatu elemen dapat digunakan sintaks berikut :

```
html()
```

Sedangkan untuk memberikan nilai digunakan sintaks berikut :

```
html(value)
```

Dan berikut ini adalah contoh penggunaan dari fungsi-fungsi ini.

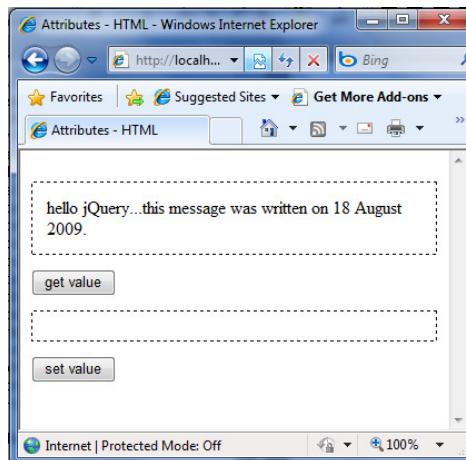
```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Attributes - HTML</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        function GetValue() {
            alert($("#Div1").html());
        }

        function SetValue() {
            $("#Div2").html('<input type="text" value="im just created" />');
        }
    </script>
    <style type="text/css">
        .box
        {
            border:dashed 1px #000;
            padding:13px;
            margin-top:13px;
            margin-bottom:13px;
        }
    </style>
</head>
<body>
    <div id="Div1" class="box">
        hello jQuery...this message was written on 18 August 2009.
    </div>
    <input type="button" value="get value" onclick="GetValue()" />

    <div id="Div2" class="box">

    </div>
    <input type="button" value="set value" onclick="SetValue()" />
</body>
</html>
```

Berikut ini adalah antarmuka dari contoh di atas.

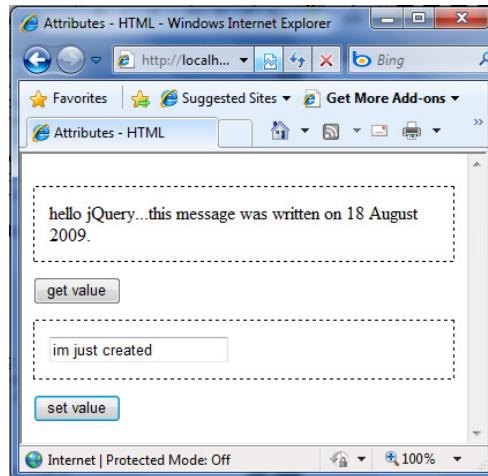


Gambar 2-40 : Antarmuka dari contoh penggunaan fungsi html().

Setelah tombol `get value` diklik maka akan dieksekusi baris berikut ini.

```
alert($("#Div1").html());
```

Hasil nya adalah dialog `alert` yang berisi nilai dari elemen `div` dengan `id=Div1`. Sedangkan bila tombol `set value` diklik maka akan dilihat hasil seperti gambar berikut ini.



Gambar 2-41 : Input bertipe text ditambahkan pada div dengan id=Div2.

Input bertipe `text` di dalam div dengan `id=Div2` terjadi karena tombol `set value` mengeksekusi baris berikut ini.

```
$("#Div2").html('<input type="text" value="im just created" />');
```

Dapat dilihat pada fungsi `html` terdapat nilai berupa tag HTML.

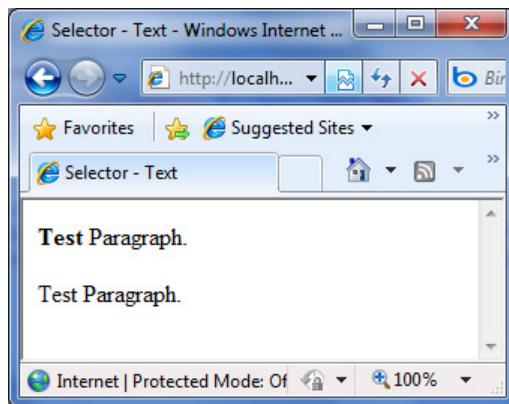
Text

Fungsi-fungsi pada kelompok ini untuk mengambil atau memberikan nilai text pada elemen yang telah dipilih.

Berikut ini contoh penggunaan fungsi `text`.

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Selector - Text</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        $(document).ready(function() {
            var str = $("p:first").text();
            $("p:last").text(str);
        });
    </script>
</head>
<body>
    <p><b>Test</b> Paragraph.</p>
    <p></p>
</body>
</html>
```

Dan berikut ini adalah hasilnya.



Gambar 2-42 : Fungsi text untuk mengambil nilai dan memberikan nilai.

Fungsi-fungsi ini tidak dapat digunakan pada semua elemen, tetapi hanya dapat bekerja pada elemen-elemen yang dapat diisi dengan text, seperti `div` atau `span`.

Value

Fungsi-fungsi ini berfungsi untuk mengambil atau memberikan nilai pada suatu elemen.

Berikut ini contoh penggunaan fungsi val.

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Selector - Val</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        $(document).ready(function() {
            $("select").change(displayVals);
            displayVals();
        });

        function displayVals() {
            var singleValues = $("#single").val();
            $("#nilai").val(singleValues);
            $("p").html("<b>Single:</b> " + singleValues);
        }
    </script>
</head>
<body>
    <p></p>
    <select id="single">
        <option>Single</option>
        <option>Single2</option>
    </select>
    <input type="text" id="nilai" />
</body>
</html>
```

Pada contoh di atas, akan diambil nilai dari elemen select dan nilai tersebut akan diberikan ke elemen input.

Traversing

Filtering

Filtering berguna untuk memilih elemen-elemen yang dipilih oleh fungsi `selector`. Pada bagian sebelumnya yaitu pada bagian pembahasan Selector, terdapat parameter yang dapat digunakan sebagai filtering.

Pada bagian ini akan diterangkan penggunaan fungsi untuk menyaring elemen-elemen yang ingin dipilih.

eq(index)

Fungsi ini berguna untuk memilih salah satu elemen dari kumpulan elemen-elemen yang telah dipilih. Elemen yang dipilih dengan menggunakan fungsi `eq(index)` adalah berdasarkan index dari kumpulan elemen yang ada.

pada contoh sebelumnya digunakan sintaks seperti berikut ini.

```
$(“element:eq(index)”) 
```

Sedangkan dengan menggunakan fungsi `eq(index)` dapat digunakan seperti contoh di bawah ini.

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Traversing - Filtering</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        $(document).ready(function() {
            $("tr").eq(0).addClass("Header");
            $("tr").eq(6).addClass("Header");
        });
    </script>
    <style type="text/css">
        body
        {
            font-family:Verdana;
            font-size:12px;
        }

        .Header
        {
            background-color:Black;
            color:White;
            font-weight:bold;
            text-align:center;
            text-transform:uppercase;
        }
    </style>
</head>
<body>
    <table border="1" cellpadding="0" cellspacing="0" width="100%">
        <tr>
            <td>No.</td>
            <td>Nama</td>
            <td>Alamat</td>
        </tr>
        <tr>
            <td>1.</td>
            <td>Nama #1</td>
            <td>Alamat #1</td>
        </tr>
        <tr>
            <td>2.</td>
            <td>Nama #2</td>
            <td>Alamat #2</td>
        </tr>
        <tr>
            <td>3.</td>
            <td>Nama #3</td>
            <td>Alamat #3</td>
        </tr>
        <tr>
```

```

        <td>4.</td>
        <td>Nama #4</td>
        <td>Alamat #4</td>
    </tr>
    <tr>
        <td>5.</td>
        <td>Nama #5</td>
        <td>Alamat #5</td>
    </tr>
    <tr>
        <td colspan="3">1 2 3</td>
    </tr>
</table>
</body>
</html>

```

Penggunaan fungsi eq(index) dapat dilihat pada baris berikut ini.

```

$("tr").eq(0).addClass("Header");
$("tr").eq(6).addClass("Header");

```

Kedua baris di atas berfungsi untuk memilih setiap elemen tr yang ke 0 dan ke 6. Sehingga dapat dilihat hasilnya seperti gambar berikut.

NO.	NAMA	ALAMAT
1.	Nama #1	Alamat #1
2.	Nama #2	Alamat #2
3.	Nama #3	Alamat #3
4.	Nama #4	Alamat #4
5.	Nama #5	Alamat #5

Gambar 2-43 : Elemen ke-0 dan ke-6 dipilih dengan fungsi eq(index).

filter(expr) dan filter(fn)

Kedua fungsi filter ini berguna untuk memilih elemen pada suatu koleksi elemen berdasarkan expression tertentu atau berdasarkan output dari suatu fungsi. Nilai parameter fungsi ini dapat berupa id dari suatu elemen, class, tag elemen dan lain-lain.

Berikut ini adalah contoh penggunaan fungsi filter(expr).

```

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Traversing - Filtering</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        $(document).ready(function() {
            $("div").filter(".middle").css("background", "red");
        });
    </script>
    <style type="text/css">
        div
        {
            width:113px;
            height:30px;

```

```

        margin:13px;
        border:dashed 1px #000;
        background-color:Gray;
    }

    .middle
    {
        text-align:center;
    }

```

</style>

</head>

<body>

<div></div>

<div class="middle"></div>

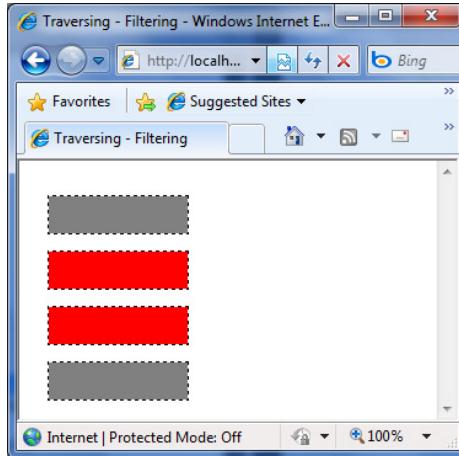
<div class="middle"></div>

<div></div>

</body>

</html>

Hasil dari kode di atas dapat dilihat pada gambar di bawah ini.



Gambar 2-44 : Fungsi filter(expr) tidak memilih elemen div pertama dan terakhir.

```
$("div").filter(".middle").css("background", "red");
```

Baris di atas akan hanya akan memilih elemen-elemen div yang menggunakan class middle, kemudian menambahkan style ke dalamnya.

Contoh berikutnya adalah penggunaan fungsi filter(fn).

```

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Traversing - Filtering</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        $(document).ready(function() {
            $("div").css("background", "#b4b0da")
                .filter(function(index) {
                    return index == 1 || $(this).attr("id") == "fourth";
                })
                .css("border", "3px double red");
        });
    </script>

    <style type="text/css">
        div
        {
            width:113px;

```

```

        height:30px;
        margin:13px;
        border:dashed 1px #000;
        background-color:Gray;
    }

```

</style>

</head>

<body>

```

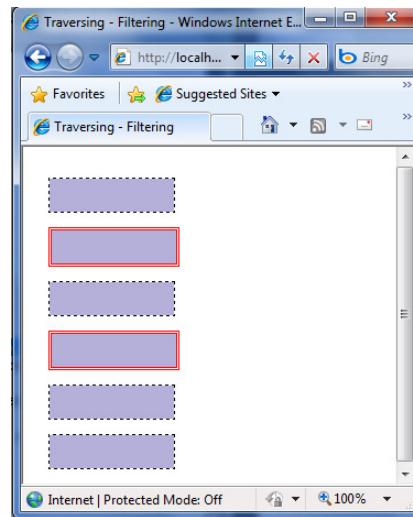
        <div id="first"></div>
        <div id="second"></div>
        <div id="third"></div>
        <div id="fourth"></div>
        <div id="fifth"></div>
        <div id="sixth"></div>

```

</body>

</html>

Hasil dari contoh ini dapat dilihat pada gambar berikut.



Gambar 2-45 : Filter dengan menggunakan filter(fn).

```

$( "div" ).css("background", "#b4b0da")
    .filter(function(index) {
        return index == 1 || $(this).attr("id") == "fourth";
    })
    .css("border", "3px double red");

```

Dari contoh di atas dapat dilihat, nilai parameter fungsi filter diisi dari output fungsi. Pada contoh ini dipilih elemen-elemen yang mempunyai index 1 atau mempunyai `id=fourth`. Sehingga dapat dilihat terdapat dua kotak dengan border berwarna merah.

hasClass(class)

Fungsi ini akan memeriksa apakah suatu elemen menggunakan class yang nilainya sama dengan nilai pada parameter `class` dari fungsi `hasClass`. Output dari fungsi ini akan bernilai true jika nilai `class` dari elemen sama dengan nilai pada parameter.

```

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Traversing - Filtering</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        $(document).ready(function() {
            $("div").click(function() {
                if ($(this).hasClass("protected"))
                    alert('Kotak ini menggunakan class=protected');
            });
        });
    </script>

```

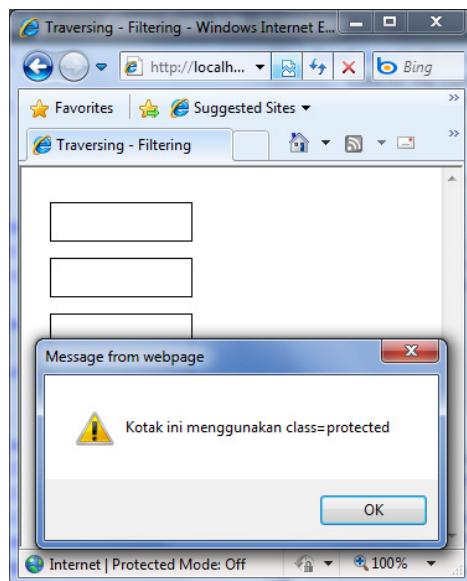
```

        });
    });
</script>
<style type="text/css">
div
{
    border:solid 1px #000;
    height:30px;
    width:113px;
    margin:13px;
}
</style>
</head>
<body>
<span></span><div class="protected"></div>
<span></span><div></div>

<span></span><div></div>
<span></span><div class="protected"></div>
</body>
</html>

```

Elemen div yang menggunakan `class=protected` akan diberikan event `click`, dan ketika elemen div tersebut diklik akan ditampilkan dialog box seperti gambar di bawah ini.



Gambar 2-46 : Dialog box yang keluar ketika elemen yang mempunyai `class=protected` diklik.

is(expr) dan not(expr)

Fungsi `is(expr)` berfungsi untuk melakukan pemeriksaan apakah elemen yang dipilih sesuai dengan nilai yang diberikan pada parameter `expr`. Bila sesuai maka output dari fungsi ini adalah `true`.

Berikut adalah contoh penggunaan fungsi ini.

```

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Traversing - Filtering</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        $(document).ready(function() {
            if ($("#kotak").is("div")) {
                $("#kotak").text("ini elemen div")
            }
        })
    </script>
</head>
<body>
<div id="kotak"></div>
</body>
</html>

```

```

        if ($("#tulisan").is("span")) {
            $("#tulisan").text("ini elemen span")
        }
    });
</script>
<style type="text/css">
div
{
    border:solid 1px #000;
    height:30px;
    width:113px;
    margin:13px;
}
</style>
</head>
<body>
    <div id="kotak"></div>
    <span id="tulisan"></span>
</body>
</html>

```

Sedangkan fungsi `not(expr)` mempunyai fungsi yang berkebalikan dengan fungsi `is(expr)`.

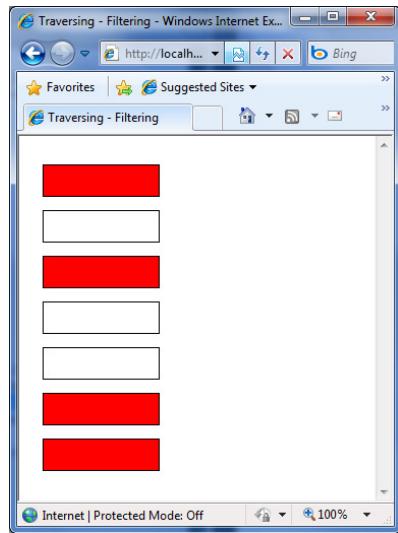
```

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Traversing - Filtering</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        $(document).ready(function() {
            $("div").not(".green, #blueone").css("background-color", "red");
        });
    </script>
    <style type="text/css">
div
{
    border:solid 1px #000;
    height:30px;
    width:113px;
    margin:13px;
}
</style>
</head>
<body>
    <div></div>
    <div id="blueone"></div>
    <div></div>
    <div class="green"></div>

    <div class="green"></div>
    <div class="gray"></div>
    <div></div>
</body>
</html>

```

Contoh di atas tidak akan memilih elemen `div` yang menggunakan `class=green` dan `id=blueone`. Sehingga dapat dilihat hasilnya seperti pada gambar di bawah ini.



Gambar : 2-47 : Fungsi `not(expr)` tidak memilih elemen kedua, keempat dan kelima.

map(callback)

Fungsi dapat digunakan untuk mengubah elemen-elemen yang dipilih menjadi suatu nilai dalam sebuah array.

Pada contoh di bawah ini akan digunakan fungsi `map(callback)` untuk mengambil nilai seluruh elemen `input` yang ada pada halaman web dan memasukkannya dalam array.

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Traversing - Filtering</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        $(document).ready(function() {
            var inputArr = $("input").map(function() {
                return $(this).val();
            });
            alert(inputArr[0]);
        });
    </script>
</head>
<body>
    <form>
        <input type="text" name="name" value="John"/>

        <input type="text" name="password" value="password"/>
        <input type="text" name="url" value="http://ejohn.org/">
    </form>
</body>
</html>
```

Pada contoh ini akan diambil nilai dari semua elemen `input` yang dipilih, kemudian nilai-nilai tersebut akan disimpan pada variable `inputArr` yang bertipe array, sehingga dengan memanggil `inputArr[0]` akan ditampilkan John sebagai nilai dari array index ke-0 ini.

Contoh lain penggunaan fungsi ini dapat dilihat pada kode berikut.

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Traversing - Filtering</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
```

```

<script language="javascript" type="text/javascript">
    $(document).ready(function() {
        $("p").append($("input").map(function() {
            return $(this).val();
        }).get().join(", "));
    });
</script>
</head>
<body>
    <p><b>Values: </b></p>
    <form>
        <input type="text" name="name" value="John"/>

        <input type="text" name="password" value="password"/>
        <input type="text" name="url" value="http://ejohn.org/">
    </form>
</body>
</html>

```

slice(start, end)

Fungsi `slice(start, end)` digunakan memilih elemen-elemen dari kumpulan elemen yang sudah dipilih oleh fungsi selector ke dalam suatu sub elemen. Sub elemen hasil output fungsi `slice(start, end)`, sebagai contoh `slice(3, 5)`, akan berisi elemen-elemen dengan index dari 3 sampai 5 (elemen ke-3 dan elemen ke-4 saja).

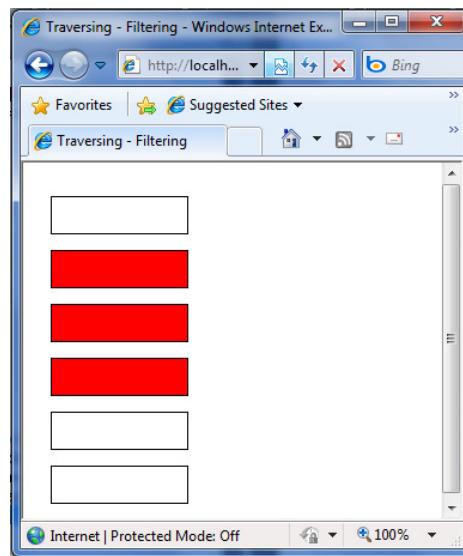
Sebagai contoh penggunaan fungsi ini bisa dilihat dari kode di bawah ini.

```

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Traversing - Filtering</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        $(document).ready(function() {
            $("div").slice(1, 4).css("background-color", "red");
        });
    </script>
    <style type="text/css">
div
{
    border:solid 1px #000;
    height:30px;
    width:113px;
    margin:13px;
}
    </style>
</head>
<body>
    <div></div>
    <div></div>
    <div></div>
    <div></div>
    <div></div>
    <div></div>
</body>
</html>

```

Hasil dari contoh ini dapat dilihat pada gambar berikut.



Gambar 2-48 : Fungsi slice(1,4) memilih elemen pertama, kedua dan ketiga.

Contoh penggunaan fungsi `slice(start, end)` yang lain dapat dilihat di bawah ini.

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Traversing - Filtering</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        $(document).ready(function() {
            $("button").click(colorEm);
        });

        function colorEm() {
            var $div = $("div");
            var start = Math.floor(Math.random() *
                $div.length);
            var end = Math.floor(Math.random() *
                ($div.length - start)) +
                start + 1;
            if (end == $div.length) end = undefined;
            $div.css("background", "");
            if (end)
                $div.slice(start, end).css("background", "yellow");
            else
                $div.slice(start).css("background", "yellow");

            $("span").text('$(\'div\').slice(' + start +
                (end ? ', ' + end : '') +
                ').css("background", "yellow");');
        }
    </script>
    <style type="text/css">
div
{
    border:solid 1px #000;
    height:30px;
    width:113px;
    margin:13px;
}
    </style>
</head>
<body>
    <button>Turn slice yellow</button>
    <span>Click the button!</span>
    <div></div>
    <div></div>
```

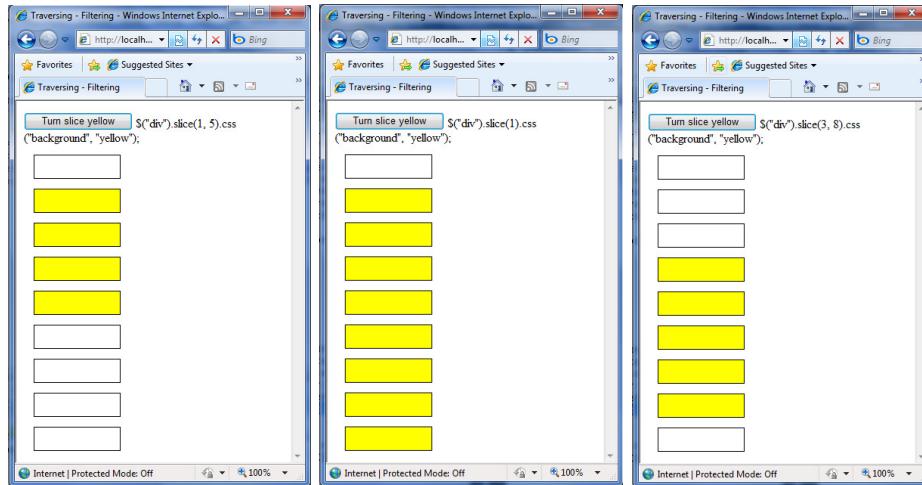
```

<div></div>
<div></div>
<div></div>
<div></div>
<div></div>

<div></div>
<div></div>
</body>
</html>

```

Hasilnya adalah gambar-gambar di bawah ini.



Gambar 2-49 : Fungsi slice dijalankan dengan nilai start dan end secara acak.

Contoh di atas memanfaatkan fungsi `slice(start, end)` untuk memilih elemen `div` secara acak ketika tombol `Turn slice yellow` diklik.

Finding

`add(expr)`

Misalnya telah dipilih semua elemen `div` dengan menggunakan fungsi selector `$("div")`, bila ingin menambahkan elemen lain di dalam kumpulan elemen yang telah dipilih tersebut maka dapat digunakan fungsi `add(expr)`.

```

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Traversing - Finding</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        $(document).ready(function() {
            $("div").add("p")
                .add("input")
                .css("background", "yellow");
        });
    </script>
    <style type="text/css">
        div
        {
            border:solid 1px #000;
            height:30px;
            width:113px;
            margin:13px;
        }
    </style>
</head>

```

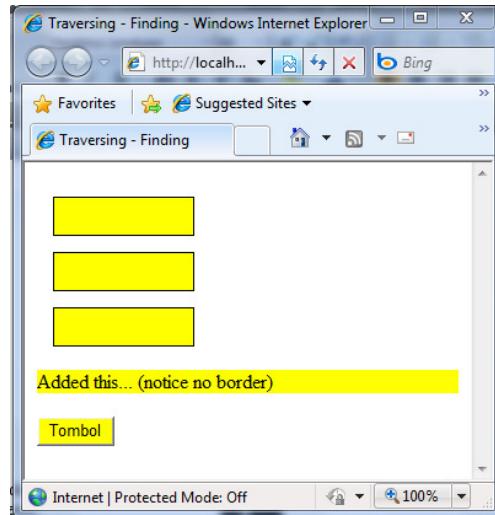
```

<body>
    <div></div>
    <div></div>
    <div></div>

    <p>Added this... (notice no border)</p>
    <input type="button" value="Tombol" />
</body>
</html>

```

Pada kode di atas, fungsi selector memilih semua elemen `div` ditambah dengan elemen `p` dan `input`. Sehingga akan didapat hasil seperti berikut.



Gambar 2-50 : Menambahkan elemen yang dipilih dengan fungsi `add(expr)`.

children(expr)

Fungsi `children(expr)` akan berguna apabila ingin memilih elemen anak dari suatu elemen yang telah dipilih oleh fungsi selector. Elemen anak dapat difilter berdasarkan nilai parameter `expr` yang ada pada fungsi ini.

```

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Traversing - Finding</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        $(document).ready(function() {
            $("div").children().css("border-bottom", "3px double red");
        });
    </script>
    <style type="text/css">
        div
        {
            border:solid 1px #000;
            height:50px;
            width:232px;
            margin:13px;
        }
    </style>
</head>
<body>
    <p>Hello (this is a paragraph)</p>

    <div><span>Hello Again (this span is a child of the a div)</span></div>
    <p>And <span>Again</span> (in another paragraph)</p>

    <div>And One Last <span>Time</span> (most text directly in a div)</div>

```

```
</body>  
</html>
```

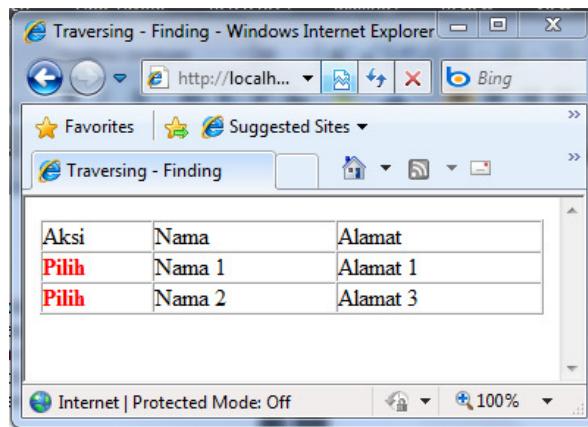
Pada contoh di atas, akan dipilih semua elemen anak yg ada di dalam elemen `div`. contoh pemakaian fungsi `children` diatas tidak mengisikan nilai parameter `expr` sama sekali.

```
<html xmlns="http://www.w3.org/1999/xhtml">  
<head>  
    <title>Traversing - Finding</title>  
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>  
    <script language="javascript" type="text/javascript">  
        $(document).ready(function() {  
            $("td").children("a").addClass("link");  
        });  
    </script>  
    <style type="text/css">  
        .link  
        {  
            color:Red;  
            text-decoration:none;  
            font-weight:bold;  
        }  
  
        .link:hover  
        {  
            color:Red;  
            text-decoration:underline;  
            font-weight:bold;  
        }  
    </style>  
</head>  
<body>  
    <table style="width: 100%;" border="1" cellpadding="0" cellspacing="0">  
        <tr>  
            <td>  
                Aksi  
            </td>  
            <td>  
                Nama  
            </td>  
            <td>  
                Alamat  
            </td>  
        </tr>  
        <tr>  
            <td>  
                <a href="#">Pilih</a>  
            </td>  
            <td>  
                Nama 1  
            </td>  
            <td>  
                Alamat 1  
            </td>  
        </tr>  
        <tr>  
            <td>  
                <a href="#">Pilih</a>  
            </td>  
            <td>  
                Nama 2  
            </td>  
            <td>  
                Alamat 3  
            </td>  
        </tr>  
    </table>  
</body>  
</html>
```

Pada contoh ini dipilih elemen `a` yang berada di dalam elemen `td` dengan cara berikut.

```
$( "td" ).children( "a" ).addClass( "link" );
```

Dan berikut ini adalah hasilnya.



Gambar 2-51 : Pemberian style pada elemen a yang berada di dalam elemen td.

find(expr)

Bila dimiliki kode HTML seperti ini.

```
<span>Hello jQuery</span>
<p><span>Hello</span>, how are you?</p>
<p>Me? I'm <span>good</span>.</p>
<div><span>Do you use jQuery?</span></div>
```

Maka untuk memilih elemen yang hanya berada di dalam elemen p dapat menggunakan fungsi selector seperti ini.

```
 $("p span")
```

Sedangkan apabila menggunakan fungsi `find(expr)`, maka dapat ditulis seperti berikut ini.

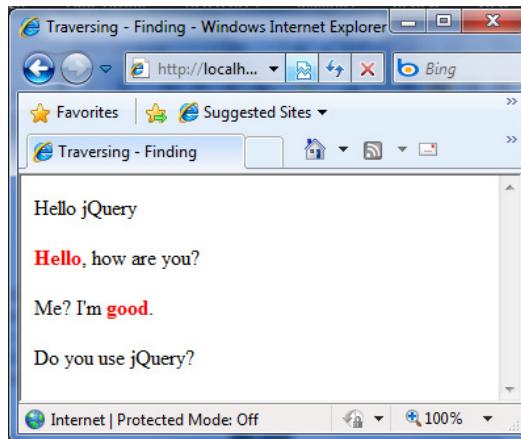
```
 $("p").find("span")
```

Berikut ini contoh kode lengkap penggunaan fungsi `find(expr)`.

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Traversing - Finding</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        $(document).ready(function() {
            $("p").find("span").addClass("link");
        });
    </script>
    <style type="text/css">
        .link {
            color:Red;
            text-decoration:none;
            font-weight:bold;
        }
    </style>
</head>
<body>
    <span>Hello jQuery</span>
    <p><span>Hello</span>, how are you?</p>
    <p>Me? I'm <span>good</span>.</p>
    <div><span>Do you use jQuery?</span></div>
</body>
```

```
</html>
```

Hasilnya dapat dilihat pada gambar di bawah ini.



Gambar 2-52 : Pemberian style pada elemen span yang berada di dalam elemen p.

contents()

Fungsi `content()` dapat digunakan untuk menemukan semua elemen anak pada suatu elemen induk yang dipilih, hapir semua elemen anak dapat dipilih oleh fungsi ini termasuk node text, isi dari dokumen bahkan elemen yang berada di dalam `iframe`.

Berikut ini contoh penggunaan fungsi `content()` untuk melakukan operasi pada elemen-elemen dalam `iframe`.

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Traversing - Finding</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        $(document).ready(function() {
            $("iframe").contents().find("body p").append("Tulisan ini ditulis
bukan dari halaman iframe");
        });
    </script>
    <style type="text/css">
        iframe
        {
            border:solid 1px #000;
        }
    </style>
</head>
<body>
    <iframe src="iframe.html" width="300" height="100"></iframe>
</body>
</html>
```

Dan berikut ini adalah kode dari file `iframe.html`.

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title></title>
</head>
<body>
    <div>
        <p>Halo jQuery. Tulisan ini berada di dalam iframe.</p>
    </div>
</body>
</html>
```

next(expr) dan nextAll(expr)

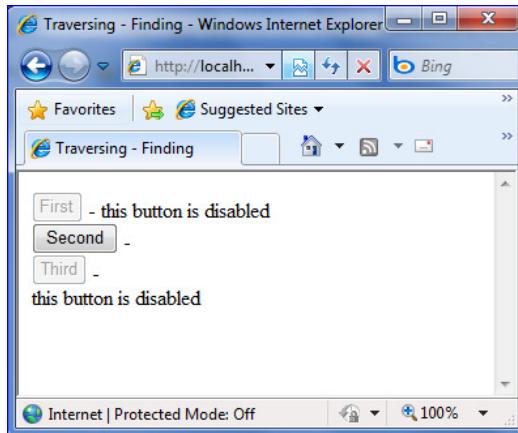
pada contoh di bawah ini, akan ditambahkan text setelah button yang mempunyai status akses disabled.

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Traversing - Finding</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        $(document).ready(function() {
            $("button[disabled]").next().text("this button is disabled");
        });
    </script>
</head>
<body>
    <div>
        <button disabled="disabled">First</button> -
        <span></span>
        <div></div>
    </div>

    <div>
        <button>Second</button> -
        <span></span>
    </div>

    <div>
        <button disabled="disabled">Third</button> -
        <div></div>
        <span></span>
    </div>
</body>
</html>
```

Dengan fungsi `next()` maka akan dipilih elemen persis setelah elemen hasil pilihan fungsi selector. Dalam hal ini adalah elemen-elemen setelah elemen button dengan status akses disabled, yaitu elemen `span` dan `div`.



Gambar 2-53 : Memilih elemen dengan menggunakan fungsi `next()`.

Apa bila ingin menambahkan text pada elemen `span` saja, maka baris ini :

```
$( "button[disabled]" ).next().text("this button is disabled");
```

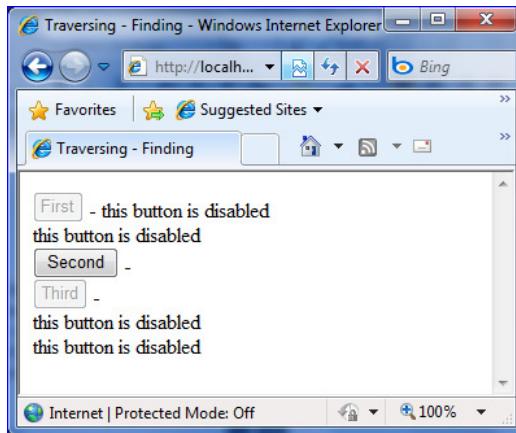
Dapat diubah menjadi berikut ini :

```
$( "button[disabled]" ).next("span").text("this button is disabled");
```

Fungsi selanjutnya adalah fungsi `nextAll(expr)`, kemampuan fungsi ini hampir sama seperti fungsi `next(expr)` kecuali fungsi ini akan memilih semua elemen setelah elemen yang dipilih dengan fungsi selector (atau fungsi lainnya). Jika fungsi `next(expr)` di atas diganti dengan `nextAll(expr)` seperti berikut ini.

```
$( "button[disabled]" ).nextAll().text("this button is disabled");
```

Maka hasilnya akan dapat dilihat pada gambar berikut.



Gambar 2-54 : Memilih semua elemen setelah elemen pilihan fungsi selector dengan menggunakan fungsi `nextAll()`.

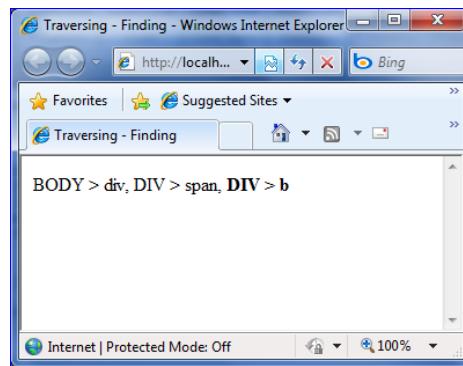
parent(expr) dan parents(expr)

Fungsi `parent(expr)` dapat digunakan untuk memilih elemen induk dari elemen yang dipilih oleh fungsi selector.

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Traversing - Finding</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        $(document).ready(function() {
            $("*", document.body).each(function() {
                var parentTag = $(this).parent().get(0).tagName;
                $(this).prepend(document.createTextNode(parentTag + " > "));
            });
        });
    </script>
</head>
<body>
    <div>div,
        <span>span, </span>
        <b>b </b>
    </div>
</body>
</html>
```

Kode di atas akan menampilkan elemen induk dari elemen yang ada.

Hasilnya dapat dilihat pada gambar berikut ini.



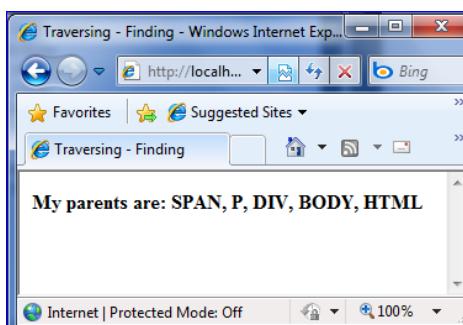
Gambar 2-55 : Menampilkan induk dari setiap elemen dengan menggunakan fungsi parent.

Dari gambar dapat dilihat bahwa induk dari elemen `div` adalah `body`, induk elemen `span` dan `b` adalah elemen `div`.

Berbeda dengan fungsi `parent(expr)`, fungsi `parents(expr)` digunakan untuk memilih semua elemen induk dari suatu elemen.

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Traversing - Finding</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        $(document).ready(function() {
            var parentEls = $("b").parents()
                .map(function() {
                    return this.tagName;
                })
                .get().join(", ");
            $("b").append("<strong>" + parentEls + "</strong>");
        });
    </script>
</head>
<body>
<div>
    <p>
        <span>
            <b>My parents are: </b>
        </span>
    </p>
</div>
</body>
</html>
```

Contoh di atas akan menampilkan seluruh elemen induk dari elemen `b`.



Gambar 2-56 : Daftar elemen induk dari elemen `b`.

prev(expr) dan prevAll(expr)

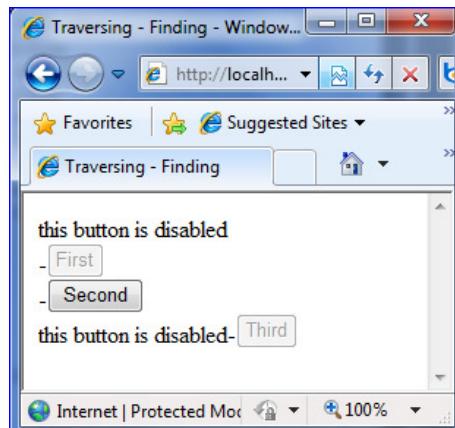
Fungsi prev(expr) mempunyai sifat yang berkebalikan dengan fungsi next(expr), sedangkan sifat fungsi prevAll(expr) berkebalikan dengan sifat fungsi nextAll(expr). Berikut ini adalah contoh yang mirip dengan kode yang digunakan saat membahas fungsi next(expr).

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Traversing - Finding</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        $(document).ready(function() {
            $("button[disabled]").prev().text("this button is disabled");
        });
    </script>
</head>
<body>
    <div>
        <span></span>
        <div></div> -
        <button disabled="disabled">First</button>
    </div>

    <div>
        <span></span> -
        <button>Second</button>
    </div>

    <div>
        <div></div>
        <span></span> -
        <button disabled="disabled">Third</button>
    </div>
</body>
</html>
```

Hasilnya mirip seperti pada contoh penggunaan fungsi next(expr) kecuali elemen yang modifikasi adalah elemen sebelum elemen yang dipilih.



Gambar 2-57 : Hasil dari fungsi prev(expr).

Sedangkan untuk mengetahui hasil dari fungsi prevAll(expr) dapat dilakukan dengan cara mengganti baris berikut ini :

```
$( "button[disabled]" ).prev().text("this button is disabled");
```

Menjadi seperti berikut.

```
$( "button[disabled]" ).prevAll().text("this button is disabled");
```

siblings(expr)

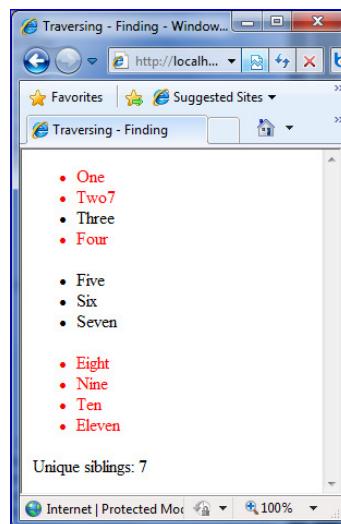
Fungsi siblings(expr) berfungsi untuk memilih semua tetangga dari suatu elemen. Untuk lebih jelasnya bisa dilihat contoh kode di bawah ini beserta hasilnya.

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Traversing - Finding</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        $(document).ready(function() {
            var len = $(".hilite").siblings()
                .css("color", "red")
                .length;
            $("b").text(len);
        });
    </script>
</head>
<body>
    <ul>
        <li>One</li>
        <li>Two</li>
        <li class="hilite">Three</li>
        <li>Four</li>
    </ul>

    <ul>
        <li>Five</li>
        <li>Six</li>
        <li>Seven</li>
    </ul>

    <ul>
        <li>Eight</li>
        <li class="hilite">Nine</li>
        <li>Ten</li>
        <li class="hilite">Eleven</li>
    </ul>

    <p>Unique siblings: <b></b></p>
</body>
</html>
```



Gambar 2-58 : Memilih elemen tetangga dengan fungsi siblings(expr).

Chaining

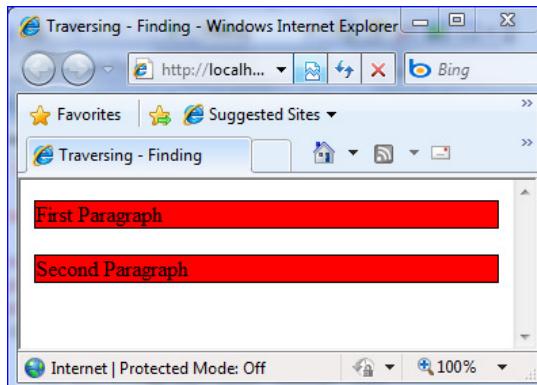
andSelf()

Untuk mengetahui fungsi dari andSelf() ini akan lebih baik melihat contoh-contoh berikut ini.

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Traversing - Finding</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        $(document).ready(function() {
            $("div").find("p").addClass("border");
            $("div").find("p").addClass("background");
        });
    </script>
    <style type="text/css">
        .border
        {
            border:solid 1px #000;
        }

        .background
        {
            background-color:Red;
        }
    </style>
</head>
<body>
    <div>
        <p>First Paragraph</p>
        <p>Second Paragraph</p>
    </div>
</body>
</html>
```

Dari contoh ini dapat diketahui bahwa elemen yang dipilih adalah elemen `p` yang berada di dalam elemen `div`. elemen-elemen `p` tersebut akan diberi class `border` dan `background` sehingga dapat dilihat pada gambar ini.



Gambar 2-59 : Pemilihan elemen p di dalam elemen div.

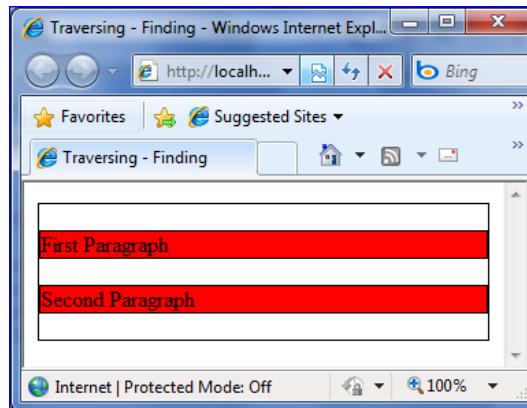
Jika baris ini :

```
$( "div" ).find( "p" ).addClass( "border" );
$( "div" ).find( "p" ).addClass( "background" );
```

Diganti menjadi seperti ini :

```
$( "div" ).find( "p" ).endSelf().addClass( "border" );
$( "div" ).find( "p" ).addClass( "background" );
```

Maka hasilnya akan dapat dilihat seperti gambar ini.



Gambar 2-60 : Dengan fungsi `endSelf()` elemen `div` juga terpilih.

Berikut penjelasan dari kode di atas, pertama akan dipilih semua elemen `div` dengan cara ini:

```
$(“div”)
```

Setelah itu akan dipilih seluruh elemen `p` yang berada di dalam elemen `div` dengan cara berikut.

```
$(“div”).find(“p”)
```

Dengan cara di atas, elemen yg terpilih hanya elemen `p` di dalam `div` saja, tetapi bila ditambahkan fungsi `andSelf()` seperti ini.

```
$(“div”).find(“p”).andSelf()
```

Dengan adanya fungsi `andSelf()` elemen sebelumnya juga terpilih, yaitu elemen `div`. Jadi fungsi `andSelf()` berfungsi memilih elemen yang telah dipilih sebelumnya (*previous element*).

end()

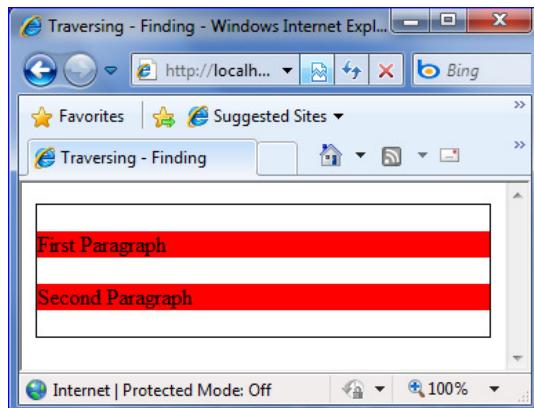
Bila contoh di atas diganti, fungsi `andSelf()` diganti dengan `end()`, maka elemen yang dipilih hanya elemen `div` saja. Sehingga kode ini.

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Traversing - Finding</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        $(document).ready(function() {
            $("div").find("p").end().addClass("border");
            $("div").find("p").addClass("background");
        });
    </script>
    <style type="text/css">
        .border
        {
            border:solid 1px #000;
        }

        .background
        {
            background-color:Red;
        }
    </style>
```

```
</head>
<body>
  <div>
    <p>First Paragraph</p>
    <p>Second Paragraph</p>
  </div>
</body>
</html>
```

Akan menghasilkan seperti gambar berikut.



Gambar 2-61 : Elemen p tanpa border.

Manipulation

Changing Content

Fungsi-fungsi yang termasuk dalam kelompok ini adalah `html()`, `html(value)`, `text()` dan `text(value)`. Keempat fungsi ini telah dibahas di atas, pada bagian Attribute.

Inserting Inside

Fungsi-fungsi berikut ini berguna untuk menyisipkan suatu content yang berupa text atau elemen ke dalam elemen. Sebagai gambaran, jika dimiliki elemen seperti contoh berikut ini.

```
<p>Hello jQuery</p>
```

Penyisipan content yang dilakukan oleh fungsi-fungsi kelompok ini akan membuat contoh di atas menjadi seperti berikut ini.

```
<p>Hello jQuery {content/elemen yang disisipkan}</p>
```

Atau berikut ini.

```
<p>{content/elemen yang disisipkan} Hello jQuery </p>
```

Dapat dilihat bahwa content yang ditambahkan masih berada di dalam elemen yang telah dipilih.

append(content) dan appendTo(element)

Fungsi ini digunakan untuk menambahkan content pada suatu elemen yang dipilih. Berikut macam-macam penggunaan fungsi `append(content)` yang bisa digunakan.

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Manipulation - Inserting Inside</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        $(document).ready(function() {
            $("p").append("<b>Hello</b>");
            $("p").append($("b"));
        });
    </script>
</head>
<body>
    <b>Hello</b>
    <p>I would like to say: </p>
</body>
</html>
```

Dan berikut ini hasilnya.



Gambar 2-62 : Menambahkan content dengan fungsi `append(content)`.

Dari contoh di atas dapat dilihat bahwa elemen dipilih terlebih dahulu kemudian content akan ditambahkan dengan fungsi `append(content)`, dimana parameter `content` berisi string atau elemen yang ingin ditambahkan pada elemen yang telah dipilih oleh fungsi selector.

Fungsi yang mirip juga dimiliki oleh fungsi `appendTo(element)`. Untuk mengetahui bagaimana fungsi ini bekerja, terlebih dahulu perhatikan contoh kode di bawah ini. Contoh di bawah ini akan membuat hasil yang sama seperti pada gambar 2-62.

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Manipulation - Inserting Inside</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        $(document).ready(function() {
            $("<b>Hello</b>").appendTo("p");
            $("b").appendTo("p");
        });
    </script>
</head>
<body>
    <b>Hello</b>
    <p>I would like to say: </p>
</body>
</html>
```

Dari contoh di atas dapat dilihat perbedaan antara :

```
$( "p" ).append( "<b>Hello</b>" );
$( "p" ).append( $( "b" ) );
```

Dengan kode berikut.

```
$( "<b>Hello</b>" ).appendTo( "p" );
$( "b" ).appendTo( "p" );
```

Pada fungsi `appendTo` terlebih dulu ditentukan content yang akan ditambahkan, penentuan content ini dapat dengan menggunakan fungsi selector, kemudian parameter pada fungsi `appendTo` adalah elemen tujuan yang akan disisipkan content.

prepend(content) dan prependTo(element)

Kedua fungsi ini mempunyai sifat yang berkebalikan bila dibandingkan dengan fungsi `append(content)` dan `appendTo(element)`. Perbedaannya, content akan disisipkan di awal elemen yang dipilih atau elemen tujuan, seperti pada gambar berikut ini.



Gambar 2-63 : Hasil fungsi `prepend(content)` dan `prependTo(element)`.

Dan berikut ini adalah contoh kode penggunaan kedua fungsi tersebut. Di bawah ini adalah penggunaan fungsi `prepend(content)`.

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Manipulation - Inserting Inside</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        $(document).ready(function() {
            $("p").prepend("<b>Hello</b>");
            $("p").prepend($( "b" ));
        });
    </script>
</head>
<body>
    <b>Hello</b>
    <p>I would like to say: </p>
</body>
</html>
```

Dan berikut ini adalah penggunaan fungsi `prependTo(element)`.

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Manipulation - Inserting Inside</title>
```

```
<script src="script/jquery-1.3.2.js" type="text/javascript"></script>
<script language="javascript" type="text/javascript">
    $(document).ready(function() {
        $("<b>Hello</b>").prependTo("p");
        $("b").prependTo("p");
    });
</script>
</head>
<body>
    <b>Hello</b>
    <p>I would like to say: </p>
</body>
</html>
```

Inserting Outside

Fungsi-fungsi pada kelompok ini mempunyai cara kerja yang mirip dengan fungsi-fungsi yang telah di bahas pada bagian Inserting Inside, perbedaannya fungsi-fungsi pada kelompok Inserting Outside akan menambahkan content atau elemen diluar elemen yang dipilih.

Sebagai gambaran, jika dimiliki elemen seperti contoh berikut ini.

```
<p>Hello jQuery</p>
```

Penyisipan content yang dilakukan oleh fungsi-fungsi kelompok ini akan membuat contoh di atas menjadi seperti berikut ini.

```
<p>Hello jQuery</p> {content/elemen yang disisipkan}
```

Atau berikut ini.

```
{content/elemen yang disisipkan} <p>Hello jQuery</p>
```

Dapat dilihat bawah content yang ditambahkan masih berada di luar elemen yang telah dipilih.

after(content) dan insertAfter(element)

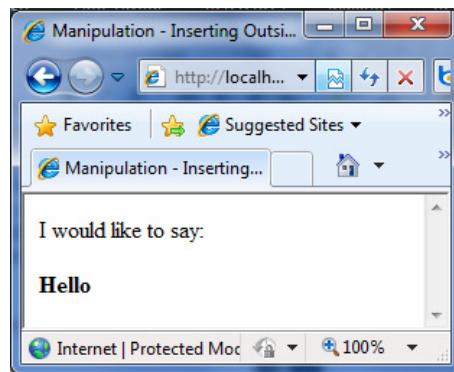
Kedua fungsi ini untuk menyisipkan content sebelum elemen yang dipilih, seperti berikut ini.

```
<p>Hello jQuery</p> {content/elemen yang disisipkan}
```

Contoh di bawah ini adalah contoh penggunaan fungsi `after(content)`.

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Manipulation - Inserting Outside</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        $(document).ready(function() {
            $("p").after("<b>Hello</b>");
        });
    </script>
</head>
<body>
    <p>I would like to say: </p>
</body>
</html>
```

Dan hasilnya seperti pada gambar di bawah ini.



Gambar 2-64 : Elemen b di sisipkan setelah elemen p.

Dapat dilihat perbedaan gambar 2-64 dengan 2-62. Pada gambar 2-62 kata *Hello* masih berada satu baris dengan kalimat *I would like to say*, karena **Hello** berada di dalam tag *p*. Berbeda dengan gambar 2-64, **Hello** berada setelah tag *p* sehingga dapat terlihat keduanya tidak berada di dalam satu baris, tetapi terpisah oleh suatu paragraf.

Untuk mendapatkan hasil yang sama dengan menggunakan fungsi `insertAfter(element)` dapat dilihat contoh berikut ini.

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Manipulation - Inserting Outside</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        $(document).ready(function() {
            $("<b>Hello</b>").insertAfter("p");
        });
    </script>
</head>
<body>
    <p>I would like to say: </p>
</body>
</html>
```

before(content) dan insertBefore(element)

Kedua fungsi ini akan menyisipkan elemen sebelum selemen yang dipilih sehingga didapatkan hasil seperti pada gambar ini.



Gambar 2-65 : Menyisipkan content sebelum elemen yang dipilih.

Berikut ini contoh penggunaan fungsi `before(content)`.

```

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Manipulation - Inserting Outside</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        $(document).ready(function() {
            $("p").before("<b>Hello</b>");
        });
    </script>
</head>
<body>
    <p>I would like to say: </p>
</body>
</html>

```

Dan berikut ini penggunaan fungsi `insertBefore(element)`.

```

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Manipulation - Inserting Outside</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        $(document).ready(function() {
            $("<b>Hello</b>").insertBefore("p");
        });
    </script>
</head>
<body>
    <p>I would like to say: </p>
</body>
</html>

```

Inserting Around

Fungsi-fungsi ini berfungsi untuk membungkus suatu elemen dengan HTML atau elemen lainnya. Sebagai gambaran bila dimiliki elemen seperti ini.

```
<p>Hello, jQuery.</p>
```

Dengan fungsi-fungsi pada kelompok Insert Around akan dilakukan pembungkusan elemen di atas seperti berikut ini.

```

<elemen_pembungkus>
    <p>Hello, jQuery.</p>
</elemen_pembungkus>

```

wrap(html) dan wrap(element)

Berikut ini contoh penggunaan fungsi `wrap(html)`, parameter html di sini dapat berisi elemen yang akan membungkus elemen yang diinginkan.

```

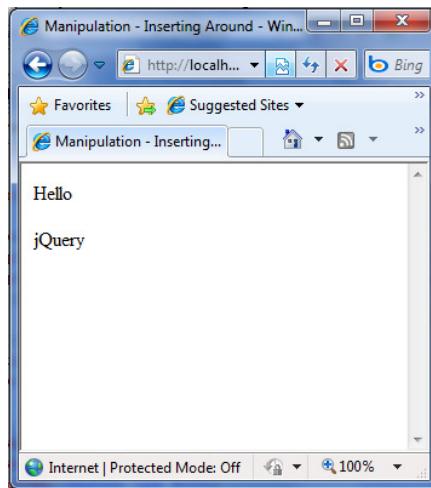
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Manipulation - Inserting Around</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        $(document).ready(function() {
            $("p").wrap("<div class='wrap'></div>");
        });
    </script>
    <style type="text/css">
        .wrap
        {
            padding:13px;
            margin:13px;
    
```

```

        border:dashed 1px #000;
    }
    </style>
</head>
<body>
    <p>Hello</p>
    <p>jQuery</p>
</body>
</html>

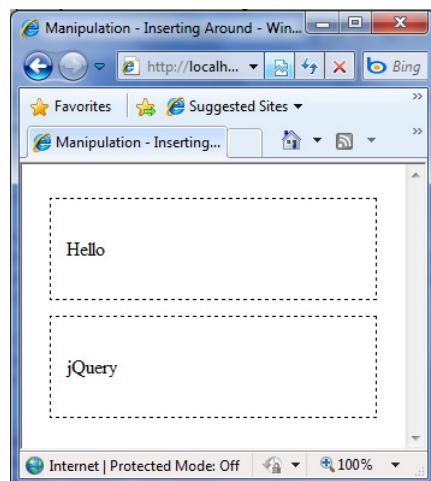
```

Sebelum menggunakan fungsi `wrap(html)` akan dilihat antarmuka seperti berikut ini.



Gambar 2-66 : Antarmuka sebelum menggunakan fungsi wrap(html).

Dan berikut ini hasil setelah elemen `p` dilingkupi oleh `<div class="wrap"></div>`.



Gambar 2-67 : Antarmuka setelah menggunakan fungsi wrap(html).

Penulisan fungsi dengan menggunakan `wrap(html)` dapat dipersingkat dengan menggunakan `wrap(element)`, sehingga kode ini.

```
$( "p" ).wrap("<div class='wrap'></div>");
```

Dapat diganti dengan kode berikut.

```
$( "p" ).wrap($(".wrap"));
```

wrapAll(html) dan wrapAll(element)

Kedua fungsi ini akan menggabungkan setiap elemen yang telah dipilih kemudian kesatuan elemen tersebut akan dibungkus dengan nilai parameter html atau element yang ada pada fungsi wrapAll.

Sebagai contoh misalnya bila dimiliki kode sebagai berikut ini.

```
<p>Hello</p>
<span>Span 1</span>
<p>Im using</p>
<span>Span 2</span>
<p>jQuery</p>
```

Dengan menggunakan baris ini.

```
$( "p" ).wrapAll("<div class='wrap'></div>");
```

Proses yang terjadi adalah, mengubah kode di atas menjadi seperti berikut ini.

```
<div class='wrap'>
    <p>Hello</p>
    <p>Im using</p>
    <p>jQuery</p>
</div>

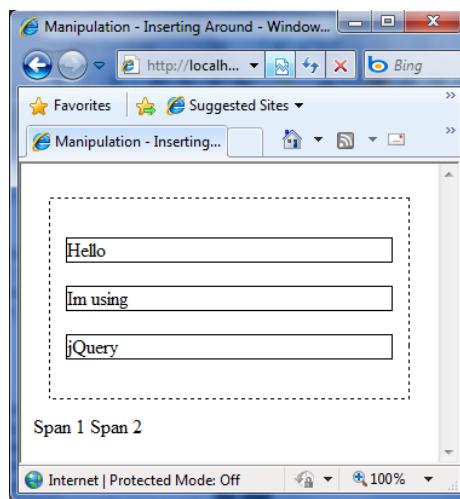
<span>Span 1</span>
<span>Span 2</span>
```

Berikut ini adalah contoh kode secara lengkap.

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Manipulation - Inserting Around</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        $(document).ready(function() {
            $("p").wrapAll("<div class='wrap'></div>");
        });
    </script>
    <style type="text/css">
        .wrap
        {
            padding:13px;
            margin:13px;
            border:dashed 1px #000;
        }

        p
        {
            border:solid 1px #000;
        }
    </style>
</head>
<body>
    <p>Hello</p>
    <span>Span 1</span>
    <p>Im using</p>
    <span>Span 2</span>
    <p>jQuery</p>
</body>
</html>
```

Dengan hasil dapat dilihat seperti pada gambar di bawah ini.



Gambar 2-68 : Hasil antarmuka penggunaan fungsi wrapAll.

Penulisan kode berikut ini.

```
$( "p" ).wrapAll("<div class='wrap'></div>")
```

Dapat diganti penulisannya dengan cara seperti ini.

```
$( "p" ).wrapAll($(".wrap"));
```

wrapInner(html) dan wrapInner(element)

Penyisipan dengan fungsi ini akan dilakukan seperti gambaran berikut ini. Berikut adalah susunan elemen awal.

```
<p>Hello</p>
<span>Span 1</span>
<p>Im using</p>
<span>Span 2</span>
<p>jQuery</p>
```

Setelah digunakan kode berikut ini.

```
$( "p" ).wrapInner("<div class='wrap'></div>");
```

Maka terjadi perubahan susunan kode di atas menjadi seperti ini.

```
<p>
<div class='wrap'>
Hello
</div>
</p>

<span>Span 1</span>

<p>
<div class='wrap'>
Im using
</div>
</p>
```

```

<span>Span 2</span>

<p>
<div class='wrap'>
jQuery
</div>
</p>

```

Berikut adalah kode lengkap yang bisa digunakan.

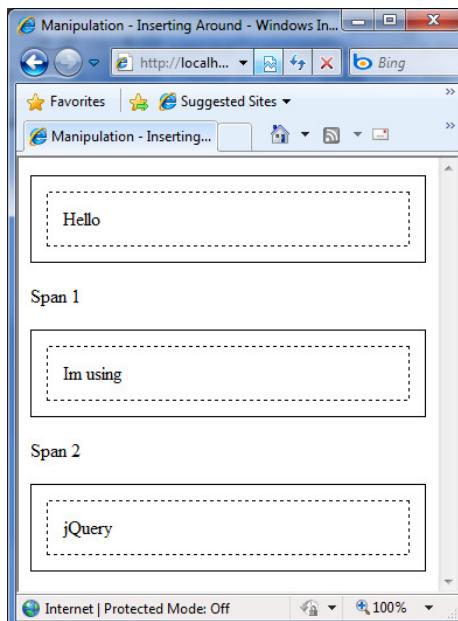
```

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Manipulation - Inserting Around</title>
<script src="script/jquery-1.3.2.js" type="text/javascript"></script>
<script language="javascript" type="text/javascript">
$(document).ready(function() {
    $("p").wrapInner("<div class='wrap'></div>");
});
</script>
<style type="text/css">
.wrap
{
    padding:13px;
    margin:13px;
    border:dashed 1px #000;
}

p
{
    border:solid 1px #000;
}
</style>
</head>
<body>
<p>Hello</p>
<span>Span 1</span>
<p>Im using</p>
<span>Span 2</span>
<p>jQuery</p>
</body>
</html>

```

Dan hasilnya dapat dilihat pada gambar berikut ini.



Gambar 2-69 : Modifikasi antarmuka on-the-fly yang dilakukan oleh fungsi wrapInner.

```
$( "p" ).wrapInner( "<div class='wrap'></div>" );
```

Dapat diganti menjadi baris kode di bawah ini.

```
$( "p" ).wrapInner( $( "<div class='wrap'>" ) );
```

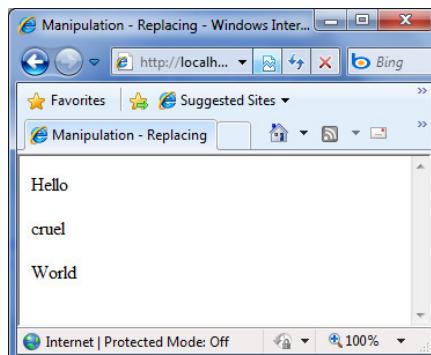
Replacing

Fungsi ini digunakan untuk mengubah suatu elemen menjadi elemen yang lain.

Sebagai contoh dapat dilihat di bawah ini.

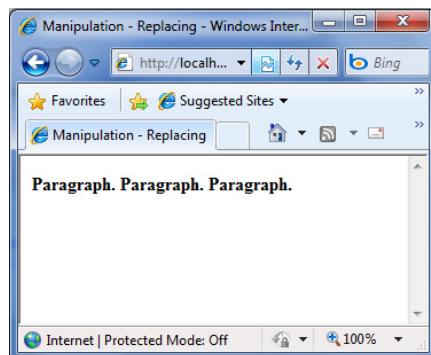
```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Manipulation - Replacing</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        $(document).ready(function() {
            $("<b>Paragraph. </b>").replaceAll("p");
        });
    </script>
</head>
<body>
    <p>Hello</p>
    <p>cruel</p>
    <p>World</p>
</body>
</html>
```

Dengan kode di atas, antarmuka awal di bawah ini.



Gambar 2-70 : Antarmuka awal.

Dan berikut ini adalah antarmuka ketika fungsi `replaceAll(selector)` digunakan.



Gambar 2-71 : Antarmuka setelah menggunakan fungsi `replaceAll(selector)`.

Selain fungsi `replaceAll(selector)`, juga dapat digunakan fungsi `replaceWith(content)` untuk kebutuhan yang sama tetapi dilakukan dengan cara yang berbeda. Baris dibawah ini.

```
$( "bParagraph. </b>" ).replaceAll("p");
```

Dapat diganti dengan baris ini.

```
$( "p" ).replaceWith("b>Paragraph. </b>");
```

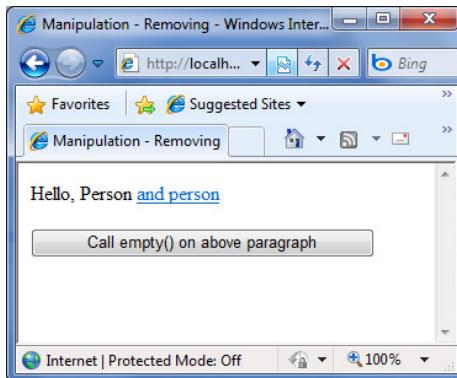
Removing

Fungsi yang termasuk dalam kelompok ini adalah `empty()` dan `remove(expr)`.

Fungsi `empty()` digunakan untuk menghapus semua anak dari suatu elemen. Bila dimiliki halaman web dengan kode seperti ini.

```
<p>
    Hello, <span>Person</span> <a href="javascript:;">and person</a>
</p>
<button>Call empty() on above paragraph</button>
```

Dengan antarmuka seperti pada gambar berikut ini.

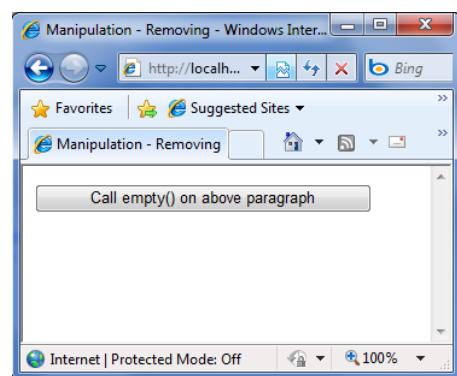


Gambar 2-71 : Antarmuka awal.

Setelah digunakan fungsi `empty()` seperti berikut ini.

```
$( "p" ).empty();
```

Maka anak dari elemen `p` akan dihapus, sehingga akan dilihat antarmuka seperti berikut.



Gambar 2-72 : Antarmuka setelah anak elemen p dihapus dengan fungsi `empty()`.

Fungsi lain yang dapat digunakan adalah `remove(expr)`. Berikut ini contoh penggunaanya.

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Manipulation - Removing</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        $(document).ready(function() {
            $("button").click(function() {
                $("p").remove(":contains('Hello')");
            });
        });
    </script>
</head>
<body>
    <p class="hello">Hello</p>
    how are
    <p>you?</p>

    <button>Remove</button>
</body>
</html>
```

Copying

Berikut ini ada dua fungsi yang tergolong dalam kelompok Copying. Salah satunya adalah `clone()`. Fungi ini berguna untuk menyalin suatu elemen ke lokasi lain di dalam DOM. Sebagai contoh adalah kode berikut ini.

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Manipulation - Copying</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        $(document).ready(function() {
            $("b").clone().prependTo("p");
        });
    </script>
</head>
<body>
    <b>Hello</b>
    <br />
    <p>, how are you?</p>
</body>
</html>
```

Selain itu juga bisa digunakan fungsi `clone(true)`, fungsi ini selain menyalin elemen beserta sifat fisiknya juga dapat menyalin event yang dimiliki oleh elemen tersebut. Untuk melihat perbedaan antara fungsi `clone()` dan `clone(true)` dapat diperhatikan kedua contoh berikut ini.

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Manipulation - Copying</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        $(document).ready(function() {
            $("button").click(function() {
                $(this).clone().insertAfter(this);
            });
        });
    </script>
</head>
<body>
    <button>Clone Me!</button>
</body>
</html>
```

Pada contoh di atas ketika tombol diklik akan dibuat tombol baru, tetapi tombol yang mempunyai respon ketika tombol diklik hanyalah tombol pertama. Tombol-tombol hasil cloning tidak mempunyai event klik, sehingga tidak akan ada respon ketika tombol-tombol tersebut diklik.

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Manipulation - Copying</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        $(document).ready(function() {
            $("button").click(function() {
                $(this).clone(true).insertAfter(this);
            });
        });
    </script>
</head>
<body>
    <button>Clone Me!</button>
</body>
</html>
```

Dengan menggunakan fungsi `clone(true)` pada setiap tombol akan mempunyai kemampuan yang sama dengan tombol aslinya, artinya setiap tombol cloning tersebut diklik maka akan terjadi respon.

Mengelola CSS

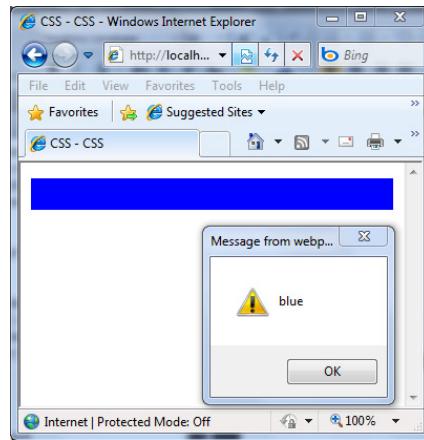
CSS

css(name)

Fungsi ini digunakan untuk mengambil nilai suatu style pada suatu elemen.

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>CSS - CSS</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        $(document).ready(function() {
            var color = $("div").css("background-color");
            alert(color);
        });
    </script>
    <style type="text/css">
        div
        {
            width:100%;
            height:30px;
        }
    </style>
</head>
<body>
    <div style="background-color:blue;"></div>
</body>
</html>
```

Contoh di atas akan memilih elemen `div` dan mengambil nilai dari property `background-color` yang ada di dalam `style`.



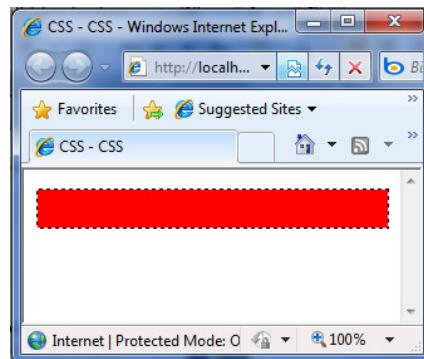
Gambar 2-73 : Mengambil property di dalam style dengan fungsi css(name).

css(properties)

Fungsi ini berfungsi untuk memberikan style pada suatu elemen.

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>CSS - CSS</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        $(document).ready(function() {
            $("div").css(
            {
                width:"100%",
                height:"30px",
                border: "dashed 1px #000",
                backgroundColor:"Red"
            });
        });
    </script>
</head>
<body>
    <div></div>
</body>
</html>
```

Hasilnya adalah elemen div seperti digambar berikut ini.



Gambar 2-74 : Pemberian style dengan fungsi css(properties).

Yang perlu diperhatikan adalah terjadi perubahan penamaan property `style`. Khusus property yang terdiri dua kata seperti `background-color` atau `font-weight` maka penulisannya menjadi `backgroundColor` dan `fontWeight`.

css(name, value)

Fungsi ini digunakan untuk memberikan sebuah property `style` beserta nilainya.

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>CSS - CSS</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        $(document).ready(function() {
            $("div").css("border", "solid 1px #000");
            $("div").css("height", "30px");
        });
    </script>
</head>
<body>
    <div></div>
</body>
</html>
```

Positioning

Fungsi yang dapat digunakan untuk mengetahui posisi suatu elemen adalah `offset()`. Berikut adalah contoh penggunaannya.

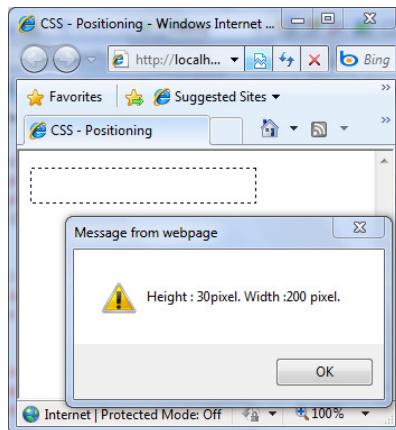
```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>CSS - Positioning</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        $(document).ready(function() {
            var p = $("p:last");
            var offset = p.offset();
            p.html("left: " + offset.left + ", top: " + offset.top);
            var div = $("div");
            var divOffset = div.offset();
            div.html("left: " + divOffset.left + ", top: " + divOffset.top);
        });
    </script>
    <style type="text/css">
        div
        {
            width:200px;
            height:30px;
            border:dashed 1px #000;
        }
    </style>
</head>
<body>
    <p>Hello</p>
    <p>Hello</p>
    <p>2nd Paragraph</p>
    <div style="position:absolute;top:32px;left:300px">test</div>
</body>
</html>
```

Height & Width

Kedua fungsi ini berguna untuk mengambil dan memberikan nilai `height` dan `width` pada suatu elemen.

Berikut ini contoh penggunaan fungsi `height()` dan `width()`.

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>CSS - Height & Width</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        $(document).ready(function() {
            var div = $("div");
            alert("Height : " + div.height() + "px Width :" + div.width()+" p");
        });
    </script>
    <style type="text/css">
        div
        {
            width:200px;
            height:30px;
            border:dashed 1px #000;
        }
    </style>
</head>
<body>
    <div>
    </div>
</body>
</html>
```



Gambar 2-75 : Menampilkan lebar dan tinggi elemen div.

Untuk memberikan nilai tinggi dan lebar suatu elemen dapat digunakan fungsi `width(value)` dan `height(value)` seperti di bawah ini.

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>CSS - Height & Width</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        $(document).ready(function() {

        });

        function Resize() {
            var lebar = $("#lebar").val();
            var tinggi = $("#tinggi").val();
            $("div").height(tinggi);
```

```

        $("div").width(lebar);
    }
</script>
<style type="text/css">
div
{
    width:200px;
    height:30px;
    border:dashed 1px #000;
}
</style>
</head>
<body>
    Lebar <input type="text" id="lebar" name="lebar" />,
    Tinggi <input type="text" id="tinggi" name="tinggi" />
    <input type="button" value="Resize" onclick="Resize()" />
    <div>
        </div>
</body>
</html>

```

Event

Page Load

Fungsi `ready()` yang telah dibahas pada bab pertama dan telah sering digunakan pada contoh-contoh di atas adalah satu-satunya fungsi yang merupakan golongan dari Page Load.

Event Handling

`bind(type, data, fn)`

Fungsi ini digunakan untuk mendaftar handler pada suatu event (seperti `click`) pada suatu elemen. Misalnya ingin diberikan event click pada tombol, dimana saat tombol diklik akan ditampilkan message box alert.

Berikut ini adalah contoh penggunaan fungsi `bind` untuk tujuan di atas.

```

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Events - Event Handling</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        $(document).ready(function() {
            $("#btn").bind("click", function() {
                alert("Hello jQuery");
            });
        });
    </script>
</head>
<body>
    <input id="btn" type="button" value="Show Alert Box" />
</body>
</html>

```

Berikut ini contoh lain penggunaan fungsi `bind`.

```

<html xmlns="http://www.w3.org/1999/xhtml">
<head>

```

```

<title>Events - Event Handling</title>
<script src="script/jquery-1.3.2.js" type="text/javascript"></script>
<script language="javascript" type="text/javascript">
    $(document).ready(function() {
        $("div").bind("dblclick", changeBG);
    });

    function changeBG() {
        $("div").toggleClass("redBG");
    }
</script>
<style type="text/css">
div
{
    border:dashed 1px #000;
    padding:13px;
    margin:13px;
}

.redBG
{
    background-color:Red;
}
</style>
</head>
<body>
    <div>
        click me!
    </div>
</body>
</html>

```

Contoh di atas akan memberikan event double clik (`dblclick`) pada elemen `div`. Event ini akan memanggil fungsi `changeBG()` yang akan mengubah warna latar elemen `div`.

Pada kedua contoh di atas, parameter yang digunakan dan diisi hanya `type` dan `fn`. Contoh berikut ini, selain menggunakan parameter `type` dan `fn`, digunakan juga parameter `data`.

```

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Events - Event Handling</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        $(document).ready(function() {
            $("#tombol").bind("click", { foo: "bar" }, handler);
        });

        function handler(event) {
            alert(event.data.foo);
        }
    </script>
</head>
<body>
    <input id="tombol" type="button" value="Show Data" />
</body>
</html>

```

unbind(type, data)

Fungsi ini mempunyai kemampuan yang berkebalikan dengan fungsi `bind(type, data, fn)`. Fungsi ini digunakan untuk menghapus handler yang pada event pada suatu elemen. Berikut ini contoh penggunaan fungsi `unbind(type, data)`.

```

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Events - Event Handling</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        $(document).ready(function() {

```

```

        $("#bind").click(function() {
            $("#theone").click(aClick)
                .text("Can Click!");
        });
        $("#unbind").click(function() {
            $("#theone").unbind('click', aClick)
                .text("Does nothing...");
        });
    });

    function aClick() {
        alert("Can Click!");
    }

```

Dengan mengklik tombol dengan `id=bind`, maka akan didaftarkan handler pada event click pada tombol dengan `id=theone`. Sedangkan tombol dengan `id=unbind` digunakan untuk menghapus handler dan event pada tombol dengan `id=theone` tersebut.

Cara lain yang bisa digunakan untuk menghapus suatu handler pada event yang ada pada tombol dengan `id=theone` adalah sebagai berikut ini.

```
$("#theone").unbind()
```

Atau.

```
$("#theone").unbind("click")
```

one(type, data, fn)

Fungsi mempunyai sifat yang sama seperti fungsi `bind(type, data, fn)`, kecuali handler pada event yang ada pada elemen hanya bisa digunakan sekali saja.

Sebagai contoh dapat digunakan kode berikut ini.

```

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Events - Event Handling</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        $(document).ready(function() {
            $("#btn").one("click", function() {
                alert("Hello jQuery");
            });

            $("div").one("dblclick", changeBG);

            $("#tombol").one("click", { foo: "bar" }, handler);
        });

        function changeBG() {
            $("div").toggleClass("redBG");
        }

        function handler(event) {
            alert(event.data.foo);
        }
    </script>
    <style type="text/css">
        div
    {

```

```

        border:dashed 1px #000;
        padding:13px;
        margin:13px;
    }

    .redBG
    {
        background-color:Red;
    }

```

</style>

</head>

<body>

<input id="btn" type="button" value="Show Alert Box" />

<div>

click me!

</div>

<input id="tombol" type="button" value="Show Data" />

</body>

</html>

Tombol dengan `id=btn`, hanya dapat diklik dan menampilkan alert satu kali, ketika tombol ini diklik kembali message box alert tidak akan ditampilkan lagi. Begitu juga elemen `div`, event double click yang mengubah background elemen ini hanya berlaku sekali.

trigger(type, data)

Fungsi ini digunakan untuk mentriger handler yang dimiliki oleh suatu even pada sebuah elemen. Berikut contoh dari penggunaan fungsi `trigger(type, data)`.

```

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Events - Event Handling</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        $(document).ready(function() {
            $("button:first").click(function() {
                update($(".span:first"));
            });
            $("button:last").click(function() {
                $("button:first").trigger('click');

                update($(".span:last"));
            });
        });

        function update(j) {
            var n = parseInt(j.text(), 0);
            j.text(n + 1);
        }
    </script>
</head>
<body>
    <button>Button #1</button>

    <button>Button #2</button>
    <div><span>0</span> button #1 clicks.</div>
    <div><span>0</span> button #2 clicks.</div>
</body>
</html>

```

Pada contoh ini, ketika tombol #2 diklik, dipanggil baris berikut ini.

```
$(“button:first”).trigger(‘click’);
```

Jadi ketika tombol #2 diklik, secara otomatis event click pada tombol #1 juga akan dilakukan, sehingga handler dari event click pada tombol #1 dieksekusi.

triggerHandler(type, data)

Berbeda dengan sifat fungsi `trigger(type, data)`, fungsi `triggerHandler(type, data)` hanya akan mentrigger event handler yang dimiliki oleh elemen, tanpa mengeksekusi aksi default dari browser.

Untuk melihat perbedaan antara sifat fungsi `trigger(type, data)` dan `triggerHandler(type, data)` dapat dilihat dari hasil contoh di bawah ini.

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Events - Event Handling</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        $(document).ready(function() {
            $("#old").click(function() {
                $("input").trigger("focus");
            });
            $("#new").click(function() {
                $("input").triggerHandler("focus");
            });
            $("input").focus(function() {
                $("<span>Focused!</span>").appendTo("body").fadeOut(1000);
            });
        });
    </script>
</head>
<body>
    <button id="old">.trigger("focus")</button>
    <button id="new">.triggerHandler("focus")</button><br/><br/>

    <input type="text" value="To Be Focused"/>
</body>
</html>
```

Ketika tombol dengan `id=old` diklik dan fungsi `trigger` dipanggil maka akan dilihat kata `Focused!`. Sedangkan bila tombol dengan `id=new` diklik, maka hanya keluar kata berikut `Focused!`. Hal ini terjadi karena fungsi `triggerHandler(type, data)` tidak memanggil aksi default dari browser.

Interaction Helper

hover(over, out)

Fungsi ini digunakan untuk mendaftarkan handler pada event hover pada suatu elemen. Parameter `over` berisi fungsi yang akan dieksekusi ketika pointer mouse berada pada area elemen yang dipilih (`over`), dan parameter `out` adalah fungsi yang akan dieksekusi ketika pointer mouse keluar dari area elemen (`out`).

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Events - Interaction Helpers </title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        $(document).ready(function() {
            $("div").hover(
                function() {
                    $(this).toggleClass("redBG");
                },
                function() {
                    $(this).toggleClass("redBG");
                }
            );
        });
    </script>
</head>
<body>
    <div>Hover Me</div>
</body>
</html>
```

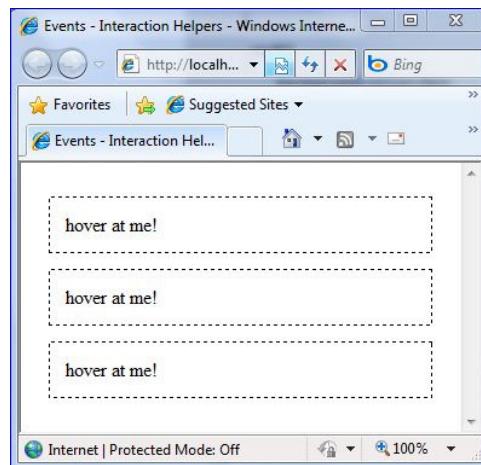
```

        });
    </script>
    <style type="text/css">
    div
    {
        border:dashed 1px #000;
        padding:13px;
        margin:13px;
    }

    .redBG
    {
        background-color:Red;
    }
    </style>
</head>
<body>
    <div>
        hover at me!
    </div>
    <div>
        hover at me!
    </div>
    <div>
        hover at me!
    </div>
</body>
</html>

```

Contoh di atas akan menghasilkan 3 buah kotak seperti gambar berikut ini.



Gambar 2-76 : Kotak menu.

Ketika pointer mouse berada di area salah satu elemen `div` tersebut, secara otomatis background elemen tersebut berubah merah, dan background akan kembali menjadi putih ketika pointer mouse berada diluar area elemen tersebut.

toggle(fn, fn)

Fungsi ini mempunyai sifat yang mirip dengan fungsi `hover(over, out)`, bedanya fungsi ini akan memberikan handler kepada event click pada suatu elemen. Contoh berikut ini mirip dengan contoh dari fungsi `hover(over, out)` kecuali fungsi `hover` diganti dengan fungsi `toggle`.

```

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
```

```

<title>Events - Interaction Helpers </title>
<script src="script/jquery-1.3.2.js" type="text/javascript"></script>
<script language="javascript" type="text/javascript">
    $(document).ready(function() {
        $("div").toggle(
            function() {
                $(this).toggleClass("redBG");
            },
            function() {
                $(this).toggleClass("redBG");
            }
        );
    });
</script>
<style type="text/css">
div
{
    border:dashed 1px #000;
    padding:13px;
    margin:13px;
}

.redBG
{
    background-color:Red;
}
</style>
</head>
<body>
    <div>
        hover at me!
    </div>
    <div>
        hover at me!
    </div>
    <div>
        hover at me!
    </div>
</body>
</html>

```

Pada contoh di atas, background elemen `div` akan berubah menjadi merah ketika diklik, dan akan berubah menjadi putih kembali ketika elemen `div` yang sama diklik kembali.

Event Helper

blur() dan blur(fn)

Beberapa elemen HTML mempunyai event `onblur`. Event ini akan dipanggil ketika pointer keluar dari elemen.

Sebagai contoh dapat dilihat pada contoh di bawah ini.

```

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Event - Event Helpers</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        function ShowMessage() {
            var message = $("#message").val();
            alert(message);
        }
    </script>
</head>
<body>
    Pesan : <input id="message" type="text" onblur="ShowMessage()" />
</body>
</html>

```

Dari contoh di atas, ketika pointer keluar dari elemen `input` dengan `id=message`, maka event `onblur` akan memanggil fungsi `ShowMessage()`.

Fungsi `blur(fn)` digunakan untuk mendaftarkan handler dan event kepada suatu elemen. Sedangkan fungsi `blur()` digunakan untuk men-trigger event `blur` pada suatu elemen yang dipilih.

Berikut ini contoh penggunaan fungsi `blur()` dan `blur(fn)`.

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Event - Event Helpers</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        $(document).ready(function() {
            $("#message").blur>ShowMessage();
            $("#btn").bind("click", function() { $("#message").blur(); });
        });

        function ShowMessage() {
            var message = $("#message").val();
            alert(message);
        }
    </script>
</head>
<body>
    Pesan : <input id="message" type="text" />
    <input id="btn" type="button" value="trigger blur" />
</body>
</html>
```

Pada contoh di atas dapat dilihat bagaimana handler dan event `blur` didaftarkan pada `input` dengan `id=message`, serta ditunjukkan juga bagaimana caranya men-trigger event `blur` saat tombol dengan `id=btn` diklik.

change() dan change(fn)

Event `onchange` pada elemen HTML akan ditrigger ketika nilai pada suatu elemen berubah. Event ini dapat dimiliki oleh input dengan type `text` atau elemen `select`. Berikut ini contoh yang dapat dilihat untuk mengetahui cara kerja event `onchange`.

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Event - Event Helpers</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        function ShowCountry() {
            var message = $("#country").val();
            alert(message);
        }
    </script>
</head>
<body>
    <select id="country" onchange="ShowCountry()">
        <option value="Indonesia">Indonesia</option>
        <option value="Jepang">Jepang</option>
        <option value="Korea">Korea</option>
    </select>
</body>
</html>
```

Ketika nilai dari elemen `select` berubah, maka fungsi `ShowCountry` akan dipanggil.

Selanjutnya adalah penggunaan fungsi `change(fn)` dan `change()`.

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
```

```

<title>Event - Event Helpers</title>
<script src="script/jquery-1.3.2.js" type="text/javascript"></script>
<script language="javascript" type="text/javascript">
    $(document).ready(function() {
        $("select").change(function() {
            var str = "";
            $("select option:selected").each(function() {
                str += $(this).text() + " ";
            });
            $("div").text(str);
        })
        .change();
    });
</script>
</head>
<body>
    <select name="sweets" multiple="multiple">
        <option>Chocolate</option>
        <option selected="selected">Candy</option>
        <option>Taffy</option>
        <option selected="selected">Carmel</option>
        <option>Fudge</option>
        <option>Cookie</option>
    </select>
    <div></div>
</body>
</html>

```

click() dan click(fn)

Fungsi `click(fn)` digunakan untuk mendaftarkan handler dan event `click` pada suatu elemen. Sedangkan fungsi `click()` dapat digunakan untuk mentrigger event `click`.

```

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Event - Event Helpers</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        $(document).ready(function() {
            $("div").click(ShowMessage);
            $("#btn").click(function() { $("div").click(); });
        });

        function ShowMessage() {
            alert("Hello jQuery");
        }
    </script>
    <style type="text/css">
div
{
    border:dashed 1px #000;
    padding:13px;
    margin:13px;
}
    </style>
</head>
<body>
    <div>Click me!</div>
    <input id="btn" type="button" value="trigger click" />
</body>
</html>

```

dblclick() dan dblclick(fn)

Event `ondblclick` akan mengeksekusi suatu fungsi jika suatu elemen diklik dua kali berturut-turut. Berikut ini contoh penggunaan kedua fungsi ini.

```
<html xmlns="http://www.w3.org/1999/xhtml">
```

```

<head>
    <title>Event - Event Helpers</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        $(document).ready(function() {
            $("div").dblclick(ShowMessage);
            $("#btn").click(function() { $("div").dblclick(); });
        });

        function ShowMessage() {
            alert("Hello jQuery");
        }
    </script>
    <style type="text/css">
    div
    {
        border:dashed 1px #000;
        padding:13px;
        margin:13px;
    }
    </style>
</head>
<body>
    <div>Click me!</div>
    <input id="btn" type="button" value="trigger dblclick" />
</body>
</html>

```

Pada contoh di atas, ketika elemen `div` diklik double, maka akan ditampilkan message box. Sedangkan ketika tombol trigger `dblclick` ditekan sekali, maka event `dblclick` pada elemen `div` akan dieksekusi.

focus() dan focus(fn)

Event `onfocus` terjadi ketika suatu elemen dipilih oleh pointer atau dengan menggunakan tombol tab. Berikut ini contoh yang bisa dilihat.

```

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Event - Event Helpers</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        $(document).ready(function() {
            $("#username").focus(ShowMessage);
            $("#btn").click(function() { $("#username").focus(); });
        });

        function ShowMessage() {
            alert("Silakan isikan nilai username.");
        }
    </script>
</head>
<body>
    <input type="text" id="username" />
    <input type="button" id="btn" value="trigger focus" />
</body>
</html>

```

Fungsi-fungsi yang sejenis dengan fungsi-fungsi yang telah dijelas di atas adalah :

- a. `keypress()` dan `keypress(fn)`, digunakan untuk menangani event `onkeypress`. Event ini dibangkitkan ketika tombol keyboard ditekan.
- b. `keyup()` dan `keyup(fn)`, digunakan untuk menangani event `onkeyup`. Event ini dibangkitkan ketika tombol keyboard diangkat setelah proses tombol keyboard ditekan.

- c. `keydown()` dan `keydown(fn)`, digunakan untuk menangani event `onkeypress`. Event ini dibangkitkan ketika tombol keyboard ditekan.
- d. `select()` dan `select(fn)`, digunakan untuk menangani event `onselect`. Event ini dibangkitkan ketika user memilih kata yang berada di dalam text field seperti `input` dan `textarea`.
- e. `submit()` dan `submit(fn)`, digunakan untuk menangani event `onsubmit`. Event ini dibangkitkan ketika form di-submit
- f. `load(fn)` dan `unload(fn)`, Fungsi load akan menangani event `onload`, event ini dibangkitkan ketika halaman telah di-load. Sedangkan fungsi `unload` menangani event `onunload` yang mempunyai sifat yg terbalik dengan event `load`.
- g. `error()` dan `error(fn)`, event error terjadi ketika elemen kehilangan `focus` dari pointer atau navigasi tab.
- h. `mousedown(fn)`, digunakan untuk menangani event `onmousedown`. Event ini dibangkitkan ketika mouse diklik pada suatu elemen.
- i. `mouseout(fn)`, digunakan untuk menangani event `onmouseout`. Event ini dibangkitkan ketika pointer mouse keluar dari area suatu elemen.
- j. `mouseover(fn)`, digunakan untuk menangani event `onmouseover`. Event ini dibangkitkan ketika pointer mouse berada dalam area suatu elemen.
- k. `mouseup(fn)`, digunakan untuk menangani event `onmouseup`. Event ini dibangkitkan ketika tombol mouse dilepas saat berada di dalam area suatu elemen.

mousemove(fn)

Fungsi ini digunakan untuk menangani ketika pointer mouse bergerak dalam suatu elemen. Sebagai contoh dapat dilihat pada kode berikut ini.

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Event - Event Helpers</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        $(document).ready(function() {
            $("div").mousemove(function(e) {
                var pageCoords = " (" + e.pageX + ", " + e.pageY + " )";
                var clientCoords = " (" + e.clientX + ", " + e.clientY + " )";
                $("span:first").text("(" + e.pageX + ", " + e.pageY + " ) - " + pageCoords);
                $("span:last").text("(" + e.clientX + ", " + e.clientY + " ) - " + clientCoords);
            });
        });
    </script>
    <style type="text/css">
        div
        {
            width:100%;
            height:300px;
            border:dashed 1px #000;
            cursor:pointer;
        }
    </style>
</head>
<body>
    <p>
        Try scrolling too.
        <span>Move the mouse over the div.</span>
        <span>&ampnbsp</span>
    </p>

    <div></div>
</body>
</html>
```

Contoh di atas akan menampilkan posisi mouse (koordinat x dan y) saat berada di dalam area elemen `div`.

resize(fn)

Event `resize` akan dibangkitkan ketika dokumen atau jendela web browser di-resize. Berikut ini contoh penggunaan fungsi `resize(fn)`.

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Event - Event Helpers</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        $(document).ready(function() {
            $(window).resize(function() { alert('Stop it!'); });
        });
    </script>
</head>
<body>
    <p>Please resize this window!</p>
</body>
</html>
```

scroll(fn)

Event `scroll` akan dibangkitkan ketika terjadi tampilan dokumen web digerakkan ke atas atau ke bawah (scroll).

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Event - Event Helpers</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        $(document).ready(function() {
            $("p").clone().appendTo(document.body);
            $("p").clone().appendTo(document.body);
            $("p").clone().appendTo(document.body);
            $(window).scroll(function() {
                $("span").css("display", "inline").fadeOut("slow");
            });
        });
    </script>
    <style type="text/css">
        div
        {
            width:100%;
            height:1000px;
            border:dashed 1px #000;
        }
    </style>
</head>
<body>
    <p>Paragraph - <span>Scroll happened!</span></p>
    <div></div>
</body>
</html>
```

Effect

Basic

hide dan show

Fungsi `hide()` digunakan untuk menyembunyikan atau menghilangkan suatu elemen dari penglihatan (⌚). Fungsi `hide(speed,callback)` mempunyai sifat yang sama, kecuali ada tambahan efek animasi saat elemen yang dipilih disembunyikan.

Sedangkan fungsi `show()` dan `show(speed, callback)` mempunyai sifat yang berkebalikan.

Berikut ini contoh penggunaan keempat fungsi di atas.

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Effects - Basics</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        $(document).ready(function() {
            $("#btn1").click(function() { $("div").hide(); });
            $("#btn2").click(function() { $("div").hide("slow"); });
            $("#btn3").click(function() { $("div").show(); });
            $("#btn4").click(function() { $("div").show("slow"); });
        });
    </script>
    <style type="text/css">
        div
        {
            width:230px;
            height:50px;
            border:dashed 1px #000;
            margin:13px 13px 13px 0;
            padding:13px;
            background-color:Red;
        }
    </style>
</head>
<body>
    <input type="button" id="btn1" value="Hide" />
    <input type="button" id="btn2" value="Hide with animation" />
    <input type="button" id="btn3" value="Show" />
    <input type="button" id="btn4" value="Show with animation" />
    <div></div>
</body>
</html>
```

toggle()

Fungsi ini berguna untuk mengubah status elemen `hide` menjadi `show`, dan apabila status elemen adalah `show` akan diubah menjadi `hide`.

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Effects - Basics</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        $(document).ready(function() {
            $("#btn").click(function() { $("div").toggle(); });
        });
    </script>
    <style type="text/css">
        div
        {
            width:230px;
            height:50px;
            border:dashed 1px #000;
            margin:13px 13px 13px 0;
        }
    </style>
</head>
<body>
    <input type="button" id="btn" value="Toggle" />
    <div></div>
</body>
</html>
```

```

        padding:13px;
        background-color:Red;
    }

```

```

</style>
</head>
<body>
    <input type="button" id="btn" value="toggle" />
    <div></div>
</body>
</html>

```

Sliding

Pada kelompok ini terdapat tiga fungsi yaitu :

- a. `slideDown(speed, callback)`, berfungsi untuk mengubah nilai tinggi suatu elemen menuju nilai maksimal, sesuai dengan nilai yang sudah diberikan.
- b. `slideUp(speed, callback)`, berfungsi untuk mengubah nilai tinggi suatu elemen menuju nol.
- c. `slideToggle(speed, callback)`, berfungsi mengubah status suatu elemen dari `slideDown` menjadi `slideUp`, dan ketika fungsi ini dieksekusi lagi maka yang terjadi adalah mengubah `slideUp` menjadi `slideDown`.

Contoh di bawah ini menjutkan penggunaan ketiga fungsi di atas.

```

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Effects - Sliding</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        $(document).ready(function() {
            $("#btn1").click(function() { $("div").slideDown("slow"); });
            $("#btn2").click(function() { $("div").slideUp("slow"); });
            $("#btn3").click(function() { $("div").slideToggle("slow"); });
        });
    </script>
    <style type="text/css">
        div
        {
            width:230px;
            height:30px;
            border:dashed 1px #000;
            padding:5px;
            margin:5px;
            cursor:pointer;
        }
    </style>
</head>
<body>
    <div></div>
    <input type="button" value="slide down" id="btn1" />
    <input type="button" value="slide up" id="btn2" />
    <input type="button" value="slide toggle" id="btn3" />
    <div></div>
</body>
</html>

```

Ketika tombol *slide up* diklik maka kedua tinggi elemen `div` akan berkurang sampai elemen `div` menghilang, dan ketika tombol *slide down* diklik maka elemen `div` akan muncul kembali secara perlahan seiring dengan meningkatnya nilai `height` dari elemen tersebut.

Fading

Fadding adalah proses pengubahan nilai `opacity` dari suatu elemen. Fungsi `fadeIn(speed, callback)` digunakan untuk menaikkan nilai `opacity` sampai ke nilai maksimal yaitu 1, fungsi `fadeOut(speed, callback)` digunakan untuk menurunkan nilai `opacity` sampai ke nilai minimal yaitu 0. Sedangkan fungsi `fadeTo(speed, opacity, callback)` digunakan untuk mengubah nilai `opacity` dari nilai default ke nilai yang diperikan pada parameter `opacity`.

Berikut ini contoh penggunaan ketiga fungsi ini.

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Effects - Fading</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        $(document).ready(function() {
            $("#btn1").click(function() { $("div").fadeIn(3000); });
            $("#btn2").click(function() { $("div").fadeOut(3000); });
            $("#btn3").click(function() { $("div").fadeTo("slow", 0.33); });
        });
    </script>
    <style type="text/css">
        div
        {
            width:230px;
            height:30px;
            border:dashed 1px #000;
            padding:5px;
            margin:5px;
            background-color:Red;
        }
    </style>
</head>
<body>
    <div></div>
    <input type="button" value="fadeIn" id="btn1" />
    <input type="button" value="fadeOut" id="btn2" />
    <input type="button" value="fadeTo" id="btn3" />
    <div></div>
</body>
</html>
```

Custom

Pada bagian ini akan dijelaskan fungsi-fungsi yang dapat digunakan untuk membuat animasi sendiri. Ada beberapa fungsi yang dapat digunakan, yaitu :

- a. `animate(params, options).`
- b. `animate(params, duration, easing, callback).`
- c. `dequeue().`
- d. `queue().`
- e. `queue(callback).`
- f. `queue(queue).`
- g. `stop().`

Berikut ini akan diberi contoh untuk membuat animasi sederhana, animasi yang mengubah ukuran font dari besar menjadi kecil dan sebaliknya.

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Effects - Custom</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        $(document).ready(function() {
```

```

        $("#go1").click(function() {
            $("#block1").animate({ width: "90%" }, { queue: false, duration:
3000 })
                .animate({ fontSize: "24px" }, 1500)
                .animate({ borderRightWidth: "15px" }, 1500);
        });

        $("#go2").click(function() {
            $("#block2").animate({ width: "90%" }, 1000)
                .animate({ fontSize: "24px" }, 1000)
                .animate({ borderLeftWidth: "15px" }, 1000);
        });

        $("#go3").click(function() {
            $("#go1").add("#go2").click();
        });

        $("#go4").click(function() {
            $("div").css({ width: "", fontSize: "", borderWidth: "" });
        });
    });

```

</script>

</head>

<body>

- <button id="go1">» Animate Block1</button>
- <button id="go2">» Animate Block2</button>
- <button id="go3">» Animate Both</button>

- <button id="go4">» Reset</button>

<div id="block1">Block1</div>

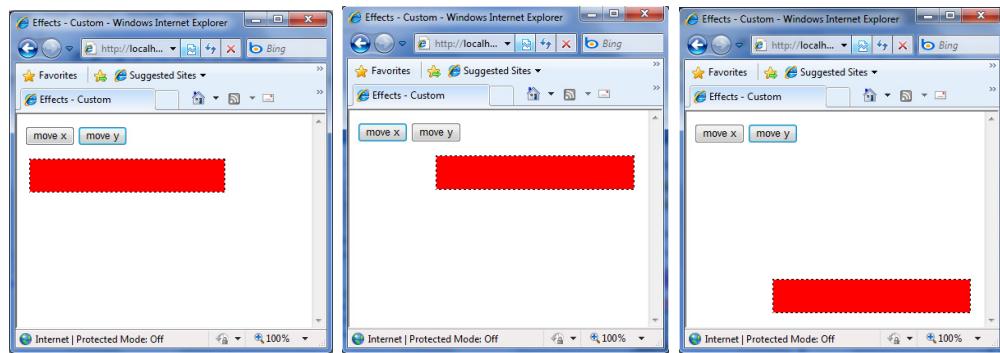
<div id="block2">Block2</div>

Contoh yang lain dapat dilihat pada kode di bawah ini.

```

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Effects - Custom</title>
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script language="javascript" type="text/javascript">
        $(document).ready(function() {
            $("#btn-x").click(
                function() {
                    $("div").animate({ left: "+=10px" }, 1000);
                }
            );
            $("#btn-y").click(
                function() {
                    $("div").animate({ top: "+=10px" }, 1000);
                }
            );
        });
    </script>
    <style type="text/css">
        div
        {
            width:230px;
            height:30px;
            border:dashed 1px #000;
            padding:5px;
            margin:5px;
            background-color:Red;
        }
    </style>
</head>
<body>
    <input id="btn-x" type="button" value="move x" />
    <input id="btn-y" type="button" value="move y" />
    <div style="position:absolute;top:50px;left:10px;"></div>
</body>
</html>

```



Gambar 2-77 : Inisial, tombol x ditekan, tombol y ditekan.

Pada contoh di atas akan dilihat hasilnya seperti pada gambar, ketika tombol *move x* ditekan maka atribut *left* dari elemen *div* akan diubah menjadi *10px* lebih besar. Sedangkan ketika tombol *move y* di tekan, maka yang terjadi adalah nilai atribut *top* akan diubah menjadi *10px* lebih besar.

3.

jQuery UI & Plug-in

jQuery UI

Bagi web developer yang menggunakan ASP.NET, dan bekerja dengan ASP.NET Ajax mungkin juga mengenal Ajax Control Toolkit (ACT). ACT digunakan di atas ASP.NET, artinya ACT tidak dapat hidup tanpa ASP.NET Ajax. Pada ACT terdapat control-control yang interaktif seperti :

1. Accordion.
2. HTMLEditor.
3. Tab.
4. Slider.
5. Dan lain-lain.

Seperti ACT, begitu pula jQuery UI. jQuery UI harus digunakan di atas jQuery, karena jQuery UI merupakan library interaksi, efek dan widget-widget seperti control-control yang ada pada ACT. Artinya pada jQuery UI akan dapat ditemui widget seperti :

1. Tab.
2. Accordion.
3. Dialog.
4. Datepicker.
5. Progressbar.
6. Dan lain-lain.

Paket library jQuery UI dapat diunduh di <http://jqueryui.com>. Setelah mengunduh paket library tersebut dan mengekstraknya, maka akan didapati file-file berekstension JS yang dapat digunakan untuk membuat widget yang diinginkan atau memanfaatkan fitur interaksi dan efek.

Interaksi

Pada library interaksi terdapat beberapa fungsi yang dapat digunakan untuk mempermudah developer membuat web yang interaktif. Contoh-contoh fungsi yang disediakan oleh jQuery UI untuk interaksi dapat dilihat pada penjelasan-penjelasan di bawah ini.

Draggable

jQuery UI mampu memberikan kemampuan *drag (draggable)* pada elemen DOM yang diinginkan. Dengan mengklik elemen yang telah diberikan kemampuan *drag*, kemudian menggerakkannya ke area yang diinginkan.

Berikut ini adalah contoh sederhana penggunaan jQuery UI untuk membuat sebuah elemen DOM dapat di-drag.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Draggable</title>
    <!-- Script -->
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script src="script/ui.core.js" type="text/javascript"></script>
    <script src="script/ui.draggable.js" type="text/javascript"></script>

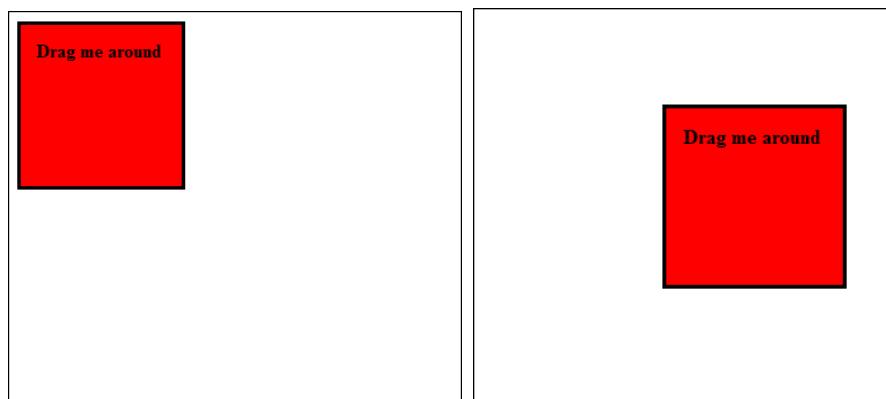
    <!-- Script memilih elemen untuk diberikan kemampuan drag -->
    <script language="javascript" type="text/javascript">
        $(document).ready(function() {
            $("#draggable").draggable();
        });
    </script>

    <link href="css/ui.all.css" rel="stylesheet" type="text/css" />
    <style type="text/css">
        #draggable
        {
            width:113px;
            height:113px;
            border:solid 3px #000;
            padding:13px;
            color:White;
            font-weight:bold;
            background-color:Red;
        }
    </style>
</head>
<body>

    <div id="draggable">
        <p>Drag me around</p>
    </div>

</body>
</html>
```

Hasil dari contoh ini akan dapat dilihat kotak seperti pada gambar di bawah ini yang dapat digerakkan pada area yang diinginkan.



Gambar 3-1 : Kotak yang mempunyai kemampuan di-drag.

Pada contoh di atas dapat dilihat diperlukan beberapa file JS yang diperlukan agar kemampuan drag dapat dilekatkan pada elemen yang diinginkan, yaitu :

1. jquery-1.3.2.js, file ini mutlak diperlukan oleh jQuery UI agar bisa digunakan.

2. ui.core.js adalah library dasar dari jQuery UI.
3. ui.draggable.js adalah library yang berisi fungsi-fungsi untuk menempelkan kemampuan drag kepada elemen.

Langkah selanjutnya tinggal memilih elemen dengan nilai `id` adalah `draggable`, kemudian berikan kemampuan drag dengan cara seperti berikut.

```
$("#draggable").draggable();
```

Bila ternyata element yang ingin diberikan kemampuan drag lebih dari satu, maka dapat digunakan cara seperti berikut ini.

```
$("#elemen-1", "#elemen-2", "#elemen-n").draggable();
```

Event

Kemampuan drag yang diberikan jQuery UI tidak terbatas pada contoh di atas. Pada contoh berikut ini akan ditunjukkan cara untuk menangkap event saat suatu elemen di-drag. Event yang bisa ditangkap adalah :

1. start, event ini terjadi saat pertama kali elemen diangkat untuk digerakkan.
2. drag, event ini terjadi saat elemen digerakkan.
3. stop, event ini terjadi saat elemen berhenti digerakkan dan tombol kiri mouse dilepaskan.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Draggable</title>
    <!-- Script -->
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script src="script/ui.core.js" type="text/javascript"></script>
    <script src="script/ui.draggable.js" type="text/javascript"></script>

    <!-- Script memilih elemen untuk diberikan kemampuan drag -->
    <script language="javascript" type="text/javascript">
        $(document).ready(function() {
            $("#draggable").draggable({
                start: function() {
                    $("#start").val(parseInt($("#start").val()) + 1);
                },
                drag: function() {
                    $("#drag").val(parseInt($("#drag").val()) + 1);
                },
                stop: function() {
                    $("#stop").val(parseInt($("#stop").val()) + 1);
                }
            });
        });
    </script>

    <link href="css/ui.all.css" rel="stylesheet" type="text/css" />
    <style type="text/css">
        #draggable
        {
            width:113px;
            height:113px;
            border:solid 3px #000;
            padding:13px;
            color:Black;
            font-weight:bold;
            background-color:Red;
        }
    </style>

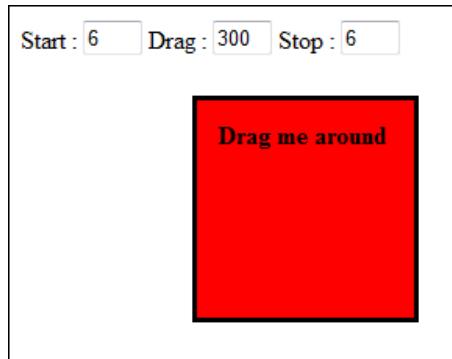
```

```

        }
        input
        {
            width:32px;
        }
    </style>
</head>
<body>
    Start : <input type="text" id="start" value="0" />
    Drag : <input type="text" id="drag" value="0" />
    Stop : <input type="text" id="stop" value="0" />
    <div id="draggable">
        <p>Drag me around</p>
    </div>
</body>
</html>

```

Hasilnya dapat dilihat seperti pada gambar di bawah ini.



Gambar 3-2 : Menggunakan event.

Pada saat kotak diklik dan digerakkan, secara otomatis nilai pada Start akan bertambah. Sedangkan nilai pada Drag akan terus bertambah selama kotak digerakkan, setelah kotak dilepas maka nilai Stop akan bertambah.

```

    $("#draggable").draggable({
        start: function() {
            // perintah yg akan dijalankan ketika event start ditangkap
        },
        drag: function() {
            // perintah yg akan dijalankan ketika event drag ditangkap
        },
        stop: function() {
            // perintah yg akan dijalankan ketika event stop ditangkap
        }
    });

```

Membatasi Gerak

Pada contoh sebelumnya elemen yang sudah memiliki kemampuan drag, dapat digerakkan ke berbagai arah. Pada bagian ini akan diberikan contoh untuk membatasi gerak elemen tersebut.

Berikut ini contoh agar elemen hanya dapat digerakkan secara vertikal atau horizontal saja.

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head>

```

```

<title>Draggable</title>
<!-- Script -->
<script src="script/jquery-1.3.2.js" type="text/javascript"></script>
<script src="script/ui.core.js" type="text/javascript"></script>
<script src="script/ui.draggable.js" type="text/javascript"></script>

<!-- Script memilih elemen untuk diberikan kemampuan drag -->
<script language="javascript" type="text/javascript">
    $(document).ready(function() {
        $("#draggable-x").draggable({ axis: 'x' });
        $("#draggable-y").draggable({ axis: 'y' });
    });
</script>

<link href="css/ui.all.css" rel="stylesheet" type="text/css" />
<style type="text/css">
    #draggable-x, #draggable-y
    {
        width:113px;
        height:113px;
        border:solid 3px #000;
        margin:13px;
        color:Black;
        font-weight:bold;
        background-color:Red;
    }
</style>
</head>
<body>

    <div id="draggable-x">
        <p>Horizontal</p>
    </div>

    <div id="draggable-y">
        <p>Vertikal</p>
    </div>

</body>
</html>

```

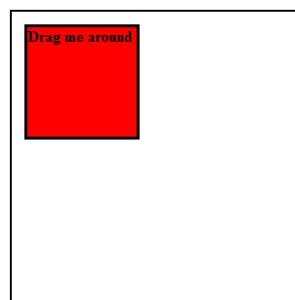
Pada contoh di atas, dapat dilihat bahwa untuk membuat elemen hanya bergerak horizontal maka digunakan baris berikut ini.

```
$("#draggable-x").draggable({ axis: 'x' });
```

Sedangkan untuk membuat elemen hanya dapat bergerak vertikal maka digunakan baris berikut.

```
$("#draggable-y").draggable({ axis: 'y' });
```

Contoh yang lain adalah membuat elemen hanya dapat bergerak dalam area tertentu saja, misalnya seperti pada gambar di bawah ini.



Gambar 3-3 : Elemen hanya dapat bergerak di dalam kotak.

Dan berikut ini adalah contoh kodanya.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Draggable</title>
    <!-- Script -->
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script src="script/ui.core.js" type="text/javascript"></script>
    <script src="script/ui.draggable.js" type="text/javascript"></script>

    <!-- Script memilih elemen untuk diberikan kemampuan drag -->
    <script language="javascript" type="text/javascript">
        $(document).ready(function() {
            $("#draggable").draggable({ containment: '#area-gerak' });
        });
    </script>

    <link href="css/ui.all.css" rel="stylesheet" type="text/css" />
    <style type="text/css">
        #draggable
        {
            width:113px;
            height:113px;
            border:solid 3px #000;
            margin:13px;
            color:Black;
            font-weight:bold;
            background-color:Red;
        }

        #area-gerak
        {
            width:300px;
            height:300px;
            border:solid 2px #000;
        }
    </style>
</head>
<body>
    <div id="area-gerak">
        <div id="draggable">
            <p>Drag me around</p>
        </div>
    </div>
</body>
</html>
```

Area gerak adalah elemen `div` dengan id bernilai `area-gerak`. Sehingga agar elemen `div` dengan id bernilai `draggable` tidak keluar dari area gerak maka digunakan baris berikut ini.

```
$("#draggable").draggable({ containment: '#area-gerak' });
```

Dengan menggunakan atribut `containment` dan menunjuk id dari elemen pembatas atau area gerak maka, elemen yang dipilih oleh fungsi selector hanya akan dapat di-drag di dalam area elemen pembatas tersebut saja.

Menunda Gerak

Gerakan elemen yang akan didrag dapat ditunda berdasarkan jarak atau waktu. Ada dua atribut yang dapat digunakan, yaitu :

1. `distance`, penundaan gerak akan dilakukan berdasarkan jarak yang diberikan sebagai nilai dari atribut ini. Misal diberikan nilai 20, maka setelah cursor mouse bergerak 20px baru elemen akan bergerak mengikuti gerak mouse.

- delay, penundaan gerak berdasarkan waktu. Bila nilai atribut ini adalah 1000 maka elemen akan bergerak setelah 1 detik.

Berikut ini adalah contohnya.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Draggable</title>
    <!-- Script -->
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script src="script/ui.core.js" type="text/javascript"></script>
    <script src="script/ui.draggable.js" type="text/javascript"></script>

    <!-- Script memilih elemen untuk diberikan kemampuan drag -->
    <script language="javascript" type="text/javascript">
        $(document).ready(function() {
            $("#draggable-distance").draggable({ distance: '32' });
            $("#draggable-delay").draggable({ delay: '1000' });
        });
    </script>

    <link href="css/ui.all.css" rel="stylesheet" type="text/css" />
    <style type="text/css">
        #draggable-distance, #draggable-delay
        {
            width:113px;
            height:113px;
            border:solid 3px #000;
            margin:13px;
            color:Black;
            font-weight:bold;
            background-color:Red;
        }
    </style>
</head>
<body>

    <div id="draggable-distance">
        <p>Horizontal</p>
    </div>

    <div id="draggable-delay">
        <p>Vertikal</p>
    </div>

</body>
</html>
```

Snap

Fitur ini memungkinkan untuk mengatur gerak elemen agar elemen tersebut dapat menempel seperti kancing magnet pada elemen yang diinginkan. Berikut ini contoh untuk kasus ini.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Draggable</title>
    <!-- Script -->
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script src="script/ui.core.js" type="text/javascript"></script>
    <script src="script/ui.draggable.js" type="text/javascript"></script>

    <!-- Script memilih elemen untuk diberikan kemampuan drag -->
    <script language="javascript" type="text/javascript">
```

```

$(document).ready(function() {
    $("#draggable").draggable();
    $("#draggable-snap").draggable({ snap: true });
});
</script>

<link href="css/ui.all.css" rel="stylesheet" type="text/css" />
<style type="text/css">
    #draggable
    {
        width:213px;
        height:213px;
        border:solid 3px #000;
        margin:13px;
        color:Black;
        font-weight:bold;
        background-color:Red;
    }

    #draggable-snap
    {
        width:113px;
        height:113px;
        border:solid 3px #000;
        margin:13px;
        color:Black;
        font-weight:bold;
        background-color:Blue;
    }
</style>
</head>
<body>

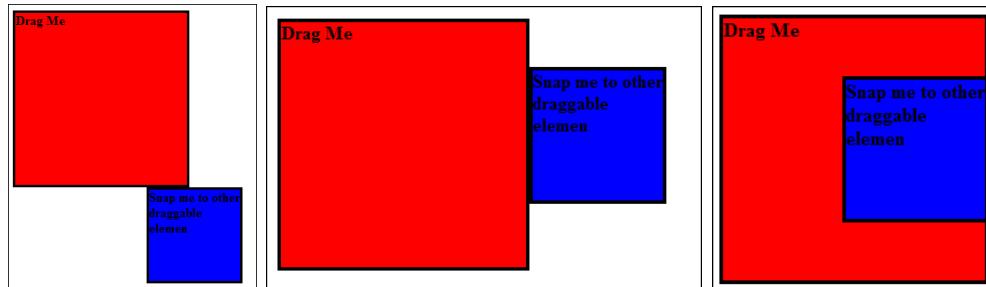
    <div id="draggable">
        <p>Drag Me</p>
    </div>

    <div id="draggable-snap">
        <p>Snap me to other draggable elemen</p>
    </div>

</body>
</html>

```

Hasil dari contoh diatas dapat dilihat pada gambar di bawah ini.



Gambar 3-4 : Kotak biru dapat menempel pada sisi luar dan dalam kotak merah.

Bila kotak biru digerakkan mendekati sisi kotak merah maka secara otomatis kotak biru seperti tertarik untuk menempel pada sisi kotak merah. Pada contoh di atas kotak biru dapat menempel pada sisi luar dan sisi dalam kotak merah.

Apabila yang menginginkan elemen hanya dapat menempel pada elemen tertentu saja maka dapat digunakan baris seperti berikut ini.

```
$("#draggable-snap").draggable({ snap: '#draggable' });
```

Artinya elemen dengan nilai `id=draggable-snap` hanya dapat menempel pada elemen yang mempunyai nilai `id=draggable`.

Pada contoh di atas, elemen dapat menempel pada sisi dalam atau luar suatu elemen, bila menginginkan elemen hanya dapat menempel pada sisi luar suatu elemen saja maka dapat digunakan cara seperti berikut ini.

```
$("#draggable-snap").draggable({ snap: true, snapMode:'outer' });
```

Sedangkan contoh berikut adalah cara apabila ingin elemen hanya bisa menempel pada sisi dalam saja.

```
$("#draggable-snap").draggable({ snap: true, snapMode:'inner' });
```

Selain itu gerak elemen dapat dibuat seperti meloncat berdasarkan jarak tertentu. Dengan menggunakan baris berikut ini, maka gerak elemen saat di-drag akan seperti meloncat-loncat setiap 80px. Atau bisa juga beranggapan bahwa elemen berada didalam grid yang dibangun oleh persegi dengan sisi 80x80.

```
$("#draggable-snap").draggable({ grid: [80, 80] });
```

Kembali ke Posisi

Pada contoh di bawah ini akan diperlihatkan cara agar elemen yang didrag akan kembali ke posisi asalnya secara otomatis.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Draggable</title>
    <!-- Script -->
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script src="script/ui.core.js" type="text/javascript"></script>
    <script src="script/ui.draggable.js" type="text/javascript"></script>

    <!-- Script memilih elemen untuk diberikan kemampuan drag -->
    <script language="javascript" type="text/javascript">
        $(document).ready(function() {
            $("#draggable").draggable({revert:true});
        });
    </script>

    <link href="css/ui.all.css" rel="stylesheet" type="text/css" />
    <style type="text/css">
        #draggable
        {
            width:73px;
            height:73px;
            border:solid 3px #000;
            padding:13px;
            color:Black;
            font-weight:bold;
            background-color:Red;
        }
    </style>
</head>
<body>

    <div id="draggable">
        <p>Drag me around</p>
    </div>

</body>
</html>
```

Cukup menggunakan baris dibawah ini maka dapat dibuat efek agar elemen kembali ke posisi semula.

```
$("#draggable").draggable({revert:true});
```

Apabila ingin mendapatkan cara yang lebih cantik maka dapat digunakan cara berikut ini.

```
$("#draggable").draggable({revert:true, helper:'clone'});
```

Dengan menggunakan baris di atas, saat elemen digerakkan masih dapat dilihat elemen kembaran yang berada pada posisi awal.

Drag handle

Pada bagian ini akan ditunjukkan bahwa suatu elemen hanya dapat digerakkan bila suatu area tertentu saja yang diklik oleh cursor mouse.



Gambar 3-5 : Drag handle.

Pada gambar diatas kotak putih di atas hanya dapat digerakkan bila cursor memilih area kotak berwarna abu-abu.

Berikut adalah contoh kode yang dapat digunakan.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Draggable</title>
    <!-- Script -->
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script src="script/ui.core.js" type="text/javascript"></script>
    <script src="script/ui.draggable.js" type="text/javascript"></script>

    <!-- Script memilih elemen untuk diberikan kemampuan drag -->
    <script language="javascript" type="text/javascript">
        $(document).ready(function() {
            $("#draggable").draggable({ handle: 'p' });
        });
    </script>

    <link href="css/ui.all.css" rel="stylesheet" type="text/css" />
    <style type="text/css">
        #draggable
        {
            width:113px;
            height:113px;
            border:solid 3px #000;
            color:Black;
            font-weight:bold;
            background-color:Red;
        }
    </style>
</head>
<body>
    <div id="draggable">
        I can be dragged only by this handle
    </div>
</body>

```

```

        }

        #draggable p
        {
            width:113px;
            height:30px;
            background-color:Gray;
            text-align:center;
        }
    </style>
</head>
<body>

<div id="draggable">
    <p>Header</p>
</div>

</body>
</html>

```

Berdasarkan baris berikut ini :

```
$("#draggable").draggable({ handle: 'p' });
```

Maka elemen dengan `id=draggable` hanya dapat di-drag apabila area yang diklik mouse adalah elemen `p`. Selain atribut `handle`, dikenal juga atribut `cancel` seperti contoh di bawah ini.

```
$("#draggable").draggable({ cancel: 'p' });
```

Bila baris di atas digunakan, maka elemen hanya dapat digerakkan apabila area yang dipilih bukan elemen `p`.

Dropable

Pada bagian ini akan dipaparkan mengenai bagaimana jQuery UI memberikan kemampuan `dropable` pada suatu elemen DOM. Caranya sama sesederhana seperti jQuery UI memberikan kemampuan `draggable` yang telah dijelaskan di atas.

Berikut ini contohnya.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Dropable</title>
    <!-- Script -->
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script src="script/ui.core.js" type="text/javascript"></script>
    <script src="script/ui.draggable.js" type="text/javascript"></script>
    <script src="script/ui.droppable.js" type="text/javascript"></script>

    <!-- Script memilih elemen untuk diberikan kemampuan drag -->
    <script language="javascript" type="text/javascript">
        $(document).ready(function() {
            $("#draggable").draggable();
            $("#droppable").droppable({
                drop: function() {
                    $(this).find('p').html('Dropped!');
                }
            });
        });
    </script>

    <link href="css/ui.all.css" rel="stylesheet" type="text/css" />
    <style type="text/css">
```

```

#draggable
{
    width:75px;
    height:75px;
    border:solid 1px #000;
    margin:13px;
    padding:3px;
    color:Black;
    font-weight:bold;
    background-color:Red;
}

#droppable
{
    width:132px;
    height:132px;
    border:solid 1px #000;
    margin:13px;
    padding:3px;
    color:Black;
    font-weight:bold;
    background-color:Yellow;
}

```

</style>

</head>

<body>

<div id="draggable">

<p>Drag me around</p>

</div>

<div id="droppable">

<p>Drop something to me</p>

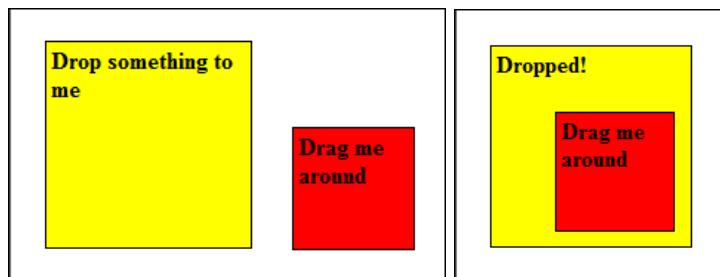
</div>

</body>

</html>

Dari contoh di atas maka dapat dilihat terdapat tambahanan library jQuery UI, yaitu :

```
<script src="script/ui.droppable.js" type="text/javascript"></script>
```



Gambar 3-6 : Kemampuan droppable pada kotak kuning.

Kotak kuning adalah elemen dengan `id=droppable`, sedangkan kotak merah adalah elemen dengan `id=draggable`. Sehingga untuk membuat kasus seperti pada gambar di atas, digunakan baris seperti berikut ini.

```

$(document).ready(function() {
    $("#draggable").draggable();
    $("#droppable").droppable({
        drop: function() {
            $(this).find('p').html('Dropped!');
        }
    });
});

```

Pada contoh di atas digunakan event `drop` yang akan memanggil suatu fungsi. Pada kasus di atas fungsi akan melakukan pengubahan text di dalam kotak kuning.

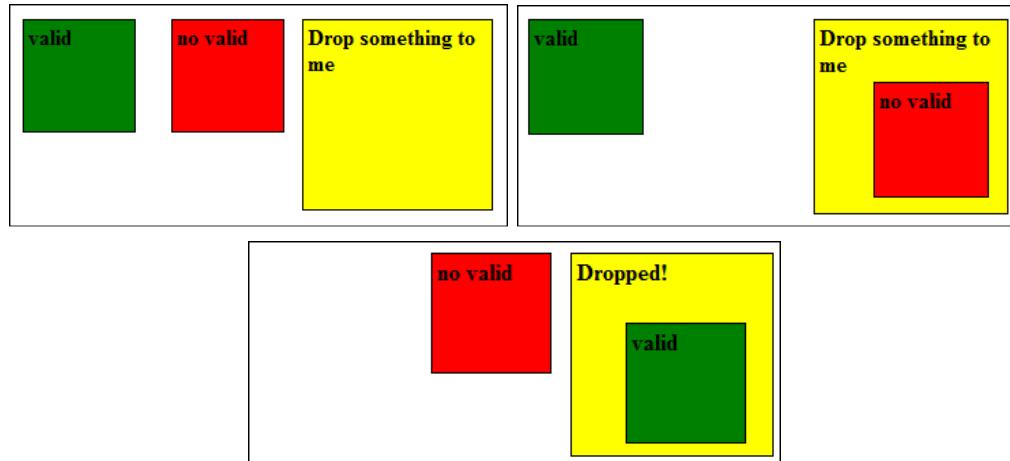
Selain mempunyai atribut `drop` yang dapat digunakan untuk menangani suatu event, ada beberapa atribut lain yang bisa digunakan diantaranya adalah seperti contoh berikut ini.

```
$("#draggable").draggable({  
    activeClass: 'active',  
    hoverClass: 'hover',  
    drop: function() {  
        $(this).find('p').html('Dropped!');  
    }  
});
```

Dapat dilihat pada contoh di atas terdapat dua atribut `activeClass` dan `hoverClass`. Atribut lain adalah `accept`. Atribut ini berisi elemen yang diijinkan untuk *di-drop* pada elemen yang mempunyai kemampuan *dropable*.

Accept

Pada contoh berikut ini akan diperlihatkan penggunaan atribut `accept`.



Gambar 3-7 : Kotak kuning hanya menerima kotak hijau.

Berikut ini adalah contoh kode yang bisa dicoba untuk membuat seperti pada contoh kasus di atas.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">  
  
<html xmlns="http://www.w3.org/1999/xhtml">  
<head>  
    <title>Dropable</title>  
    <link href="css/ui.all.css" rel="stylesheet" type="text/css" />  
    <style type="text/css">  
        #draggable, #draggable-novalid  
        {  
            width:75px;  
            height:75px;  
            border:solid 1px #000;  
            margin:13px;  
            padding:3px;  
            color:Black;  
            font-weight:bold;  
            background-color:Green;  
            float:left;  
        }  
    </style>
```

```

#draggable-novalid
{
    background-color:Red;
}

#droppable
{
    width:132px;
    height:132px;
    border:solid 1px #000;
    margin:13px;
    padding:3px;
    color:Black;
    font-weight:bold;
    background-color:Yellow;
}
</style>

<!-- Script -->
<script src="script/jquery-1.3.2.js" type="text/javascript"></script>
<script src="script/ui.core.js" type="text/javascript"></script>
<script src="script/ui.draggable.js" type="text/javascript"></script>
<script src="script/ui.droppable.js" type="text/javascript"></script>

<!-- Script memilih elemen untuk diberikan kemampuan drag -->
<script language="javascript" type="text/javascript">
$(document).ready(function() {
    $("#draggable, #draggable-novalid").draggable();
    $("#droppable").droppable({
        accept: '#draggable',
        drop: function() {
            $(this).find('p').html('Dropped!');
        }
    });
});
</script>
</head>
<body>

<div id="draggable">
    <p>valid</p>
</div>

<div id="draggable-novalid">
    <p>no valid</p>
</div>

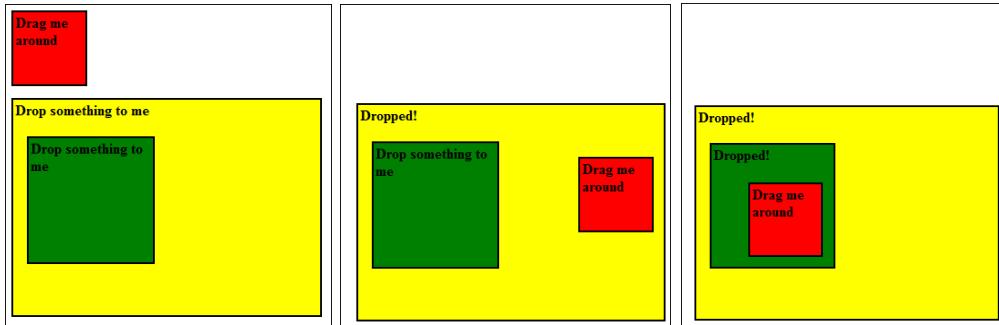
<div id="droppable">
    <p>Drop something to me</p>
</div>

</body>
</html>

```

Greedy

Pada bagian ini akan diperlihatkan penggunaan atribut `greedy` pada fungsi `dropable`. Misalnya dimiliki elemen dengan struktur seperti gambar di bawah ini.



Gambar 3-8 : Tanpa atribut greedy.

Bila kotak merah diletakkan pada area kuning maka event drop pada kotak kuning akan dieksekusi, tetapi event drop pada kotak hijau tidak dijalankan. Tetapi bila kotak merah diletakkan pada kotak hijau maka event drop pada kotak kuning dan hijau akan dijalankan.

Berikut ini contoh kode untuk kasus di atas.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Dropable</title>
    <link href="css/ui.all.css" rel="stylesheet" type="text/css" />
    <style type="text/css">
        #draggable
        {
            width:75px;
            height:75px;
            border:solid 2px #000;
            margin:13px;
            padding:2px;
            color:Black;
            font-weight:bold;
            background-color:Red;
        }

        #droppable, #droppable-inner
        {
            width:332px;
            height:232px;
            border:solid 2px #000;
            margin:13px;
            padding:2px;
            color:Black;
            font-weight:bold;
            background-color:Yellow;
        }

        #droppable-inner
        {
            width:132px;
            height:132px;
            background-color:Green;
        }
    </style>

    <!-- Script -->
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script src="script/ui.core.js" type="text/javascript"></script>
    <script src="script/ui.draggable.js" type="text/javascript"></script>
    <script src="script/ui.droppable.js" type="text/javascript"></script>
```

```

<!-- Script memilih elemen untuk diberikan kemampuan drag -->
<script language="javascript" type="text/javascript">
    $(document).ready(function() {
        $("#draggable").draggable();
        $("#droppable, #droppable-inner").droppable({
            drop: function() {
                $(this).find('> p').html('Dropped!');
            }
        });
    });
</script>
</head>
<body>

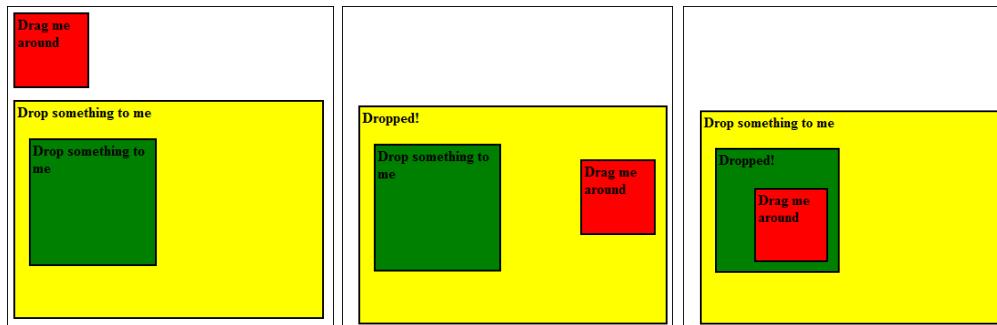
    <div id="draggable">
        <p>Drag me around</p>
    </div>

    <div id="droppable">
        <p>Drop something to me</p>
        <div id="droppable-inner">
            <p>Drop something to me</p>
        </div>
    </div>

</body>
</html>

```

Bila diinginkan kasus seperti pada gambar di bawah ini.



Gambar 3-8 : Menggunakan atribut greedy.

Perbedaan contoh ini dengan contoh sebelumnya adalah dapat dilihat pada gambar ketiga. Yaitu saat kotak merah diletakkan di dalam kotak hijau, maka event drop kuning tidak dieksekusi tapi hanya event drop pada kotak hijau saja yang dijankan.

Perbedaan kodennya hanya dengan menambahkan atribut greedy=true, seperti yang dilihat pada kode di bawah ini.

```

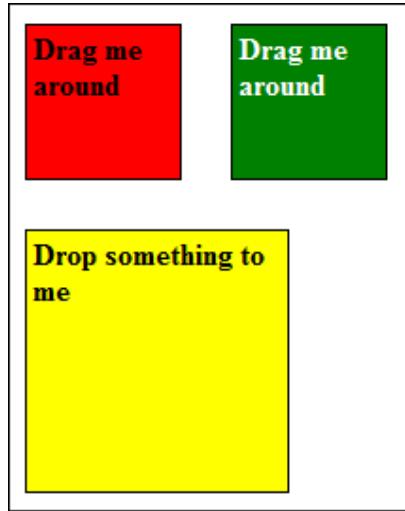
$("#droppable, #droppable-inner").droppable({
    greedy:true,
    drop: function() {
        $(this).find('> p').html('Dropped!');
    }
});

```

Posisi

Pada pembahasan draggable, telah dikenal atribut revert pada elemen draggable. Bila menggunakan atribut ini maka dapat dilihat elemen draggable akan kembali ke posisi awal setelah di-drag.

Elemen dropable dapat menangkap elemen draggable ketika elemen ini di-drop di dalam area elemen dropable. Artinya elemen draggable tidak akan kembali ke posisi awal. Elemen ini hanya akan kembali ke posisi awal bila tidak diletakkan di dalam area dropable.



Gambar 3-9 : Atribut revert pada elemen draggable.

Bila kotak merah diletakkan di dalam kotak kuning maka kotak merah akan kembali ke posisi awal. Sedangkan bila kotak hijau diletakkan pada kotak kuning maka kotak kuning akan menangkapnya.

Berikut ini contoh kode yang dapat digunakan.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Dropable</title>
    <link href="css/ui.all.css" rel="stylesheet" type="text/css" />
    <style type="text/css">
        #draggable-valid, #draggable-invalid
        {
            width:75px;
            height:75px;
            border:solid 1px #000;
            margin:13px;
            padding:3px;
            color:Black;
            font-weight:bold;
            background-color:Red;
            float:left;
        }

        #draggable-invalid
        {
            background-color:Green;
            color:White;
        }

        #droppable
        {
            width:132px;
            height:132px;
            border:solid 1px #000;
            margin:13px;
            padding:3px;
            color:Black;
            font-weight:bold;
            background-color:Yellow;
        }
    </style>
```

```

<!-- Script -->
<script src="script/jquery-1.3.2.js" type="text/javascript"></script>
<script src="script/ui.core.js" type="text/javascript"></script>
<script src="script/ui.draggable.js" type="text/javascript"></script>
<script src="script/ui.droppable.js" type="text/javascript"></script>

<!-- Script memilih elemen untuk diberikan kemampuan drag -->
<script language="javascript" type="text/javascript">
    $(document).ready(function() {
        $("#draggable-valid").draggable({ revert: 'valid' });
        $("#draggable-invalid").draggable({ revert: 'invalid' });
        $("#droppable").droppable({
            activeClass: 'active',
            hoverClass: 'hover',
            drop: function(event, ui) {
                $(this).find('p').html('Dropped!');
            }
        });
    });
</script>
</head>
<body>

    <div id="draggable-valid">
        <p>Drag me around</p>
    </div>

    <div id="draggable-invalid">
        <p>Drag me around</p>
    </div>

    <div style="clear:both"></div>

    <div id="droppable">
        <p>Drop something to me</p>
    </div>

</body>
</html>

```

Tidak ada penambahan atribut pada fungsi dropable, tetapi justru dilakukan penambahan pada fungsi draggable, seperti pada contoh di bawah ini.

```

$("#draggable-valid").draggable({ revert: 'valid' });
$("#draggable-invalid").draggable({ revert: 'invalid' });

```

Resizable

Berikut ini adalah contoh sederhana penggunaan fungsi resizable.

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Resizable</title>
    <!-- Script -->
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script src="script/ui.core.js" type="text/javascript"></script>
    <script src="script/ui.resizable.js" type="text/javascript"></script>

    <style type="text/css">
        #resizable { width: 150px; height: 150px; padding: 0.5em; border:solid 1px #000 }
        #resizable h3 { text-align: center; margin: 0; }
    </style>
    <link href="css/ui.resizable.css" rel="stylesheet" type="text/css" />

    <script type="text/javascript">
        $(document).ready(function() {
            $("#resizable").resizable();

```

```

        });
    </script>
</head>
<body>
    <div id="resizable" class="ui-widget-content">
        <h3>Resizable</h3>
    </div>
</body>
</html>

```

Pada kode di atas, digunakan beberapa tambahan file, yaitu :

1. ui.resizable.js, berisi fungsi-fungsi yang digunakan untuk implementasi fitur resizable pada elemen.
2. ui.resizable.css, berisi style-style yang akan membuat antarmuka elemen mudah dikenali sebagai elemen yang dapat di-resize, misalnya bentuk icon yang berubah saat pada sisi elemen yang ingin di-resize.

kedua file ini terdapat pada paket jQuery UI yang telah didownload.

Aspect Ratio

Contoh berikut ini adalah implementasi aspect ratio pada elemen yang dapat di-resize. Artinya pembesaran panjang akan berpengaruh ke tinggi, begitu juga sebaliknya. Nilai pembesaran panjang dan tinggi akan sesuai dengan rasio masing-masing.

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Resizable</title>
    <!-- Script -->
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script src="script/ui.core.js" type="text/javascript"></script>
    <script src="script/ui.resizable.js" type="text/javascript"></script>

    <style type="text/css">
        #resizable
        {
            width: 150px;
            height: 150px;
            padding: 0.5em;
            border:solid 1px #000
        }

        #resizable h3
        {
            text-align: center;
            margin: 0;
        }
    </style>
    <link href="css/ui.resizable.css" rel="stylesheet" type="text/css" />

    <script type="text/javascript">
        $(document).ready(function() {
            $("#resizable").resizable({ aspectRatio: 1 / 1 });
        });
    </script>
</head>
<body>
    <div id="resizable" class="ui-widget-content">
        <h3>Resizable</h3>
    </div>
</body>
</html>

```

Pada contoh di atas, dapat dilihat penggunaan atribut `aspectRatio` seperti ini.

```
$("#resizable").resizable({ aspectRatio: 1 / 1 });
```

Dari baris di atas dapat diketahui aspect ratio yang digunakan adalah 1:1.

Mengatur Ukuran

Untuk mengatur tinggi dan lebar maka dapat digunakan atribut-atribut seperti maxHeight, minHeight, maxWidth atau minWidth.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Resizable</title>
    <!-- Script -->
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script src="script/ui.core.js" type="text/javascript"></script>
    <script src="script/ui.resizable.js" type="text/javascript"></script>

    <style type="text/css">
        #resizable
        {
            width: 150px;
            height: 150px;
            padding: 0.5em;
            border:solid 1px #000
        }

        #resizable h3
        {
            text-align: center;
            margin: 0;
        }
    </style>
    <link href="css/ui.resizable.css" rel="stylesheet" type="text/css" />

    <script type="text/javascript">
        $(document).ready(function() {
            $("#resizable").resizable({
                maxHeight: 250,
                maxWidth: 350,
                minHeight: 150,
                minWidth: 150
            });
        });
    </script>
</head>
<body>
    <div id="resizable" class="ui-widget-content">
        <h3>Resizable</h3>
    </div>
</body>
</html>
```

Pembatasan ukuran elemen juga dapat dilakukan berdasarkan ukuran elemen induk atau rumahnya. Berikut ini adalah contohnya.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Resizable</title>
    <!-- Script -->
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script src="script/ui.core.js" type="text/javascript"></script>
    <script src="script/ui.resizable.js" type="text/javascript"></script>
```

```

<style type="text/css">
    #resizable
    {
        width: 150px;
        height: 150px;
        padding: 0.5em;
        border:solid 1px #000;
        background-color:Red;
    }

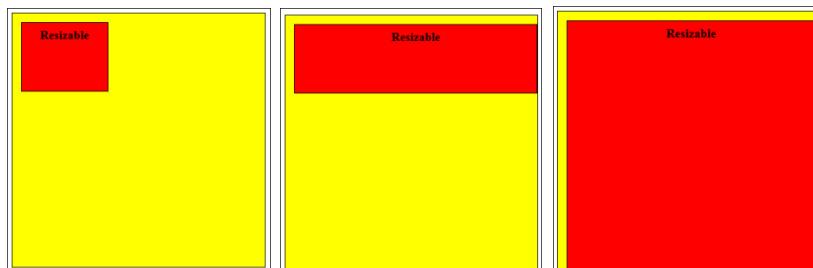
    #parent
    {
        width: 350px;
        height: 350px;
        border:solid 1px #000;
        padding:13px;
        background-color:Yellow;
    }

    #resizable h3
    {
        text-align: center;
        margin: 0;
    }
</style>
<link href="css/ui.resizable.css" rel="stylesheet" type="text/css" />

<script type="text/javascript">
    $(document).ready(function() {
        $("#resizable").resizable({
            containment: '#parent'
        });
    });
</script>
</head>
<body>
    <div id="parent">
        <div id="resizable" class="ui-widget-content">
            <h3>Resizable</h3>
        </div>
    </div>
</body>
</html>

```

Cara kerja contoh di atas adalah seperti gambar berikut ini.



Gambar 3-10 : Penggunaan atribut constraint.

Dari gambar di atas dapat diketahui bahwa ukuran maksimal kotak merah tidak akan melebihi kotak merah. Kotak merah adalah elemen dengan `id=resizable`, sedangkan kotak kuning adalah elemen dengan `id=parent`. Cukup dengan baris di bawah ini, maka kasus di atas bisa terjadi.

```

    $("#resizable").resizable({
        containment: '#parent'
    });

```

Pengaturan ukuran juga dapat dilakukan dengan menggunakan atribut `grid`. Penggunaan atribut ini membuat cara pembesaran elemen bergerak meloncat berdasarkan nilai pada atribut `grid`.

Berikut ini contoh penggunaannya.

```
$("#resizable").resizable({
    containment: '#parent',
    grid:32
});
```

Synchronous Resize

Untuk menggunakan fitur ini dapat dilakukan dengan cara menggunakan atribut `alsoResize`. Nilai atribut bisa diisi dengan id dari elemen lain yang akan di-resize mengikuti ukuran elemen ini.

Berikut ini adalah contoh penggunaannya.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Resizable</title>
    <!-- Script -->
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script src="script/ui.core.js" type="text/javascript"></script>
    <script src="script/ui.resizable.js" type="text/javascript"></script>

    <style type="text/css">
        #resizable, #resizable-child
        {
            width: 113px;
            height: 113px;
            padding: 0.5em;
            border:solid 1px #000;
            background-color:Red;
            margin:13px;
        }

        #resizable-child
        {
            background-color:Yellow;
        }

        #resizable h3
        {
            text-align: center;
            margin: 0;
        }
    </style>
    <link href="css/ui.resizable.css" rel="stylesheet" type="text/css" />

    <script type="text/javascript">
        $(document).ready(function() {
            $("#resizable").resizable({
                alsoResize: '#resizable-child'
            });

            $("#resizable-child").resizable();
        });
    </script>
</head>
<body>
    <div id="resizable" class="ui-widget-content">
        <h3>Resizable</h3>
    </div>

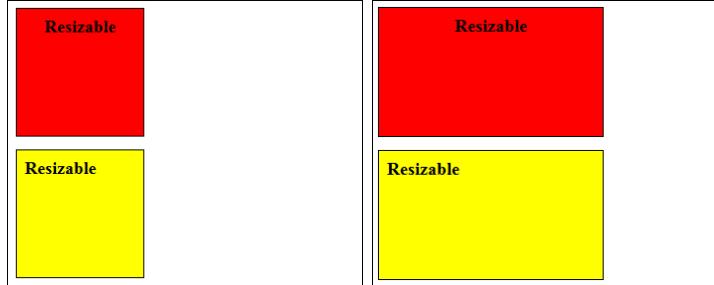
    <div id="resizable-child" class="ui-widget-content">
        <h3>Resizable</h3>
```

```

        </div>
    </body>
</html>

```

Maka akan dilihat seperti pada gambar berikut ini.



Gambar 3-11 : Synchronous resize.

Pembesaran ukuran kotak kuning akan mengikuti ukuran dari kotak merah.

Animasi

Dengan menggunakan atribut animate maka cara perubahan saat elemen diperbesar atau diperkecil akan memiliki efek animasi.

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Resizable</title>
    <!-- Script -->
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script src="script/ui.core.js" type="text/javascript"></script>
    <script src="script/ui.resizable.js" type="text/javascript"></script>

    <style type="text/css">
        #resizable
        {
            width: 150px;
            height: 150px;
            padding: 0.5em;
            border:solid 1px #000
        }

        #resizable h3
        {
            text-align: center;
            margin: 0;
        }

        .ui-resizable-helper { border: 1px dotted gray; }
    </style>
    <link href="css/ui.resizable.css" rel="stylesheet" type="text/css" />

    <script type="text/javascript">
        $(document).ready(function() {
            $("#resizable").resizable({ animate:true });
        });
    </script>
</head>
<body>
    <div id="resizable" class="ui-widget-content">
        <h3>Resizable</h3>
    </div>
</body>
</html>

```

Pada contoh ini selain menambahkan atribut animate seperti berikut.

```
$("#resizable").resizable({ animate:true });
```

Juga ditambahkan style berikut ini, yang berfungsi untuk menampilkan style saat elemen di-resize.

```
.ui-resizable-helper { border: 1px dotted gray; }
```

Selectable

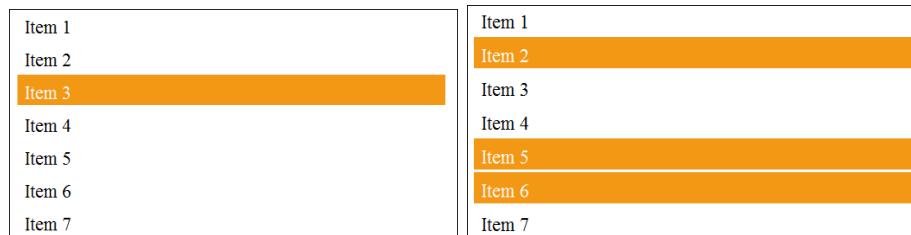
Fungsi selectable merupakan fungsi untuk memilih item-item yang berada di dalam elemen yang diberikan kemampuan selectable. Berikut ini contohnya.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Selectable</title>
    <!-- Script -->
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script src="script/ui.core.js" type="text/javascript"></script>
    <script src="script/ui.selectable.js" type="text/javascript"></script>

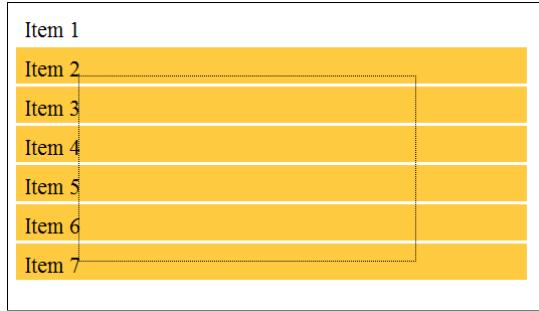
    <style type="text/css">
        #feedback { font-size: 1.4em; }
        #selectable .ui-selecting { background: #FECA40; }
        #selectable .ui-selected { background: #F39814; color: white; }
        #selectable { list-style-type: none; margin: 0; padding: 0; width: 60%; }
        #selectable li { margin: 3px; padding: 0.4em; font-size: 1.4em; height: 18px; }
    </style>
    <script type="text/javascript">
        $(function() {
            $("#selectable").selectable();
        });
    </script>
</head>
<body>
    <ol id="selectable">
        <li class="ui-widget-content">Item 1</li>
        <li class="ui-widget-content">Item 2</li>
        <li class="ui-widget-content">Item 3</li>
        <li class="ui-widget-content">Item 4</li>
        <li class="ui-widget-content">Item 5</li>
        <li class="ui-widget-content">Item 6</li>
        <li class="ui-widget-content">Item 7</li>
    </ol>
</body>
</html>
```

Dari contoh di atas, selain memilih salah satu item, dapat juga memilih lebih dari satu item yang diinginkan. Seperti pada gambar berikut ini.



Gambar 3-12 : Memilih item yang berada di dalam elemen selectable.

Cara pemilihan item juga dapat dilakukan dengan memblok item-item dengan cursor mouse.



Gambar 3-13 : Memilih item dengan cara diblok.

Untuk mengetahui item-item mana saja yang telah dipilih, dapat digunakan event `stop` seperti contoh di bawah ini.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Selectable</title>
<!-- Script -->
<script src="script/jquery-1.3.2.js" type="text/javascript"></script>
<script src="script/ui.core.js" type="text/javascript"></script>
<script src="script/ui.selectable.js" type="text/javascript"></script>

<style type="text/css">
#feedback { font-size: 1.4em; }
#selectable .ui-selecting { background: #FECA40; }
#selectable .ui-selected { background: #F39814; color: white; }
#selectable { list-style-type: none; margin: 0; padding: 0; width: 60%; }
#selectable li { margin: 3px; padding: 0.4em; font-size: 1.4em; height: 18px; }
</style>
<script type="text/javascript">
$(function() {
    $("#selectable").selectable({
        stop: function() {
            var result = $("#select-result").empty();
            $(".ui-selected", this).each(function() {
                var index = $("#selectable li").index(this);
                result.append(" #" + (index + 1));
            });
        }
    });
    </script>
</head>
<body>
<p id="feedback">
    You've selected: <span id="select-result">none</span>.
</p>

<ol id="selectable">
    <li class="ui-widget-content">Item 1</li>
    <li class="ui-widget-content">Item 2</li>
    <li class="ui-widget-content">Item 3</li>
    <li class="ui-widget-content">Item 4</li>
    <li class="ui-widget-content">Item 5</li>
    <li class="ui-widget-content">Item 6</li>
    <li class="ui-widget-content">Item 7</li>
</ol>

</body>
</html>
```

Sortable

Berikut ini contoh penggunaan fungsi sortable.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

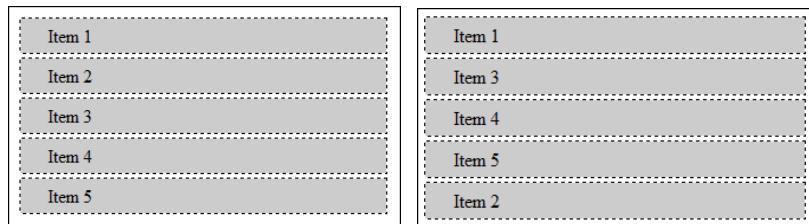
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Sortable</title>
    <!-- Script -->
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script src="script/ui.core.js" type="text/javascript"></script>
    <script src="script/ui.sortable.js" type="text/javascript"></script>

    <style type="text/css">
        #sortable { list-style-type: none; margin: 0; padding: 0; width: 60%; }
        #sortable li
        {
            margin: 0 3px 3px 3px;
            padding: 0.4em;
            padding-left: 1.5em;
            font-size: 1em;
            height: 18px;
            border:dashed 1px #000;
            background-color:#ccc;
        }

        #sortable li span { position: absolute; margin-left: -1.3em; }
    </style>

    <script type="text/javascript">
        $(function() {
            $("#sortable").sortable();
        });
    </script>
</head>
<body>
    <ul id="sortable">
        <li class="ui-state-default">
            <span class="ui-icon ui-icon-arrowthick-2-n-s"></span>Item 1
        </li>
        <li class="ui-state-default">
            <span class="ui-icon ui-icon-arrowthick-2-n-s"></span>Item 2
        </li>
        <li class="ui-state-default">
            <span class="ui-icon ui-icon-arrowthick-2-n-s"></span>Item 3
        </li>
        <li class="ui-state-default">
            <span class="ui-icon ui-icon-arrowthick-2-n-s"></span>Item 4
        </li>
        <li class="ui-state-default">
            <span class="ui-icon ui-icon-arrowthick-2-n-s"></span>Item 5
        </li>
    </ul>
</body>
</html>
```

Dari contoh di atas maka akan dilihat gambar seperti berikut ini.

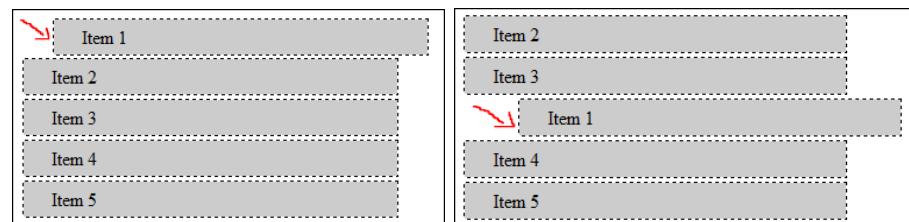


Gambar 3-14 : Item 2 dipindahkan keposisi paling bawah.

Pada gambar di atas dapat dilihat perbedaan antara gambar kiri dan kanan yang terletak pada posisi Item 2. Untuk memindahkannya cukup dengan memilih Item 2 dan ditarik ke lokasi yang diinginkan maka secara otomatis Item 2 tersusun dengan rapi pada barisan.

Placeholder

Placeholder adalah lokasi tempat asal item atau lokasi tujuan item, seperti yang ditunjuk oleh panah berwarna merah.



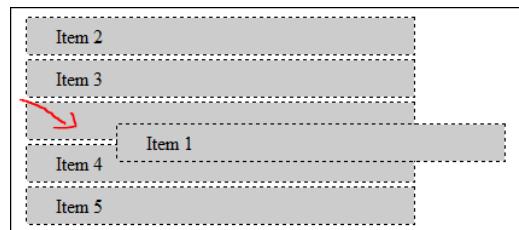
Gambar 3-15 : Placeholder.

Pada gambar di atas, tidak terlihat lokasi placeholder, dengan menggunakan atribut `placeholder` maka dapat dibuat lokasi ini dilihat oleh mata. Nilai dari atribut placeholder adalah class style pada CSS.

Misalnya telah dibuat sebuah class style dengan nama `.placeholder-item`, maka dapat dilihat penulisan atribut `placeholder` seperti berikut ini.

```
$("#sortable").sortable({ placeholder: 'placeholder-item' });
```

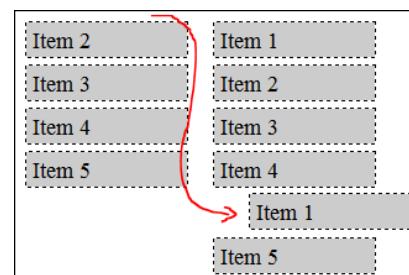
Hasilnya dapat dilihat pada gambar di bawah ini. Style placeholder berwarna abu-abu.



Gambar 3-15 : Placeholder dengan style.

Koneksi

Misalnya terdapat lebih dari satu elemen yang mempunyai kemampuan sortable, maka elemen-elemen tersebut dapat dihubungkan satu sama lain agar item-itemnya dapat berpindah barisan. Untuk lebih jelasnya bisa dilihat gambar di bawah ini.



Gambar 3-16 : Koneksi item antar elemen-elemen.

Berikut ini kode yang bisa diikuti.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Sortable</title>
    <!-- Script -->
    <script src="script/jquery-1.3.2.js" type="text/javascript"></script>
    <script src="script/ui.core.js" type="text/javascript"></script>
    <script src="script/ui.sortable.js" type="text/javascript"></script>

    <style type="text/css">
        #sortable1, #sortable2 { list-style-type: none; margin: 0; padding: 0;
float: left; margin-right: 10px; }
        #sortable1 li, #sortable2 li { margin: 0 5px 5px 5px; padding: 5px; font-
size: 1.2em; width: 120px; }
        li
        {
            margin: 0 3px 3px 3px;
            padding: 0.4em;
            padding-left: 1.5em;
            font-size: 1em;
            height: 18px;
            border:dashed 1px #000;
            background-color:#ccc;
        }
    </style>

    <script type="text/javascript">
        $(function() {
            $("#sortable1, #sortable2").sortable({
                connectWith: '.connectedSortable'
            }).disableSelection();
        });
    </script>
</head>
<body>
    <ul id="sortable1" class="connectedSortable">
        <li class="ui-state-default">Item 1</li>
        <li class="ui-state-default">Item 2</li>
        <li class="ui-state-default">Item 3</li>
        <li class="ui-state-default">Item 4</li>
        <li class="ui-state-default">Item 5</li>
    </ul>

    <ul id="sortable2" class="connectedSortable">
        <li class="ui-state-highlight">Item 1</li>
        <li class="ui-state-highlight">Item 2</li>
        <li class="ui-state-highlight">Item 3</li>
        <li class="ui-state-highlight">Item 4</li>
        <li class="ui-state-highlight">Item 5</li>
    </ul>
</body>
</html>
```