MODUL
WEB PROGRAMMING 2

Oleh :

Agus Heryanto, ST., M.Kom

STTB SEKOLAH TINGGI TEKNOLOGI BANDUNG	**TEKNIK INFORMATIKA** **SEKOLAH TINGGI TEKNOLOGI BANDUNG** **2019/2020**

Membuat Fitur CRUD

Pada tutorial ini, kita akan mengerjakan banyak hal. Mulai dari membuat database, menyiapkan library, membuat model, sampai membuat CRUD.

CRUD *(Create, Read, Update Delete)* adalah fitur dasar yang harus kita buat saat bekerja dengan database.

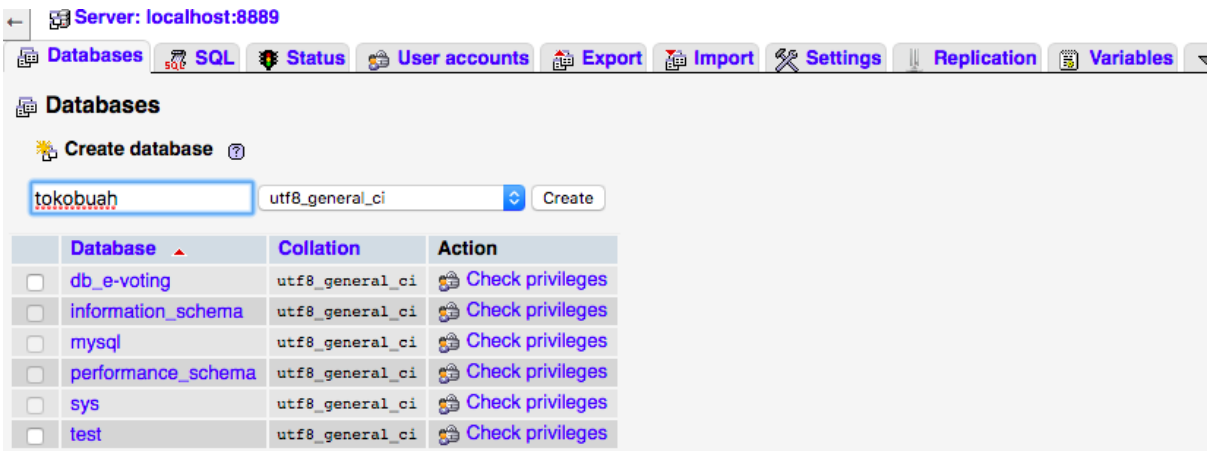
Berikut ini daftar pekerjaannya...

TODO:

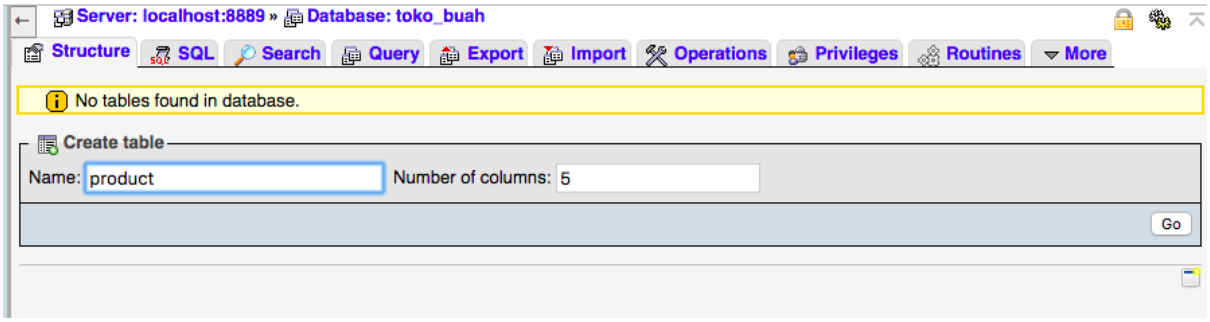
- Membuat Database;
- Konfigurasi Codeigntier
- Membuat Model untuk Tabel;
- Membuat Controller;
- Membuat View;
- Membuat Form Add;
- Membuat Form Edit;
- Membuat Fitur Hapus Data;

Membuat Database untuk Codeigniter

Silahkan buka PHPMyadmin, kemudian buatlah database baru dengan nama tokobuah.

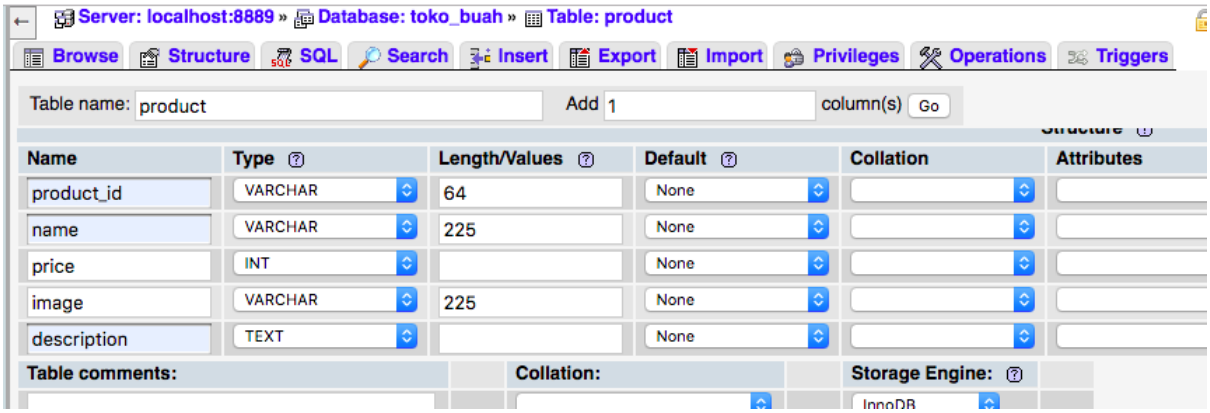


Setelah itu, buat tabel products dengan 5 kolom. Tabel ini nanti akan menyimpan data produk.

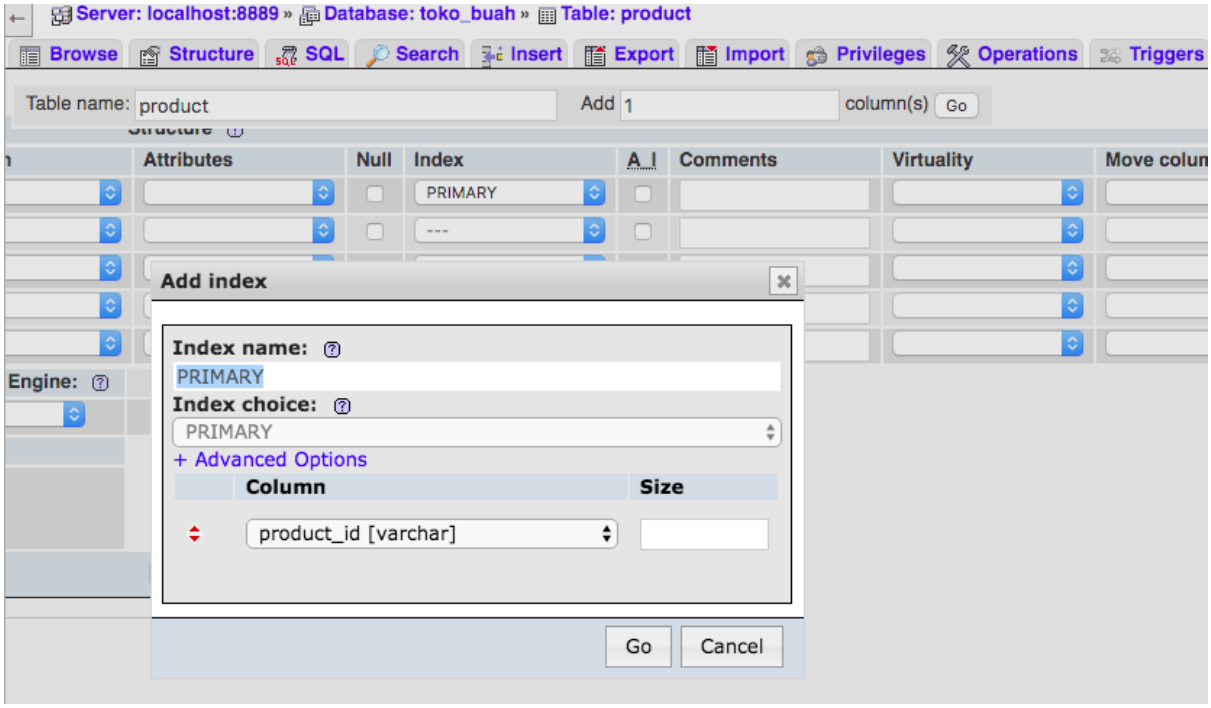


Kolom yang dibutuhkan:

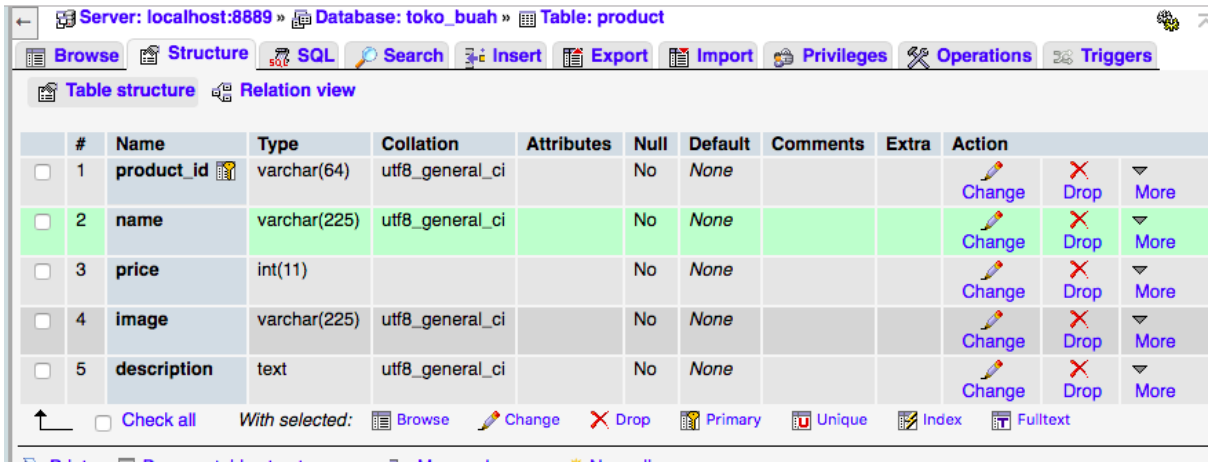
1. product_id (Primary Key) bertipe string dengan panjang 64;
2. name bertipe string dengan panjang 255.
3. price bertipe integer.
4. image bertipe string dengan panjang 255.
5. description bertipe TEXT.



...dan jangan lupa jadikan kolom product_id sebagai **primary key**.



Sehingga kita sekarang punya tabel products dengan struktur seperti ini:



Terakhir, klik **Save** untuk menyimpan.

Kode SQL-nya:

```
CREATE TABLE `products` (
  `product_id` varchar(64) NOT NULL,
  `name` varchar(255) NOT NULL,
  `price` int(11) NOT NULL,
  `image` varchar(255) NOT NULL DEFAULT 'default.jpg',
  `description` text NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Konfigurasi Codeigniter

Silahkan buka config/database.php, kemudian isi seperti ini:

```
$db['default'] = array(
        'dsn'   => '',
        'hostname' => 'localhost',
        'username' => 'root',
        'password' => '',
        'database' => 'tokobuah',
        'dbdriver' => 'mysqli',
        'dbprefix' => '',
        'pconnect' => FALSE,
        'db_debug' => (ENVIRONMENT !== 'production'),
        'cache_on' => FALSE,
        'cachedir' => '',
        'char_set' => 'utf8',
        'dbcollat' => 'utf8_general_ci',
        'swap_pre' => '',
        'encrypt' => FALSE,
        'compress' => FALSE,
        'stricton' => FALSE,
        'failover' => array(),
        'save_queries' => TRUE
);
```

Silahkan ubah sesuai dengan konfigurasi server mysql pada komputermu.

Jika kamu menggunakan XAMPP, password-nya biasanya tidak ada dan user yang digunakan adalah root.
Berikutnya, silahkan buka config/autoload.php.

Kemudian cari $autoload['libraries'] dan tambahkan database dan session di sana.

```
$autoload['libraries'] = array('database', 'session');
```

Ini artinya, kita akan me-load library database dan session secara otomatis.

Apa fungsinya?

- Library database akan menyediakan fungsi-fungsi untuk operasi database. Kita butuh ini, karena kita akan menggunakan database dalam aplikasi;
- Library session menyediakan fungsi-fungsi untuk mengakses variable $_SESSION. Kita butuh ini untuk menampilkan *flash message* dan membuat login.

Dengan demikian konfigurasi selesai...

Jika ada yang ingin kita konfigurasi lagi atau mau tambah library lagi, nanti kita bisa ubah konfigurasinya.

Untuk saat ini, kita cukup butuh konfigurasi *database* dan *autoload library* saja.

Berikutnya, kita akan mulai menulis kode untuk model.

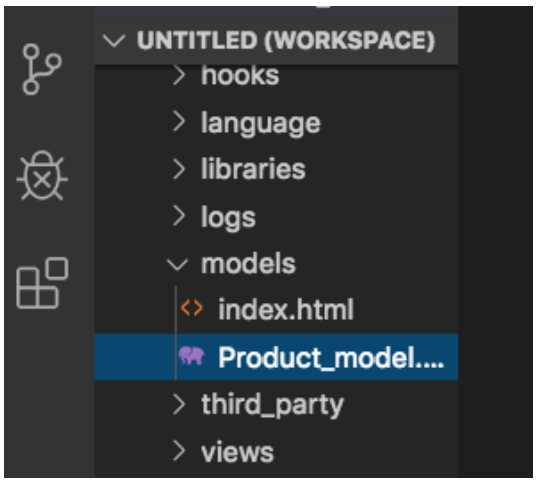
Membuat Model untuk Tabel

Model merupakan *class* atau kode yang berhubungan dengan data.

Di dalam model, kita akan membuat pemodelan data dari database. Sehingga kita akan lebih mudah mengaksesnya.

Biasanya satu tabel, dibuatkan satu modelnya.

Silahkan buat file baru di dalam direktori application/model/ dengan nama Product_model.php.



Perhatikan, namanya harus diawali dengan huruf kapital.

Pada contoh di atas, P adalah huruf kapital.

Lalu untuk akhiran _model ini bersifat opsional.

Nanti kita juga akan membuat class *Controller* yang bernama Products.php, karena itu kita sebaiknya menggunakan akhiran _model pada class *Model*.

Setelah membuat file, lalu apa lagi?

Selanjutnya kita akan mulai menulis kode untuk Product_model.php.

Silahkan **ketik** kode berikut

```php
<?php defined('BASEPATH') OR exit('No direct script access allowed');

class Product_model extends CI_Model
{
    private $_table = "products";

    public $product_id;
    public $name;
    public $price;
    public $image = "default.jpg";
    public $description;

    public function rules()
    {
        return [
            ['field' => 'name',
            'label' => 'Name',
            'rules' => 'required'],

            ['field' => 'price',
            'label' => 'Price',
            'rules' => 'numeric'],

            ['field' => 'description',
            'label' => 'Description',
            'rules' => 'required']
        ];
    }

    public function getAll()
    {
        return $this->db->get($this->_table)->result();
    }

    public function getById($id)
    {
        return $this->db->get_where($this->_table, ["product_id" =>
$id])->row();
    }

    public function save()
    {
        $post = $this->input->post();
        $this->product_id = uniqid();
        $this->name = $post["name"];
        $this->price = $post["price"];
        $this->description = $post["description"];
        $this->db->insert($this->_table, $this);
    }

    public function update()
    {
        $post = $this->input->post();
        $this->product_id = $post["id"];
        $this->name = $post["name"];
        $this->price = $post["price"];
        $this->description = $post["description"];
```

```
        $this->db->update($this->_table, $this, array('product_id' =>
$post['id']));
    }

    public function delete($id)
    {
        return $this->db->delete($this->_table, array("product_id" =>
$id));
    }
}
```

Kode di atas memang belum bisa kita coba, karena ini hanya pemodelan data saja.

Nanti kita akan gunakan fungsi-fungsi atau method yang ada di dalam kode ini pada class *Controller*.

Pertama silahkan perhatikan bagian ini:

```
private $_table = "products"; //nama tabel

// nama kolom di tabel, harus sama huruf besar dan huruf kecilnya!
public $product_id;
public $name;
public $price;
public $image = "default.jpg";
public $description;
```

Ini adalah properti atau variabel yang kita butuhkan dalam model *Product*

Pada properti $_tabel kita memberikan modifier private, karena property ini hanya akan digunakan pada class ini saja.

Jika kamu pernah belajar OOP, pasti paham.

Lalu pada properti $image, kita langsung mengisi nilainya dengan "default.jpg".

Ini nanti akan menjadi nilai default-nya, sebenarnya kita bisa saja tidak isi demikian.

Karena di tabel sudah kita berikan nilai default-nya.

Selanjutnya silahkan perhatikan method rules():

```
public function rules()
{
    return [
        ['field' => 'name',
        'label' => 'Name',
        'rules' => 'required'],

        ['field' => 'price',
```

```
            'label' => 'Price',
            'rules' => 'numeric'],

            ['field' => 'description',
            'label' => 'Description',
            'rules' => 'required']
        ];
    }
```

Method ini akan mengembalikan sebuah array yang berisi *rules*.

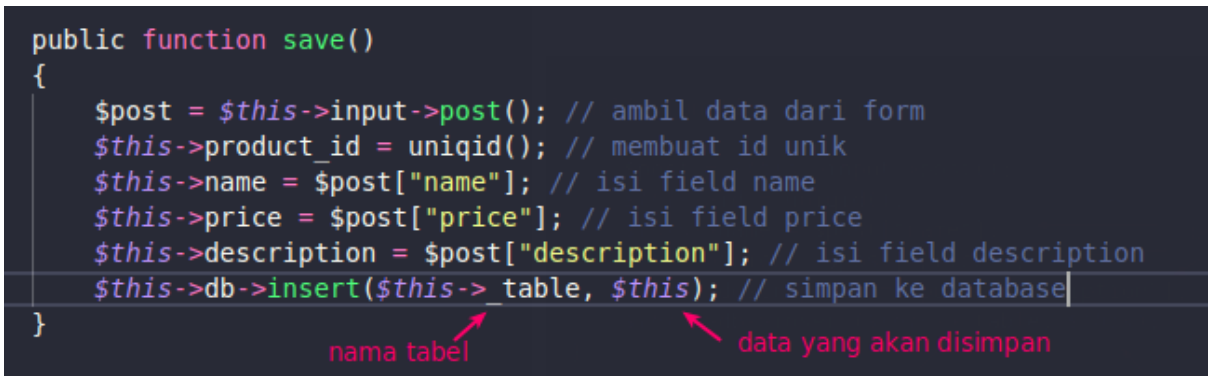
Rules ini nanti kita butuhkan untuk validasi input.

Pada *Rules* di atas, kita memberikan aturan untuk wajib mengisi *(required)* field name, price, dan description.

Berikutnya, silahkan perhatikan method get() dan getAll(). Kedua method ini akan kita gunakan untuk mengambil data dari database.


```
public function getAll()
{
    return $this->db->get($this->_table)->result();
    // ini sama artinya seperti:
    // SELECT * FROM products
    // method ini akan mengembalikan sebuah array
    // yang berisi objek dari row
}

public function getById($id)
{
    return $this->db->get_where($this->_table, ["product_id" => $id])->row();
    // ini sama artinya seperti:
    // SELECT * FROM products WHERE product_id=$id
    // method ini akan mengembalikan sebuah objek
}
```

Berikutnya perhatikan method save():




```
public function save()
{
    $post = $this->input->post(); // ambil data dari form
    $this->product_id = uniqid(); // membuat id unik
    $this->name = $post["name"]; // isi field name
    $this->price = $post["price"]; // isi field price
    $this->description = $post["description"]; // isi field description
    $this->db->insert($this->_table, $this); // simpan ke database
}
```

Method ini akan kita gunakan untuk menyimpan data ke tabel product.

Kita mengambil input yang dikirim dari form menggunakan $this->input->post().

Mengapa ini ditulis di model?

Biasanya orang menuliskannya pada *Controller*. Namun, biar *Controller* lebih fokus mengatur hubungan Model dengan View, maka sebaiknya ini kita tulis di Model.

Karena nanti pada *Controller*, kita tinggal validasi saja inputannya.

Untuk method berikutnya hampir sama.

Method update() untuk update data dan delete() untuk menghapus data.

Pada method update(), kita mengisi $this->product_id dengan **id** yang didapatkan dari *form* ($post['id']). Karena ini untuk update.

Sedangkan pada method save(), kita mengisinya dengan fungsi uniqid(). Karena kita akan membuat baru.

Fungsi ini nantinya akan menghasilkan karakter unik.

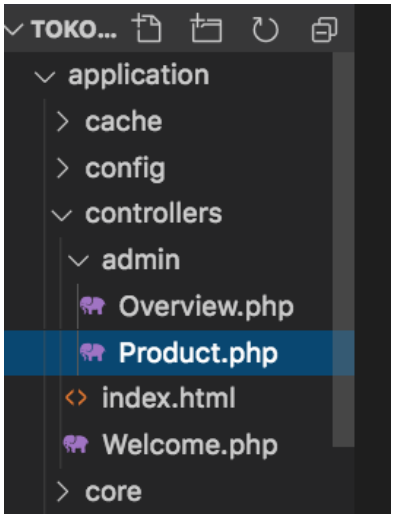
Membuat Controller

Controller adalah bagian dari CI yang bertugas untuk menangani *HTTP request* dan menghubungan Model dengan View.

Pada *Controller*, kita akan memanggil method-method yang ada di dalam model untuk mendapatkan data.

Setelah itu data tersebut di-render ke dalam view dengan me-load-nya.

Silahkan buat file baru di dalam folder application/controllers/admin/ dengan nama Products.php.



Kemudian isi file Products.php dengan kode berikut:

```php
<?php

defined('BASEPATH') OR exit('No direct script access allowed');

class Products extends CI_Controller
{
    public function __construct()
    {
        parent::__construct();
        $this->load->model("product_model");
        $this->load->library('form_validation');
    }

    public function index()
    {
        $data["products"] = $this->product_model->getAll();
        $this->load->view("admin/product/list", $data);
    }

    public function add()
    {
        $product = $this->product_model;
        $validation = $this->form_validation;
        $validation->set_rules($product->rules());

        if ($validation->run()) {
            $product->save();
            $this->session->set_flashdata('success', 'Berhasil
disimpan');
        }

        $this->load->view("admin/product/new_form");
    }

    public function edit($id = null)
    {
        if (!isset($id)) redirect('admin/products');

        $product = $this->product_model;
        $validation = $this->form_validation;
        $validation->set_rules($product->rules());
```

```
        if ($validation->run()) {
            $product->update();
            $this->session->set_flashdata('success', 'Berhasil
disimpan');
        }

        $data["product"] = $product->getById($id);
        if (!$data["product"]) show_404();

        $this->load->view("admin/product/edit_form", $data);
    }

    public function delete($id=null)
    {
        if (!isset($id)) show_404();

        if ($this->product_model->delete($id)) {
            redirect(site_url('admin/products'));
        }
    }
}
```

Ada lima method dalam class tersebut:

1. Method __construct()

Method __construct() merupakan sebuah konstruktor. Method ini yang akan dieksekusi pertama kali saat *Controller* diakses.

Pada method ini, kita melakukan load *model* (porduct_model) dan *library* (form_validation).

```
public function __construct()
{
    parent::__construct();
    $this->load->model("product_model");
    $this->load->library('form_validation');
}
```

Library form_validation akan kita gunakan untuk memvalidasi input pada method add() dan edit().

Mengapa harus divalidasi?

Karena bisa saja pengguna menginputkan data sembarang. Misalnya, sengaja mengisi dengan data kosong, script jahat seperti: serangan XSS, dll.

Intinya:

Sebelum menyimpan data ke database, pastikan data tersebut sudah benar.

2. Method index()

Pada method ini, kita akan mengambil data dari *model* dengan memanggil method product_model->getAll().

Lalu kita me-rendernya ke dalam view admin/product/list.

```
public function index()
{
    $data["products"] = $this->product_model->getAll();
    $this->load->view("admin/product/list", $data);
}
```

View admin/product/list belum ada. Nati kita akan membuatnya.

3. Method add()

Method ini bertugas untuk menampilkan *form add* dan menyimpan data ke database. Tentunya dengan memanggil method save() yang ada pada model.

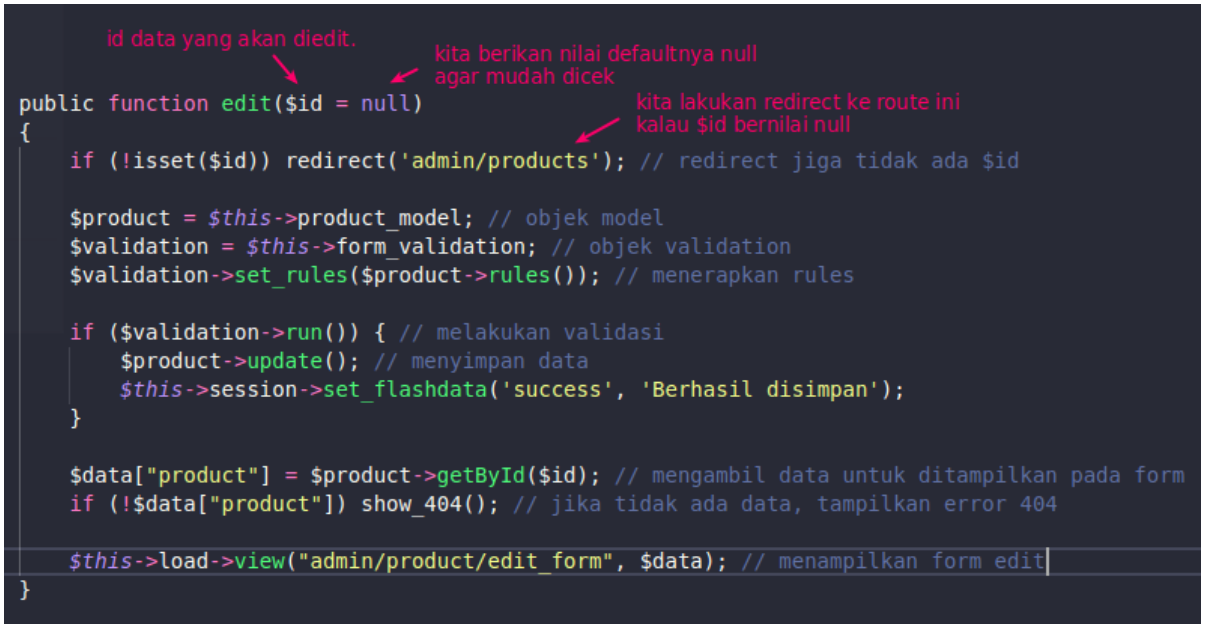
Namun, sebelum memanggil method save(), kita lakukan validasi terlebih dahulu dengan mengeksekusi method run() pada objek $validation.

Jika berhasil, maka pesan "Berhasil disimpan" akan ditampilkan.

Terakhir, kita akan menampilkan view product/new_form. View ini berisi sebuah form untuk menambah data.

4. Method edit()

Hampir sama dengan method add(), method edit() juga bertugas untuk menampilkan form dan menyimpan data.




```php
public function edit($id = null)
{
    if (!isset($id)) redirect('admin/products'); // redirect jiga tidak ada $id

    $product = $this->product_model; // objek model
    $validation = $this->form_validation; // objek validation
    $validation->set_rules($product->rules()); // menerapkan rules

    if ($validation->run()) { // melakukan validasi
        $product->update(); // menyimpan data
        $this->session->set_flashdata('success', 'Berhasil disimpan');
    }

    $data["product"] = $product->getById($id); // mengambil data untuk ditampilkan pada form
    if (!$data["product"]) show_404(); // jika tidak ada data, tampilkan error 404

    $this->load->view("admin/product/edit_form", $data); // menampilkan form edit
}
```

Nilai $id akan didapatkan dari mana?

Nilai $id akan kita dapatkan dari parameter atau argumen pada URL.

5. Method delete()

Method delete() befungsi untuk menangni penghapusan data.

Prinsipnya hampir sama seperti method edit(), method delete() juga membutuhkan $id untuk menentukan data mana yang akan dihapus.

```php
public function delete($id=null)
{
    if (!isset($id)) show_404();

    if ($this->product_model->delete($id)) {
        redirect(site_url('admin/products'));
    }
}
```

Apabila data berhasil dihapus, maka kita langsung alihkan (redirect()) menuju ke halaman admin/products/.

Membuat View

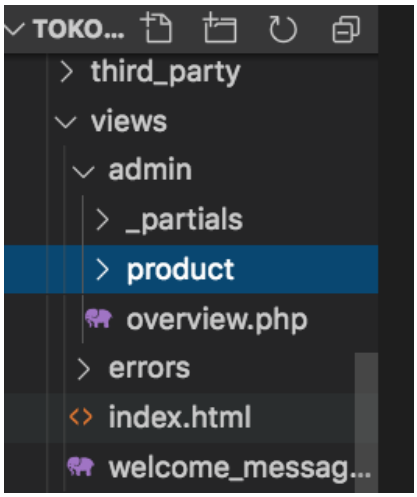
View merupakan bagian yang bertugas mengurus tampilan.

Ada tiga macam view yang harus kita buat dalam aplikasi ini:

1. list.php untuk menampilkan data;
2. new_form.php untuk menampilkan form tambah data;

3. dan edit_form.php untuk menampilkan form edit data.

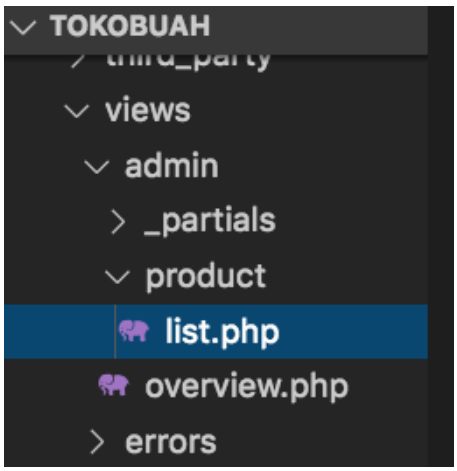
Tapi sebelum itu, silahkan buat folder baru pada direktori views/admin dengan nama product.



Setelah itu, kita akan membuat ketiga view di dalam folder views/admin/product/.

1. View List Data

Buatlah file baru dengan nama list.php di dalam folder views/admin/product/



Setelah itu, isi dengan kode berikut:

```
<!DOCTYPE html>
<html lang="en">

<head>
        <?php $this->load->view("admin/_partials/head.php") ?>
</head>

<body id="page-top">

        <?php $this->load->view("admin/_partials/navbar.php") ?>
        <div id="wrapper">
```

```
                <?php $this->load->view("admin/_partials/sidebar.php") ?>

                <div id="content-wrapper">

                        <div class="container-fluid">

                                <?php $this->load->view("admin/_partials/breadcrumb.php") ?>

                                <!-- DataTables -->
                                <div class="card mb-3">
                                        <div class="card-header">
                                                <a href="<?php echo site_url('admin/products/add') ?>"><i class="fas fa-plus"></i> Add New</a>
                                        </div>
                                        <div class="card-body">

                                                <div class="table-responsive">
                                                        <table class="table table-hover" id="dataTable" width="100%" cellspacing="0">
                                                                <thead>
                                                                        <tr>

        <th>Name</th>

        <th>Price</th>

        <th>Photo</th>

        <th>Description</th>

        <th>Action</th>
                                                                        </tr>
                                                                </thead>
                                                                <tbody>
                                                                        <?php foreach ($products as $product): ?>
                                                                        <tr>

        <td width="150">

                <?php echo $product->name ?>

        </td>

        <td>

                <?php echo $product->price ?>

        </td>

        <td>

                <img src="<?php echo base_url('upload/product/'.$product->image) ?>" width="64" />

        </td>
```

```
        <td class="small">

                <?php echo substr($product->description, 0, 120)
?>...</td>

        <td width="250">

                <a href="<?php echo
site_url('admin/products/edit/'.$product->product_id) ?>"

                 class="btn btn-small"><i class="fas fa-edit"></i>
Edit</a>

                <a onclick="deleteConfirm('<?php echo
site_url('admin/products/delete/'.$product->product_id) ?>')"

                 href="#!" class="btn btn-small text-danger"><i class="fas
fa-trash"></i> Hapus</a>

        </td>
                                                                    </tr>
                                                                    <?php
endforeach; ?>

                                                                </tbody>
                                                            </table>
                                                        </div>
                                                    </div>
                                                </div>

                                            </div>
                                            <!-- /.container-fluid -->

                                            <!-- Sticky Footer -->
                                            <?php $this->load-
>view("admin/_partials/footer.php") ?>

                </div>
                <!-- /.content-wrapper -->

        </div>
        <!-- /#wrapper -->


        <?php $this->load->view("admin/_partials/scrolltop.php") ?>
        <?php $this->load->view("admin/_partials/modal.php") ?>

        <?php $this->load->view("admin/_partials/js.php") ?>

</body>

</html>
```

View porduct/list.php bertugas untuk menampilkan data pada halaman admin.

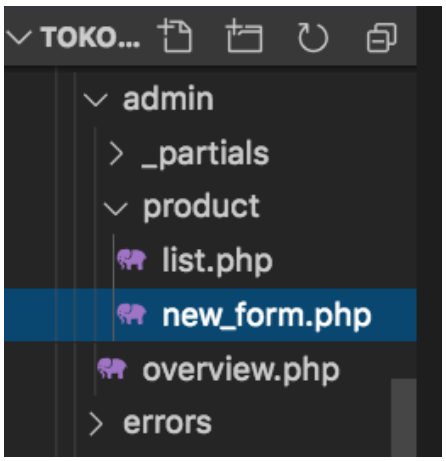
Datanya kita dapat dari *Controller*, coba saja perhatikan method index() pada controller Product.php.

```
public function index()
{
    $data["products"] = $this->product_model->getAll(); // ambil data
dari model
    $this->load->view("admin/product/list", $data); // kirim data ke view
}
```

2. Membuat Form Add

View berikutnya yang harus kita buat adalah new_form.php.

Silahkan buat file baru di dalam folder view/admin/product/ dengan nama new_form.php.



Setelah itu, isi dengan kode berikut:

```
<!DOCTYPE html>
<html lang="en">

<head>
        <?php $this->load->view("admin/_partials/head.php") ?>
</head>

<body id="page-top">

        <?php $this->load->view("admin/_partials/navbar.php") ?>
        <div id="wrapper">

                <?php $this->load->view("admin/_partials/sidebar.php") ?>

                <div id="content-wrapper">

                        <div class="container-fluid">

                                <?php $this->load-
>view("admin/_partials/breadcrumb.php") ?>
```

```
                                <?php if ($this->session->flashdata('success')): ?>
                                <div class="alert alert-success" role="alert">
                                        <?php echo $this->session->flashdata('success'); ?>
                                </div>
                                <?php endif; ?>

                                <div class="card mb-3">
                                        <div class="card-header">
                                                <a href="<?php echo site_url('admin/products/') ?>"><i class="fas fa-arrow-left"></i> Back</a>
                                        </div>
                                        <div class="card-body">

                                                <form action="<?php base_url('admin/product/add') ?>" method="post" enctype="multipart/form-data" >
                                                        <div class="form-group">
                                                                <label for="name">Name*</label>
                                                                <input class="form-control <?php echo form_error('name') ? 'is-invalid':'' ?>" type="text" name="name" placeholder="Product name" />
                                                                <div class="invalid-feedback">
                                                                        <?php echo form_error('name') ?>
                                                                </div>
                                                        </div>

                                                        <div class="form-group">
                                                                <label for="price">Price*</label>
                                                                <input class="form-control <?php echo form_error('price') ? 'is-invalid':'' ?>" type="number" name="price" min="0" placeholder="Product price" />
                                                                <div class="invalid-feedback">
                                                                        <?php echo form_error('price') ?>
                                                                </div>
                                                        </div>


                                                        <div class="form-group">
                                                                <label for="name">Photo</label>
                                                                <input class="form-control-file <?php echo form_error('price') ? 'is-invalid':'' ?>" type="file" name="image" />
```

```
                                                    <div
class="invalid-feedback">
                                                        <?php
echo form_error('image') ?>
                                                    </div>
                                                </div>

                                                <div class="form-
group">
                                                    <label
for="name">Description*</label>
                                                    <textarea
class="form-control <?php echo form_error('description') ? 'is-
invalid':'' ?>"
name="description" placeholder="Product description..."></textarea>
                                                    <div
class="invalid-feedback">
                                                        <?php
echo form_error('description') ?>
                                                    </div>
                                                </div>

                                                <input class="btn
btn-success" type="submit" name="btn" value="Save" />
                                            </form>

                                        </div>

                                        <div class="card-footer small text-
muted">
                                            * required fields
                                        </div>


                                    </div>
                                    <!-- /.container-fluid -->

                                    <!-- Sticky Footer -->
                                    <?php $this->load-
>view("admin/_partials/footer.php") ?>

                                </div>
                                <!-- /.content-wrapper -->

                            </div>
                            <!-- /#wrapper -->


                            <?php $this->load->view("admin/_partials/scrolltop.php")
?>

                            <?php $this->load->view("admin/_partials/js.php") ?>

</body>

</html>
```

View new_form.php betugas untuk menampilkan form input untuk pembuatan data baru.

Form ini akan mengirim data ke: /admin/products/add (controller products, method add).

Perhatikan pada form-nya.

```
<form action="<?php base_url('admin/product/add') ?>" method="post"
enctype="multipart/form-data" >
...
</form>
```

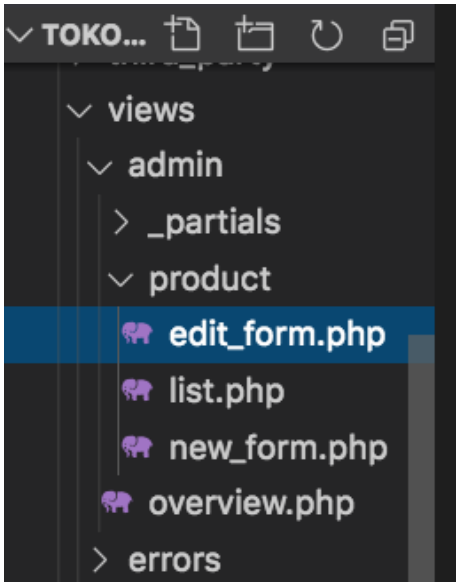
Pada form tersebut, kita menggunakan enctype="multipart/form-data". Ini nanti akan kita gunakan untuk upload file.

3. Membuat Form Edit

View terakhir yang harus kita buat adalah edit_form.php. Isi kodenya hampir sama seperti new_form.php.

Bedanya, di edit_form.php kita menampilkan nilai untuk setiap field-nya.

Silahkan buat file baru di dalam folder views/admin/product/ dengan nama edit_form.php.



Setelah itu, isi dengan kode berikut:

```
<!DOCTYPE html>
<html lang="en">

<head>
```

```
        <?php $this->load->view("admin/_partials/head.php") ?>
</head>

<body id="page-top">

        <?php $this->load->view("admin/_partials/navbar.php") ?>
        <div id="wrapper">

                <?php $this->load->view("admin/_partials/sidebar.php") ?>

                <div id="content-wrapper">

                        <div class="container-fluid">

                                <?php $this->load-
>view("admin/_partials/breadcrumb.php") ?>

                                <?php if ($this->session-
>flashdata('success')): ?>
                                <div class="alert alert-success"
role="alert">
                                        <?php echo $this->session-
>flashdata('success'); ?>
                                </div>
                                <?php endif; ?>

                                <!-- Card  -->
                                <div class="card mb-3">
                                        <div class="card-header">

                                                <a href="<?php echo
site_url('admin/products/') ?>"><i class="fas fa-arrow-left"></i>
                                                        Back</a>
                                        </div>
                                        <div class="card-body">

                                                <form action="<?php
base_url('admin/product/edit') ?>" method="post" enctype="multipart/form-
data">

                                                        <input type="hidden"
name="id" value="<?php echo $product->product_id?>" />

                                                        <div class="form-
group">
                                                                <label
for="name">Name*</label>
                                                                <input
class="form-control <?php echo form_error('name') ? 'is-invalid':'' ?>"
                                                                 type="text"
name="name" placeholder="Product name" value="<?php echo $product->name
?>" />
                                                                <div
class="invalid-feedback">
                                                                        <?php
echo form_error('name') ?>
                                                                </div>
                                                        </div>
```

```
                                                    <div class="form-
group">
                                                        <label
for="price">Price</label>
                                                        <input
class="form-control <?php echo form_error('price') ? 'is-invalid':'' ?>"
type="number" name="price" min="0" placeholder="Product price"
value="<?php echo $product->price ?>" />
                                                        <div
class="invalid-feedback">
                                                            <?php
echo form_error('price') ?>
                                                        </div>
                                                    </div>


                                                    <div class="form-
group">
                                                        <label
for="name">Photo</label>
                                                        <input
class="form-control-file <?php echo form_error('image') ? 'is-invalid':''
?>"
                                                         type="file"
name="image" />
                                                        <input
type="hidden" name="old_image" value="<?php echo $product->image ?>" />
                                                        <div
class="invalid-feedback">
                                                            <?php
echo form_error('image') ?>
                                                        </div>
                                                    </div>

                                                    <div class="form-
group">
                                                        <label
for="name">Description*</label>
                                                        <textarea
class="form-control <?php echo form_error('description') ? 'is-
invalid':'' ?>"
name="description" placeholder="Product description..."><?php echo
$product->description ?></textarea>
                                                        <div
class="invalid-feedback">
                                                            <?php
echo form_error('description') ?>
                                                        </div>
                                                    </div>

                                                    <input class="btn
btn-success" type="submit" name="btn" value="Save" />
                                                </form>

                                            </div>

                                            <div class="card-footer small text-
muted">
```

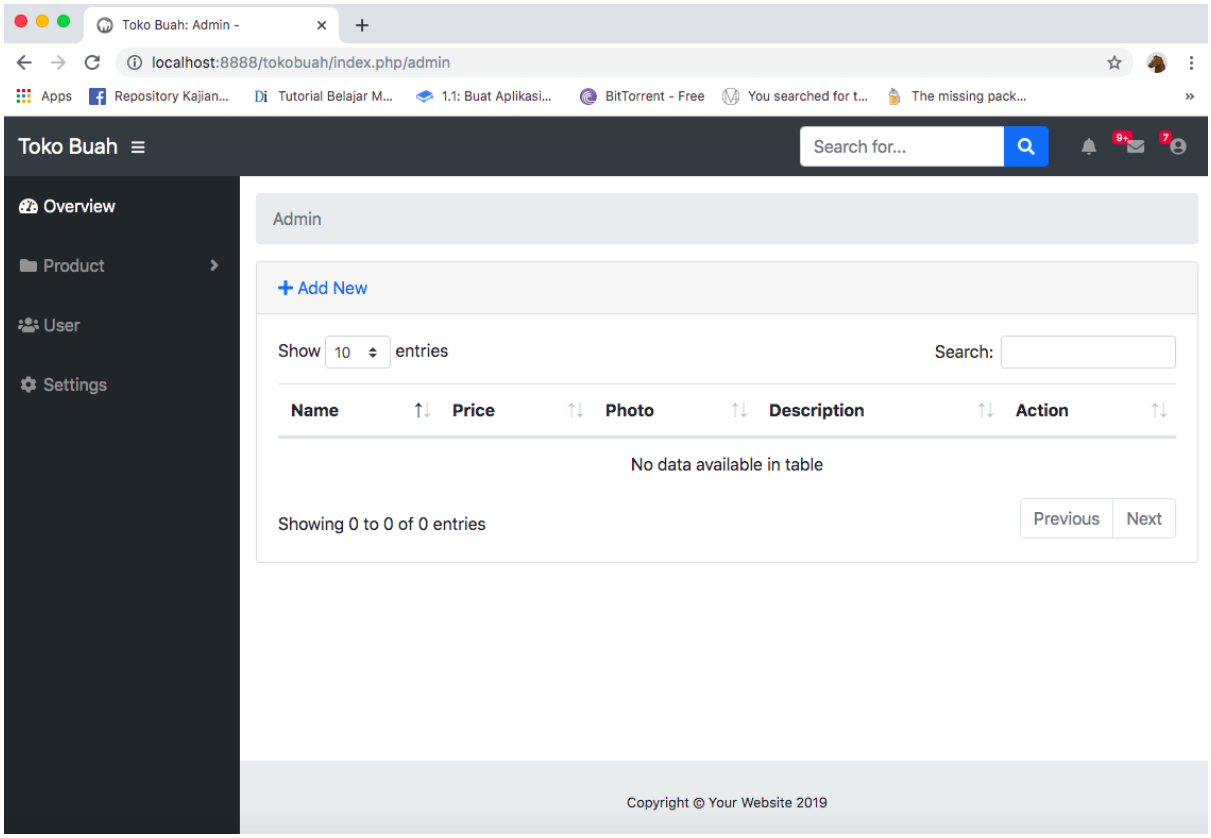
```
                                        * required fields
                                    </div>


                                </div>
                                <!-- /.container-fluid -->

                                <!-- Sticky Footer -->
                                <?php $this->load-
>view("admin/_partials/footer.php") ?>

                            </div>
                            <!-- /.content-wrapper -->

                    </div>
                    <!-- /#wrapper -->

                    <?php $this->load->view("admin/_partials/scrolltop.php")
?>

                    <?php $this->load->view("admin/_partials/js.php") ?>

</body>

</html>
```

Tapi sebelum itu, kita coba dulu aplikasinya.

Percobaan Aplikasi

Coba buka: http://localhost/tokobuah/index.php/admin/products/

Jika kamu tidak melihat error, maka itu artinya berhasil.

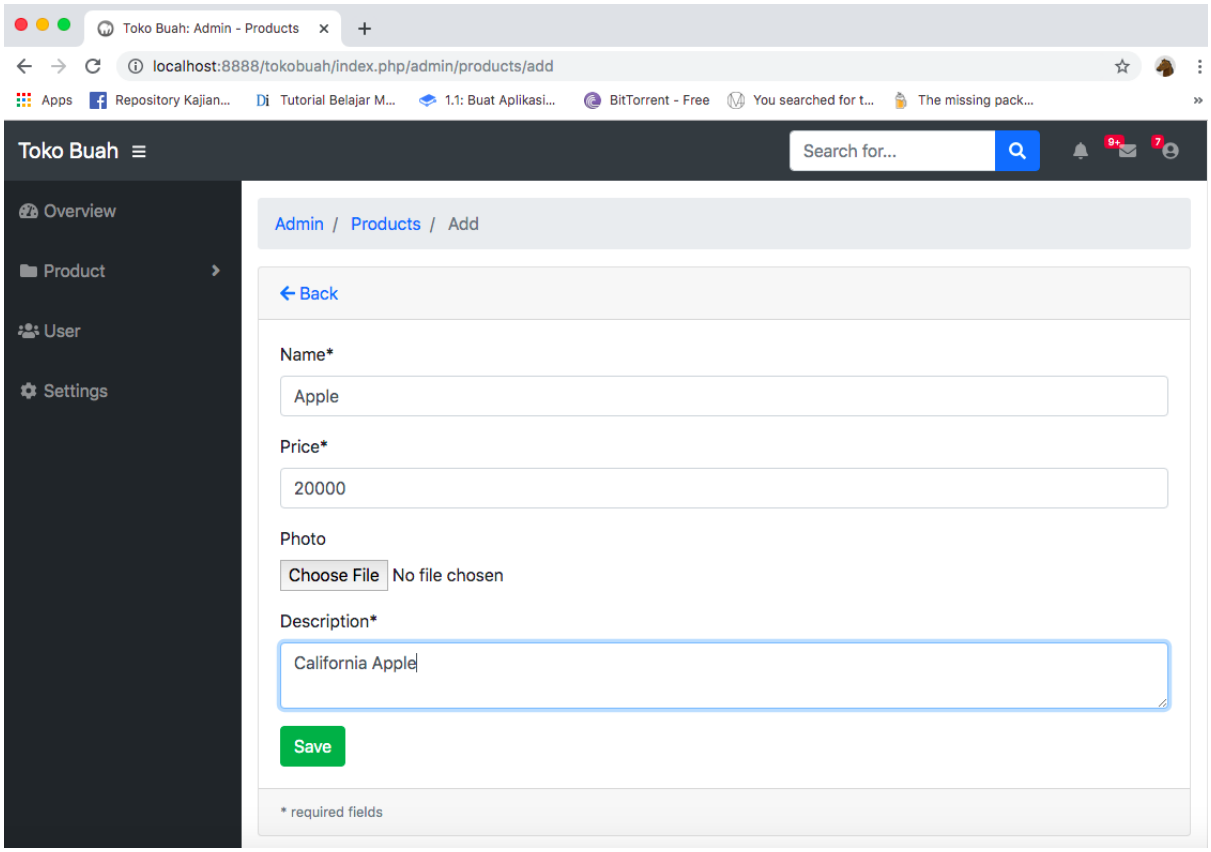


Belum ada data yang ditampilkan di dalam halaman list product, karena kita belum menambahkan data apapun.

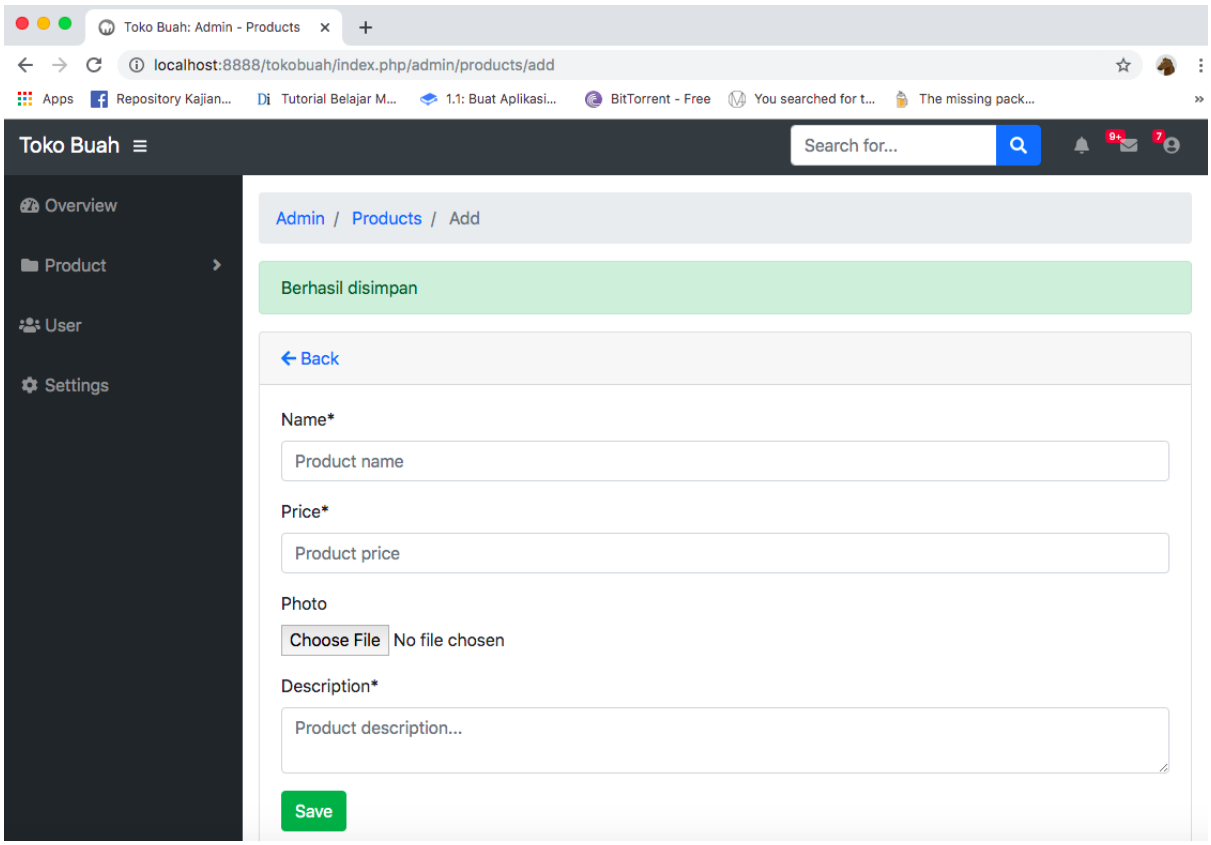
Mari kita coba buat data baru.

Silahkan buka: http://localhost/tokobuah/index.php/admin/products/add atau klik link **+ Add New**.

Kemudian isi dengan data baru.

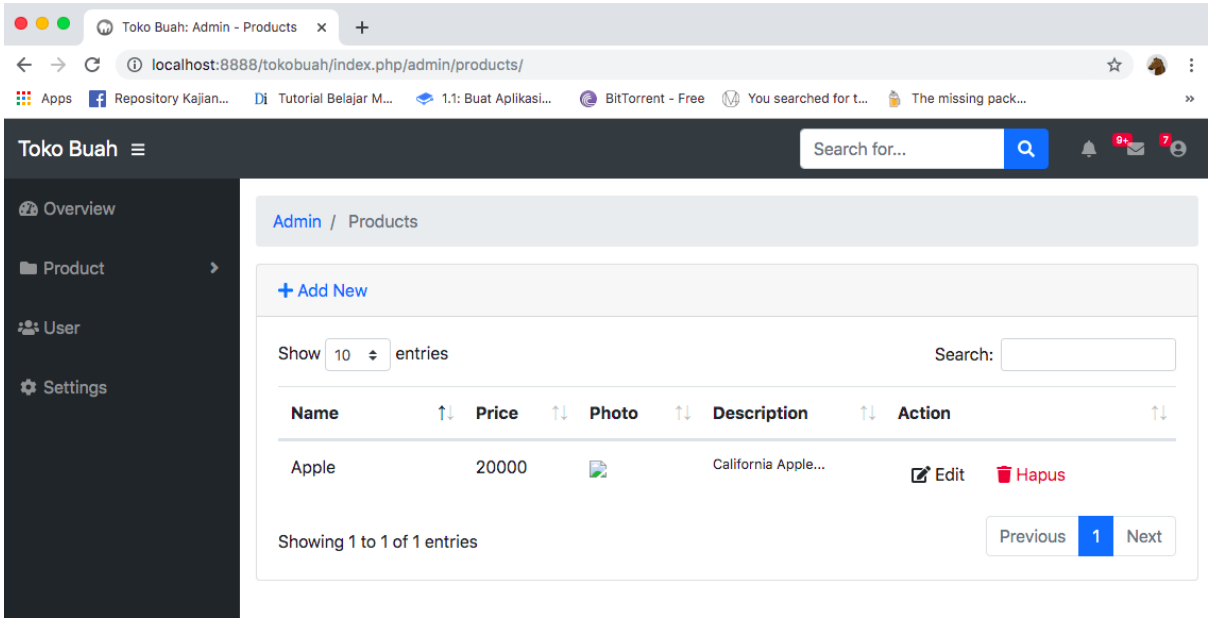


Klik **Save**, jika muncul seperti ini...

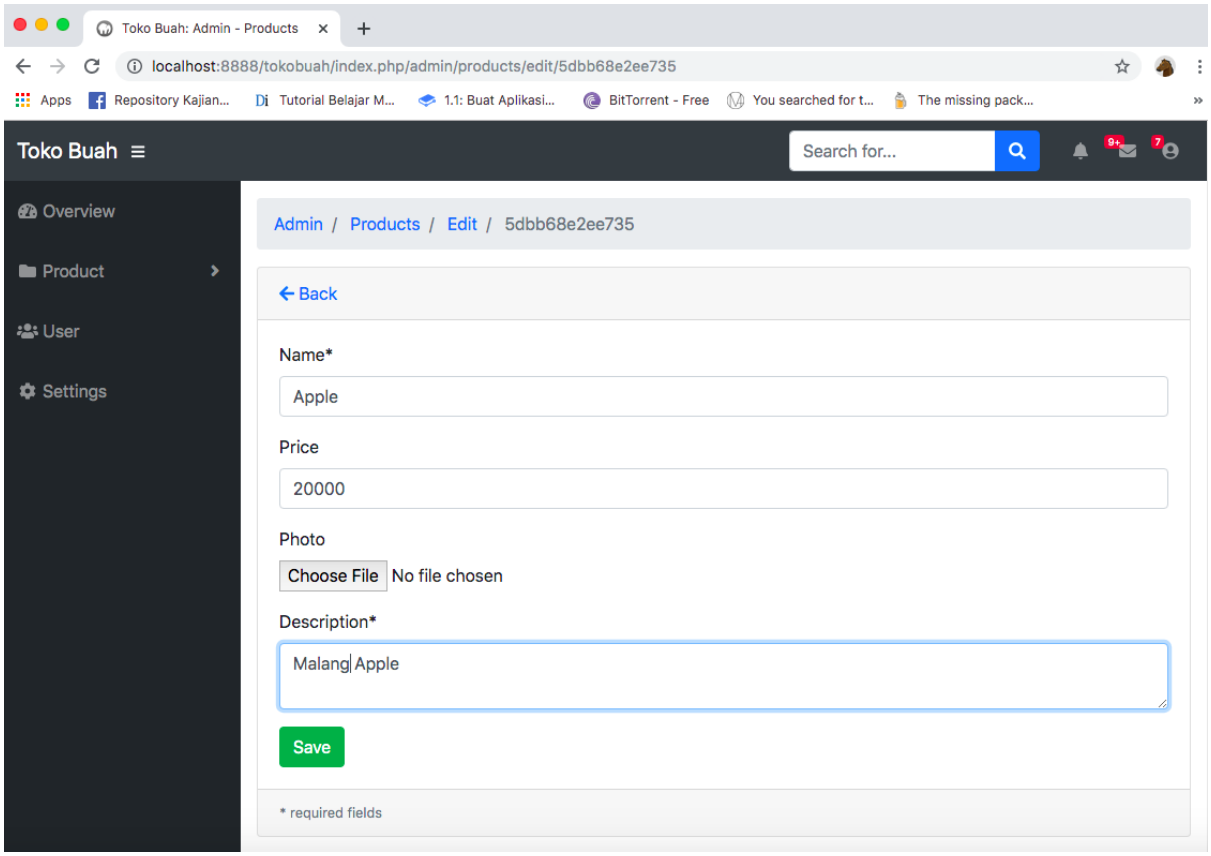


...artinya data berhasil ditambahkan.

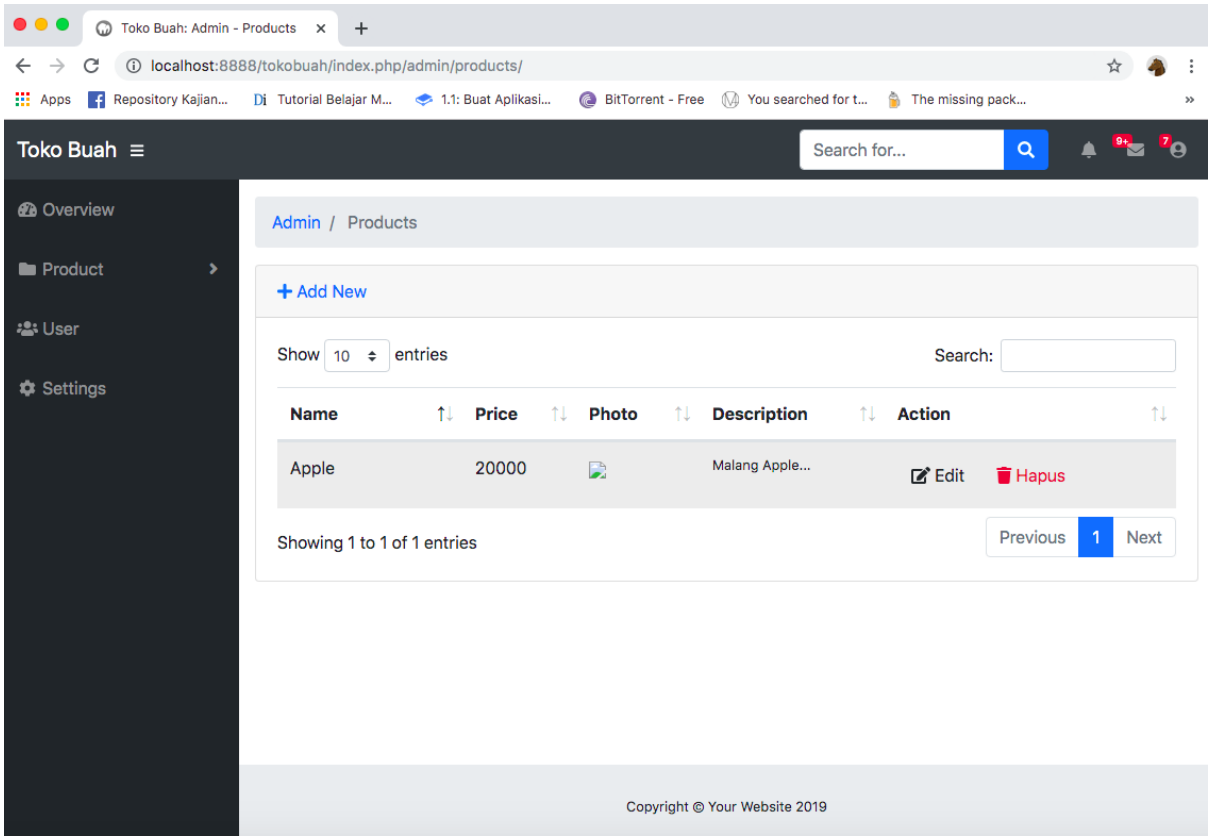
Coba periksa lagi halaman **products**.



Berikutnya coba lakukan update data. Klik tombol **Edit** yang ada di dekat tombol **Hapus**.



Hasilnya:



Fitur untuk Menghapus Data

Kita belum membuat fungsi konfirmasi pada tombol **Hapus**.

Mengapa kita perlu konfirmasi?

Karena tindakan ini berbahaya. Bisa saja nanti terjadi salah klik, kalau tidak dikonfirmasi data bisa hilang. Dan ini tentu akan menjadi pengalaman buruk bagi pengguna.

Sebenarnya pada tahapan ini, kita hanya akan membuat satu fungsi Javascript saja.

Karena kita memanggilnya pada tombol **Hapus**.

Perhatikan view list.php.

```
<a onclick="deleteConfirm('<?php echo
site_url('admin/products/delete/'.$product->product_id) ?>')"
    href="#!" class="btn btn-small text-danger"><i class="fas fa-
trash"></i> Hapus</a>
```

Di sana ada *event onclick* yang akan memanggil fungsi deleteConfirm(). Ini adalah fungsi javascript, bukan PHP.

Fungsi ini nanti akan menampilkan sebuah modal konfirmasi.

Oke, tapi di mana kita akan menulis kode Javascript?

Ada dua tempat yang bisa kita gunakan:

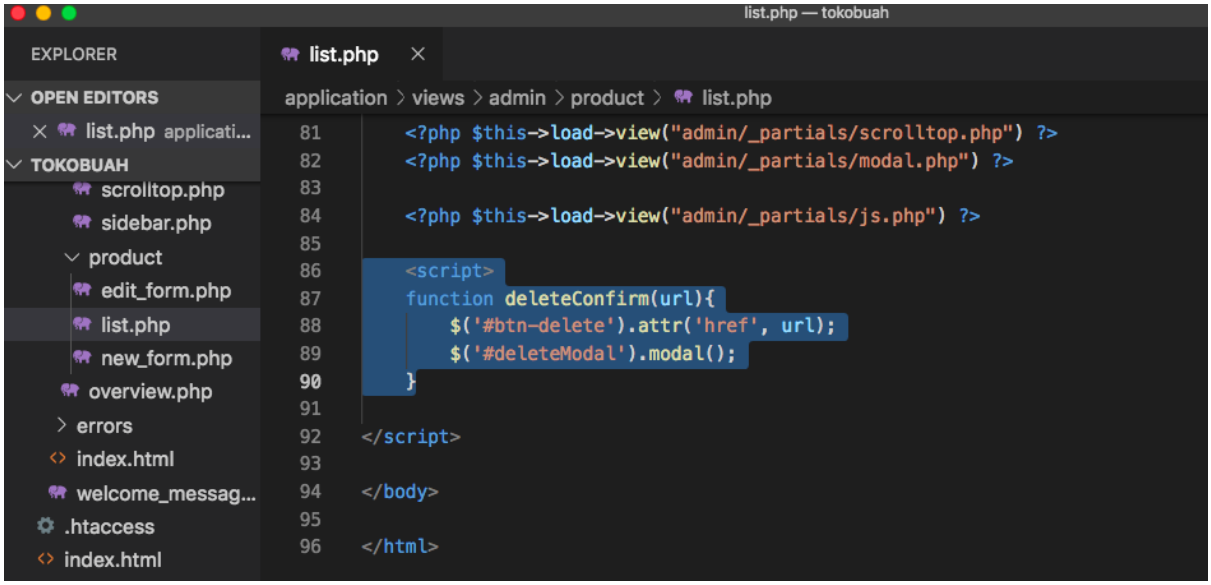
1. *Embed* langsung di dalam view;
2. Membuat file Javascript terpisah.

Kita akan menggunakan cara yang pertama, karena fungsi yang kita butuhkan hanya akan digunakan pada view product/list.php saja.

Jika nanti ada fungsi Javascript yang digunakan di beberapa view, maka sebaiknya membuat file terpisah.

Baiklah, silahkan buka file views/admin/product/list.php, kemudian tambahkan kode berikut di bagian bawah, sebelum tutup <body>.

```
<script>
function deleteConfirm(url){
        $('#btn-delete').attr('href', url);
        $('#deleteModal').modal();
}
</script>
```

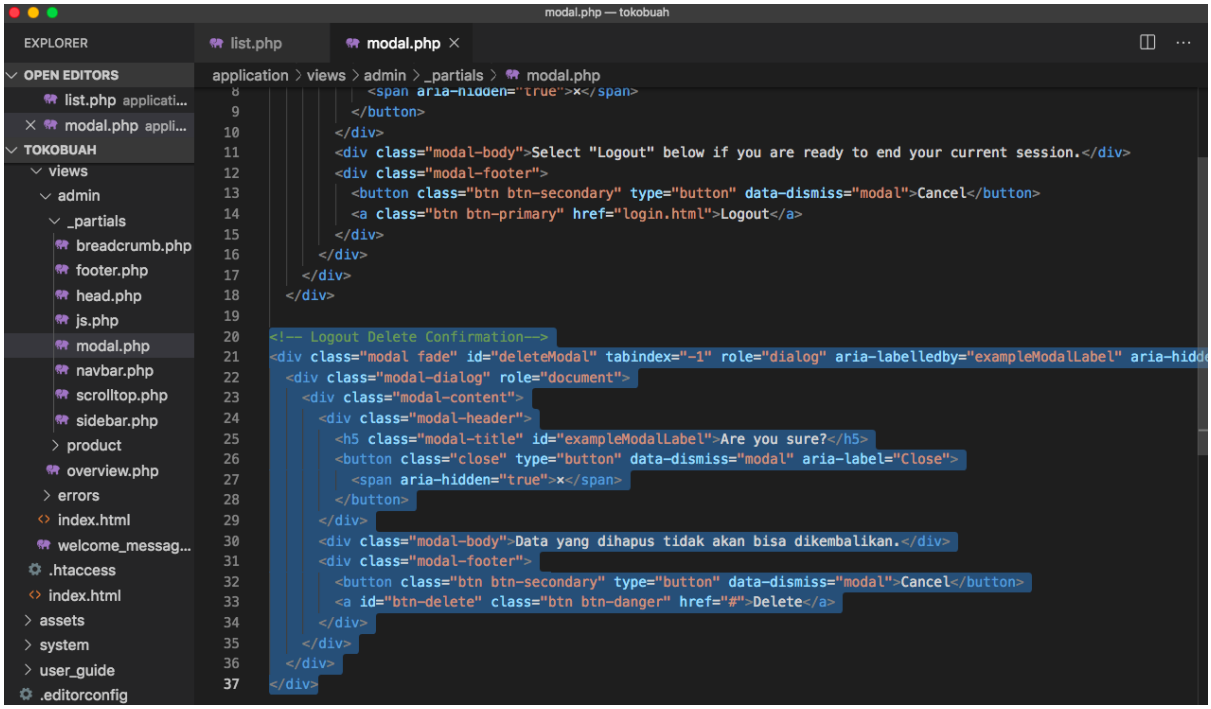


Setelah itu, tambahkan sebuah modal untuk delete di dalam file views/admin/_partials/modal.php.

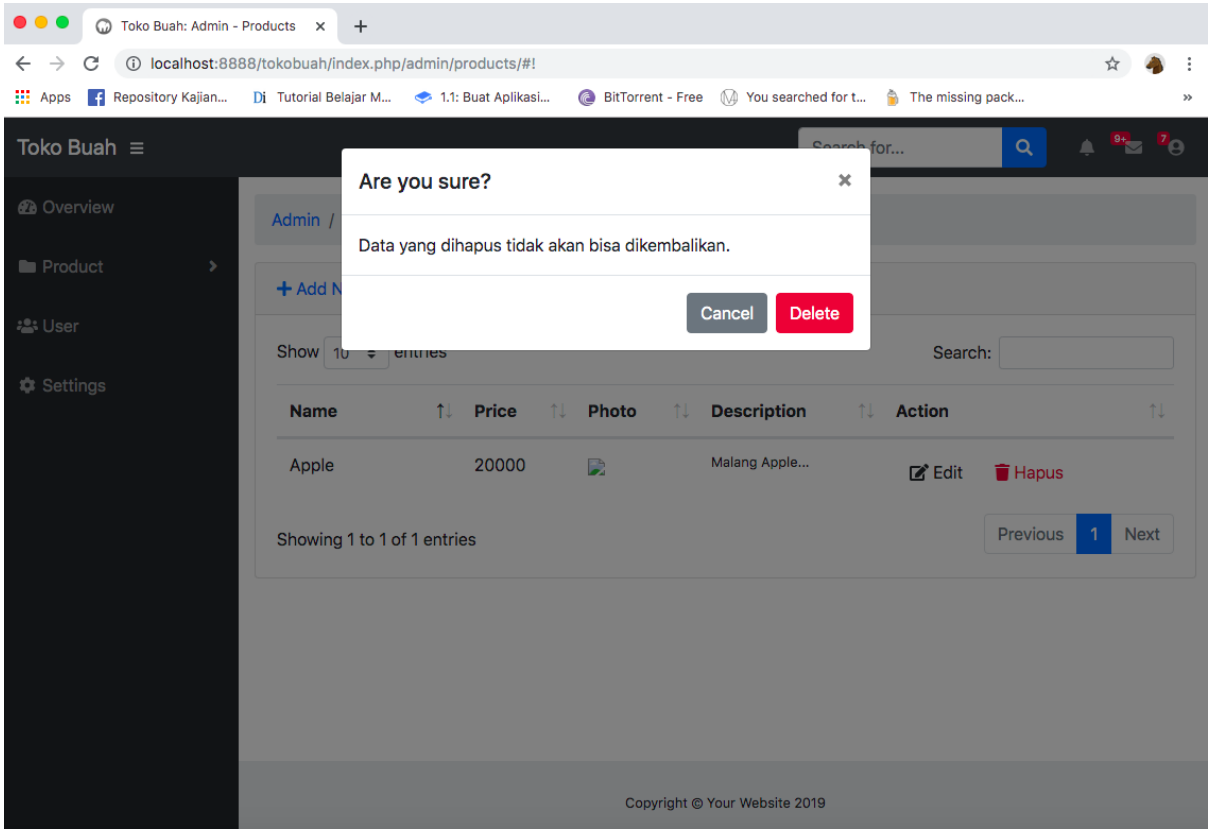
```
<!-- Logout Delete Confirmation-->
<div class="modal fade" id="deleteModal" tabindex="-1" role="dialog"
aria-labelledby="exampleModalLabel" aria-hidden="true">
```

```
  <div class="modal-dialog" role="document">
    <div class="modal-content">
      <div class="modal-header">
        <h5 class="modal-title" id="exampleModalLabel">Are you sure?</h5>
        <button class="close" type="button" data-dismiss="modal" aria-
label="Close">
          <span aria-hidden="true">×</span>
        </button>
      </div>
      <div class="modal-body">Data yang dihapus tidak akan bisa
dikembalikan.</div>
      <div class="modal-footer">
        <button class="btn btn-secondary" type="button" data-
dismiss="modal">Cancel</button>
        <a id="btn-delete" class="btn btn-danger" href="#">Delete</a>
      </div>
    </div>
  </div>
</div>
```

Sehingga akan menjadi seperti ini:



Oke, setelah itu coba hapus sebuah data.



Selanjutnya?

Kita memang sudah selesai membuat CRUD *(Create, Read, Update, Delete)* untuk products dan sudah dapat ditampilkan dengan baik menggunakan datatables.

Tapi masih ada fitur yang belu kita buat, seperti upload file untuk gambar products.