

VALIDASI FORM CODEIGNITER

Memvalidasi form bertujuan untuk memfilter inputan data yang dikirimkan melalui form. Sebelum sebuah data diproses, terlebih dahulu dilakukan pengecekan sesuai format yang diinginkan.

Pada framework codeigniter, mekanisme validasi form telah disediakan. Codeigniter telah mendukung berbagai bentuk validasi yang umum digunakan, seperti harus terisi (**required**), harus angka (**numeric**), format email harus valid (**valid_email**), dan lain sebagainya. Untuk lebih detil terkait *set rule reference* nya dapat anda baca di *user guide* yang telah disertakan pada framework codeigniter.

LANGKAH-LANGKAH PEMBUATAN VALIDASI FORM DI CODEIGNITER

- ✓ Load library form_validation pada construct() atau menambahkan form_validation pada file **autoload.php**

Contoh load library form validation di construct

```
public function __construct()
{
    parent::__construct();

    $this->load->library('form_validation');
}
```

contoh load otomatis library form validation di file autoload.php

```
60 */
61 $autoload['libraries'] = array('form_validation');
62
```

- ✓ Menambahkan **set rule** pada setiap form input
- ✓ Menjalankan proses validasi form sesuai dengan set rule yang ditentukan

Contoh:

```
if ($this->form_validation->run() == TRUE) {
    //jika validasi sukses
} else {
    //jika validasi gagal
}
```

- ✓ Untuk menampilkan notifikasi validasi error dapat mengirimkan "**validation_error()**" ke sebuah view.
-

LATIHAN

1. Konfigurasi file “**autoload.php**”, dengan menambahkan “**form_validation**” pada bagian **\$config['libraries']** - line 61.

```
60 */
61 $autoload['libraries'] = array('form_validation');
62
```

2. Untuk file view menggunakan halaman **pendaftaran_view.php** yang telah kalian buat pada praktikum sebelumnya
3. Buat method baru dengan nama “**simpan**” pada controller “**pendaftaran**” untuk memproses data pendaftaran.

```
public function simpan()
{
}
```

4. Buat kondisi di dalam method “**simpan**” untuk mendeteksi ketika tombol/button submit diklik.

```
//jika button submit/daftar di klik
if($this->input->post('submit')){

}
}
```

*ket: 'submit' merupakan **name** button submit

5. Menambahkan **set rule** untuk menentukan aturan inputan yang dikehendaki.

```
//set rule di setiap form input
$this->form_validation->set_rules('nama', 'Nama', 'trim|required');
$this->form_validation->set_rules('alamat', 'Alamat', 'trim|required');
$this->form_validation->set_rules('telp', 'No.Telp', 'trim|required|numeric');
$this->form_validation->set_rules('asal_smp', 'Asal SMP', 'trim|required');
```

6. Menjalankan proses validasi form menggunakan *form validation run*

```
if ($this->form_validation->run() == TRUE) {
    //jika sukses
    $data['notif'] = 'Validasi sukses';

    $data['main_view'] = 'pendaftaran_view';
    $this->load->view('template', $data);
} else {
    //jika gagal
    $data['notif'] = validation_errors();
    $data['main_view'] = 'pendaftaran_view';
    $this->load->view('template', $data);
}
```

7. Untuk menampilkan notifikasi gagal memvalidasi dapat mengirimkan data ke view berupa “**validation_errors()**”.
8. Tambahkan script berikut tepat di atas tag <form> pada file “**pendaftaran_view.php**” untuk menampilkan notif validasi.

```
<?php
    if(!empty($notif)){
        echo '<div class="alert alert-danger">';
        echo $notif;
        echo '</div>';
    }
?>
```

TUGAS

Berikan fungsi validasi pada halaman login, yang telah anda buat pada praktikum sebelumnya.