**Javascript & Jquery**

**Json, memperhatikan indentansi. Jika salah indentasi maka akan terjadi error.**

Link belajar object javascript

<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array>

Cara mengambil nilai radio button yg lagi di checked/ pilih.

var result = $('#dlg\_program:checked');

contoh dalam penggunaan system:

$("#select\_program").click(function() {

            var result = $('#dlg\_program:checked');

            if (result.length > 0){

                console.log(result.val());

                $("#program").val(result.val());

            }

            return true;

        })

Dalam program di atas kita mengisi element yg ber id program dengan value radio button (memiliki id: dlg\_program yg di pilih).

Cara menutup dialog box bootstrap setelah di klik button.

$('#programModal').modal('toggle');

**Cara menampilkan dan menghide sebuah element bisa juga pakai toggle.**

Contoh:

<button>Toggle</button>

<p>Hello</p>

<p style="display: none">Good Bye</p>

<script>

$( "button" ).click(**function**() {

$( "p" ).toggle();

});

</script>

Catatan: untuk toggle itu berlaku dengan property **display.** kalau misalnya kita menghilangkannya dengan menambah attribute hidden, maka toglle itu tidak akan berjalan.

Contoh:

<p hidden>Good Bye</p>

Kalaupun kita tambah toggle itu tidak akan berjalan.

Cara menampilkan halaman lain ketika klik button di jquery ajax (namun berlaku juga dengan event yg lain).

Format:

$(*selector*).load(*URL,data,callback*);

Contoh:

$("#div1").load("demo\_test.txt");

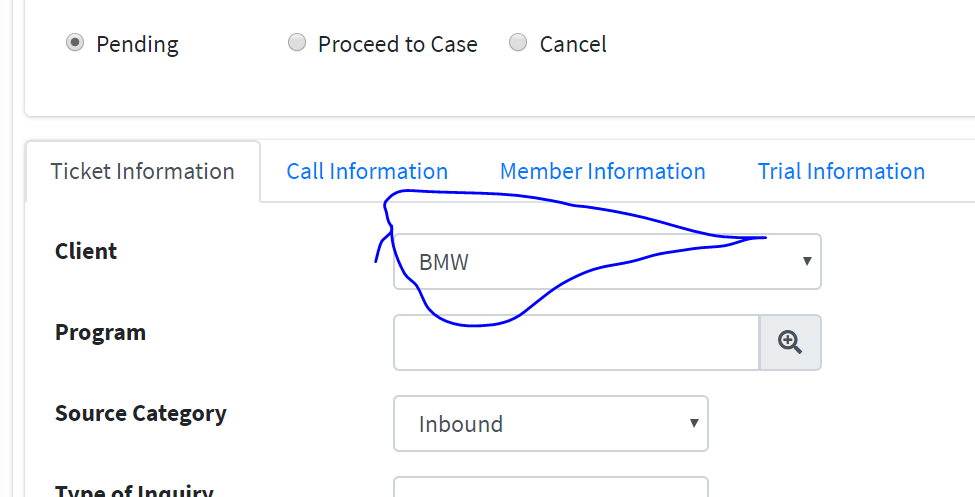
Itu maksudnya adalah element dengan id div1 akan di isi dengan isi dari content demo\_test.txt

url = parameter required, merupakan halaman yang akan kita tuju/ panggil.

data = data yg akan kita kirim ke halaman baru yg kita tuju tadi dimana nantinya bisa diolah di halaman yg kita tuju itu. Misalnya adalah id untuk di panggil nanti dari database untuk keperluan data.

Contoh:

Kita punya 1 field selection dengan nama client, dimana nilainya sudah berisi dengan client id.



Kemudian kita ingin memanggi/me- load sebuah halaman. Maka kita bisa kirimkan id dari client tersebut ke halaman yg kita tuju di urlnya. Perhatikan contoh berikut:

**Array**

Cara membuat array di javascript:

Var nama\_array = [];

Contoh :

var car = ['kijang', 'phanter', 'fortuner'];

var numb = [1,2,4,3,7];

**Cara mengambil isi dari array**

rumus : nama\_array[index].

Contoh: car[0], car[1], dst.

**Cara mereplace isi dari array.**

Rumus : nama\_array[index] = value.

car[0] = 'Kia'

hasilnya:

var car = ['Kia', 'phanter', 'fortuner'];

**Menambahkan element ke dalam array**

Rumus : namaArray.push(‘newElement’)

Contoh :

car.push('kuda')

itu akan menambah element kuda ke dalam array, letaknya yg paling terakhir.

Kita juga bisa menambah

**Menghapus element dari dalam array.**

Rumus : namaArray.pop();

Ini sebenarnya tidak menghapus, tapi mengambil nilai dari belakang. Dengan demikian kita bisa meletakkannya ke dalam satu variable.  
Contoh:

var lastElement = car.pop();

**shift dan unshift**ini hampir sama denga push dan pop. Tapi yg menjadi pembeda adalah kalau push dan pop menghapus dan menambah di belakang sedangkan shift dan unshift di bagian depan.

1. Shift. Rumus: namaArray.shift ,.

contoh:

const car = ['kijang', 'panter', 'fortuner'];

car.shift();

output:

[

"panter",

"fortuner"

]

1. unshift . Rumus : namaArray.unshift()

contoh:

const car = ['kijang', 'panter', 'fortuner'];

car.unshift('toyota');

output

[

"toyota",

"kijang",

"panter",

"fortuner"

]

**Loop for**

let arr = ["Apple", "Orange", "Pear"];

for (let i = 0; i < arr.length; i++) {

alert( arr[i] );

}

Tapi ada juga cara lain untuk melakukan perulangan di javascript

Yaitu dengan let of

for (let fruit of fruits) {

alert( fruit );

}

**Loop forEach**

const car = ['kijang', 'panter', 'fortuner'];

car.forEach(function(e)

{

console.log(e);

})

parameter e adalah element yang ada di dalam array car.  
outpuatnya sama dengan pakai **for** disini kita bisa menuliskan codenya lebih singkat.

'kijang'

'panter'

'fortuner'

**Loop map**Dengan perulangan map ini, kita bisa membuat function lagi di dalam parameternya. Hasil dari perulangan ini adalah array.

1. Sort

Mengurutkan isi dari array. Contoh: numb.sort(); hasilnya: [1,2,3,4,6]

Rumus: namaArray.map(function);

Contoh:

var numb = [1,2,4,3,7];

var res = numb.map(function(n){

return n \* 2;

})

console.log(res);

jika kita perhatikan contoh di atas di dalam map itu, kita kasih parameter dengan function aninomous dan kita bisa memanpilasinya lagi dengan return n \* 2. Parameter n adalah satu per satu element array. Kalua di loop kita tidak bisa

jika jquery is not found, maka pastikan dari semua library js , jquery ada di list awal.

Contoh :

<script src="https://code.jquery.com/jquery-3.3.1.js"></script>

<script src="https://cdn.datatables.net/1.10.20/js/jquery.dataTables.min.js"></script>

<script src="https://cdn.datatables.net/fixedcolumns/3.3.0/js/dataTables.fixedColumns.min.js"></script>

<script src="https://cdn.datatables.net/scroller/2.0.1/js/dataTables.scroller.min.js"></scr

List yg di atas sudah benar. Kita perhatikan baris jquerynya paling atas, baru setelah itu diikuti dengan list javascript-javascript yang lain.

<https://forum.jquery.com/topic/uncaught-referenceerror-jquery-is-not-defined-21-6-2013>

**Jquery AJAX**

Berikut contoh menggunakan jquery ajax:

<script>

        $.ajax({

            type: 'get',

            url: 'http://omdbapi.com',

            dataType: 'json',

            data: {

                'apikey': '21b88b86',

                's': $('#search-input').val()

            }

        })

    </script>

Penjelasan

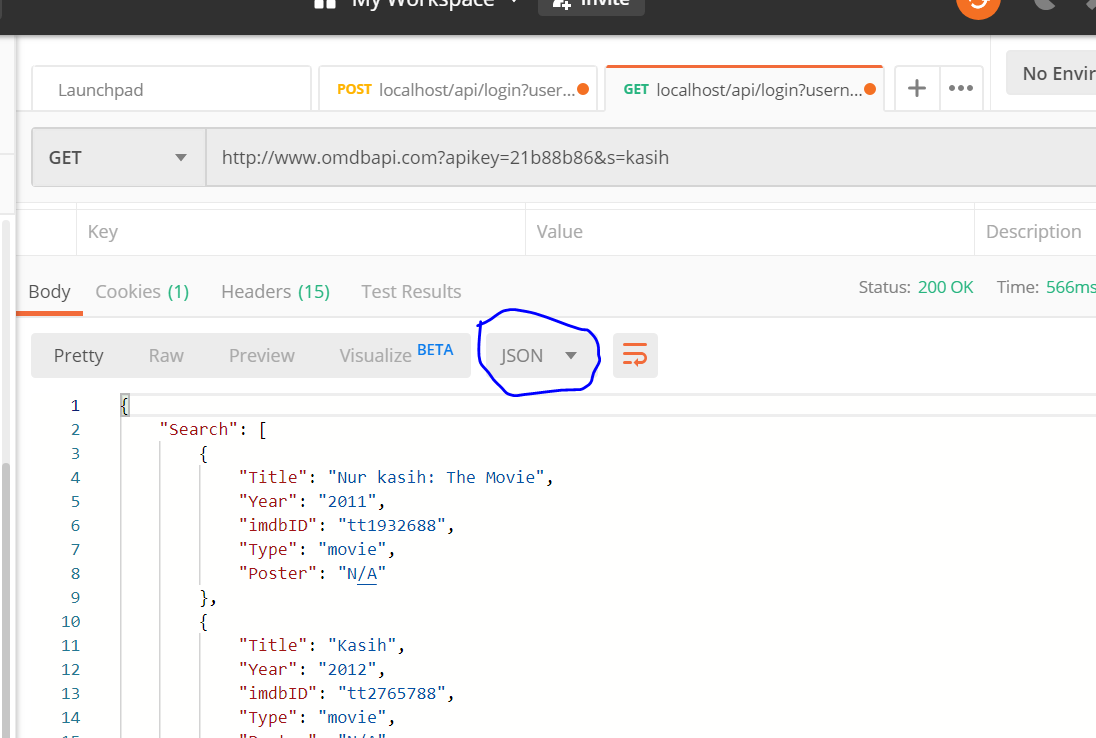
**type** adalah jenis method yang akan kita pakai di http.

Kalau di form html itu sama dengan method :

<form action="fileupload.php" method="post" enctype="multipart/form-data">

**url :** ke link mana kita akan tuju untuk mengambil datanya ini sama dengan actionnya.

**dataType:** jenis data yang akan kita terima dari response link itu. Kalau di postman itu adalah seperti gambar ini:



**Datatable**

Syarat yang harus ada di datatable:

1. Format tablenya harus benar.

Harus ada tag:

1. <table>
2. <thead><tr>
3. <tbody><tr>
4. Jumlah td di head harus sama dengan jumlah td di body

Contoh:

<table>

        <thead>

            <tr>

                <th>header 1</th>

                <th>header 2</th>

            </tr>

        </thead>

        <tbody>

            <tr>

                <td>data 1</td>

                <td>data 2</td>

            </tr>

        </tbody>

    </table>

Kita perhatikan th yang ada di thead sama dengan td yang ada di tbody. Itu sudah benar

Cara membuat sebuah element hide dan show, kita bisa dengan cara berikut saja:

Kita bisa langsung menggunakan :

toggle('')

contoh:

button yg di click

<div class="container row mb-1 ">

              <div class="col-12 float-right">

                <a href="#" id="ad\_search"> Advance search

                  <i class="fas fa-search-plus"></i>

                </a>

              </div>

            </div>

Element yang akan di hide dan di show

<div class="row border mb-2" id="advance\_search">

              <div class="col-12">

                <div class="form-group mt-1">

                  <div class="row">

                    <div class="col col-2">

                      <label for="id">Case Id</label>

                    </div>

                    <div class="col col-1">

                      <input name="case\_id" id="case\_id" type="text">

                    </div>

                  </div>

</div>

                  </div>

                </div>

Jquerynya:

$('#ad\_search').on('click', function() {

      $('#advance\_search').toggle('')

    })

**Menambah button di datatables bawaannya javascript**:

Kita bisa menambah dengan bawaan javascript:

dom : blfrtip, (kalau di dokumentasinya ini hanya bfrtip, tanpa huruf l kecil diantara b dan f nya)

buttons: [{

text : ‘my button’,

action: function (e, dt, node, config){

}

}]

Contoh:

$(document).ready(function() {

        $('#d\_group\_table').DataTable({

            dom: 'Blfrtip',

            buttons: [

            {

                text: '<i class="fas fa-plus"></i>',

                action: function ( e, dt, node, config ) {

                    window.location.href = "<?php echo base\_url(); ?>dealer\_group\_c/add\_d\_group";

                }

            }

        ]

        });

    });

Ketika di klik, button di atas membuka halaman yang lain menuju :

window.location.href = "<?php echo base\_url(); ?>dealer\_group\_c/add\_d\_group";

Untuk memanggil/memasukkan halaman lain ke dalam satu element. Misalnya div, kita bisa menggukan fungsi **load**.

Contoh:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>jQuery load() Demo</title>

<script src="https://code.jquery.com/jquery-1.12.4.min.js"></script>

<script>

$(document).ready(function(){

    $("button").click(function(){

        $("#box").load("test-content.html");

    });

});

</script>

</head>

<body>

    <div id="box">

        <h2>Click button to load new content inside DIV box</h2>

    </div>

    <button type="button">Load Content</button>

</body>

</html>

Cara buat function anynous di dalam function anynous:

Kita cukup hanya menambahkan/membungkus function itu di dalam tanda kurung : ()

Contoh:

$(document).ready(function() {

            // ##########

            // ### function anynomous

            // ##########

            var f\_anyno = (function(x, y) {

                return x + y;

            })

            $('#function\_anynomous').html(f\_anyno(5, 7));

        })

Cara menjalankan function anynomous secara langsung tanpa harus simpan ke dalam variable:

(**function**() {

*// private*

**var** x = 1;

console.log( x ); *// 1*

})();

Membuat form di html tidak bisa di submit:

$( "#myform" ).submit(**function**() {

**return** false;

});

Untuk membuat menu terbuka dengan cara menambah sebuah class untuk mendisplay jadi block.

Dalam hal ini nama classnya adalah menu-open. Fungsi berikut kita buat fungsi if ternary

<?php echo (in\_array($info, ['Ticket', 'Ticket2']) ? 'menu-open' :'' ); ?>"

Contoh:

<li id="tickets" class="nav-item has-treeview <?php echo (in\_array($info, ['Ticket', 'Ticket2']) ? 'menu-open' :'' ); ?>">

          <a href="" class="nav-link">

            <i class="fas fa-ticket-alt"></i>

            <p>Tickets</p>

          </a>

          <ul class="nav-treeview mleft\_min-20">

            <li class="nav-item">

              <ul>

                <li style="list-style-type: none;">

                  <a href="<?php echo base\_url(); ?>ticket\_c/add\_ticket" class="nav-link">

                    <i class="far fa-plus-square"></i>

                    <p>New</p>

                  </a>

                </li>

              </ul>

            </li>

            <li id="ticket\_search">

              <ul>

                <li style="list-style-type: none;">

                  <a href="<?php echo base\_url(); ?>ticket\_c/index" class="nav-link">

                    <i class="fas fa-search"></i>

                    <p>Search</p>

                  </a>

                </li>

              </ul>

            </li>

          </ul>

        </li>

Kalau kita lihat contoh di atas kita langsung letakin di dalam menu li nya.

Sebenarnya untuk menambah class menu-open kita bisa juga buat di dalam javascript, dengan catatan kita harus cek info dari halaman tersebut. Oleh karena itu kita harus menambah sebuah info untik ke dalam halaman itu , untuk nanti di cek di javascriptnya.

Contoh:

Pada saat ingin meload halama ticket di controller kita sertakan data variable info tersebut. Perhatikan contoh code berikut:

public function index()

    {

        $data['info'] = 'Ticket';

        $this->load->view('template/asset\_header\_datatabel');

        $this->load->view('template/sidebar', $data);

        $this->load->view('ticket/ticket', $data);

        $this->load->view('template/asset\_footer\_datatabel');

    }

Disana kita perhatikan :

$data['info'] = 'Ticket';

Variable data[‘info’] memiliki nilai unik yaitu: Ticket. Dan kita parsing ke dalam halaman sidebar. Nanti itulah yang akan kita pakai sebagai identifier untuk menambah class yang kita inginkan. Dalam hal ini adalah menu-open miliknya admin lte-3

array(10) {

[0]=>

object(stdClass)#26 (12) {

["id"]=>

string(1) "1"

["client"]=>

string(3) "219"

["company"]=>

string(13) "AON INDONESIA"

["member\_name"]=>

string(25) "NIPPON SHOKUBAI INDONESIA"

["rsa\_vehicle\_chassis\_no"]=>

**Contoh membuat calendar di field dengan jquery**

<!doctype html>

<html lang="en">

<head>

    <meta charset="utf-8">

    <meta name="viewport" content="width=device-width, initial-scale=1">

    <title>jQuery UI Datepicker - Default functionality</title>

    <link rel="stylesheet" href="http://code.jquery.com/ui/1.12.1/themes/base/jquery-ui.css">

    <link rel="stylesheet" href="/resources/demos/style.css">

    <script src="https://code.jquery.com/jquery-1.12.4.js"></script>

    <script src="https://code.jquery.com/ui/1.12.1/jquery-ui.js"></script>

    <script>

        $(function() {

            $("#datepicker").datepicker();

        });

    </script>

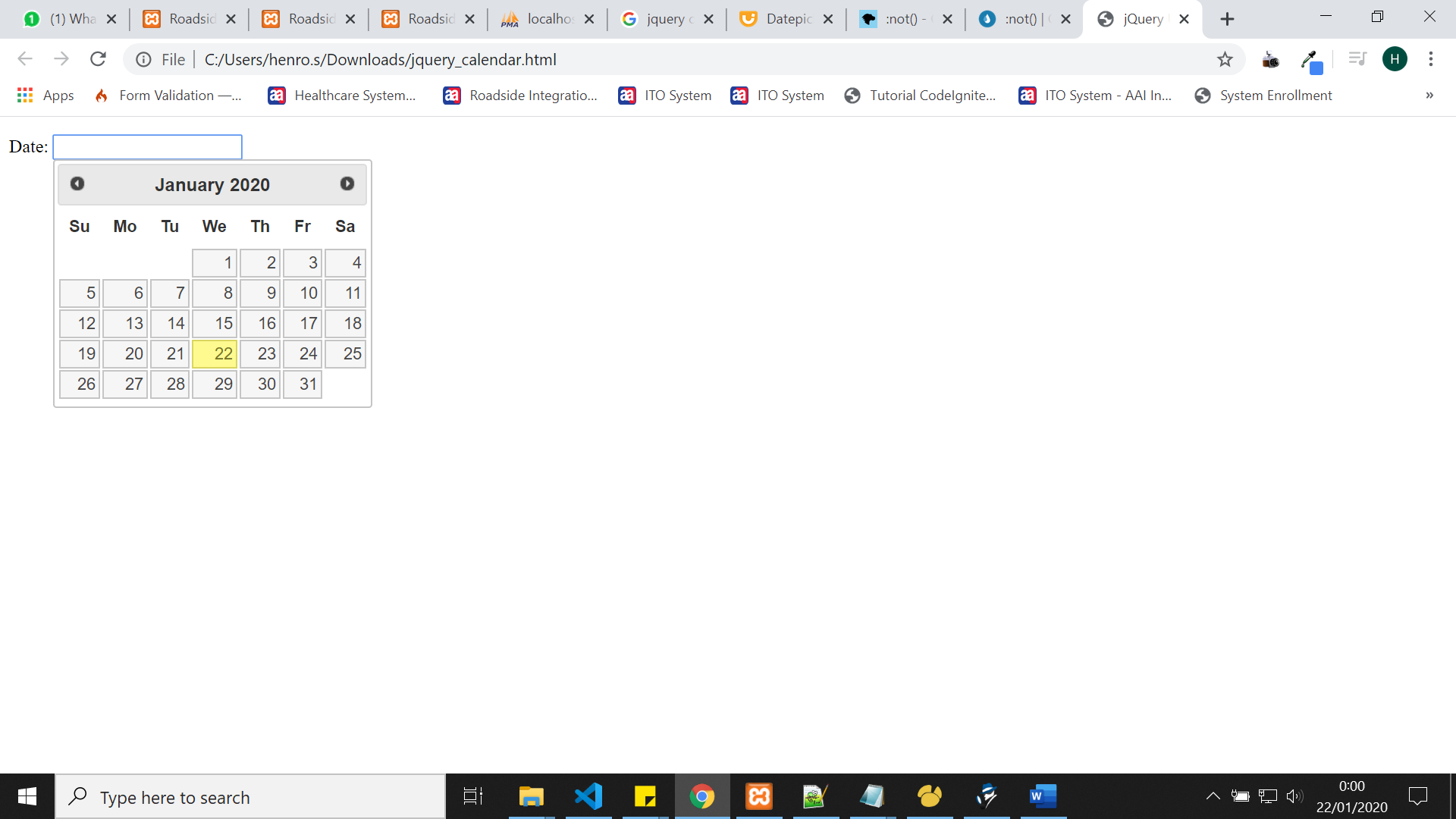
</head>

<body>

    <p>Date: <input type="text" id="datepicker"></p>

</body>

Hasil:



Note: yang paling penting dari membuat jquery calendar itu adalah,

Fungsi javascriptnya harus ada di dalam function.

Contoh salah:

$("#dob").datepicker({

            dateFormat: 'yy-mm-dd'

        });

Itu tidak akan jalan, dan di javascriptnya aka nada error, datepicker is not a function.

contoh yang benar:

$(function() {

        $("#dob").datepicker({

            dateFormat: 'yy-mm-dd'

        });

    });

Menampilkan element:

$('#create\_date').show();

Menhilang element:

$('#create\_date').hide();

Menampilkan dan menghilangkan:

$('#create\_date').toggle();

**Cara mengambil semua nilai input field html dimana di group dalam 1 element.**

Kita bisa cari dengan :

:input

Itu akan mengambil semua type element untuk tujuan input data (sudah termasuk juga tag <select>).

Contoh:

                        <div id="form1">

                          <input id="nama" name="nama">

                          <input id="alamat" name="alamat">

                          <label for="client">Hobbi</label>

                          <select>

                            <option>Bola</option>

                            <option>Badminton</option>

                          </select>

                        </div>

jika kita ingin mendapatkan nilai-nilai dari field inputan yang ada di dalam element div dengan id = form1 di dalam javascript, maka secara biasa kita akan ambil 1 demi 1 dengan id dari element masing-masing. Contoh:

var nama = document.getElementById(‘nama’).value(); atau dengan jquery: $(‘#nama’).val();

*Note: di sini kita akan menggunakan jquery untuk mempersingkat pekerjaan.*

Bagaimana jika fieldnya banyak? Kita harus menulisnya 1 demi 1. Dengan cara seperti itu, code kita akan bertambah banyak dan memakan waktu yang cukup lama. Maka untuk mengatasi hal seperti itu kita bisa menggunakan cara sperti berikut:

$('#filter').click(function() {

      var args = {};

      $('#advance\_search :input').each(function()

      {

        var input\_id = $(this).context.id;

        var input\_val = $(this).val();

        console.log(input\_id);

        args[input\_id] = input\_val;

      });

      console.log(args);

}

Penjelasan:

Pada saat element filter di click, akan mencari id advance\_search, setelah ketemu , akan menyeleksi semuat element untuk tujuan input data .

#advance\_search :input

Dan langsung melakukan loop,

each

dan kemudian langsung memanggil function anynomous di dalam nya:

function()

      {

        var input\_id = $(this).context.id;

        var input\_val = $(this).val();

        console.log(input\_id);

        args[input\_id] = input\_val;

      });

$this, dalam hal ini adalah, object element yang lagi di loop.

Untuk lebih spesifik, merubah mengambil element dengan type kita bisa pakai:

$("form#formID input[type=text]")//...

Contoh: <input type=”text”>

Contoh:

Cek sebuah key di dalam object:

Contoh:

var operator = $('#operator').val();

      if (['is\_null','is\_not\_null'].includes(operator)) {

        $('#create\_date').hide();

      }

Function javascript

Cara 1:

function advance\_search (args) {

      var args = {};

      $('#advance\_search :input').each(function() {

        var input\_id = $(this).context.id;

        var input\_val = $(this).val();

        args[input\_id] = input\_val;

      });

      return args;

    }

Cara 2:

advance\_search = function (args) {

      var args = {};

      $('#advance\_search :input').each(function() {

        var input\_id = $(this).context.id;

        var input\_val = $(this).val();

        args[input\_id] = input\_val;

      });

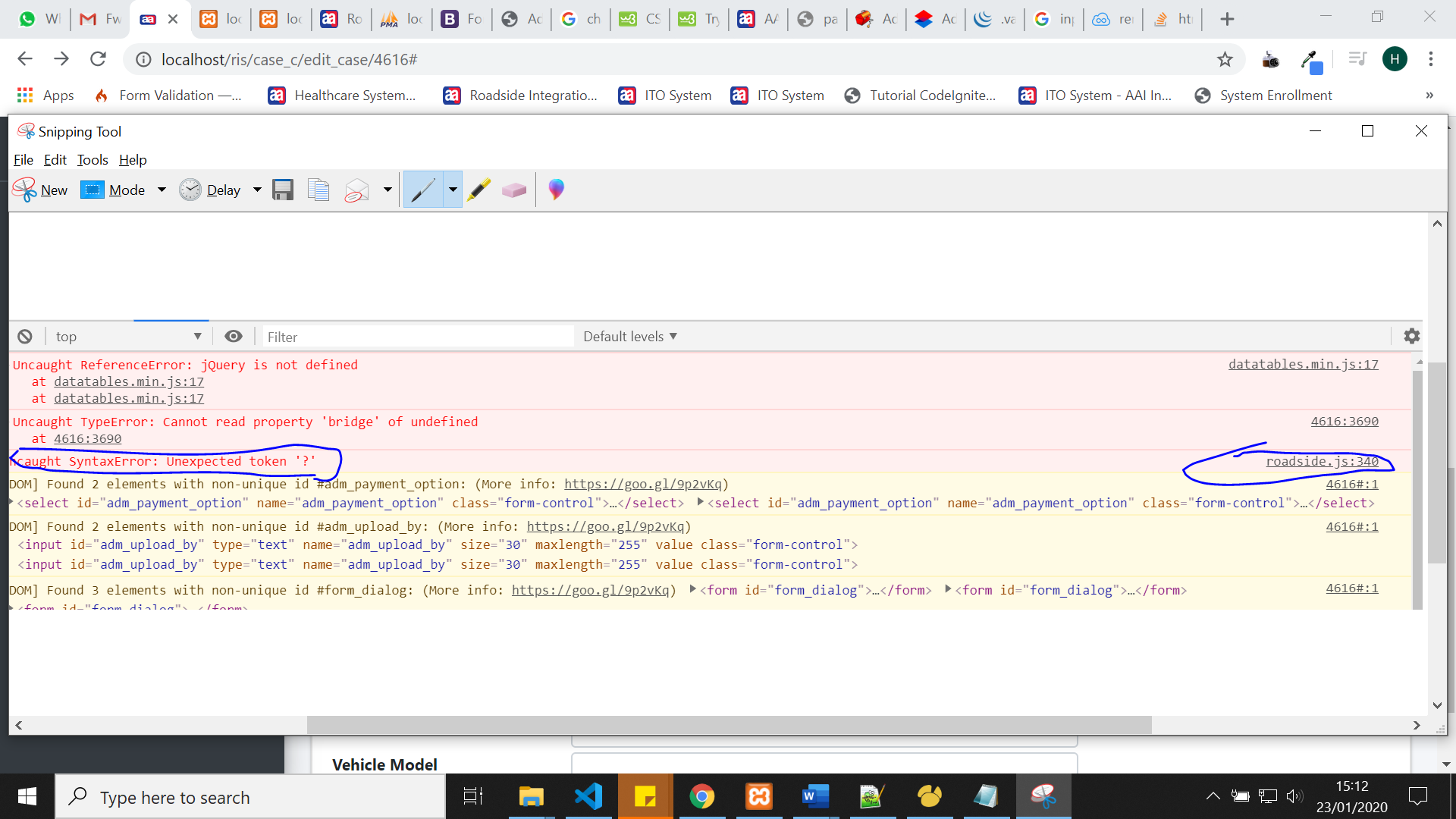
      return args;

    }

Dari 2 function di atas memiliki hasil yang sama dan cari panggilnya juga sama.

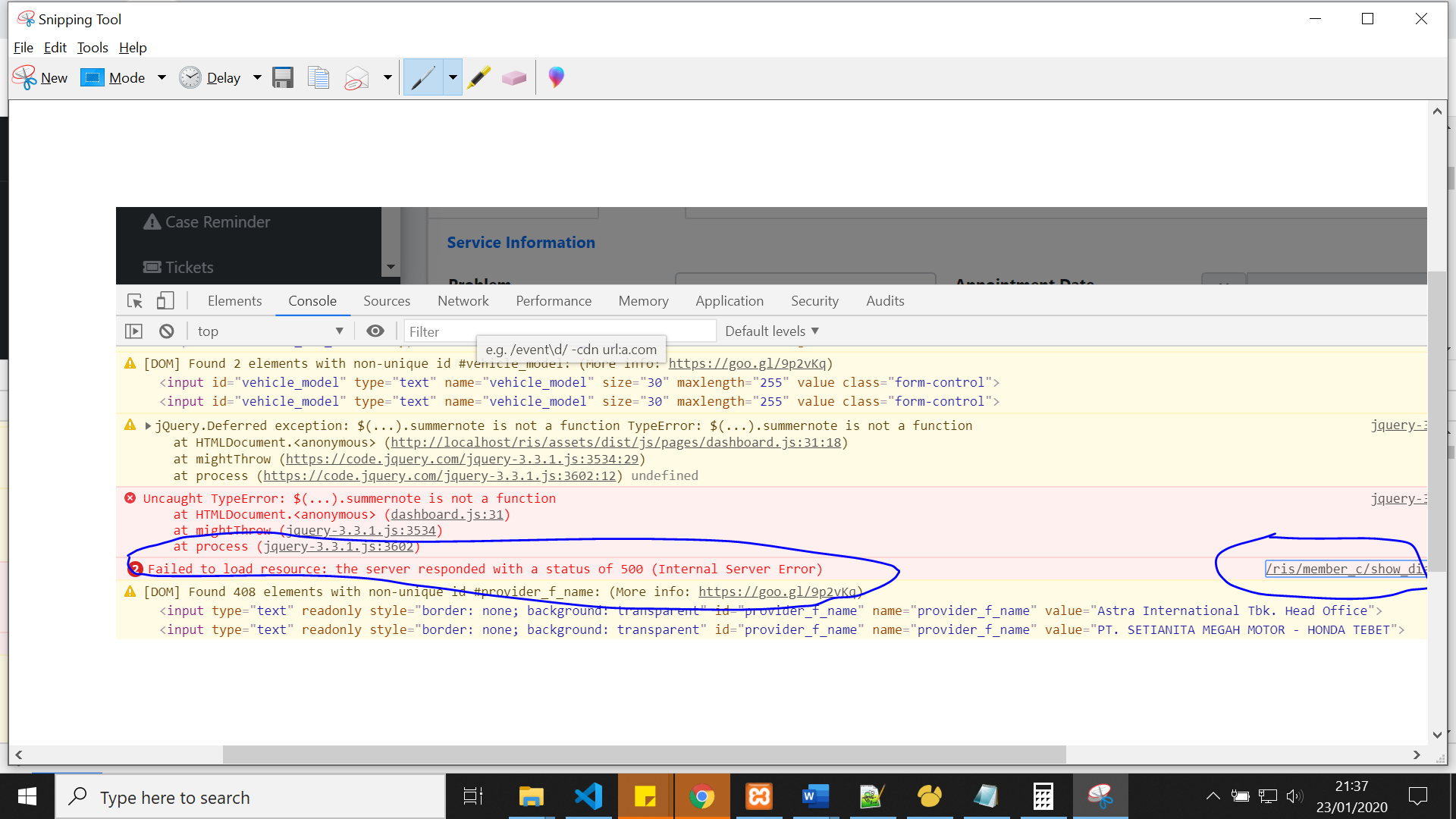
advance\_search()

Error experience:

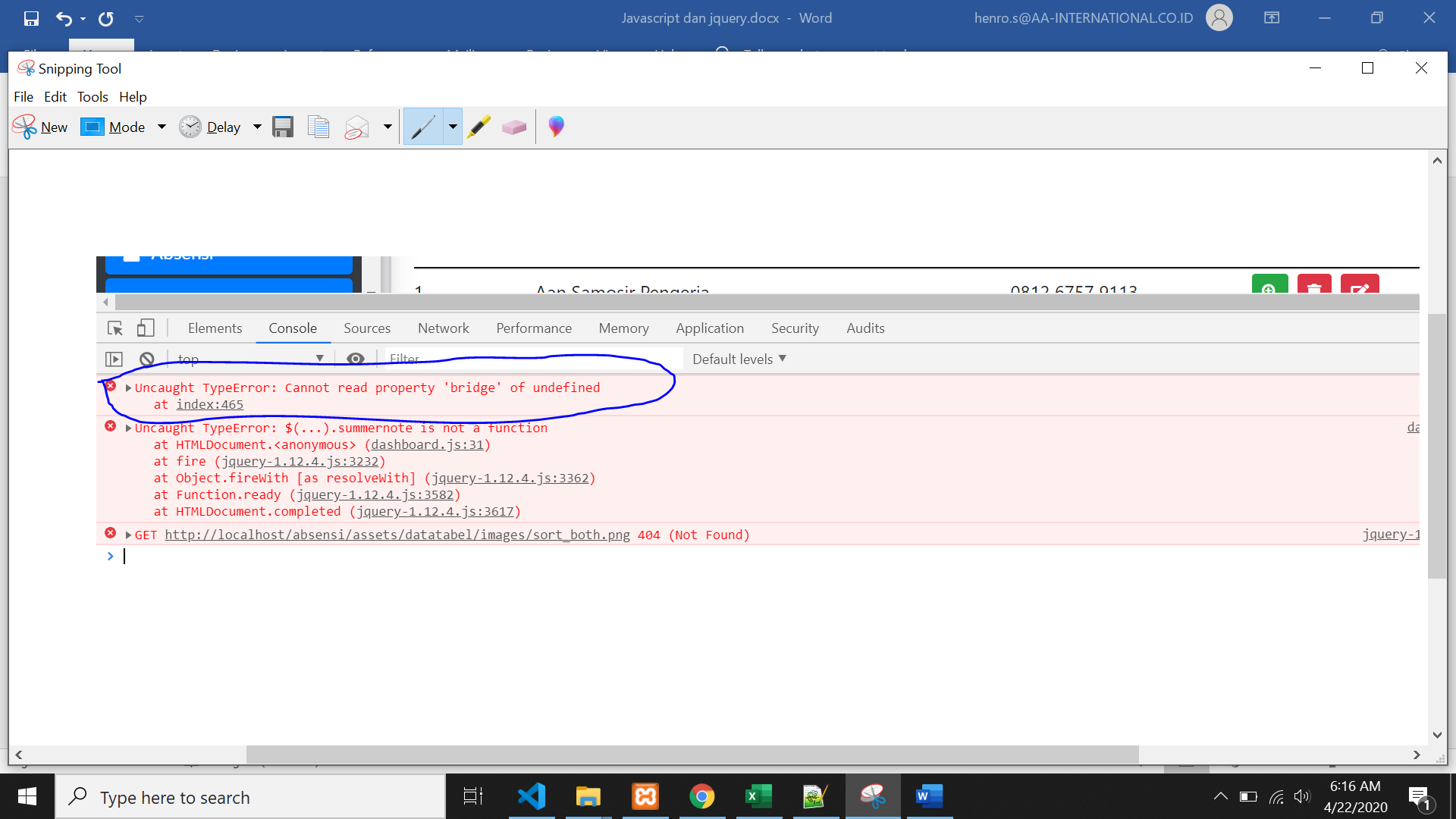
Untuk mengetahui sebuah error javascript, kita harus lihat di console, baca mulai dari atas, dan lihat dimana dia mulai error. (ini hampir sama dengan di command line, sebab memang ini konsepnya sama, procedural . Yaitu dibaca dari line yang paling atas, kemudian ke bawah. Itulah kenapa kita bisa lihat di line mana dia error dan code error mulai muncul).Perhatikan contoh berikut : 

Javascript tidak akan bisa dijalankan kalau ada error, maka kita bisa langsung bisa lihat di consolenya perhatikan, conth di bawah.

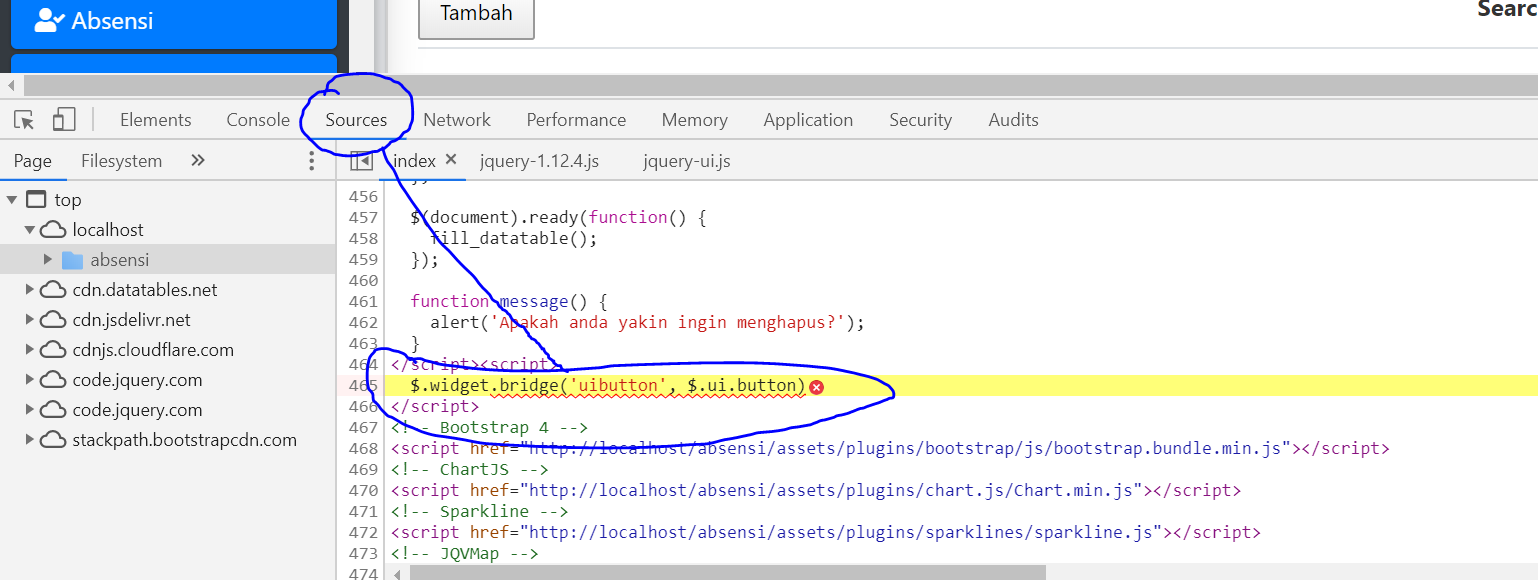
Dan contoh berikut:



Contoh error lain yang pernah saya temui:

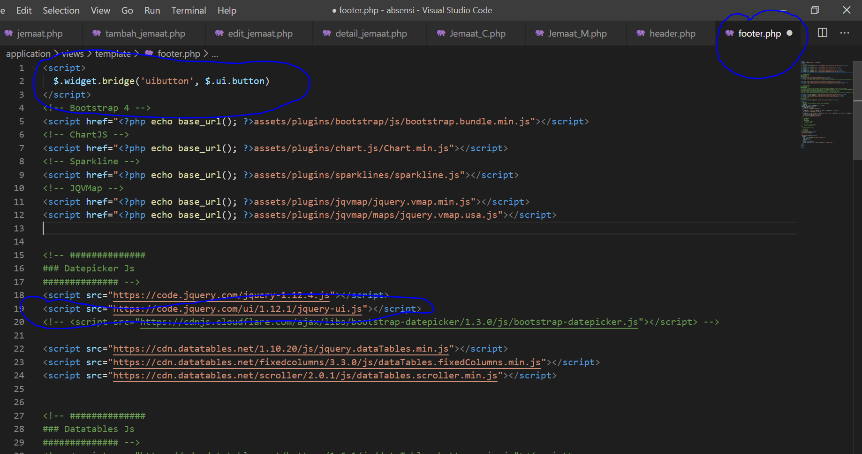


Jika kita klik at index:465 tersebut, maka source code penyebab error akan muncul:



Gambar : 1.2

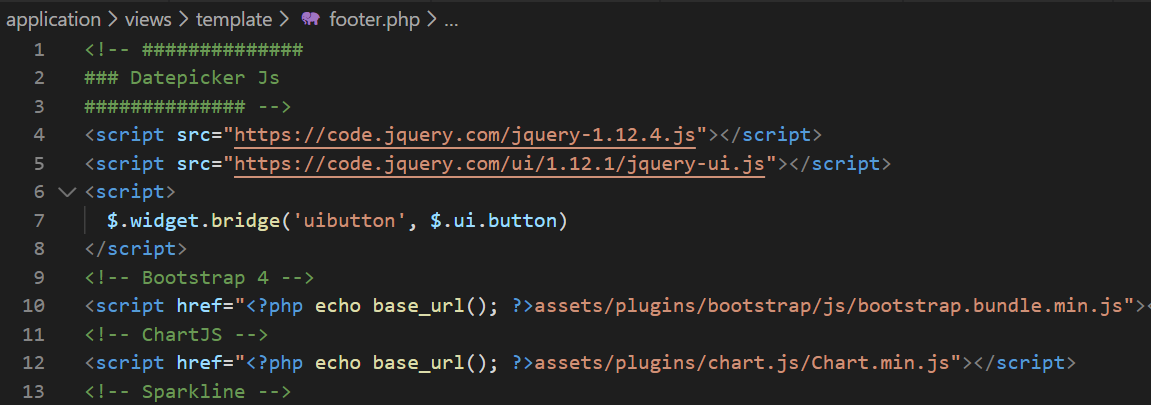
Jika kita perhatikan, code tersebut sebenarnya sumbernya dari code kita sendiri. Yaitu:



Gambar : 1.3

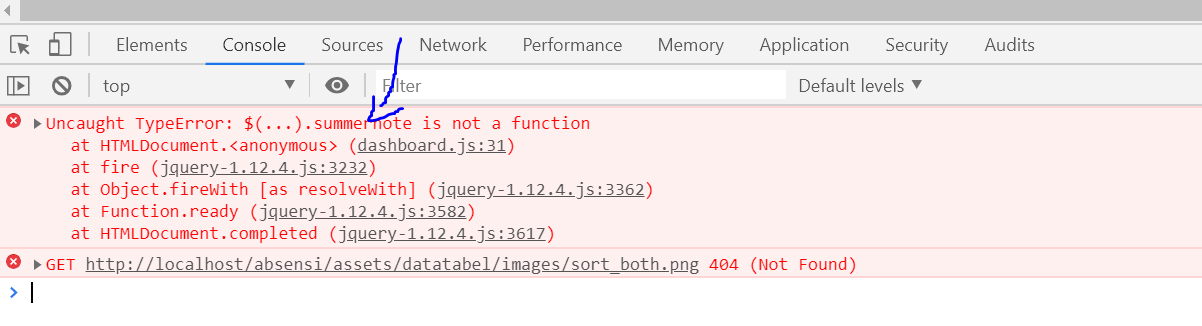
Jika kita, perhatikan di gambar 1.2 diatas, itu menunjuk ke code kita yang ada di gambar 1.3 di line 2.

Sementara librarynya baru kita input di line 19. Jadi penyebabnya adalah, kita sudah panggil fungsi UI sebelum kita include librarynya. Maka untuk fixnya adalah, kita harus memindahkan library ui dan jquery itu sebelum pemanggilan fungsi ui (.bridge) tersebut. Seperti gambar berikut:



Gambar : 1.3

Contoh error lain:



Gambar : 1.4

Untuk gambar di atas bisa disebabkan version jquery yang tidak cocok dengan dengan library summernote nya. Fungsi summernote kemungkinan mulai berlaku di versi jquery 3.1.

Untuk menutup sebuah dialog box, setelah tombol submit/ save, supaya dialog box itu, tertutup kembali kita bisa tambah code id\_dialog.modal(‘hide’).

Contoh:

<button id="btnSave" type="submit" class="btn btn-success"><i class="glyphicon glyphicon-trash"></i> Save</button>

$('#btnSave').click(function() {

$('#StudentModal').modal('hide');

});

Di bootstrap kita bisa langsung gunakan attribute seperti berikut:

data-dismiss="modal"

contoh:

<button type="submit" class="btn btn-success" data-dismiss="modal"><i class="glyphicon glyphicon-ok"></i> Save</button>

**Jquery ajax**

Untuk mendapatkan hasil response, di function php kita harus membuat hasil akhirnya echo atau pun var\_dump bukan return seperti function biasa atau dengan yang penting memprint hasilnya.

Sebab, functionnya dipanggil bukan dari function tapi dari http ajax, jadi yang diterima hanya text.

Contoh:

$.ajax({

          url: "<?php echo base\_url() . ('provider\_c/get\_name\_val'); ?>",

          method: 'POST',

          data: 'provider\_id=' + provider\_id,

          dataType: 'html'

        })

        .done(function(data) {

          try {

            $('#service\_provider').append('<option value="' + provider.val() + '" selected="selected">' + data + '</option>');

          } catch (err) {

            alert(err.message);

          }

        })

Di function php nya

public function get\_name\_val()

    {

        $provider\_id = $this->input->post('provider\_id');

        $full\_name = $this->provider\_m->get\_name\_val('provider', $provider\_id);

        $full\_name = json\_decode(json\_encode($full\_name), True);

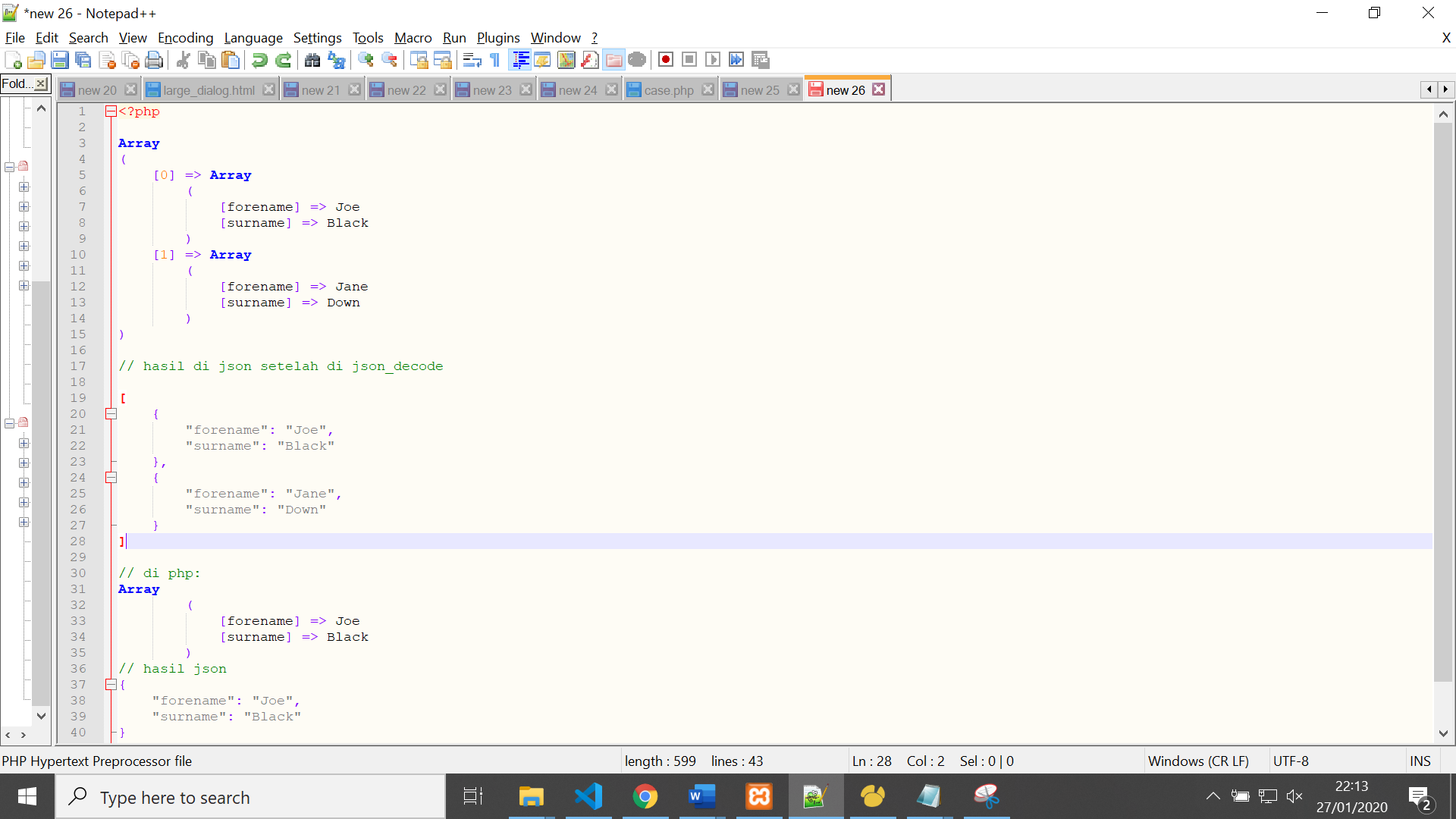
        $full\_name = ($full\_name['full\_name']);

        echo $full\_name;

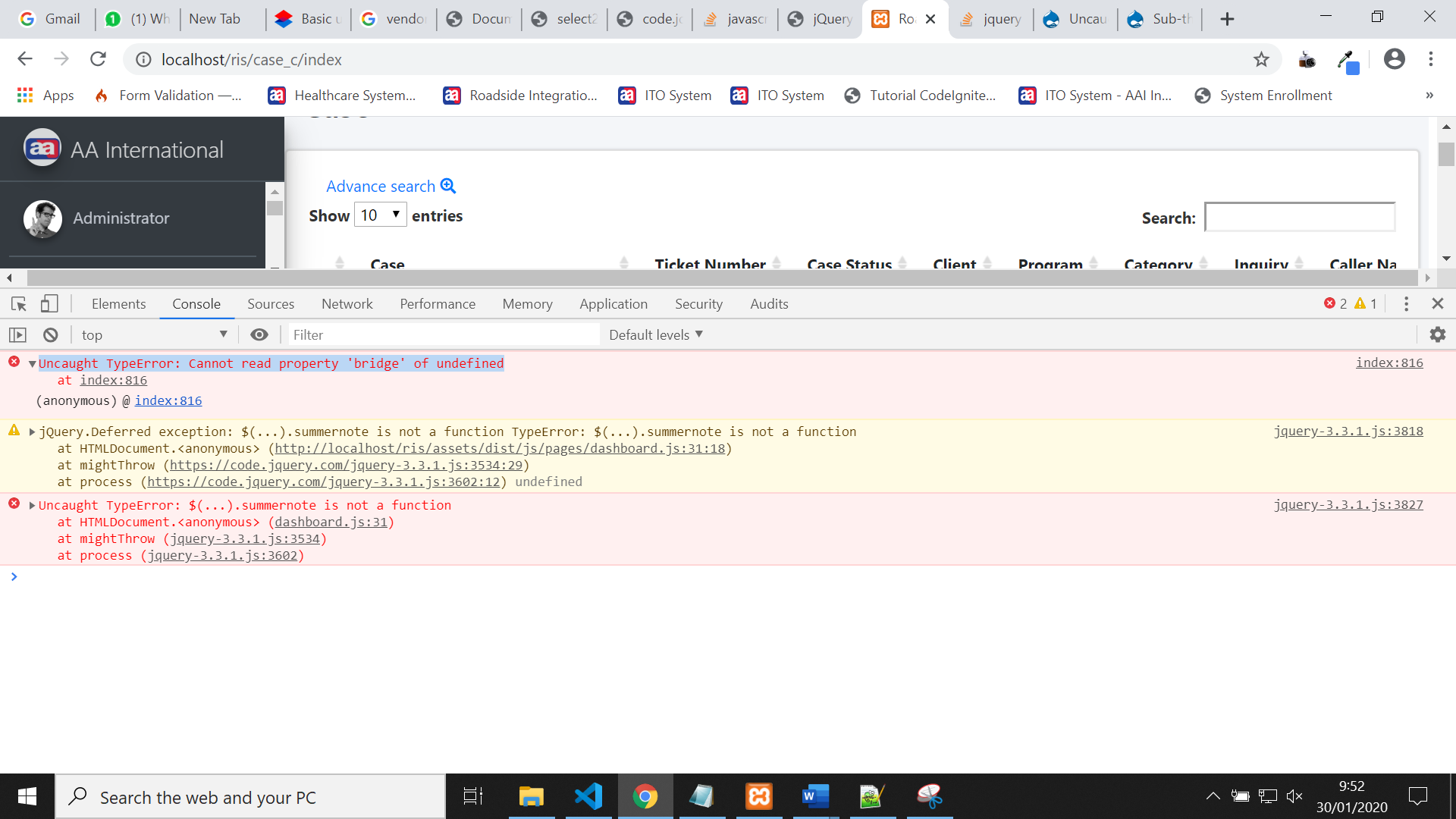
    }

Kalau kita perhatikan di akhir function itu adalah echo $full\_name bukan return.

JSON



Penyebab terjadi error di javascript not defined:



Penyebab terjadinya:

1. Mungkin librarynya belum di include
2. Mungkin librarynya sudah dimasukkan, tapi posisinya paling bawah (fungsinya sudah dipakai, tapi karena librarynya di bawah library yang memakai, maka tidak ke panggil)

**Jquery Selector**

Memilih element dengan nama element,attribute,value dari attribute.

Contoh:

<div class="wrap-input100 " data-validate="Enter username">

   <input class="input100" type="text" name="username" placeholder="Username">

   <span class="focus-input100" data-placeholder="&#xf207;"></span>

</div>

untuk memilih field username kita bisa dengan cara:

$("input[name='username']")

untuk mengambil valuenya, kita bisa dengan cara:

var username = $("input[name='username']").val();

Untuk supaya bisa function .done diajax jalan, itu response dari callbacknya harus ada.  
setidaknya echo 123.

Contoh kasus :

Saya ingin membuat sebuh meredirect ke halaman setelah klik 1 button, dimana disini saya akan pakai jquery ajax. Dalam kasus ini adalah untuk function logout.

Berikut codenya:

Untuk halaman menu logout.

<!-- #############

    ### Logout session

    ############# -->

    <div id="logout" style="margin-top: -17px; display: none">

      <div class="user-panel mt-3 pb-3 mb-3 d-flex" style="background-color: hsl(180, 15%, 41%)">

        <ul class="logout">

          <!-- The user image in the menu -->

          <li class="user-header"> <img src="<?php echo base\_url(); ?>assets/dist/img/user2-160x160.jpg" class="img-circle" alt="User Image">

            <p>Administrator<br>

              <font size="3"><i class="fa fa-circle text-success"></i> Online </font>

            </p>

          </li>

          <li class="user-footer">

            <div class="pull-right">

              <a class="btn btn-default btn-flat" id="signout" type="submit">Logout</a>

            </div>

          </li>

        </ul>

      </div>

    </div>

Halaman javascriptnya:

$("#signout").click(function() {

    $.ajax({

      url: "<?php echo base\_url('login\_c/logout') ?>",

      type: "post",

      dataType: "json",

      data: ''

    }).done(function(data){

      window.location.href = "<?php echo base\_url(); ?>login\_c/index";

    });

  });

Kita perhatikan ke url nya, itu menuju ke controller login\_c/logout.

Berikut codenya:

public function logout()

    {

        session\_destroy();

        echo 123;

    }

kalau kita lihat di function javascriptnya, dibagian done.

Itu kita buat function aninomus untuk meredirect ke function login\_c/index. Dimana function itu akan meredirect halam ke halaman login. Kalau dilihat dari function logout itu, function sudah jalan dengan baik. Itu karena di function itu ada sebuah respon, meskipun itu hanya echo 123. Yang dimana itu akan menjadi respon yang akan di antar ke function .done di javascript itu. Jika function logout tidak memiliki response, sperti dibawah ini:

public function logout()

    {

        session\_destroy();

    }

maka function .done tidak akan jalan, dan halamannya tidak akan di redirect ke halaman login.

Notenya itu tidak boleh string. Harus json atau setidaknya number. Kalau string itu tidak akan jalan.

Contoh

public function logout()

    {

        session\_destroy();

        echo ‘ok’;

    }

Itu tidak akan jalan. Tapi kalau json bisa seperti berikut:

public function logout()

    {

        session\_destroy();

        $status = array(

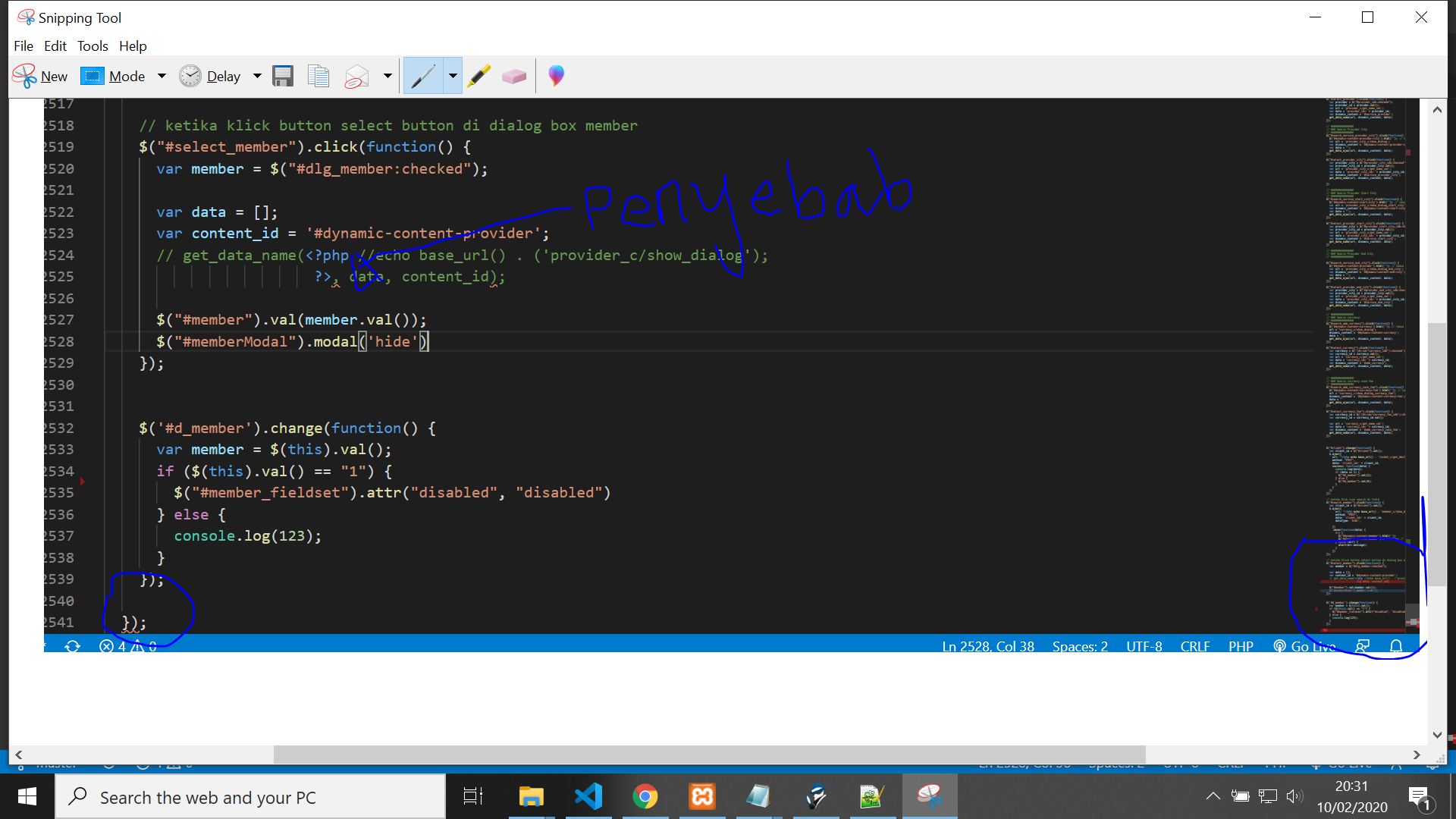
          'status' => 'ok'

        );

        echo json\_encode($status);

    }

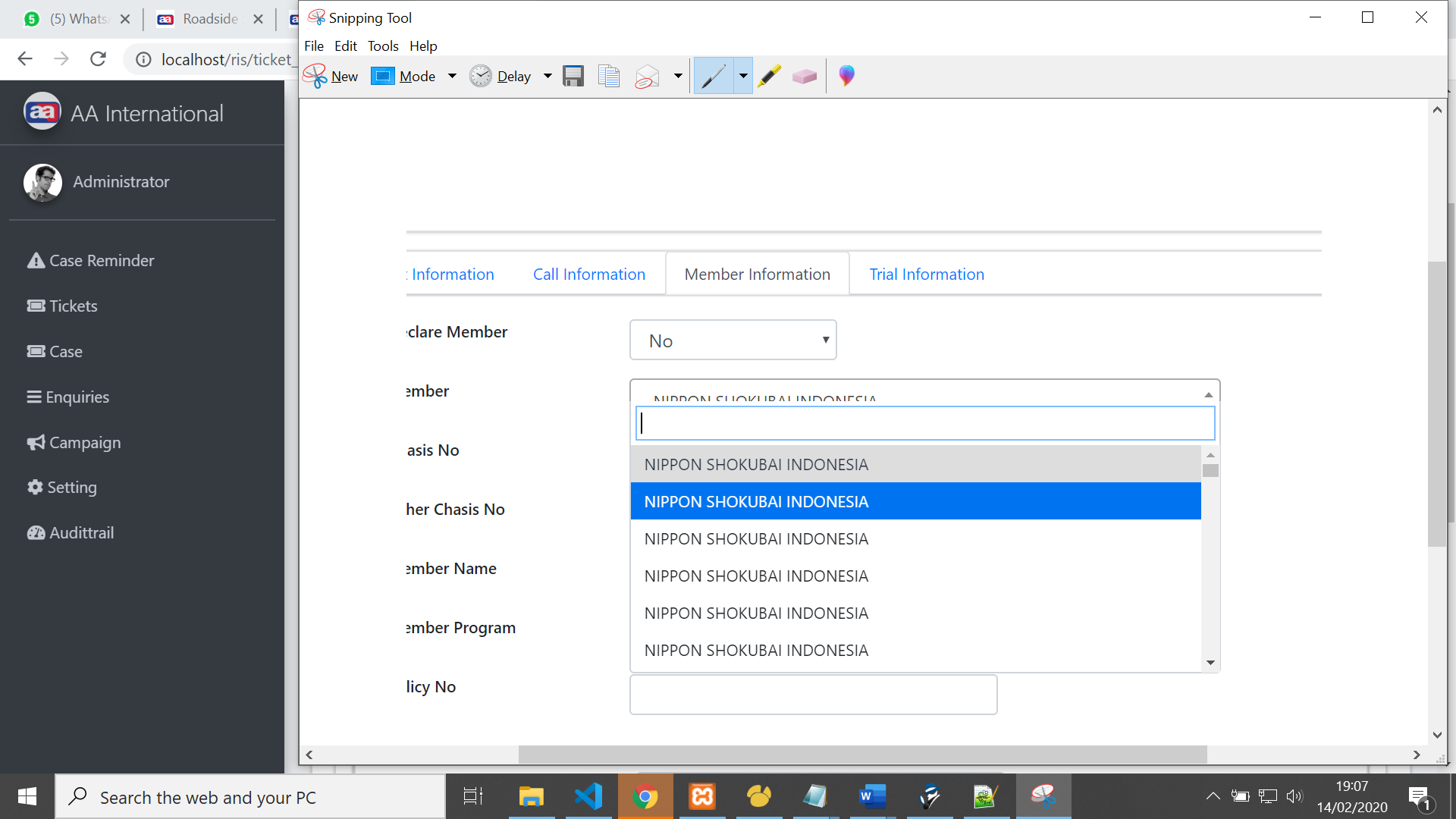
Salah satu penyebab error di javasciript:



**Select2**

Select 2 adalah sebuah library untuk membuat tampilan field selection jadi lebih bagus.

Contoh:



Seperti kita lihat disana ada fitur untuk search juga.

Cara membuatnya:

Untuk librarynya tambahkan link berikut di tag header

<!-- #############

### Select2

############# -->

  <link href="https://cdn.jsdelivr.net/npm/select2@4.0.13/dist/css/select2.min.css" rel="stylesheet" />

  <script src="https://cdn.jsdelivr.net/npm/select2@4.0.13/dist/js/select2.min.js"></script>

  <script src="//cdnjs.cloudflare.com/ajax/libs/select2/4.0.3/js/select2.full.js"></script>

Field:

<div class="form-group">

    <div class="input-group">

    <select class="js-example-basic-single" name="member" id="member" style="width: 520px">

    </select>

   </div>

</div>

Di javascript

<script>

$('#member').select2();

</script>

$.ajax({

      url: "<?php echo base\_url() . 'member\_c/select\_complete' ?>",

      method: "POST",

      data: [],

      dataType: 'html',

    })

    .done(function(data) {

      try {

        $('#member').html(data);

      } catch (err) {

        alert(err.message);

      }

    });

Styling select2

Menambah height select2 field selection

.select2-selection\_\_rendered {

line-height: 31px !important;

}

.select2-container .select2-selection--single {

height: 35px !important;

}

.select2-selection\_\_arrow {

height: 34px !important;

}

Contohnya filenya ada di : ris di file view : add\_member.php

Cara membuat edit di select2 sehingga valuenya itu tetap memilih yang ada di data.

$program\_id = "<?php echo $member->pr\_id?>";

  $.ajax({

      url: "<?php echo base\_url() . 'program\_c/edit\_autocomplete' ?>",

      method: "GET",

      data: 'program\_id=' + $program\_id,

      dataType: 'html',

    })

    .done(function(data) {

      try {

        $('#program').html(data);

      } catch (err) {

        alert(err.message);

      }

    });

$program\_id = $\_GET['program\_id'];

        // echo $program\_id;

        $programs = $this->program\_m->autocomplete();

        $index = 0;

        foreach ($programs as $program) {

            if ($program\_id == '' && $index == 0) {

                echo "<option value='' selected></option>";

            }

            echo "<option value='$program->id'>$program->name</option>";

            $index++;

        }

**String**

String adalah Type data yang dibuat dengan text. Bisa berupa huruf, angka, symbol, tanda baca, dan bahkan emotion.

Untuk membuat string kita kita harus menambah simbol " atau ' diawal dan diakhir dari suatu pernyataan.

Contoh:

"Ini adalah text".

Jika kita ingin menyertakan sebuah tanda kutip di dalam string.

Contoh:

"Ini adalah "text" ". Ini merupakan contoh yang salah.

ada 2 cara mengatasi hal diatas. 2 caranya yaitu:

1. "Ini adalah 'text' "

2. "Ini adalah \"text\" "

**Properti dan Methods**

String memiliki built-in variable dan function sendiri. Dimana disebut juga dengan Properti dan Methods.

1. length

    Length memiliki fungsi untuk menghitung jumlah karakter string.

Contoh:

"buku".length() output: 4

2. toLowerCase

   Fungsi ini berguna untuk membuat string menjadi huruf kecil semua.

Contoh:

"BUKU".toLowerCase() output: "buku"

3. toUpperCase

   Fungsi ini berguna untuk membuat string menjadi huruf besar semua.

Contoh:

"buku".toUpperCase() output: "BUKU"

4. trim

    Fungsi ini berguna untuk menghapus karakter kosong di dalam string (kecuali 1 spasi pertiap kata).

Contoh:

"   Saya belajar javascript    ".     trim()  output: "Saya belajar javascript"

**Number**

Number adalah nilai yang mana bisa dibuat untuk perhitungan matematika.

Contoh:

10 + 3.1234

Output:

13.1234

Fraction tidak ada dalam javascript, tapi kita bisa solve itu dengan pembagian seperti berikut ini.

1/3

Output:

00.333333333

**Booleans**

Boolean Merupakan type data yang hanya bisa menampung 2 nilai. Yaitu: true dan false.

contoh:

var menikah = true

**Operator**

    Operator adalah symbol yang berada diatara value diamana akan melakukan perhitungan terhadap ke 2 operand tersebut (fungsi aritmatika).

Penjumlahan

5 + 2

output : 7

kita juga bisa menggunakan simbol + untuk menggabungkan 2 string. Istilah ini disebut concatenation.

Contoh:

"Saya" + "sedang" + "belajar."

Pengurangan

5 - 2

output : 3

Perkalian

5 \* 2

output: 10

Pembagian

4/2

output: 2

gabungan

1 + 2 \* 3

output: 7

ket: Pola perhitungan di Js itu mengikuti pola matematika. Maksudnya adalah operator yang memiliki prioritas lebih tinggi akan dikerjakan terlebih dahulu. Dari contoh di atas perkalian (\*) lebih tinggi prioritasnya dari pada penjumlahan. Itu sebabnya hasilnya 7. Jika kita ingin hasilnya itu menjadi 9, dimana yang dikerjakan adalah 1 + 2 terlebih dahulu, maka kita bisa mengatasinya dengan pengelempokan (grouping). (1 + 2 ) \* 3 output: 9

**Variable**

Variable adalah sebuah nilai yang bisa berubah-ubah. Untuk membuat sebuah variable di Javascript, kita menggunakan kata kunci: Let dan Var.

Contoh:

var X = 10.

ket:

var merupakan kata kunci dari javascript

X merupakan nama variable

= merupakan operator dalam hal ini assigment

10 merupakan nilai yang di assign/diberikan kepada variable X

Setelah kita mendeklarasikan variable tersebut, sekarang kita sudah bisa memakainya di code kita.

contoh:

var x = 10

x + 5, ouput 15

Pemberian nilai ulang (reassigning) ke variable.

var x = 10

x = 5

ketika kita print x dengan : console.log(x) maka outputnya adalah 5.

Tata cara pemberian nama ke variable:

- diawali dengan abjad, \_underscore(\_), atau $

- setelah abjad, underscore, atau $ kita bisa menlanjutkan dengan huruf, angka.

- tidak boleh menggunakan key yang sudah dipakai oleh javascript (reserved key). contoh: var, on, dll.

**Function**

Function merupakan block kode yang bisa diberi nama dan digunakan ulang.

Contoh:

function penjumlahan(x, y) {

   return x + y

}

Ket:

function : keyword untuk membuat function di javascript

penjumlahan : nama function , penamaan itu terserah kita

x, y : parameter untuk function itu (sebuah function tidak harus ada parameter).

return : keyword untuk mengembalikan nilai function di javascript

Menggunakan/ memanggil function.

contoh:

function buatText() {

   return "Hello world";

}

echo buatText();

output:

"Hello World"

function penjumlahan(x, y){

   return x + y;

}

echo penjumlahan (5, 2);

output:

7

**Condition**

Condition menentukan sebuah code akan dijalankan atau tidak. Istilah sehari-harinya, condition adalah pemerikasaan sebuah kondisi.

Berikut beberapa condition di javascript:

if : jika benar sebuah block code akan dijalankan.

elif : jika di kondisi sebelumnya (if maupun elif) tidak sama, maka elif akan dilanjutkan ke pemeriksaan ini. Jika kondisinya benar maka block code di dalamnya akan dijalankan .

else : jika semua kondisi tidak ada yang benar, maka kondisi ini akan di jalankan.

contoh penggunaan:

1. if ( 10 > 5)

    {

       console.log("ini block if");

    }

output:

"ini block if"

2. if ( 10 < 5)

    {

       console.log("ini block if");

    }

   elif ( 10 > 5)

    {

      console.log("ini block elif")

     }

output:

"ini block elif"

3.  if ( 10 < 5)

    {

       console.log("ini block if");

    }

   elif ( 3 > 5)

    {

      console.log("ini block elif")

     }

   else {

      console.log("ini block else")

    }

output:

"ini block else"

**JS HTML DOM (Document Object Model)**

Berikut adalah gambaran DOM.

****

Gambar 1 Document Object Model (DOM)Penjelasan:

Ketika kita membuat 1 halaman html, maka kita itu sudah memulai code dari yang Namanya root element (dimulai dari tag html). Kemudian mengikut ke structure berikutnya. Sperti yang kita lihat di gambar yang sebelah kiri itu merupakan tag <head> dan semua element yang adi dalamnya. Dan yang sebelah kanan merupakan bagian tag **body.**

Dengan DOM, Javascript bisa membuat content maupun tag HTML secara dynamic. Berikut yang bisa kita buat menggunakan javascript terhadap DOM:

1. Mengganti semua element HTML di sebuah halaman
2. Mengganti attribute HTML disebuah halaman
3. Mengganti style css disebuah halaman
4. Menghapus dan menambah element dan attribute element HTML yang ada
5. Bisa bereaksi dengan semua event HTML yang ada
6. Membuat element html baru di sebuah halaman.

Sebenarnya apa sih DOM itu?

DOM adalah sebuah standarisasi dari w3c (World Wide Web) Consortium.

DOM dibuat untuk mengakses document.

*"The W3C Document Object Model (DOM) is a platform and language-neutral interface that allows programs and scripts to dynamically access and update the content, structure, and style of a document."*

Maksudnya adalah W3C DOM adalah sebuah platform dan Bahasa neutral interface dimana mengijinkan program dan script diakses, update content, struktur, dan style dari document secara dinamis.

W3C DOM dibagi menjadi 3 bagian yaitu:

1. Core DOM : model standard untuk semua type document
2. XML DOM : model standard untuk XML type document
3. HTML DOM : model standard untuk HTML type document

Nah disini kita akan membahas hanya bagian no 3 saja, yaitu HTML DOM.

HTML DOM

Seperti yang sudah dijelaskan di atas, bahwasannya HTML DOM adalah model standard untuk document object model untuk HTML. HTML DOM dibagi menjadi 4 bagian, yaitu:

1. Html elements sebagai Object
2. Property untuk semua element HTML
3. Method/ function untuk mengakses semua element HTML
4. Events untuk semua element HTML

Dengan demikian kita bisa mengartikan HTML DOM ini digunakan untuk mendapatkan, mengubah, menambah, dan menghapus element HTML.

1. Methods HTML DOM

* Method di HTML DOM merupakan sebuah aksi yang bisa kita gunakan di element html.
* Property adalah nilai dari sebuah element yang bisa diubah-ubah.

Perhatikan contoh berikut:

<html>  
<body>  
  
<p id="demo"></p>  
  
<script>  
document.**getElementById**("demo").**innerHTML** = "Hello World!";  
</script>  
  
</body>  
</html>

Dari snippet code di atas, kita bisa melihat getElementById adalah method dan innerHTML adalah properti.

**Javascript HTML DOM DOCUMENT**

Document adalah owner atau pembungkus dari semua object yang ada di halaman html kita. Seperti yang bisa kita lihat di gambar 1 mengenai gambaran tree dari HTML DOM. Jika kita lihat digambarnya, document adalah yang terluar/ paling atas. Itu sebabnya untuk mengakses element halaman, kita selalu mengawali dengan **object document**. Contoh : document.getElementById(“demo”), document.getElementByTagsName(“p”), dll.

Mengganti Element HTML.

Category porperti:

1. element.innerHTML = new html content : mengganti inner html sebuat element.

Contoh:

Var element = getElementById(‘demo’)

element.innerHTML = ‘Ini adalah nilai baru’.

1. element.attribut = new Value : mengganti value dari sebuah attribute
2. element.style.property = new style: mengganti style dari sebuah element.

Category method:

1. element.setAttribute(attribute, value) : mengganti nilai attribute dari sebuah element.

catatan: **element** dalam hal ini adalah variable dari sebuah element yang udah kita ambil dengan menggunakan method DOM. Baik itu getElementByID, getElementsByTagName, dll.

**Menghapus dan Menambah Elements**

1. document.createElement(element) : membuat sebuah element html
2. document.removeChild(element) : menghapus sebuat element html
3. document.appendChild(element) : menambah sebuah element html
4. document.replaceChild(new, old) : Mengganti/ replace sebuah element html
5. document.write(text) : menulis/ write ke dalam output sebuah html

**DOM ELEMENTS**

Disini kita akan mempelajari bagaimana cara mengakses element menggunakan javascript. Jika kita ingin memanipulasi element sebuah halaman html, hal yang harus kita lakukan pertama adalah mendapatkan element tersebut. Baik dari id, tagName, class, dll.

Adapun javascript menyediakan beberapa cara untuk mendapatkan element tersebut, diantaranya:

1. Berdasarkan id. Menggunakan **getElementById**
2. Berdasarkan tagName: Menggunakan **getElementsByTagName**
3. Berdasarkan className: menggunakan **getElementsByClassName**
4. Berdasarkan CSS selector : pelajarin css selector
5. Dengan HTML object collections

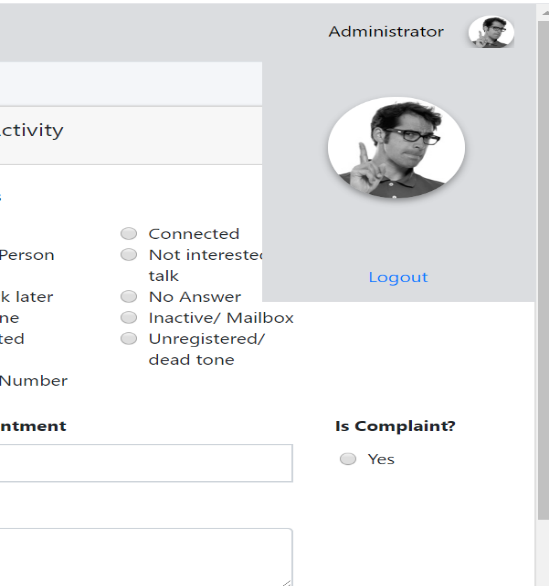
Setiap halaman akan berbeda, akan mendeteksi element,tag, id yang berbeda. Maksudnya adalah Ketika kita meload sebuah halaman, maka dia hanya akan meload semua id yang ada pada dia saja. Ini akan menjadi masalah Ketika kita membuat 1 file js yang sama untuk banyak halaman. Contoh: halaman tambah ticket dan halaman product. Di kedua halaman tersebut kita membuat javascript di 1 file javascript yang sama. Biasanya kita di dalam file js langsung panggil element (baik berdasarkan id, class, dll). Contoh di halaman ticket, ada input dengan id = ticket\_id. Kemudian di file js nya itu kita memanggilnya: getElementById(‘ticket\_id’). Nah, Ketika membuka halaman ticket, itu tidak menjadi masalah. Tapi akan menjadi masalah Ketika kita membuka halaman product. Di consolenya akan ada error yang mengatakan ticket\_id undefined.

**Event**

HTML events are **"things"** that happen to HTML elements (source: W3school).

Events adalah hal-hal yang terjadi terhadap element html. Contoh: click, dobule click, onchange, onmouse over, dll.  
Ada banyak event di dalam html maupun javascript.

Beriku contoh menampilkan dan menghide sebuah element. Dalam contoh ini kita akan membua sebuah case dimana akan menampilkan log-out Ketika mouse di arahkan gambar user yang sedang login.



Code di element html:

<ul class="navbar-nav ml-auto">

    <form class="form-inline">

      <div class="user-panel d-flex">

        <div class="info">

          <a href="#" class="d-block">Administrator</a>

        </div>

        <div class="image" id="image" onmouseenter="show()">

          <img src="<?php echo base\_url(); ?>assets/dist/img/user2-160x160.jpg" class="img-circle elevation-2" alt="User Image">

        </div>

      </div>

      <div class="log-out" id="log-out" onmouseleave="hide()">

        <div class="image-logout mt-5 ml-5" id="image-logout">

          <img src="<?php echo base\_url(); ?>assets/dist/img/user2-160x160.jpg" class="img-circle elevation-2" alt="User Image">

        </div>

        <p><a href="">Logout</a></p>

      </div>

    </form>

  </ul>

Yang perlu kita perhatikan di sini adalah pada element id image ada event onmouseenter=”show()”. Dimana ketika mouse diarahkan ke atas element itu akan menjalankan fungsi javascript.

Fungsi tersebut adalah :

function show() {

    $(".log-out").css('display', 'block');

}

Kemudian kita ingin menghide element tersebut kita mouse tidak di element id=”log-out” log-out tersebut. Disana kita buat event onmouseleave=”hide()”.

Fungsi js nya:

function hide() {

    $(".log-out").css('display', 'none');

}

**Property**

Property merupakan bagian terpenting di Javascript yang ada didalam object sebuah javascript. Property bisa ditambah, diganti, di hapus. Namun ada juga yang hanya bisa dibaca saja dengan kata lain tidak bisa di hapus maupun diganti.

Contoh : object person

var person = {

firstname:"John",

lastname:"Doe",

age:50,

eyecolor:"blue"

};

Object person di atas memiliki property : firstname, lastname, age, eyecolor.