

PHP Admin Template

Jagowebdev.com

I. Pengantar

PHP Admin Template Jagowebdev dikembangkan untuk memudahkan dalam pengembangan aplikasi. Anda tidak perlu membuat dari awal manajemen usernya, seperti bagaimana user login, hak akses pada user tersebut, dll sehingga Anda dapat fokus dalam pengembangan aplikasi.

Berikut ini beberapa hal penting yang perlu diketahui:

II. Install Aplikasi

Untuk install aplikasi ikuti langkah berikut:

- a. Copy file php ke folder htdocs, misal jika di copy ke folder htdocs/admin_template maka struktur foldernya akan tampak seperti berikut:

DATA (D:) > xampp > htdocs > admin_template >

Name	Type
app	File folder
public	File folder
system	File folder
writable	File folder
.env	ENV File
.htaccess	HTACCESS File
admin-template-codeigniter-4.sql	SQL File
index.php	PHP File
license.txt	Text Document
README.md	MD File
spark	File

- b. Buat database dengan nama yang dikehendaki, misal penjualan, selanjutnya load file admin-template-codeigniter-4.sql yang disertakan pada file download ke database tersebut.
- c. Edit file app/Config/App.php edit bagian \$baseUrl sesuai dengan url dimana aplikasi diinstall dan edit app/Config/Database.php sesuai dengan konfigurasi database Anda. Untuk variabel lainnya bersifat opsional. Misal jika kita ekstrak file ke folder htdocs/admin_template seperti contoh diatas, maka \$baseUrl nya adalah http://localhost/admin_template/

d. System requirements

Codeigniter 4 mensyaratkan versi PHP minimal adalah versi 7.2 sehingga untuk menjalankan Admin Template ini juga diperlukan PHP versi > 7.2. Direkomendasikan menggunakan PHP versi > 7.4 yang bisa diinstal menggunakan xampp portable versi 7.4.10 yang bisa didownload di <https://sourceforge.net/projects/xampp/files/XAMPP%20Windows/>

Catatan: Agar lebih aman, sebaiknya menggunakan database MariaDB bukan MySQL, contoh bundle yang menggunakan database MariaDB adalah XAMPP

Berikut ini beberapa hal penting yang perlu diketahui:

III. Setup Awal Aplikasi

1. Membuat Menu

a. Membuat menu

Untuk membuat menu, klik menu website > Menu

b. Menampilkan menu

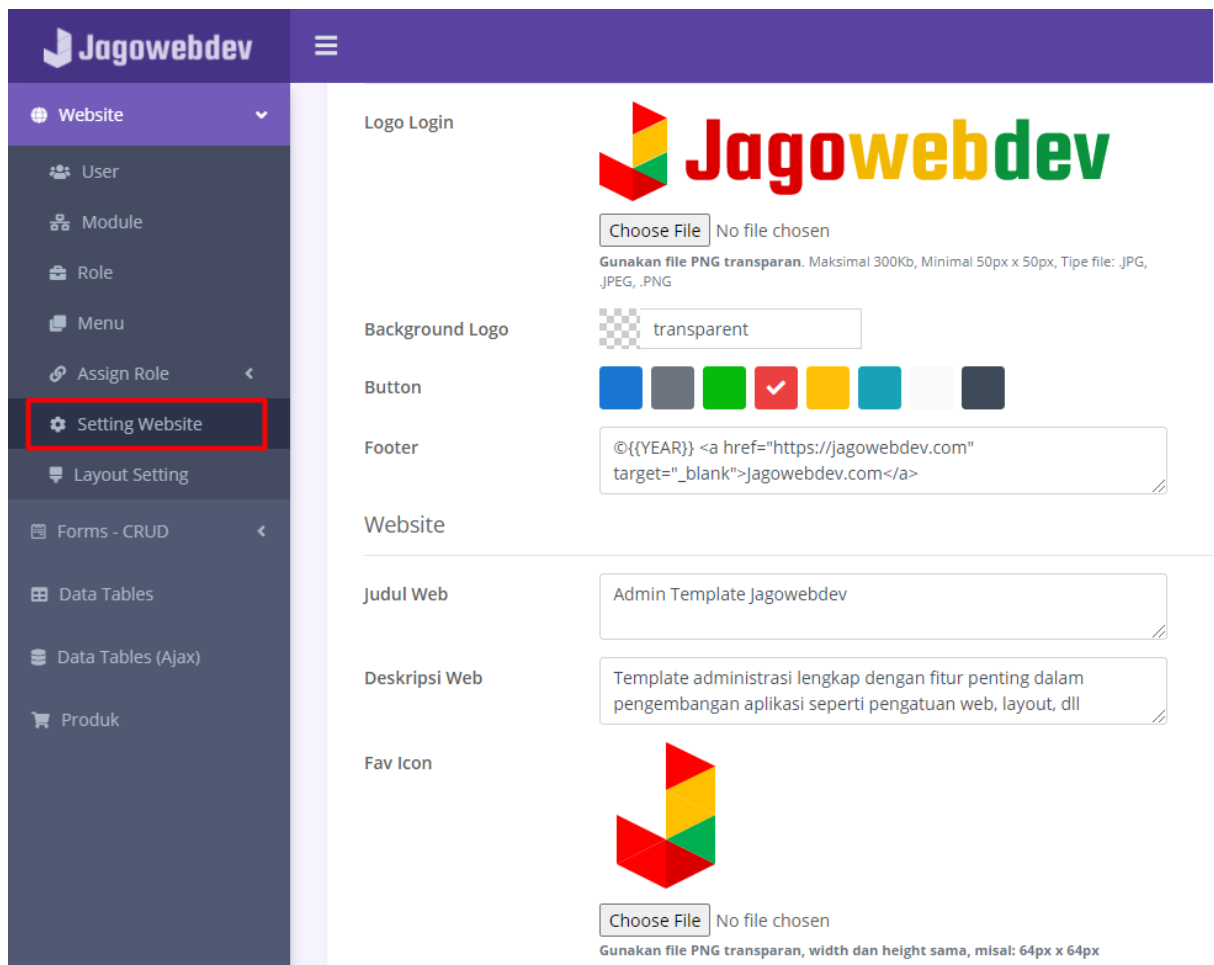
Menu tidak serta merta tampil, agar dapat tampil, menu perlu di assign terlebih dahulu ke role, untuk assign ke role, klik menu Website > Assign Role > Menu Role

c. Menampilkan Highlight Menu ketika halaman dibuka

Agar menu terhighlight ketika membuka halaman tertentu yang artinya module tertentu, kita perlu assign menu tersebut ke module, klik menu Website > Menu kemudian pilih menu yang ingin diedit kemudian pada bagian module, pilih module yang ingin diassign ke menu tersebut

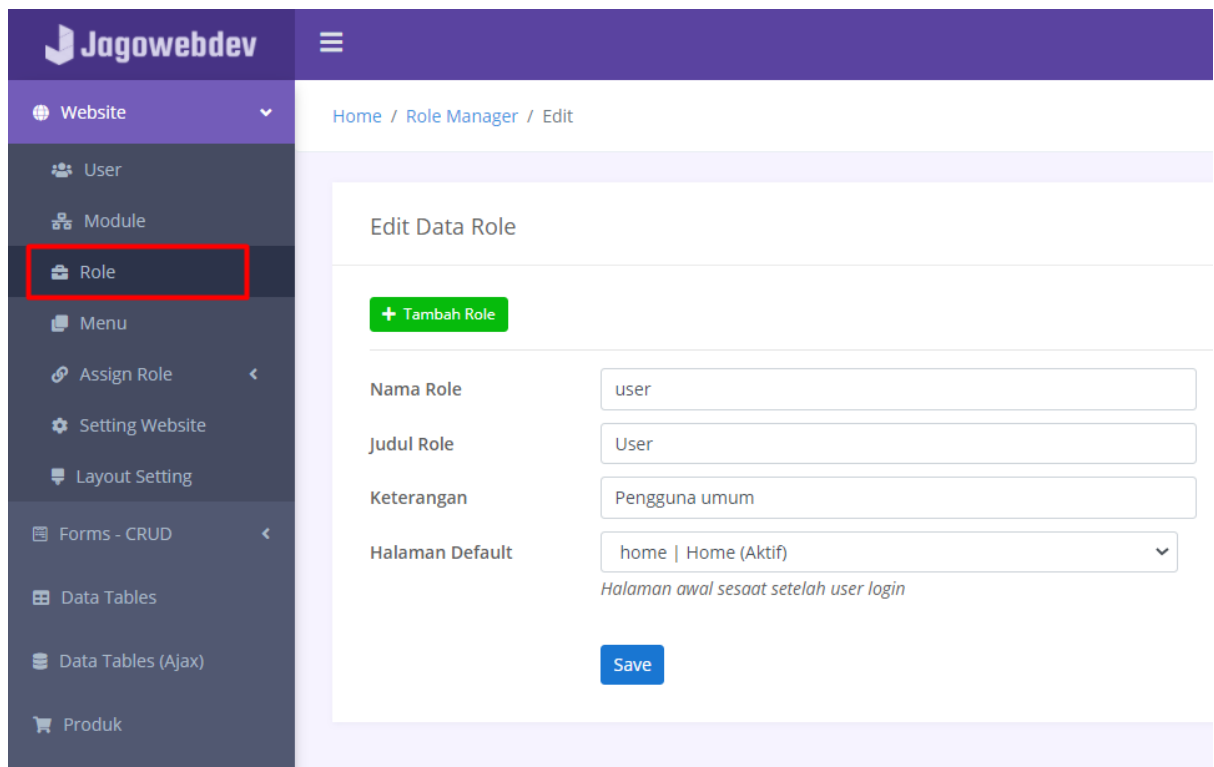
2. Mengubah logo

Pengaturan logo, baik logo pada favicon, halaman login, maupun halaman aplikasi dapat dilakukan melalui menu Setting Website, contoh sebagai berikut:



3. Membuat halaman default

Ketika user login, dapat langsung diarahkan ke halaman default sesuai dengan role user tersebut. Untuk membuat halaman default tersebut, masuk ke menu role, edit role yang ada kemudian pada halaman default pilih halaman default untuk role tersebut.



4. Mode Pengembangan

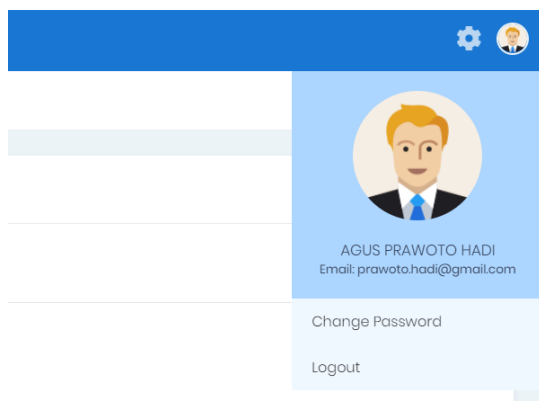
Secara default jika terjadi error, maka aplikasi akan memunculkan pesan error tersebut, Anda dapat menonaktifkan pesan error tersebut dan hanya menampilkan pesan error secara umum dengan cara mengedit file .htaccess kemudian ganti SetEnv CI_ENVIRONMENT development menjadi SetEnv CI_ENVIRONMENT production.

5. Login

Login pertama kali adalah username: admin, password: admin

6. Ubah Password

Untuk menjaga kerahasiaan dan keamanan password, ubah password hanya bisa dilakukan oleh user pemilik password sendiri. Caranya yaitu login ke akun, kemudian masuk ke menu Change Password yang ada di menu akun pojok kanan atas



IV. Membuat Module

Pada admin template ini, module tidak bisa dibuat secara otomatis, melainkan dibuat secara manual sesuai dengan lingkungan pengembangan CodeIgniter 4

A. Module Tanpa Database

Seperti ketika mengembangkan aplikasi menggunakan codeigniter, untuk membuat module/halaman, kita perlu membuat controller, misal kita akan membuat module produk. Pertama tama kita buat file controller dengan nama Produk.php di folder app/Controllers sebagai berikut:

DATA (D:) > xampp > htdocs > admin_template > app > Controllers		
Name	Type	Size
Builtin	File folder	
Base.php	PHP File	4 KB
BaseController.php	PHP File	8 KB
Data_tables.php	PHP File	5 KB
Data_tables_ajax.php	PHP File	5 KB
Home.php	PHP File	1 KB
Image_upload.php	PHP File	5 KB
Input_dinamis.php	PHP File	5 KB
Login.php	PHP File	3 KB
Multiple_fileupload.php	PHP File	7 KB
Options_dinamis.php	PHP File	6 KB
Produk.php	PHP File	1 KB
Welcome.php	PHP File	1 KB

Misal file Produk.php tersebut kita isi script sebagai berikut:

```
<?php
namespace App\Controllers;

class Produk extends BaseController
{
    public function index()
    {
        echo 'Tes Module Produk';
    }
}
```

Selanjutnya, agar module dapat diakses, kita perlu mendaftarkan module ke sistem, caranya masuk kemenu module kemudian tambahkan module produk sebagai berikut:

Jagowebdev

Website

User

Module

Role

Menu

Assign Role

Setting Website

Layout Setting

Forms - CRUD

Data Tables

Home / Module Manager / Add

Tambah Module

+ Tambah Module Daftar Module

Nama Module produk Sesuai nama yang ada di URL

Judul Module Produk

Deskripsi Produk

Status Aktif

Save

Selanjutnya kita tentukan, siapa yang boleh mengakses module tersebut, caranya masuk ke menu Assign Role, sub menu Module Role seperti tampak pada gambar berikut:

Jagowebdev

Website

User

Module

Role

Menu

Assign Role

User Role

Module Role

Menu Role

Setting Website

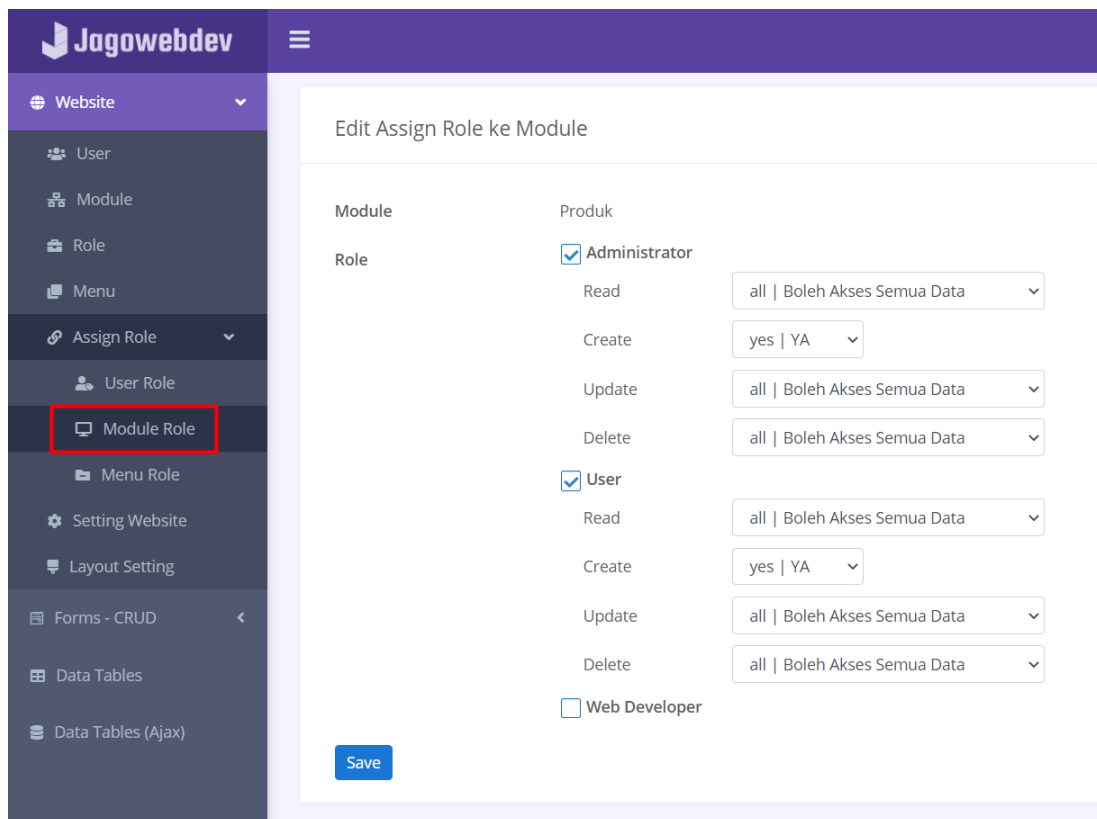
Layout Setting

Forms - CRUD

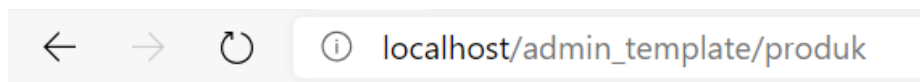
Data Tables

8	menu-role	Menu - Role	Administrator	Edit	Detail
9	image-upload	Image Upload	Administrator User	Edit	Detail
15	setting	Web Setting	Administrator User	Edit	Detail
16	setting-web	Setting Web	Administrator	Edit	Detail
21	options-dinamis	Options Dinamis	Administrator User	Edit	Detail
22	input-dinamis	Input Dinamis	Administrator User	Edit	Detail
25	home	Home	Administrator User Web Developer	Edit	Detail
26	multiple-fileupload	Multiple File Upload	Administrator User	Edit	Detail
27	datatables	Data Tables	Administrator User Web Developer	Edit	Detail
28	datatables-ajax	Data Tables Ajax	Administrator User Web Developer	Edit	Detail
29	produk	Produk		Edit	Detail

Selanjutnya pilih tombol edit pada module produk. Pada halaman Edit Assign Role ke Module, pilih role yang ingin diberi akses sebagai berikut:



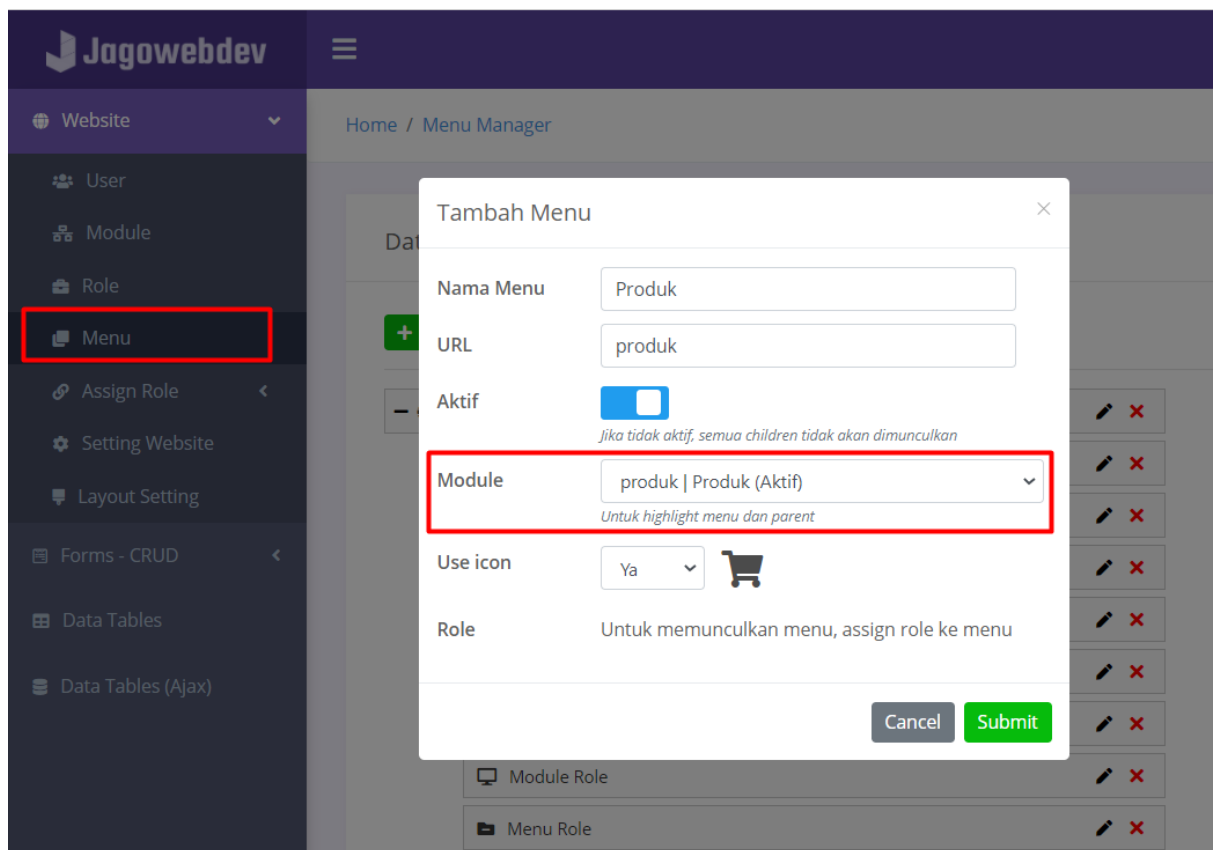
Setelah telah berhasil, maka module tersebut sudah bisa diakses, pada contoh ini, karena aplikasi saya letakkan di folder htdocs/admin_template, maka alamat module produk adalah http://localhost/admin_template/produk, contoh sebagai berikut:



Tes Module Produk

B. Membuat Menu Untuk Module

Agar module mudah diakses, maka alamat module tersebut perlu kita tambahkan pada menu, caranya, masuk ke menu "Menu" kemudian tambahkan klik Tambah Menu, pada isian menu, isikan detail menu, misal seperti contoh berikut:

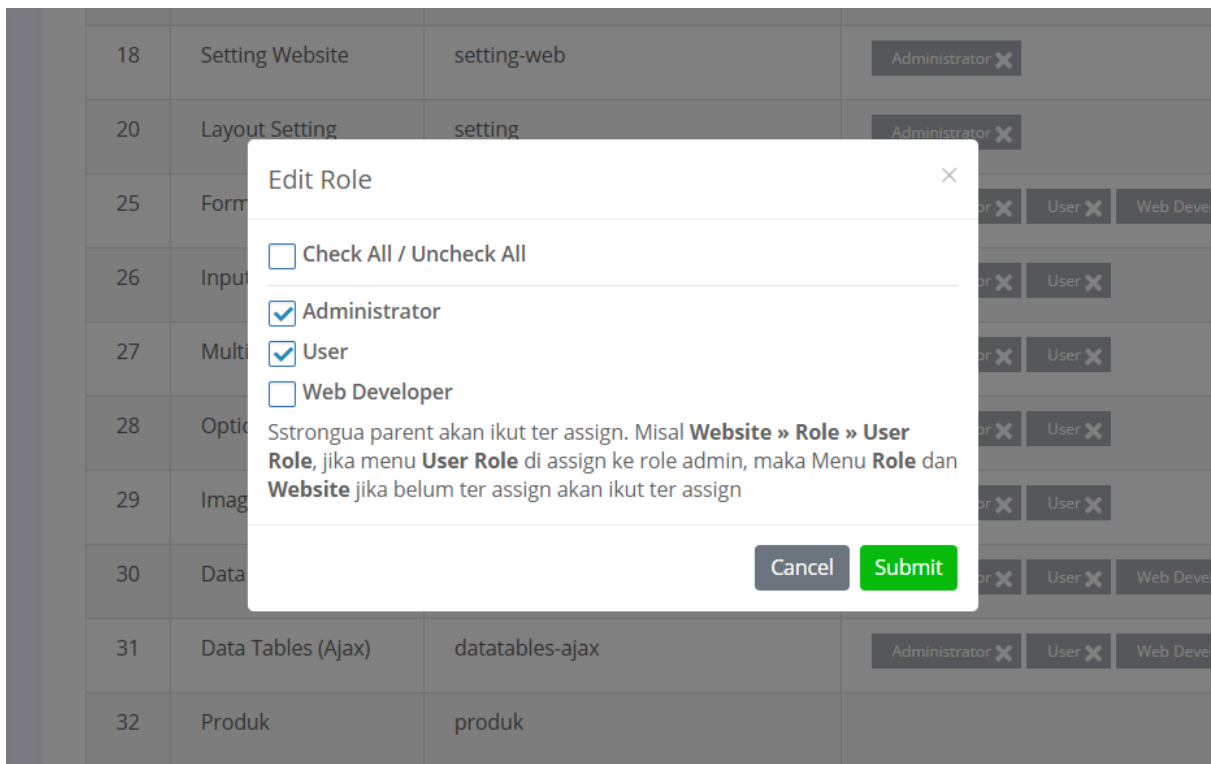


Penting diperhatikan bahwa pada bagian Module kita pilih module produk yang telah kita buat sebelumnya, hal ini bertujuan agar ketika module/halaman produk dibuka, menu Produk dapat ter highlight.

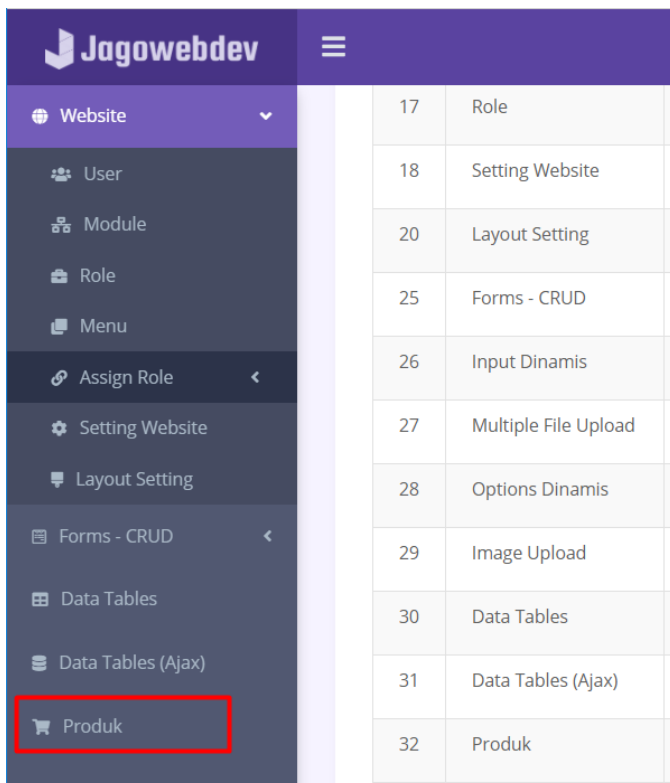
Menu yang telah kita buat tidak serta merta muncul di daftar menu, untuk memunculkannya, kita perlu meng assign menu tersebut ke role yang ada, untuk melakukannya, masuk ke menu Assign Role > Menu Role, kemudian pada menu produk, klik menu Edit sebagai berikut:

Website	18	Setting Website	setting-web	Administrator	Edit
User	20	Layout Setting	setting	Administrator	Edit
Module	25	Forms - CRUD	input-dinamis	Administrator User Web Developer	Edit
Role	26	Input Dinamis	input-dinamis?action=add	Administrator User	Edit
Menu	27	Multiple File Upload	multiple-fileupload?action=edit&id=1	Administrator User	Edit
Assign Role	28	Options Dinamis	options-dinamis?action=edit&id=2	Administrator User	Edit
User Role	29	Image Upload	image-upload?action=edit&id=1	Administrator User	Edit
Module Role	30	Data Tables	datatables	Administrator User Web Developer	Edit
Menu Role	31	Data Tables (Ajax)	datatables-ajax	Administrator User Web Developer	Edit
Setting Website	32	Produk	produk		Edit
Layout Setting					
Forms - CRUD					
Data Tables					

Selanjutnya pilih role yang ingin di assign dan simpan, misal seperti gambar berikut:



Setelah berhasil maka menu akan muncul disebelah kiri (jangan lupa refresh halaman)



C. Module Dengan Database

Pada codeigniter, untuk berinteraksi dengan database, kita perlu membuat model. Pada contoh kali ini kita buat model dengan nama ProdukModel.php dan kita letakkan di dalam folder app/Models sebagai berikut:

DATA (D:) > xampp > htdocs > admin_template > app > Models >		
Name	Type	Size
Builtin	File folder	
.gitkeep	GITKEEP File	0 KB
BaseModel.php	PHP File	6 KB
DataTablesAjaxModel.php	PHP File	5 KB
DataTablesModel.php	PHP File	4 KB
ImageUploadModel.php	PHP File	4 KB
InputDinamisModel.php	PHP File	2 KB
MultipleFileuploadModel.php	PHP File	7 KB
OptionsDinamisModel.php	PHP File	6 KB
ProdukModel.php	PHP File	1 KB

Sebagai contoh, file ProdukModel.php kita isi script sebagai berikut:

```
<?php
namespace App\Models;

class ProdukModel extends \App\Models\BaseModel
{
    public function __construct() {
        parent::__construct();
    }

    public function getDataProduk() {
        $sql = 'SELECT * FROM produk';
        $result = $this->db->query($sql)->getResultArray();
        return $result;
    }
}
```

Selanjutnya kita ubah file controller Produk.php menjadi berikut:

```
<?php
namespace App\Controllers;
use App\Models\ProdukModel;

class Produk extends BaseController
{
    private $produkModel;

    public function __construct() {
        parent::__construct();
        $this->produkModel = new ProdukModel;
    }

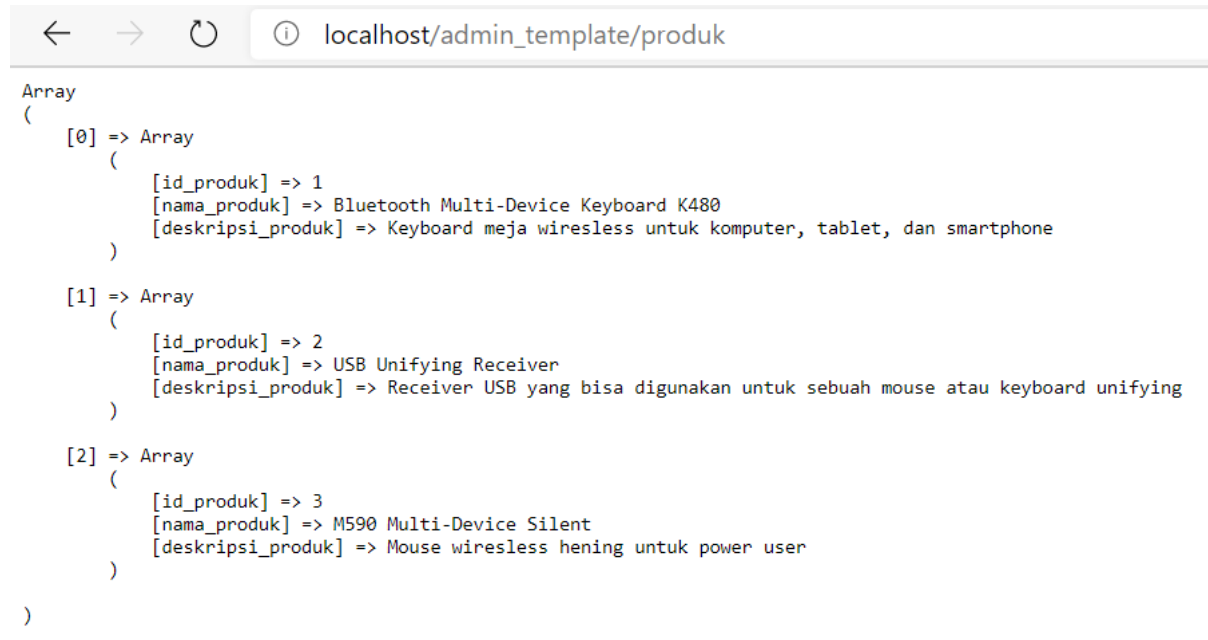
    public function index()
    {
    }
```

```

        $produk = $this->produkModel->getDataProduk();
        echo '<pre>'; print_r($produk);
    }
}

```

Jika kita buka module produk, hasil yang kita peroleh adalah sebagai berikut:




















D. Module Dengan View

Output module yang kita bahas sebelumnya merupakan output apa adanya. Anda dapat menggunakan header dan footer bawaan aplikasi sehingga tampilan module lebih menarik, untuk menggunakannya, Anda cukup memanggil method `$this->view()`. Method ini merupakan method bawaan aplikasi yang didesain agar dapat otomatis me-load file header.php dan footer.php yang ada di folder `app/Views/themes/modern/header`. Source code method ini dapat dilihat di file `app/Controllers/BaseController.php`

Selanjutnya mari kita membuat file view yang kita beri nama `produk-result.php`, letakkan file tersebut di folder `app/Views/themes/modern` sebagai berikut:

DATA (D:) > xampp > htdocs > admin_template > app > Views > themes > modern

Name	Type	Size
 builtin	File folder	
 data-tables-ajax-result.php	PHP File	2 KB
 data-tables-result.php	PHP File	2 KB
 error.php	PHP File	1 KB
 error-data-notfound.php	PHP File	1 KB
 footer.php	PHP File	1 KB
 header.php	PHP File	8 KB
 image-upload-form.php	PHP File	5 KB
 image-upload-result.php	PHP File	2 KB
 multiple-fileupload-form.php	PHP File	8 KB
 multiple-fileupload-result.php	PHP File	3 KB
 options-dinamis-form.php	PHP File	5 KB
 options-dinamis-result.php	PHP File	3 KB
 penghadap-form.php	PHP File	3 KB
 penghadap-result.php	PHP File	2 KB
 produk-result.php	PHP File	1 KB
 welcome.php	PHP File	1 KB

Selanjutnya kita isi file produk-result.php dengan script sebagai berikut:

```
<table class="table table-striped table-bordered table-hover">
  <thead>
    <tr>
      <th>No</th>
      <th>Nama Produk</th>
      <th>Deskripsi Produk</th>
    </tr>
  </thead>
  <tbody>
    <?php
    $no = 1;
    foreach ($result as $val) {
      echo '<tr>
        <td>' . $no . '</td>
        <td>' . $val['nama_produk'] . '</td>
        <td>' . $val['deskripsi_produk'] . '</td>
      </tr>';
      $no++;
    }
    ?>
  </tbody>
</table>
```

Sedangkan script pada controller Produk.php kita ubah menjadi berikut:

```
<?php
namespace App\Controllers;
```

```

use App\Models\ProdukModel;

class Produk extends BaseController
{
    private $produkModel;

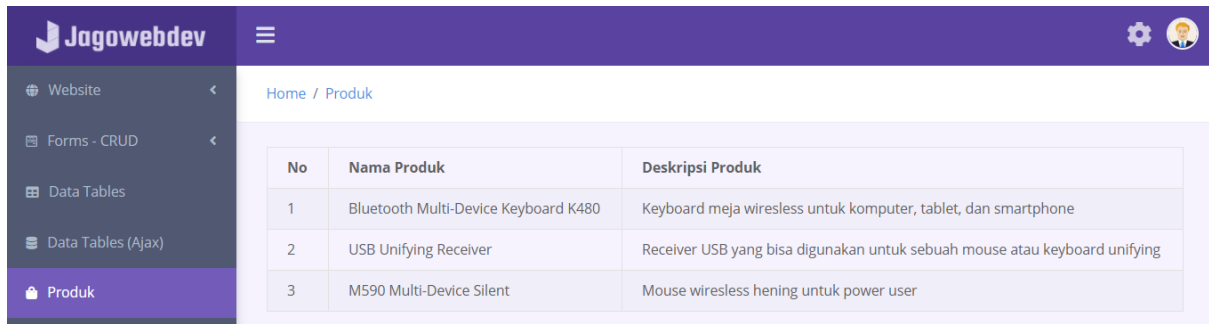
    public function __construct() {
        parent::__construct();
        $this->produkModel = new ProdukModel;
    }

    public function index()
    {
        $data = $this->data;
        $data['result'] = $this->produkModel->getDataProduk();
        $this->view('produk-result.php', $data);
    }
}

```

Perhatikan bahwa pada file controller Produk.php kita definisikan variabel \$data dengan properti \$this->data. Properti \$this->data ini berisi berbagaimacam data yang dapat kita gunakan pada file view, diantaranya \$config dan \$session, data data ini dapat dilihat pada script app/Controllers/BaseController.php selanjutnya variabel \$data kita isi data result hasil dari query tabel produk dari file model

Jika kita jalankan module produk, hasil yang kita peroleh adalah sebagai berikut:



No	Nama Produk	Deskripsi Produk
1	Bluetooth Multi-Device Keyboard K480	Keyboard meja wireless untuk komputer, tablet, dan smartphone
2	USB Unifying Receiver	Receiver USB yang bisa digunakan untuk sebuah mouse atau keyboard unifying
3	M590 Multi-Device Silent	Mouse wireless hening untuk power user

Perhatikan bahwa breadcrumb yang muncul adalah Home / Produk, menyesuaikan dengan nama module yang didaftarkan

Agar tampilan lebih menarik, kita tambahkan beberapa elemen dan style pada output diatas sebagai berikut:

```

<div class="card">
    <div class="card-header">
        <h5 class="card-title"><?=$current_module['judul_module']?></h5>
    </div>

    <div class="card-body">
        <table class="table table-striped table-bordered table-hover">
            <thead>
                <tr>
                    <th>No</th>
                    <th>Nama Produk</th>
                    <th>Deskripsi Produk</th>
                </tr>
            </thead>

```

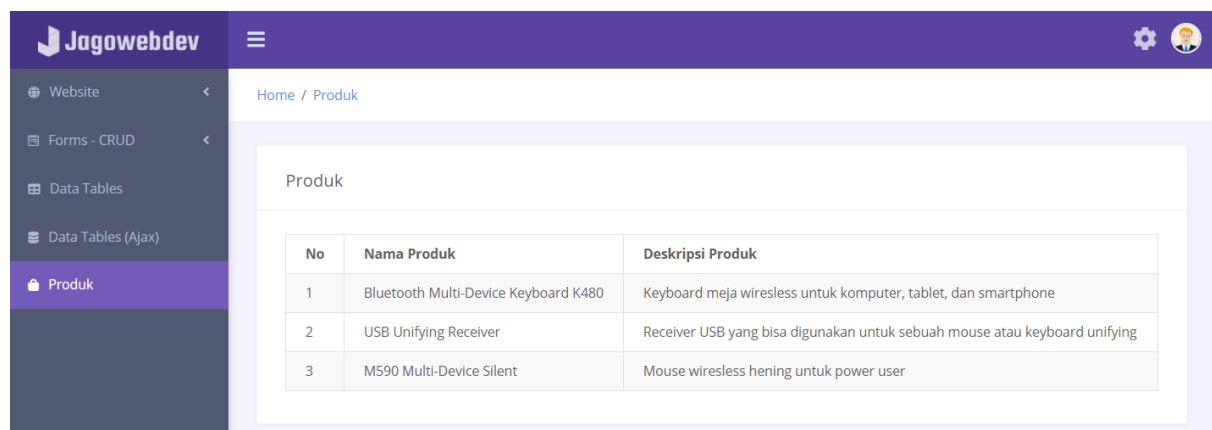
```

<tbody>
<?php
$no = 1;
foreach ($result as $val) {
    echo '<tr>
        <td>' . $no . '</td>
        <td>' . $val['nama_produk'] . '</td>
        <td>' . $val['deskripsi_produk'] . '</td>
    </tr>';
    $no++;
}
?>
</tbody>
</table>
</div>
</div>

```

Class CSS card, card-header, card-body merupakan style bawaan aplikasi admin template, anda dapat memodifikasi style ini pada file public/themes/modern/buiulin/css/site.css

Output yang dihasilkan adalah sebagai berikut:



E. Handling Error Data Tidak Ditemukan

Jika data tidak ditemukan, Anda dapat dengan mudah memunculkan pesan error bahwa data tidak ditemukan, yaitu cukup dengan memanggil method `$this->errorDataNotFound()` method ini ada di file `app/Controllers/BaseController.php`

Sebagai contoh script controller `Produk.php` kita ubah sebagai berikut:

```

<?php
namespace App\Controllers;
use App\Models\ProdukModel;

class Produk extends BaseController
{
    private $produkModel;

    public function __construct() {
        parent::__construct();
        $this->produkModel = new ProdukModel;
    }

    public function index()
    {

```

```

        $data = $this->data;
        $data['result'] = $this->produkModel->getDataProduk();

        if (!$data['result'])
            $this->errorDataNotFound();

        $this->view('produk-result.php', $data);
    }
}

```

Maka, jika data produk tidak ditemukan, akan muncul pesan error sebagai berikut:

