

MEMBUAT CRUD SEDERHANA DENGAN FRAMEWORK LARAVEL 5.4

Pendukung yang harus di siapkan sbb :

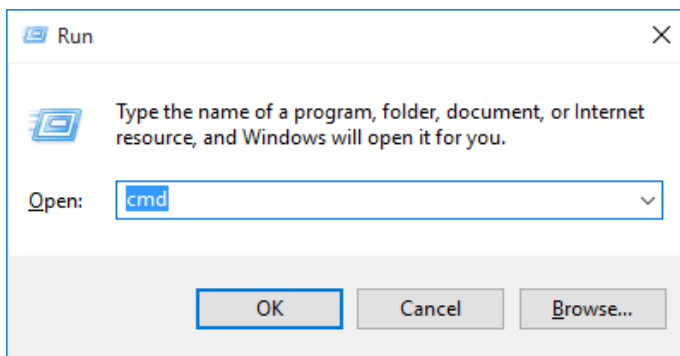
- Text editor seperti sublimeText, Notepad++ dll.
- Download server xampp untuk windows: <https://www.apachefriends.org/index.html>
- Untuk windows: <https://getcomposer.org/doc/00-intro.md#installation-windows>

Server Requirements untuk laravel 5.4

- PHP >= 5.6.4
- OpenSSL PHP Extension
- PDO PHP Extension
- Mbstring PHP Extension
- Tokenizer PHP Extension
- XML PHP Extension

Langkah – langkah membuat CRUD (Create Read Update Delete) di Laravel 5.4

Membuka Commandprompt di window dengan mengklik simbol windows + R masuk CMD



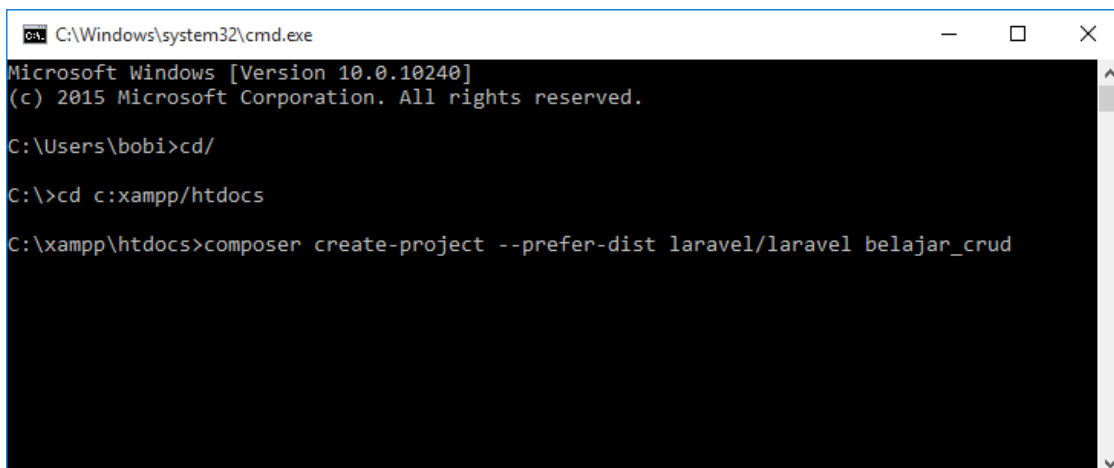
Masuk ke CMD – masuk ke direktory xampp/htdocs (untuk lokasi sebenarnya bisa diletakan dimanapun)

pastikan composer sudah di instal

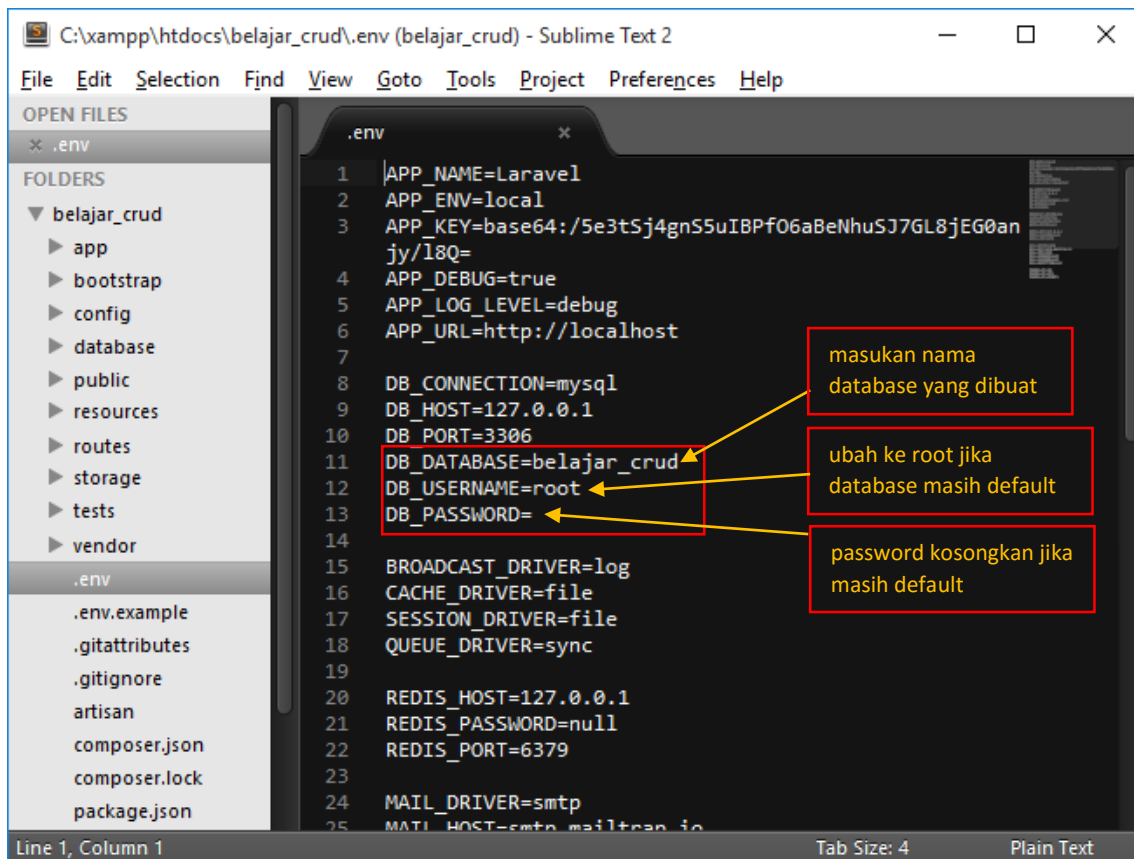
setelah sudah masuk ke direktory yang ditentukan , masukan kode berikut

Step 1: Instalasi Project Laravel

```
composer create-project --prefer-dist laravel/laravel belajar_crud
```



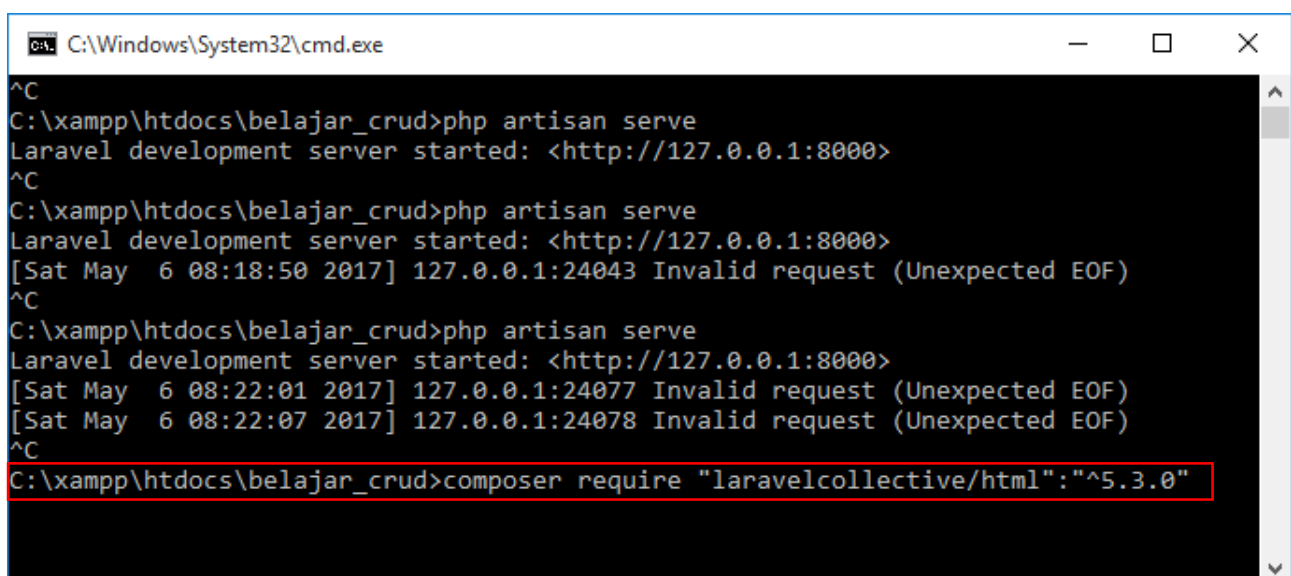
setelah project selesai dibuat , buka file .env untuk merubah koneksi ke database.



setelah membuat file project laravel 5.4 , kita harus menginstal laravelcollective/html untuk Form class, berikut ini :

untuk cara installnya bisa langsung dari CMD tersebut

```
composer require "laravelcollective/html":"^5.3.0"
```



selanjutnya buka folder belajar_curd/config/app.php:

masukan kode tersebut di 'providers' =>

```
Collective\Html\HtmlServiceProvider::class,
```

selanjutnya masih belajar_crud/config/app.php:

masukan kode tersebut di 'aliases' =>

```
'aliases' => [

    // ...

    'Form' => Collective\Html\FormFacade::class,

    'Html' => Collective\Html\HtmlFacade::class,

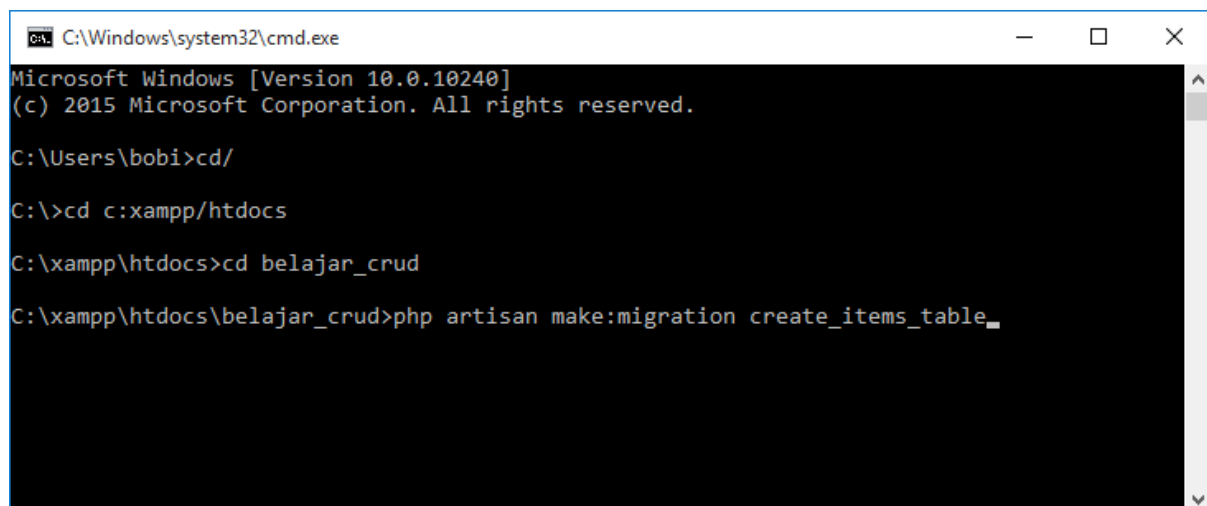
    // ...

],
```

Step 2: Membuat Tabel Item dan Model

Pada langkah ini kita akan membuat tabel migrasi untuk Tabel Item menggunakan Laravel 5.4

```
php artisan make:migration create_items_table
```



```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 10.0.10240]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\Users\bobi>cd/

C:\>cd c:xampp\htdocs

C:\xampp\htdocs>cd belajar_crud

C:\xampp\htdocs\belajar_crud>php artisan make:migration create_items_table_
```

setelah mengetikan commad ini masuk ke folder
xampp\htdocs/belajar_crud/database/migration/ buka tabel dan isi dengan kode berikut

```
use Illuminate\Database\Schema\Blueprint;

use Illuminate\Database\Migrations\Migration;
```

```

class CreateItemsTable extends Migration
{

    public function up()
    {
        Schema::create('items', function (Blueprint $table) {

            $table->increments('id');

            $table->string('title');

            $table->text('description');

            $table->timestamps();

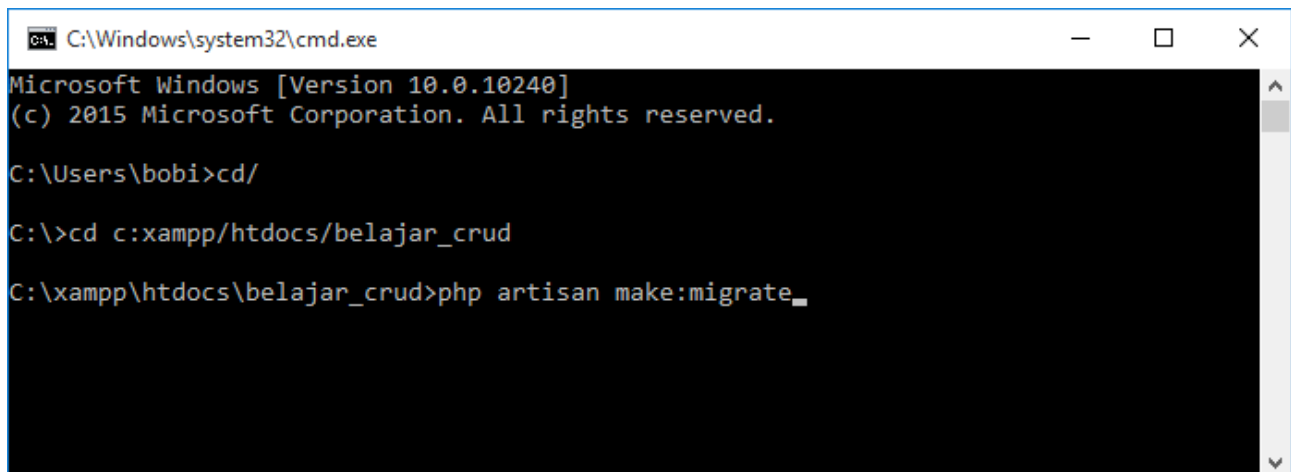
        });
    }

    public function down()
    {
        Schema::drop("items");
    }
}

```

setelah selesai ketikan di CMD

`php artisan make:migrate`



The screenshot shows a Windows Command Prompt window titled "C:\Windows\system32\cmd.exe". The text inside the window is as follows:

```

Microsoft Windows [Version 10.0.10240]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\Users\bobi>cd/

C:\>cd c:xampp\htdocs\belajar_crud

C:\xampp\htdocs\belajar_crud>php artisan make:migrate_

```

perintah tersebut berfungsi untuk migrasi tabel Item ke database “belajar_crud”

setelah membuat tabel "Items" anda harus membuat Items model for table items. pertama membuat path **app/Item.php** dan masukan kode berikut di Item.php :

app/Item.php

```
namespace App;

use Illuminate\Database\Eloquent\Model;

class Item extends Model
{

    public $fillable = ['title','description'];

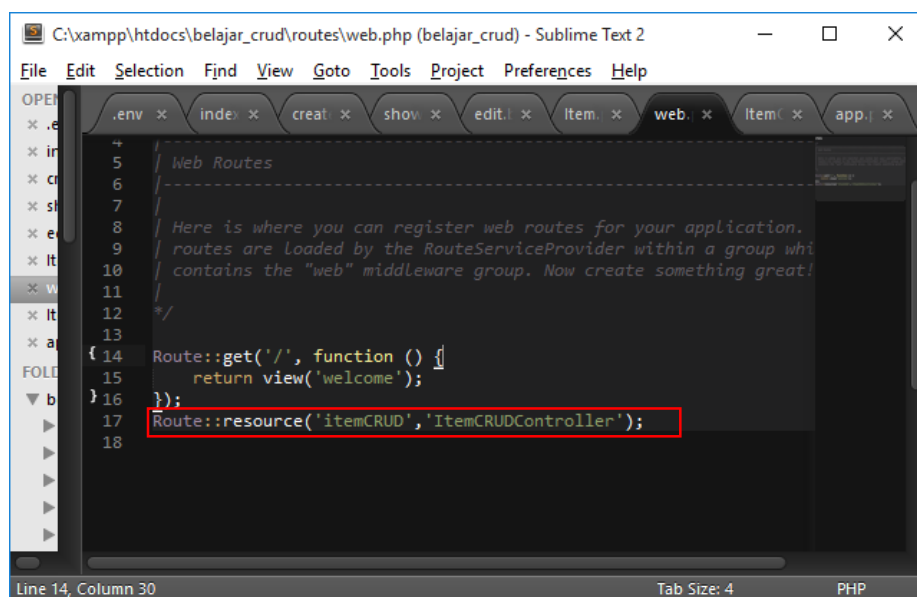
}
```

Step 3 : Tambah Route and Controller

Kita harus menambahkan route untuk items CRUD, Pada contoh ini kita akan menambahkan sumber untuk aplikasi routenya , masukan kode berikut di

belajar_crud\routes\web.php

```
Route::resource('itemCRUD','ItemCRUDController');
```



Jadi , kita sekarang harus membuat controller yaitu **ItemCRUDController.php**:

setelaah dibuat masukan kode berikut :

app/Http/Controllers/ItemCRUDController.php

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Http\Controllers\Controller;
use App\Item;

class ItemCRUDController extends Controller
{

    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index(Request $request)
    {
        $items = Item::orderBy('id','DESC')->paginate(5);

        return view('ItemCRUD.index',compact('items'))

            ->with('i', ($request->input('page', 1) - 1) * 5);
    }

    /**
     * Show the form for creating a new resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function create()
    {
        return view('ItemCRUD.create');
```

```

}

/**
 * Store a newly created resource in storage.
 *
 * @param \Illuminate\Http\Request $request
 * @return \Illuminate\Http\Response
 */
public function store(Request $request)
{
    $this->validate($request, [
        'title' => 'required',
        'description' => 'required',
    ]);

    Item::create($request->all());

    return redirect()->route('itemCRUD.index')
        ->with('success','Item created successfully');
}

/**
 * Display the specified resource.
 *
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function show($id)
{
    $item = Item::find($id);

    return view('ItemCRUD.show',compact('item'));
}

```

```

}

/**
 * Show the form for editing the specified resource.
 *
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function edit($id)
{
    $item = Item::find($id);

    return view('ItemCRUD.edit',compact('item'));
}

```

```

/**
 * Update the specified resource in storage.
 *
 * @param \Illuminate\Http\Request $request
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function update(Request $request, $id)
{
    $this->validate($request, [

        'title' => 'required',

        'description' => 'required',

    ]);

    Item::find($id)->update($request->all());

    return redirect()->route('itemCRUD.index')

```



```

        ->with('success','Item updated successfully');

    }

    /**
     * Remove the specified resource from storage.
     *
     * @param int $id
     * @return \Illuminate\Http\Response
     */
    public function destroy($id)
    {
        Item::find($id)->delete();

        return redirect()->route('itemCRUD.index')

        ->with('success','Item deleted successfully');
    }
}

```

Step 4: Membuat file blade.

Pada langkah ini kita akan membuat file blade , dimana semua file listing yang ada di laravel harus menggunakan “.blade.php” kita akan membuat listing file create, edit and show item dan juga membuat default blade file untuk pengaturan tema. Jadi kita harus membuat folder layouts di **resources/views/layouts** untuk membuat default.blade.php.

resources/views/layouts/default.blade.php

```

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="utf-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

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

    <title>Laravel CRUD</title>

```

```

    <link href="https://cdnjs.cloudflare.com/ajax/libs/twitter-bootstrap/4.0.0-alpha/css/bootstrap.css" rel="stylesheet">

</head>

<body>

<div class="container">

    @yield('content')

</div>

</body>

</html>

```

Oke sekarang kita membuat folder ItemCRUD di **resources/views/ItemCRUD** pertama buat index.blade.php di dalam folder tersebut.

untuk index,create,show dan edit dibuat di folder **ItemCRUD**.

resources/views/ItemCRUD/index.blade.php

```

@extends('layouts.default')

@section('content')

    <div class="row">

        <div class="col-lg-12 margin-tb">

            <div class="pull-left">

                <h2>Items CRUD</h2>

            </div>

            <div class="pull-right">

                <a class="btn btn-success" href="{{ route('itemCRUD.create') }}"> Create New Item

            </a>

            </div>

        </div>

    </div>

```

```

@if ($message = Session::get('success'))

    <div class="alert alert-success">

        <p>{{ $message }}</p>

    </div>

@endif

<table class="table table-bordered">

    <tr>

        <th>No</th>

        <th>Title</th>

        <th>Description</th>

        <th width="280px">Action</th>

    </tr>

    @foreach ($items as $key => $item)

    <tr>

        <td>{{ ++$i }}</td>

        <td>{{ $item->title }}</td>

        <td>{{ $item->description }}</td>

        <td>

            <a class="btn btn-info" href="{{ route('itemCRUD.show',$item->id) }}">Show</a>

            <a class="btn btn-primary" href="{{ route('itemCRUD.edit',$item->id) }}">Edit</a>

            {!! Form::open(['method' => 'DELETE','route' => ['itemCRUD.destroy', $item->id], 'style'=>'display:inline']) !!}

            {!! Form::submit('Delete', ['class' => 'btn btn-danger']) !!}

            {!! Form::close() !!}

        </td>

    </tr>

    @endforeach

</table>

```

```
{!! $items->render() !!}
```

```
@endsection
```

Selanjutnya membuat create.blade.php di dalam folder "ItemCRUD".

```
resources/views/ItemCRUD/create.blade.php
```

```
@extends('layouts.default')
```

```
@section('content')
```

```
<div class="row">
```

```
<div class="col-lg-12 margin-tb">
```

```
<div class="pull-left">
```

```
<h2>Create New Item</h2>
```

```
</div>
```

```
<div class="pull-right">
```

```
<a class="btn btn-primary" href="{{ route('itemCRUD.index') }}"> Back</a>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
@if (count($errors) > 0)
```

```
<div class="alert alert-danger">
```

```
<strong>Whoops!</strong> There were some problems with your input.<br><br>
```

```
<ul>
```

```
@foreach ($errors->all() as $error)
```

```
<li>{{ $error }}</li>
```

```
@endforeach
```

```
</ul>
```

```
</div>
```

```
@endif
```

```

{!! Form::open(array('route' => 'itemCRUD.store','method'=>'POST')) !!}

<div class="row">

    <div class="col-xs-12 col-sm-12 col-md-12">

        <div class="form-group">

            <strong>Title:</strong>

            {!! Form::text('title', null, array('placeholder' => 'Title','class' => 'form-control')) !!}

        </div>

    </div>

    <div class="col-xs-12 col-sm-12 col-md-12">

        <div class="form-group">

            <strong>Description:</strong>

            {!! Form::textarea('description', null, array('placeholder' => 'Description','class' => 'form-control','style'=>'height:100px')) !!}

        </div>

    </div>

    <div class="col-xs-12 col-sm-12 col-md-12 text-center">

        <button type="submit" class="btn btn-primary">Submit</button>

    </div>

</div>

{!! Form::close() !!}

@endsection

```

Selanjutnya membuat file show.blade.php untuk detail record , di folder ItemCRUD .

resources/views/ItemCRUD/show.blade.php

```
@extends('layouts.default')
```



```
</div>
```

```
@endsection
```

Dan terakhir membuat edit.blade.php untuk edit item di folder ItemCRUD.

resources/views/ItemCRUD/edit.blade.php

```
@extends('layouts.default')
```

```
@section('content')
```

```
<div class="row">
```

```
<div class="col-lg-12 margin-tb">
```

```
<div class="pull-left">
```

```
<h2>Edit New Item</h2>
```

```
</div>
```

```
<div class="pull-right">
```

```
<a class="btn btn-primary" href="{{ route('itemCRUD.index') }}"> Back</a>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
@if (count($errors) > 0)
```

```
<div class="alert alert-danger">
```

```
<strong>Whoops!</strong> There were some problems with your input.<br><br>
```

```
<ul>
```

```
@foreach ($errors->all() as $error)
```

```
<li>{{ $error }}</li>
```

```
@endforeach
```

```
</ul>
```

```
</div>
```

```
@endif
```

```

{!! Form::model($item, ['method' => 'PATCH', 'route' => ['itemCRUD.update', $item->id]]) !!}

<div class="row">

    <div class="col-xs-12 col-sm-12 col-md-12">

        <div class="form-group">

            <strong>Title:</strong>

            {!! Form::text('title', null, array('placeholder' => 'Title', 'class' => 'form-control')) !!}

        </div>

    </div>

    <div class="col-xs-12 col-sm-12 col-md-12">

        <div class="form-group">

            <strong>Description:</strong>

            {!! Form::textarea('description', null, array('placeholder' => 'Description', 'class' => 'form-control', 'style' => 'height:100px')) !!}

        </div>

    </div>

    <div class="col-xs-12 col-sm-12 col-md-12 text-center">

        <button type="submit" class="btn btn-primary">Submit</button>

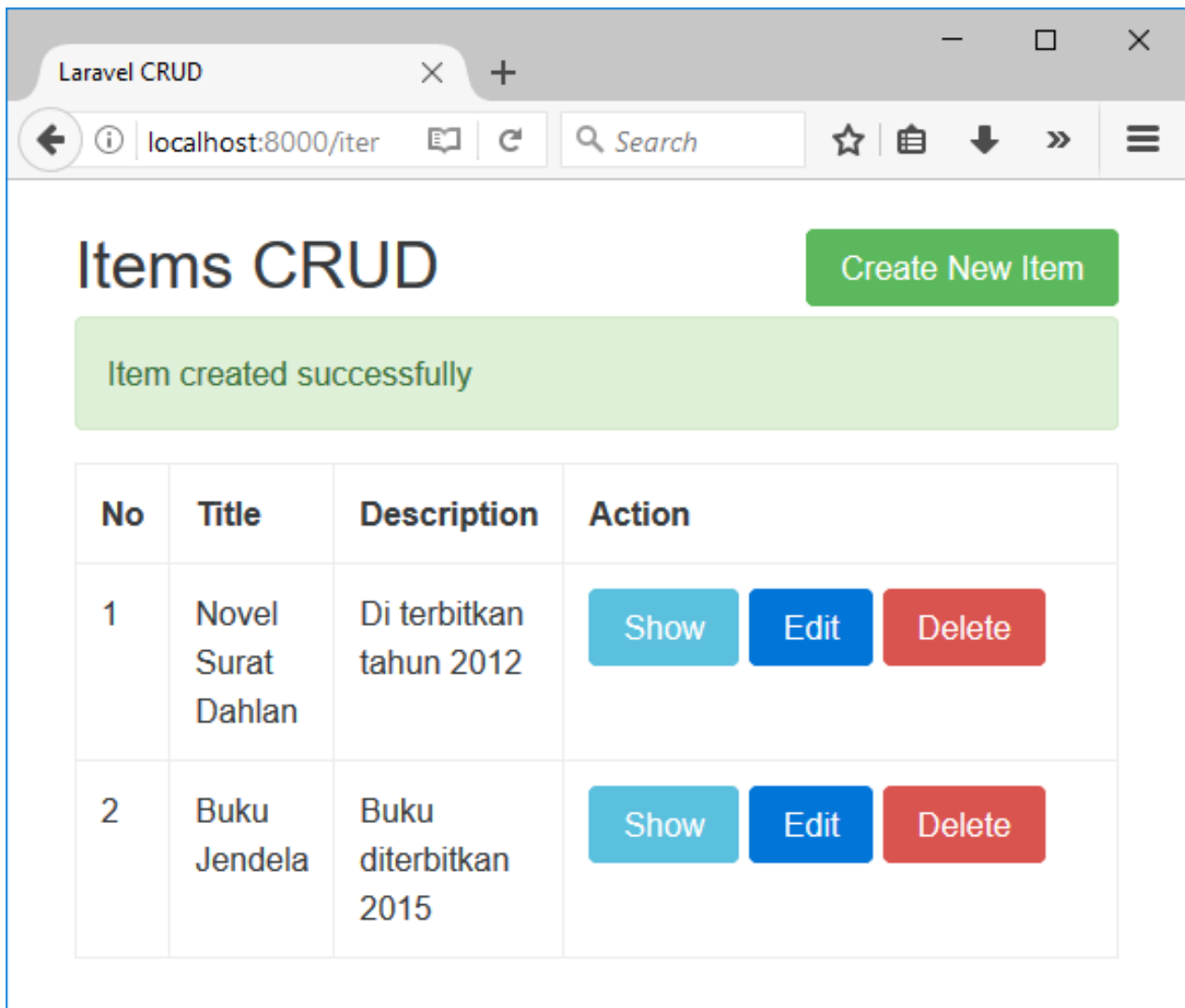
    </div>

</div>

{!! Form::close() !!}

```

@endsection



Hasil dari Item Crud

sumber :

<http://www.itsolutionstuff.com/post/crud-create-read-update-delete-example-in-laravel-52-from-scratchexample.html>

<https://laravelcollective.com/docs/5.3/html>