

**UTS**  
**PEMROGRAMAN WEB LANJUT**



**FARADISHA ALDINA PUTRI**

**TI-2I / 07**

**2141720159**

**D-IV TEKNIK INFORMATIKA**  
**JURUSAN TEKNOLOGI INFORMASI**  
**POLITEKNIK NEGERI MALANG**

**2023**

## Praktikum CRUD Dengan ORM

### 1. Membuat Project

> This PC > Local Disk (C:) > xampp > htdocs > projectweb >

Name	Date modified
07_Faradisha A_pwl_uts	4/13/2023 5:10 AM

### 2. Membuat Database

## Databases

Create database

crudlv8 utf8mb4\_general\_ci Create

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=crudlv8
DB_USERNAME=root
DB_PASSWORD=
```

### 3. Buatlah Model dan Migration dengan menggunakan satu perintah saja.

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE

PS C:\xampp\htdocs\projectweb\crudlv8> php
artisan make:model Barang -mc

INFO Model [C:\xampp\htdocs\projectweb\
crudlv8\app\Models\Barang.php] created succ
essfully.

INFO Migration [C:\xampp\htdocs\project
web\crudlv8\database\Migrations\2023_04_10_
152904_create_barangs_table.php] created su
ccessfully.
```

### 4. Kemudian modifikasi file migration table Barang dengan kolom-kolomnya seperti kode, kategori, nama barang, harga, dan qty

```

public function up()
{
    Schema::create('barangs', function (Blueprint $table) {
        $table->id('id');
        $table->string('kode_barang');
        $table->string('nama_barang');
        $table->enum('kategori_barang', ['makanan', 'minuman', 'snack', 'sembako']);
        $table->integer('harga');
        $table->integer('qty');
        $table->timestamps();
    });
}

```

5. Lalu menjalankan perintah php artisan migrate

```

PS C:\xampp\htdocs\projectweb\crudlv8> php
artisan migrate

[INFO] Preparing database.

Creating migration table .... 39ms DONE

[INFO] Running migrations.

2014_10_12_000000_create_users_table 67ms
s DONE
2014_10_12_100000_create_password_resets_
table 88ms DONE
2019_08_19_000000_create_failed_jobs_tabl
e 70ms DONE
2019_12_14_000001_create_personal_access_
tokens_table 82ms DONE
2023_04_10_152904_create_barangs_table 2
5ms DONE

```

6. Lakukan pengecekan kepada phpMyAdmin apakah tabel sudah masuk kedalam database

Server: 127.0.0.1 » Database: crudlv8 » Table: barangs

Table structure Relation view

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	id	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT	Change  Drop  More
<input type="checkbox"/>	2	kode_barang	varchar(255)	utf8mb4_unicode_ci		No	None			Change  Drop  More
<input type="checkbox"/>	3	nama_barang	varchar(255)	utf8mb4_unicode_ci		No	None			Change  Drop  More
<input type="checkbox"/>	4	kategori_barang	varchar(255)	utf8mb4_unicode_ci		No	None			Change  Drop  More
<input type="checkbox"/>	5	harga	int(11)			No	None			Change  Drop  More
<input type="checkbox"/>	6	qty	int(11)			No	None			Change  Drop  More
<input type="checkbox"/>	7	created_at	timestamp			Yes	NULL			Change  Drop  More
<input type="checkbox"/>	8	updated_at	timestamp			Yes	NULL			Change  Drop  More

## 7. Membuat dan menjalankan Seeder

```
15
16 public function run()
17 {
18     DB::table('barang')->insert([
19         'kode_barang' => 'PRD001',
20         'nama_barang' => 'Indomie',
21         'kategori_barang' => 'Makanan',
22         'harga' => '3000',
23         'qty' => '100',
24     ]);
25 }
26
27 }
28
```

PROBLEMS OUTPUT **TERMINAL** DEBUG CONSOLE

```
PS C:\xampp\htdocs\projectweb\crudlv8> php
artisan make:seeder BarangSeeder

INFO Seeder [C:\xampp\htdocs\projectweb\crudlv8\database\seeders\BarangSeeder.php]
created successfully.

PS C:\xampp\htdocs\projectweb\crudlv8>
```

```
PS C:\xampp\htdocs\projectweb\crudlv8> php
artisan db:seed --class=BarangSeeder

INFO Seeding database.
```

## 8. Lalu penambahan kode program pada web.php

```
10
17 Route::get('/', function () {
18     return view('welcome');
19 });
20 Route::get('/barang', [BarangController::class, 'index'])->name('barang');
```

9. Pada file Controller yang kita dapat secara otomatis akan terdapat 7 fungsi berikut yang dapat kita gunakan untuk membuat operasi CRUD dan jangan lupa sekaligus menyesuaikan dengan fungsi-fungsi yang terdapat di web.php agar dapat ter-routing.

- index() dan dapat untuk search

```
public function index(Request $request){
    if($request->has('search')){
        $data = Barang::where('nama_barang', 'LIKE', '%' . $request->search. '%')->paginate(5);
    } else {
        $data = Barang::paginate(5);
    }

    return view('databarang', compact('data'));
}
```

- tambahbarang()

```
public function tambahbarang(){
    return view('tambahdata');
}
```

- insert()

```
public function insertdata(Request $request){
    // dd($request->all());
    Barang::create($request->all());
    return redirect()->route('barang')->with('success', 'Data Successfully Added!');
}
```

- tampilKandata()

```
public function tampilKandata($id){
    $data = Barang::find($id);
    return view('tampildata', compact('data'));
}
```

- update()

```
public function updatedata(Request $request, $id){
    $data = Barang::find($id);
    $data->update($request->all());
    return redirect()->route('barang')->with('success', 'Data Successfully Updated!');
}
```

- delete()

```
public function delete($id){
    $data = Barang::find($id);
    $data->delete();
    return redirect()->route('barang')->with('success', 'Data Successfully Deleted!');
}
```

- show()

```
public function show($id){
    $data = Barang::find($id);
    return view('show', compact('data'));
}
```

- Routing pada web.php

```
Route::get('/', function () {
    return view('welcome');
});
Route::get('/barang', [BarangController::class, 'index']->name('barang');
Route::get('/tambahbarang', [BarangController::class, 'tambahbarang']->name('tambahbarang');
Route::post('/insertdata', [BarangController::class, 'insertdata']->name('insertdata');

Route::get('/tampilKandata/{id}', [BarangController::class, 'tampilKandata']->name('tampilKandata');
Route::post('/updatedata/{id}', [BarangController::class, 'updatedata']->name('updatedata');

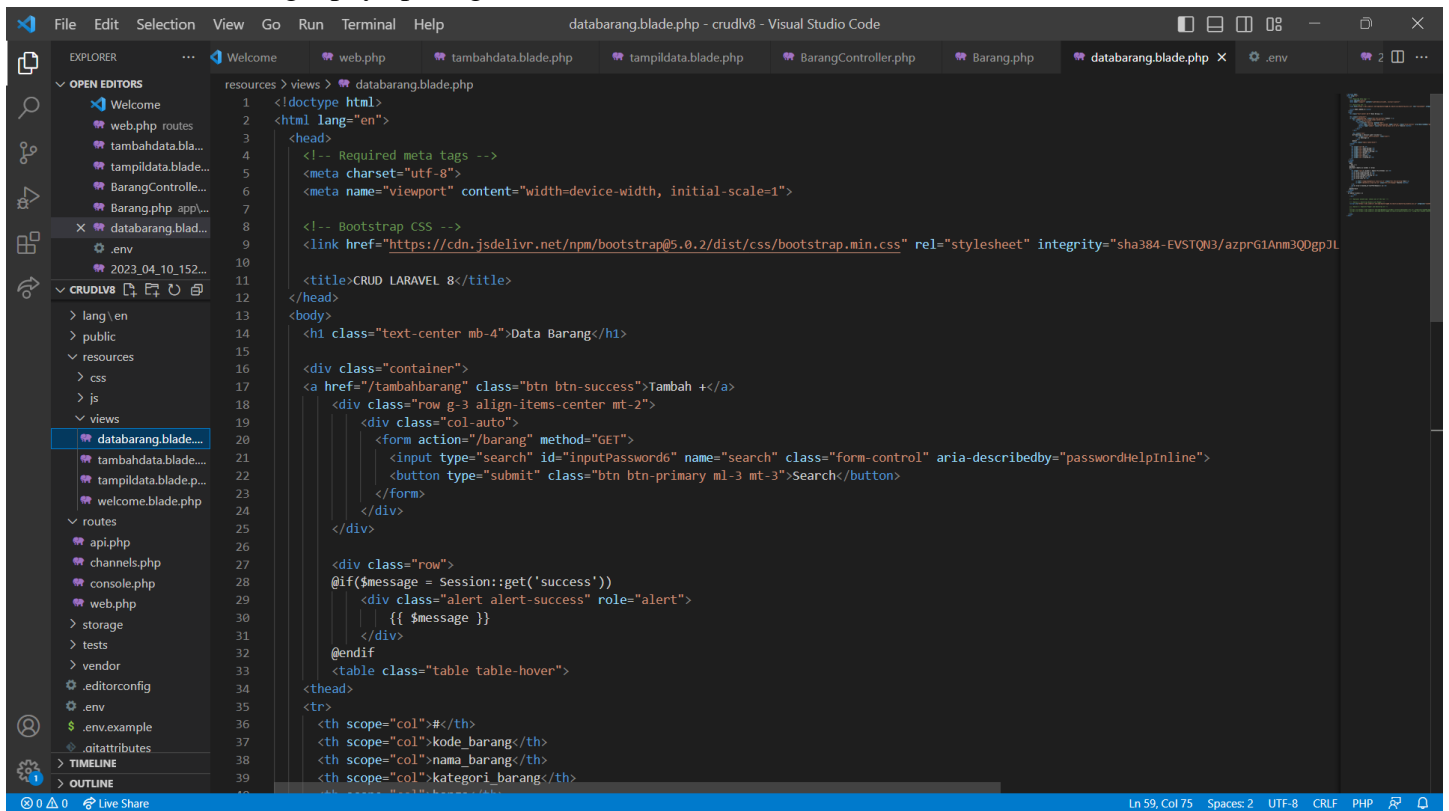
Route::get('/delete/{id}', [BarangController::class, 'delete']->name('delete');
Route::get('/show/{id}', [BarangController::class, 'show']->name('show');
```

10. Setting nilai \$fillable di Model User tambahkan syntax seperti ini agar semua data dapat terinput dan dapat tertampilkan tanpa harus memvalidasi satu per satu

```
app > Models > Barang.php
1  <?php
2
3  namespace App\Models;
4
5  use Illuminate\Database\Eloquent\Factories\HasFactory;
6  use Illuminate\Database\Eloquent\Model;
7
8  class Barang extends Model
9  {
10     use HasFactory;
11     protected $guarded = [];
12     protected $dates = ['created_at'];
13 }
14
```

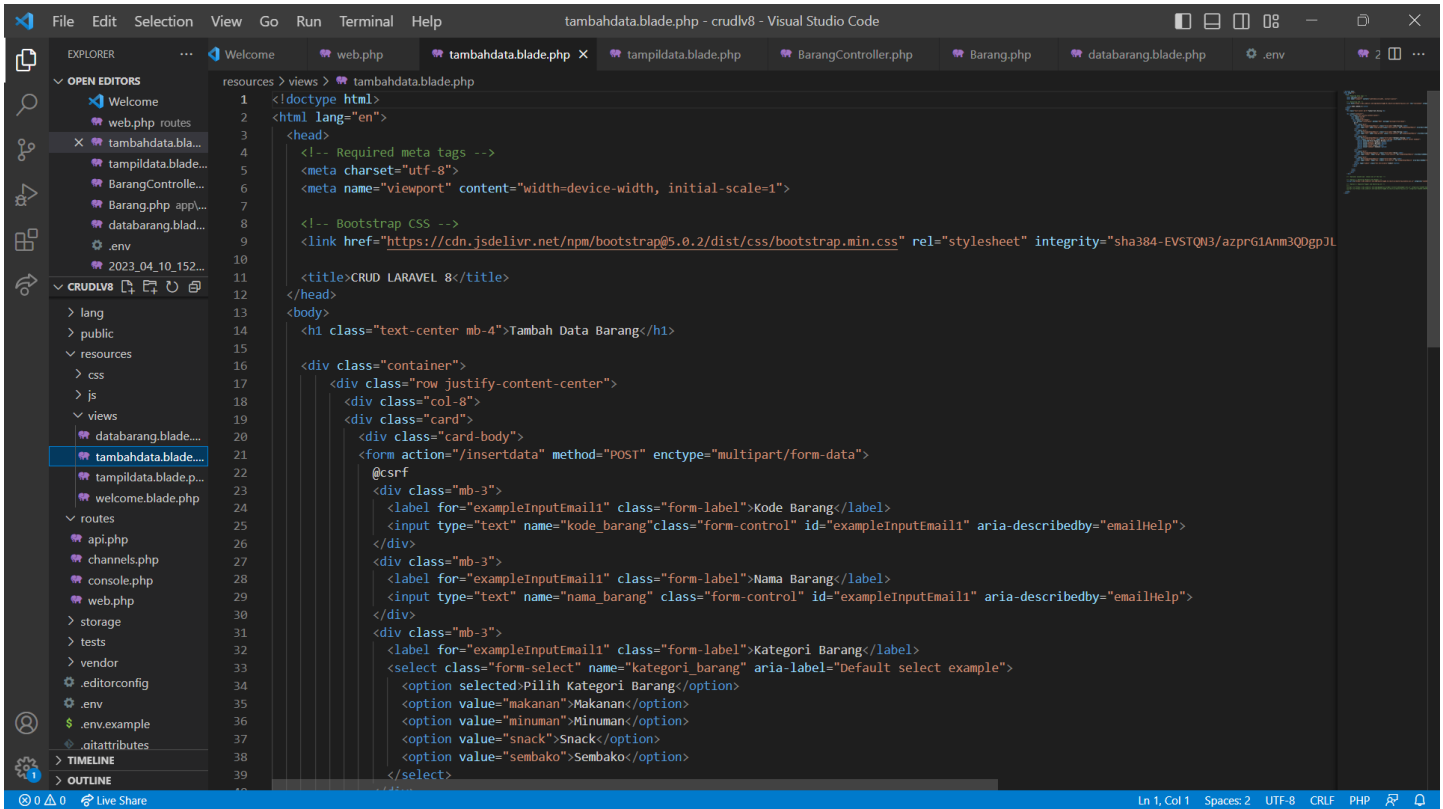
11. Selanjutnya buatlah folder users pada resources/views. Terakhir buat file-file bladenya sesuai form-form yang dibutuhkan

- databarang.blade.php tampilan awal, semua data ditampilkan pada halaman ini (selengkapnya pada github)



```
resources > views > databarang.blade.php
1  <!doctype html>
2  <html lang="en">
3  <head>
4      <!-- Required meta tags -->
5      <meta charset="utf-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1">
7
8      <!-- Bootstrap CSS -->
9      <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-EVSTQN3/azprG1Anm3QDgpJL...
10
11  <title>CRUD LARAVEL 8</title>
12 </head>
13 <body>
14  <h1 class="text-center mb-4">Data Barang</h1>
15
16  <div class="container">
17      <a href="/tambahbarang" class="btn btn-success">Tambah </a>
18      <div class="row g-3 align-items-center mt-2">
19          <div class="col-auto">
20              <form action="/barang" method="GET">
21                  <input type="search" id="inputPassword6" name="search" class="form-control" aria-describedby="passwordHelpInline">
22                  <button type="submit" class="btn btn-primary ml-3 mt-3">Search</button>
23              </form>
24          </div>
25      </div>
26
27      <div class="row">
28          @if($message = Session::get('success'))
29              <div class="alert alert-success" role="alert">
30                  {{ $message }}
31              </div>
32          @endif
33      <table class="table table-hover">
34          <thead>
35              <tr>
36                  <th scope="col">#</th>
37                  <th scope="col">kode_barang</th>
38                  <th scope="col">nama_barang</th>
39                  <th scope="col">kategori_barang</th>
40          </thead>
41      </table>
42  </div>
43 </body>
44 </html>
```

- tambahdata.blade.php untuk memunculkan form untuk menambahkan data barang baru setelah menekan tombol “Tambah+” (selengkapnya pada github)

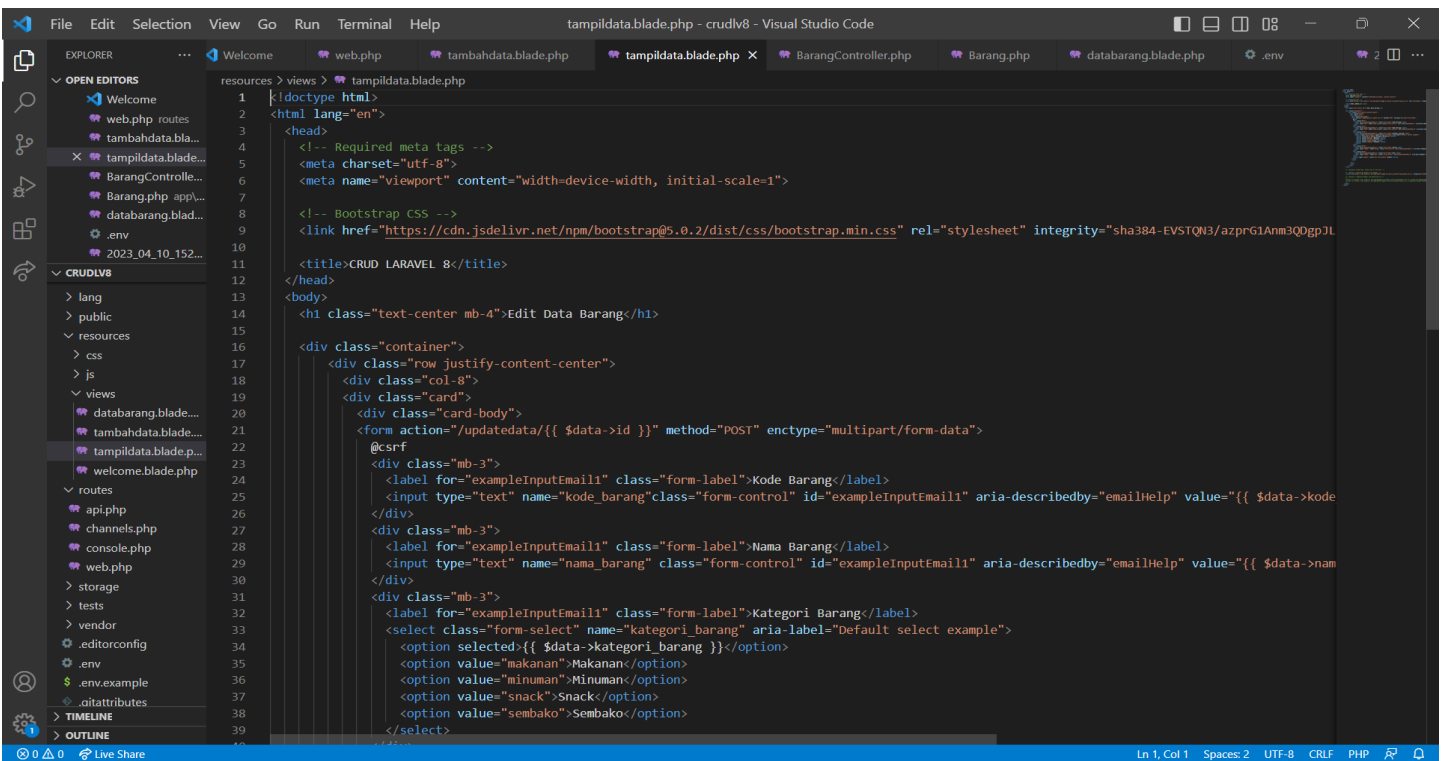


```

1 <!doctype html>
2 <html lang="en">
3 <head>
4 <!-- Required meta tags -->
5 <meta charset="utf-8">
6 <meta name="viewport" content="width=device-width, initial-scale=1">
7
8 <!-- Bootstrap CSS -->
9 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-EVSTQN3/azprG1Anm3Q0gpJL
10
11 <title>CRUD LARAVEL 8</title>
12 </head>
13 <body>
14 <h1 class="text-center mb-4">Tambah Data Barang</h1>
15
16 <div class="container">
17 <div class="row justify-content-center">
18 <div class="col-8">
19 <div class="card">
20 <div class="card-body">
21 <form action="/insertdata" method="POST" enctype="multipart/form-data">
22 @csrf
23 <div class="mb-3">
24 <label for="exampleInputEmail1" class="form-label">Kode Barang</label>
25 <input type="text" name="kode_barang" class="form-control" id="exampleInputEmail1" aria-describedby="emailHelp">
26 </div>
27 <div class="mb-3">
28 <label for="exampleInputEmail1" class="form-label">Nama Barang</label>
29 <input type="text" name="nama_barang" class="form-control" id="exampleInputEmail1" aria-describedby="emailHelp">
30 </div>
31 <div class="mb-3">
32 <label for="exampleInputEmail1" class="form-label">Kategori Barang</label>
33 <select class="form-select" name="kategori_barang" aria-label="Default select example">
34 <option selected>Pilih Kategori Barang</option>
35 <option value="makanan">Makanan</option>
36 <option value="minuman">Minuman</option>
37 <option value="snack">Snack</option>
38 <option value="sembako">Sembako</option>
39 </select>
40

```

- Tampildata.blade.php untuk memunculkan form update/edit data saat kita menekan tombol edit. (selengkapnya pada github)

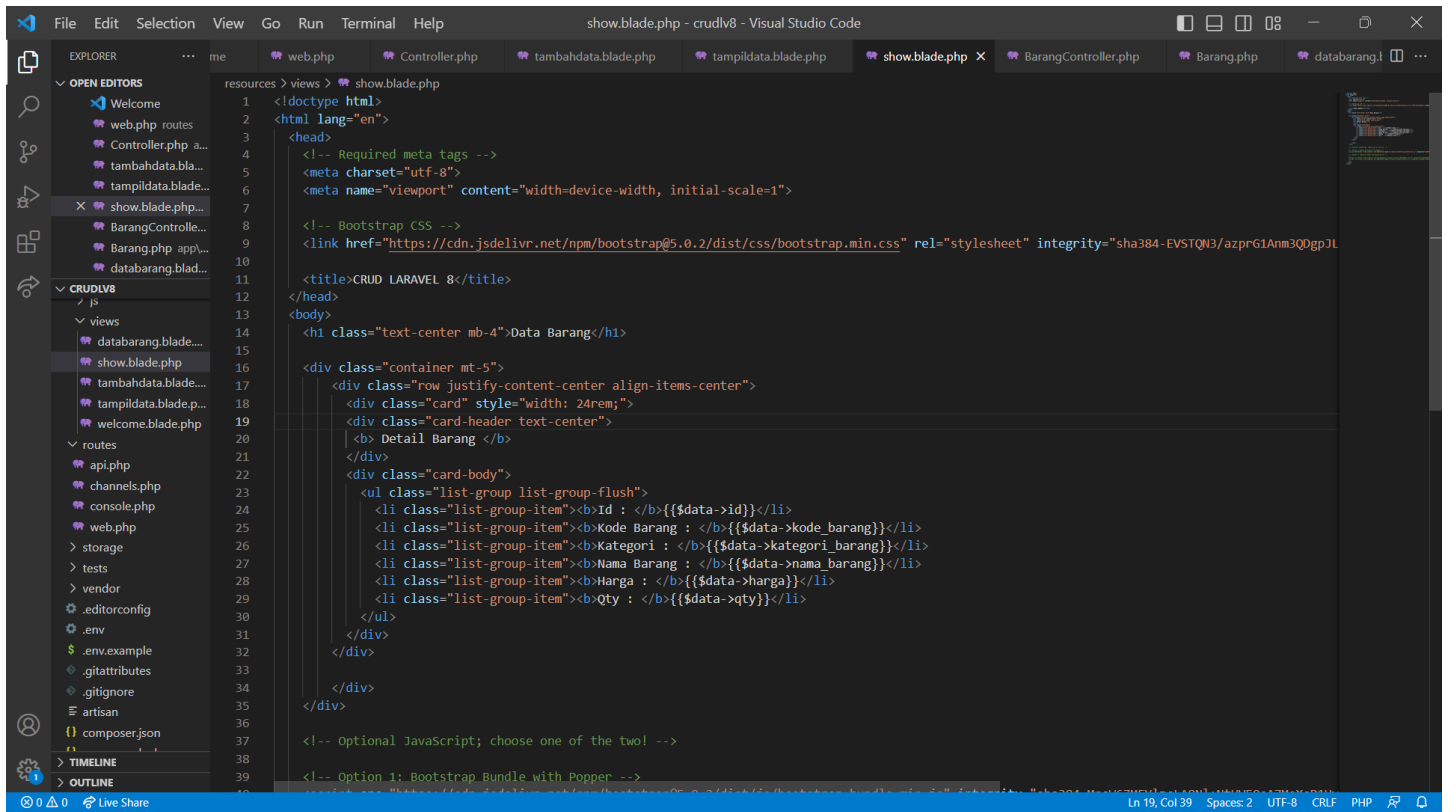


```

1 <!doctype html>
2 <html lang="en">
3 <head>
4 <!-- Required meta tags -->
5 <meta charset="utf-8">
6 <meta name="viewport" content="width=device-width, initial-scale=1">
7
8 <!-- Bootstrap CSS -->
9 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-EVSTQN3/azprG1Anm3Q0gpJL
10
11 <title>CRUD LARAVEL 8</title>
12 </head>
13 <body>
14 <h1 class="text-center mb-4">Edit Data Barang</h1>
15
16 <div class="container">
17 <div class="row justify-content-center">
18 <div class="col-8">
19 <div class="card">
20 <div class="card-body">
21 <form action="/updatedata/{{ $data->id }}" method="POST" enctype="multipart/form-data">
22 @csrf
23 <div class="mb-3">
24 <label for="exampleInputEmail1" class="form-label">Kode Barang</label>
25 <input type="text" name="kode_barang" class="form-control" id="exampleInputEmail1" aria-describedby="emailHelp" value="{{ $data->kode
26 </div>
27 <div class="mb-3">
28 <label for="exampleInputEmail1" class="form-label">Nama Barang</label>
29 <input type="text" name="nama_barang" class="form-control" id="exampleInputEmail1" aria-describedby="emailHelp" value="{{ $data->nam
30 </div>
31 <div class="mb-3">
32 <label for="exampleInputEmail1" class="form-label">Kategori Barang</label>
33 <select class="form-select" name="kategori_barang" aria-label="Default select example">
34 <option selected>{{ $data->kategori_barang }}</option>
35 <option value="makanan">Makanan</option>
36 <option value="minuman">Minuman</option>
37 <option value="snack">Snack</option>
38 <option value="sembako">Sembako</option>
39 </select>
40

```

- Show.blade.php untuk menampilkan detail barang saat menekan tombol detail



## ➤ Paginator

```
public function boot()
{
    Paginator::useBootstrap();
}
```

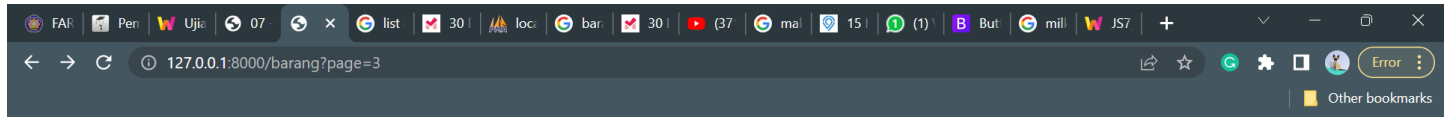


```
>paginate(5);
```



## 12. Hasil :

### ➤ Tampilan awal



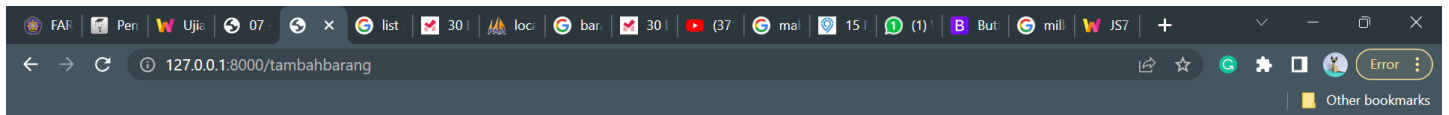
## Data Barang

Tambah +

Search

#	kode_barang	nama_barang	kategori_barang	harga	qty	action			created_at
11	PRD011	Air Mineral	minuman	3000	30	<a href="#">Details</a>	<a href="#">Edit</a>	<a href="#">Delete</a>	1 hour ago
12	PRD012	Bengbeng	snack	2500	20	<a href="#">Details</a>	<a href="#">Edit</a>	<a href="#">Delete</a>	1 hour ago
13	PRD013	Roti Aoka	makanan	2500	30	<a href="#">Details</a>	<a href="#">Edit</a>	<a href="#">Delete</a>	1 hour ago
14	PRD014	Sari Roti Tawar	makanan	13000	10	<a href="#">Details</a>	<a href="#">Edit</a>	<a href="#">Delete</a>	1 hour ago
15	PRD015	Milku	minuman	3000	20	<a href="#">Details</a>	<a href="#">Edit</a>	<a href="#">Delete</a>	1 hour ago

### ➤ Tambah barang



## Tambah Data Barang

Kode Barang

Nama Barang

Kategori Barang

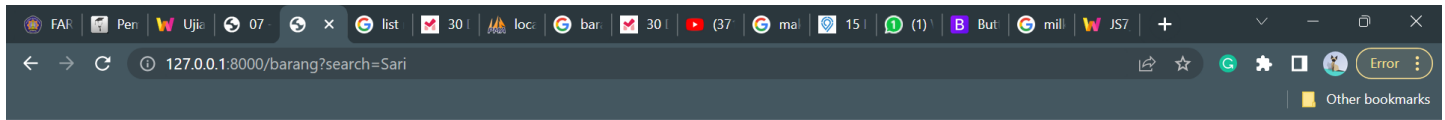
Pilih Kategori Barang

Harga

Qty

Submit

## ➤ Cari Barang



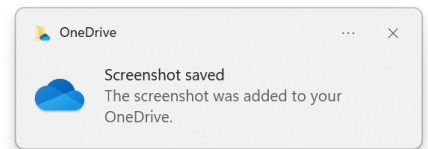
## Data Barang

Tambah +

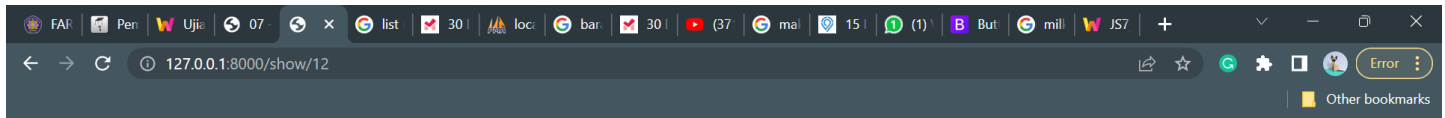
Sari

Search

#	kode_barang	nama_barang	kategori_barang	harga	qty	action	created_at
1	PRD014	Sari Roti Tawar	makanan	13000	10	<a href="#">Details</a> <a href="#">Edit</a> <a href="#">Delete</a>	1 hour ago



## ➤ Detail Barang



## Data Barang

### Detail Barang

Id : 12

Kode Barang : PRD011

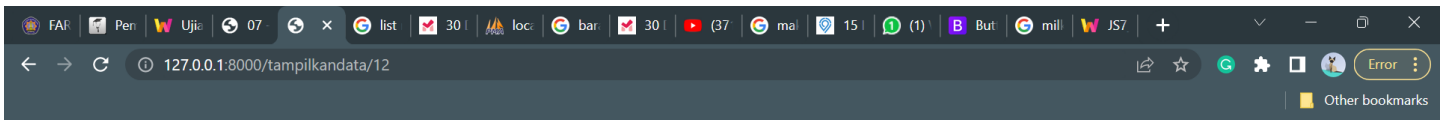
Kategori : minuman

Nama Barang : Air Mineral

Harga : 3000

Qty : 30

## ➤ Update Barang



## Edit Data Barang

Kode Barang

PRD011

Nama Barang

Air Mineral

Kategori Barang

minuman

Harga

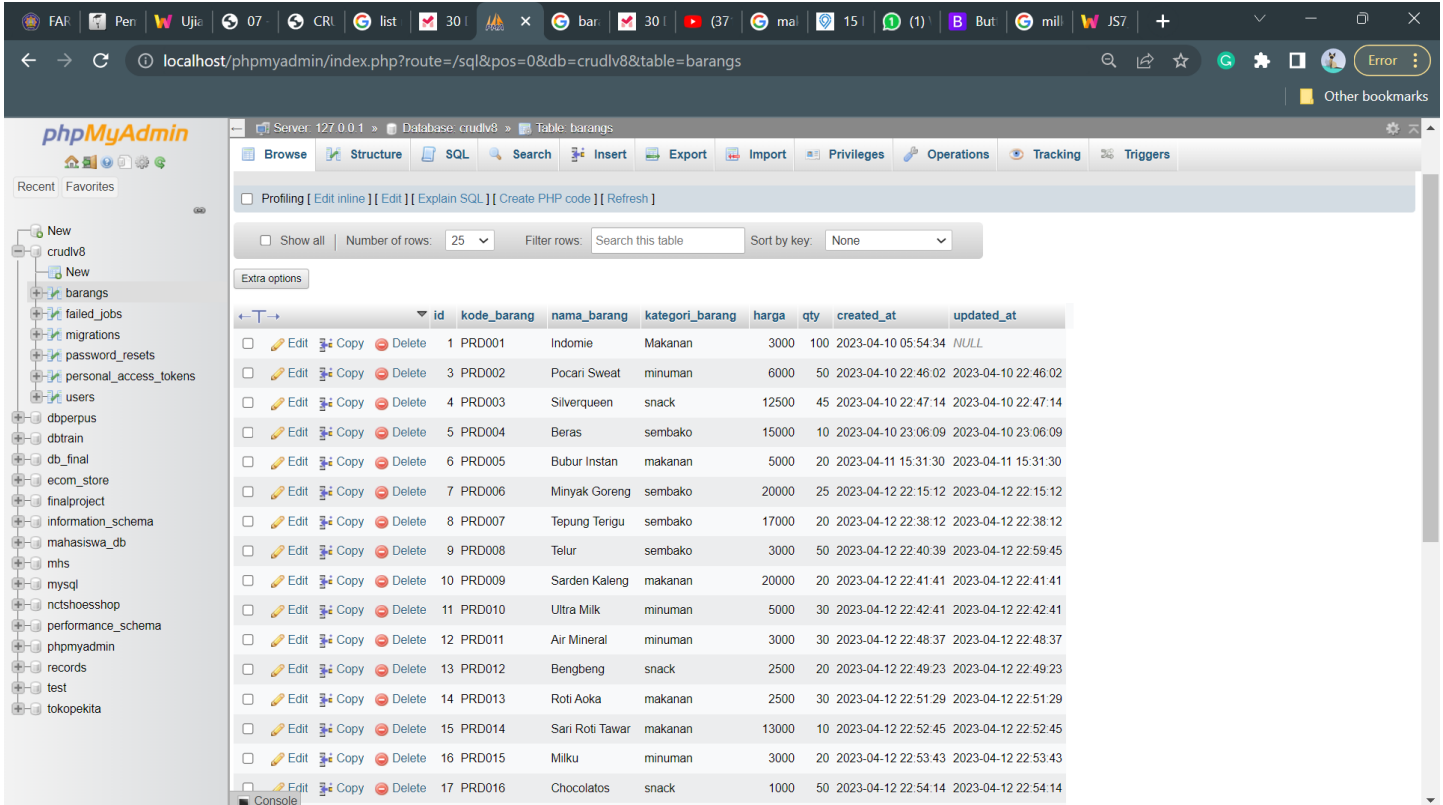
3000

Qty

30

Submit

## ➤ Database



➤ Successfully Added

Tambah +

Search

Data Successfully Added!

➤ Successfully Updated

Tambah +

Search

Data Successfully Updated!

➤ Successfully Deleted

Tambah +

Search

Data Successfully Deleted!