

LAPORAN PRAKTIKUM PENGENALAN APLIKASI BERBASIS INTERNET

CRUD LARAVEL



Agus Pranata Marpaung

13323033

DIII TEKNOLOGI KOMPUTER

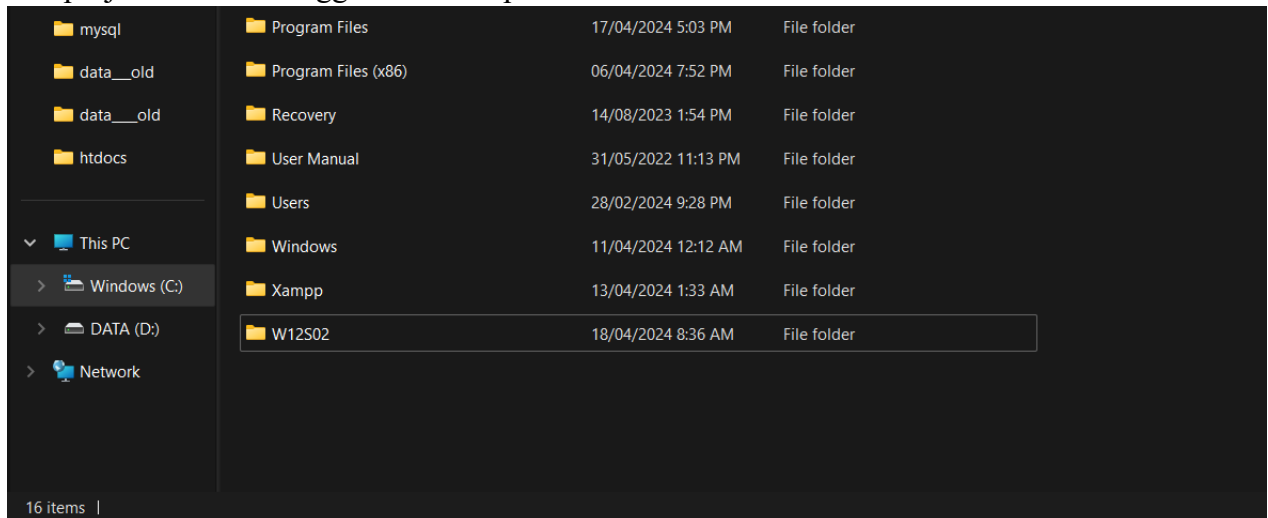
**INSTITUT TEKNOLOGI DEL
FAKULTAS VOKASI**

Judul Praktikum

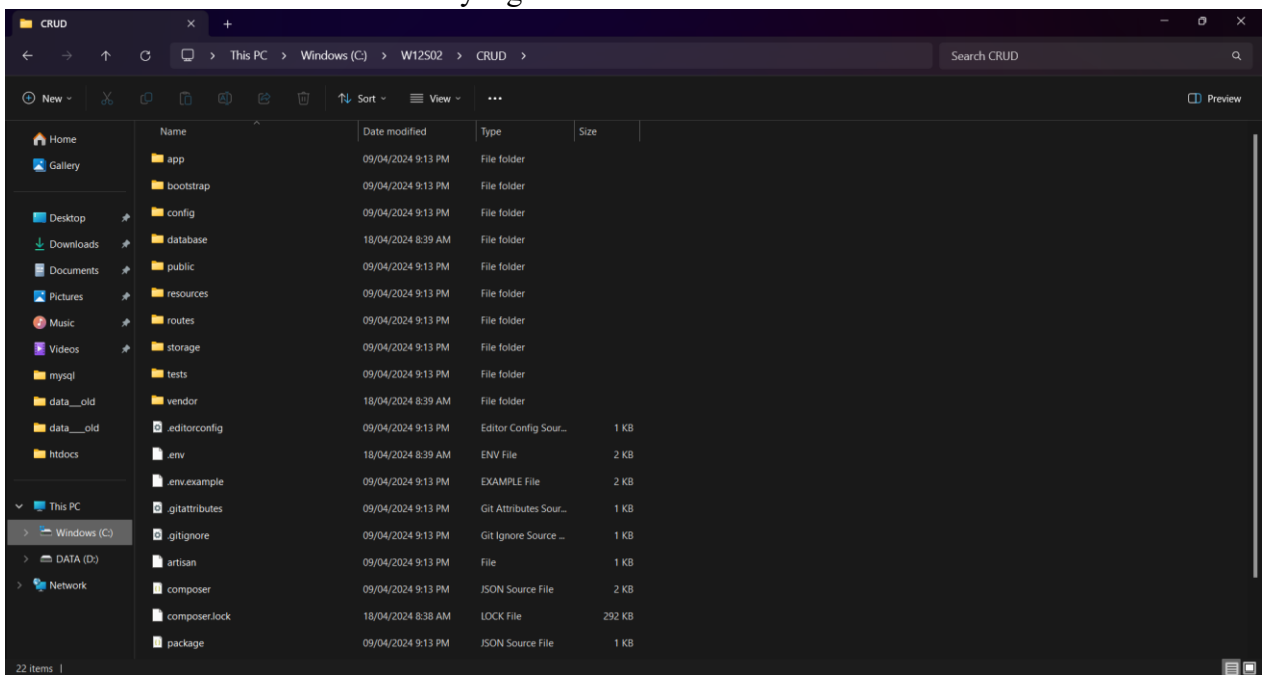
Minggu/Sesi	:	XII/2
Kode Mata Kuliah	:	1331204
Nama Mata Kuliah	:	Pengenalan Aplikasi Berbasis Internet
Setoran	:	Softcopy
Batas Waktu Setoran	:	Akhir sesi
Tujuan	:	1. Mahasiswa dapat melakukan praktikum mengenai CRUD pada laravel 11

Membuat CRUD dengan Laravel

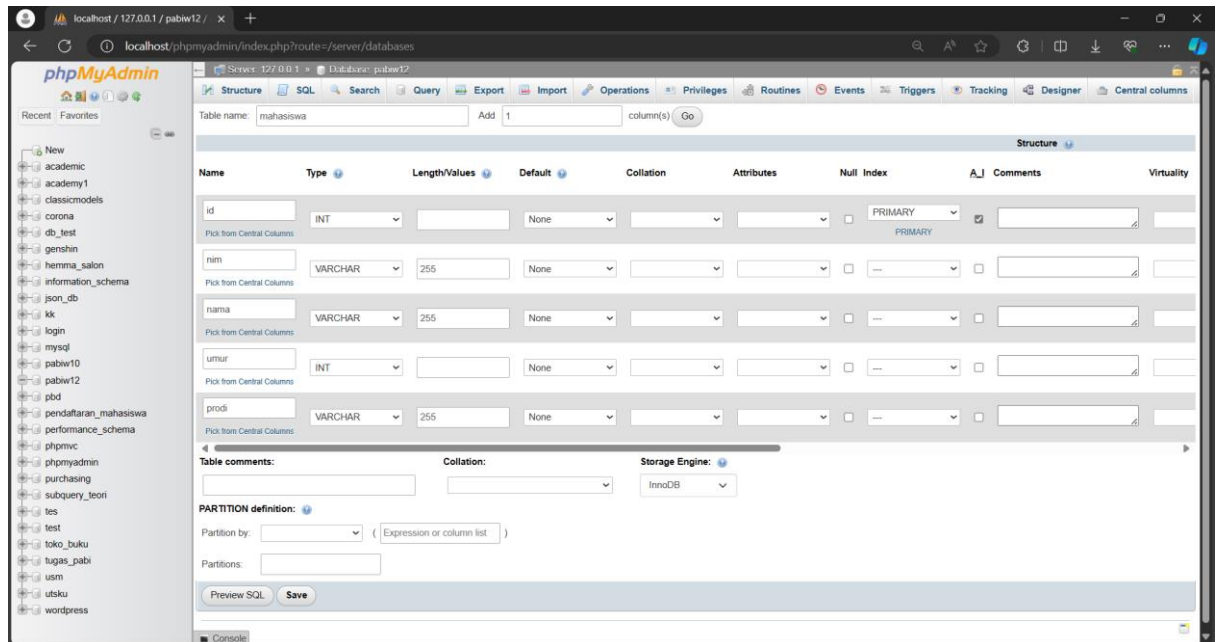
1. Buatlah Folder baru dengan nama **W12S02**, kemudian lakukan instalasi Laravel dengan nama project **CRUD** menggunakan composer.



Berikut folder hasil instalasi laravel yang dilakukan:



2. Kemudian aktifkan **Apache** dan **MySQL** melalui **XAMPP**, dan tambahkan database baru dengan nama **pabiw12**. Buat tabel **mahasiswa** dengan ketentuan sebagai berikut “id” dibuat auto increment.



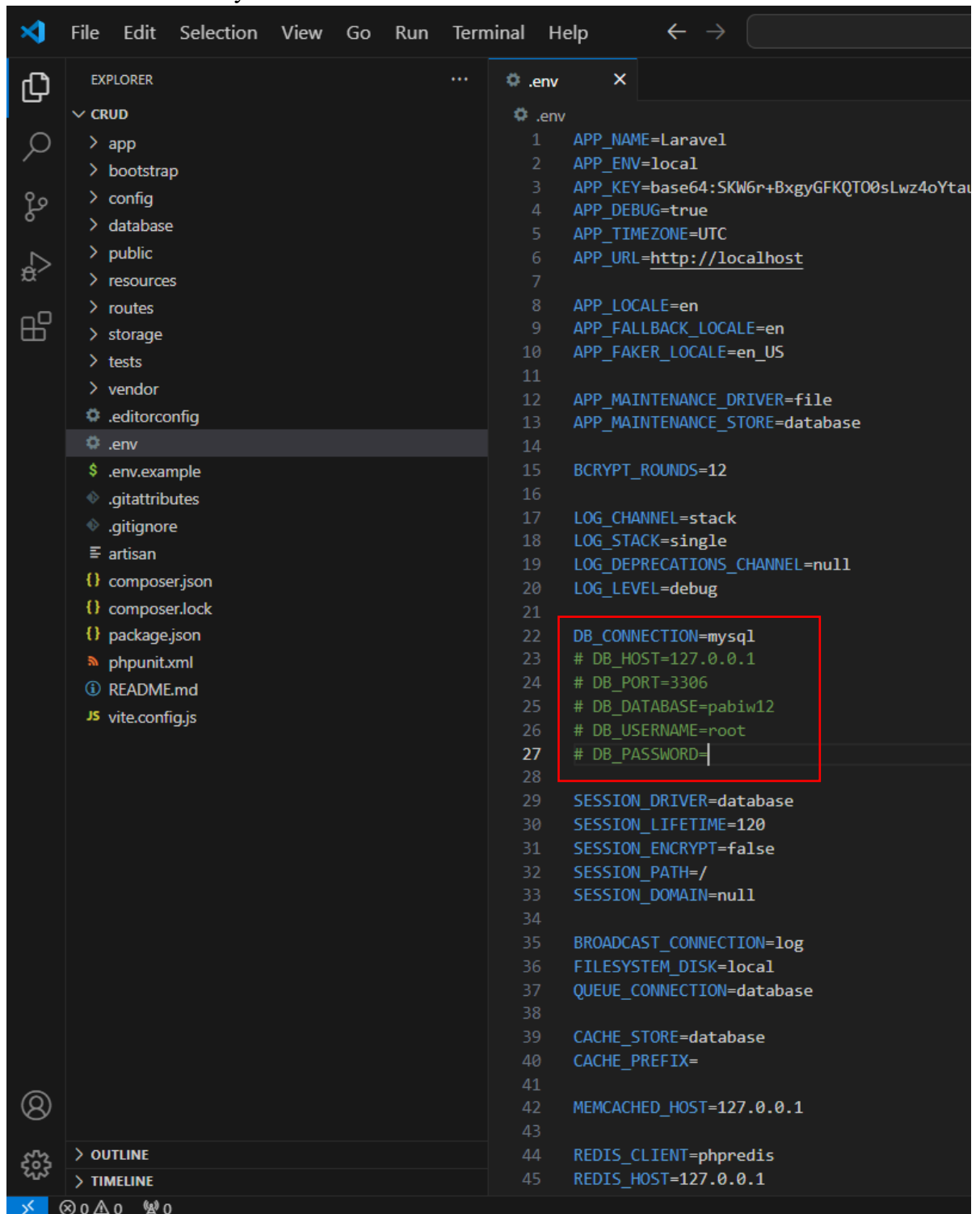
Tambahkan beberapa data pada tabel tersebut, anda dapat menggunakan data sebagai berikut.

insert into mahasiswa values

(null,'1333001','Any a Forger','18','D3 Teknologi Komputer'),

(null,'1333002','Damian Desmond','17','D3 Teknologi Komputer'), (null,'1333003','Becky Blackbell','19','D3 Teknologi Komputer')

3. Kemudian lakukan edit pada file `.env` menggunakan editor sesuaikan dengan database yang telah dibuat sebelumnya.



The screenshot shows the Visual Studio Code editor interface. On the left, the Explorer sidebar displays a file tree for a project named 'CRUD'. The file tree includes folders like 'app', 'bootstrap', 'config', 'database', 'public', 'resources', 'routes', 'storage', 'tests', and 'vendor', along with files like '.editorconfig', '.env', '.env.example', '.gitattributes', '.gitignore', 'artisan', 'composer.json', 'composer.lock', 'package.json', 'phpunit.xml', 'README.md', and 'vite.config.js'. The '.env' file is selected and its contents are displayed in the main editor area. The file contains various environment variables for a Laravel application. A red rectangular box highlights the database configuration section, which includes the following lines:

```
22 DB_CONNECTION=mysql
23 # DB_HOST=127.0.0.1
24 # DB_PORT=3306
25 # DB_DATABASE=pabiw12
26 # DB_USERNAME=root
27 # DB_PASSWORD=
```

The rest of the file contains other configuration variables such as `APP_NAME=Laravel`, `APP_ENV=local`, `APP_KEY`, `APP_DEBUG=true`, `APP_TIMEZONE=UTC`, `APP_URL=http://localhost`, `APP_LOCALE=en`, `APP_FALLBACK_LOCALE=en`, `APP_FAKER_LOCALE=en_US`, `APP_MAINTENANCE_DRIVER=file`, `APP_MAINTENANCE_STORE=database`, `BCRYPT_ROUNDS=12`, `LOG_CHANNEL=stack`, `LOG_STACK=single`, `LOG_DEPRECATED_CHANNELS=null`, `LOG_LEVEL=debug`, `SESSION_DRIVER=database`, `SESSION_LIFETIME=120`, `SESSION_ENCRYPT=false`, `SESSION_PATH=/`, `SESSION_DOMAIN=null`, `BROADCAST_CONNECTION=log`, `FILESYSTEM_DISK=local`, `QUEUE_CONNECTION=database`, `CACHE_STORE=database`, `CACHE_PREFIX=`, `MEMCACHED_HOST=127.0.0.1`, `REDIS_CLIENT=phpredis`, and `REDIS_HOST=127.0.0.1`.

4. Pada folder CRUD tambahkan controller baru menggunakan command berikut pada cmd.

```
C:\Windows\System32\cmd.e x + v
C:\W12S02\CRUD>php artisan make:controller mahasiswacontroller

INFO Controller [C:\W12S02\CRUD\app\Http\Controllers\mahasiswacontroller.php] created successfully.

C:\W12S02\CRUD>
```

Controller digunakan untuk menghubungkan antara Model dengan View dan biasanya controller akan berisi logika-logika dari aplikasi yang akan dibangun. Jika perintah berhasil dijalankan akan mendapatkan file controller baru yang berada di dalam folder `app\Http\Controllers\mahasiswacontroller.php`.

5. Kemudian lakukan edit pada `mahasiswacontroller.php` sehingga menjadi seperti berikut.

```
mahasiswacontroller.php x
app > Http > Controllers > mahasiswacontroller.php
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6  use Illuminate\Support\Facades\DB;
7
8  class mahasiswacontroller extends Controller
9  {
10
11      public function index()
12      {
13          // mengambil data dari tabel mahasiswa
14          $mahasiswa = DB::table('mahasiswa') -> get();
15
16          // mengirim data mahasiswa ke view index
17          return view('mahasiswacontroller/index', ['mahasiswa' => $mahasiswa]);
18      }
19  }
```

Penjelasan:

- Fungsi `index`, akan diambil data dari table 'mahasiswa' dengan perintah `$mahasiswa = DB::table('mahasiswa')->get();`
- Line 13: Fungsi `table()` kita menentukan nama table yang ingin dipilih. Fungsi `get()` berguna untuk mengambil data dari table yang dipilih. - Line 16: Passing data atau kirim data ke view agar ditampilkan.

```
19      // method untuk menampilkan view form tambah mahasiswa
20      public function tambah()
21      {
22
23          // memanggil view tambah
24          return view('tambah');
25      }
26  }
27  }
```

Penjelasan:

Method tambah() berfungsi untuk menampilkan view tambah, yang merupakan form penginputan data mahasiswa.

```
27 // method untuk insert data ke table mahasiswa
28 public function store(Request $request)
29 {
30     // insert data ke table mahasiswa
31     DB::table('mahasiswa')->insert([
32         'nim' => $request->nim,
33         'nama' => $request->nama,
34         'umur' => $request->umur,
35         'prodi' => $request->prodi
36     ]);
37     // alihkan halaman ke halaman mahasiswa
38     return redirect('/mahasiswa');
39 }
```

Penjelasan:

Method store digunakan untuk menangkap data yang dikirimkan dari form tambah pegawai. Cara menangkap datanya di dalam parameter fungsi store(). Diberi keterangan **Request** diikuti dengan nama variabel yang ingin kita jadikan sebagai penyimpan data yang dikirim dari form **\$request**.

Kemudian insert datanya ke tabel mahasiswa dengan menggunakan query builder laravel untuk insert data. Fungsi tabel() untuk memberitahukan nama tabel. Fungsi insert() bertujuan untuk menginsert data dan menetapkan data apa saja yang ingin di insert.

Kemudian setelah kita menginput data ke tabel mahasiswa, maka diberikan perintah untuk mengalihkan halaman ke halaman mahasiswa atau route 'mahasiswa' kembali.

```
41 public function edit($id)
42 {
43     // mengambil data mahasiswa berdasarkan id yang dipilih
44     $mahasiswa = DB::table('mahasiswa')->where('id', $id)->get();
45
46     // passing data mahasiswa yang didapat ke view edit.blade.php
47     return view('edit', ['mahasiswa' => $mahasiswa]);
48 }
```

Penjelasan:

Parameter method edit berfungsi untuk menangkap data id yang dikirimkan dari route '/mahasiswa/edit/id'.

Selanjutnya akan diambil data mahasiswa dari database sekaligus menseleksi data id yang ingin di ambil dengan menyeleksi id mahasiswa. Kemudian passing data ke view edit.blade.php

```

50     public function update (Request $request)
51     {
52         //Update data mahasiswa
53         DB::table('mahasiswa')->where('id', $request->id)->update([
54             'id' => $request->id,
55             'nim' => $request->nim,
56             'nama' => $request->nama,
57             'umur' => $request->umur,
58             'prodi' => $request->prodi
59         ]);
60         // alihkan halaman ke halaman mahasiswa
61         return redirect('/mahasiswa');
62     }
63 }
64

```

Penjelasan:

Seperti pada saat membuat proses penginputan data ke database, data yang dikirim dari form ditangkap dengan mendeklarasikan Request dalam parameter method update.

Data-data yang dikirim dari form edit akan tersimpan didalam variabel \$request. Untuk update data digunakan query builder (line 54-59). Kemudian dialihkan kembali halamannya ke route '/mahasiswa' yang menampilkan semua data mahasiswa.

```

public function hapus($id)
{
    // menghapus data mahasiswa berdasarkan id yang dipilih
    DB::table('mahasiswa')->where('id', $id)->delete();

    // alihkan halaman ke halaman mahasiswa
    return redirect('/mahasiswa');
}

```

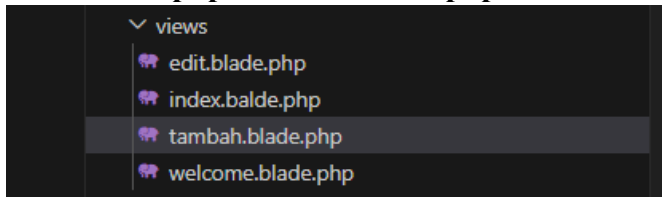
Penjelasan:

Pada method hapus, data id akan ditangkap dalam parameter method hapus, sama seperti kita menangkap data id pada method edit.

Kemudian dibuat query builder untuk menghapus data dari tabel mahasiswa. Pilih tabel dalam fungsi tabel(), kemudian seleksi dengan fungsi where() untuk menentukan data mana yang ingin dihapus. Lalu gunakan fungsi delete() untuk menghapus datanya().

Setelah penghapusan data, diberikan perintah untuk mengalihkan halaman nya kembali ke route '/mahasiswa'.

6. Kemudian pada folder CRUD/resources/views tambahkan 3 file baru dengan nama **index.blade.php** , **tambah.blade.php** , dan **edit.blade.php**



7. Isi file **index.blade.php** dengan potongan kode berikut.

```
index.blade.php X
resources > views > index.blade.php
1 <!DOCTYPE html>
2 <html>
3
4 <head>
5 </head>
6
7 <body>
8   <h3>Data mahasiswa</h3>
9
10  <a href="/mahasiswa/tambah"> + Tambah mahasiswa Baru</a>
11
12  <br />
13  <br />
14
15  <table border="1">
16    <tr>
17      <th>Nim</th>
18      <th>Nama</th>
19      <th>Umur</th>
20      <th>Prodi</th>
21      <th>Ops</th>
22    </tr>
23    @foreach($mahasiswa as $p)
24      <tr>
25        <td>{{ $p->nim }}</td>
26        <td>{{ $p->nama }}</td>
27        <td>{{ $p->umur }}</td>
28        <td>{{ $p->prodi }}</td>
29        <td>
30          <a href="/mahasiswa/edit/{{ $p->id }}">Edit</a>
31          <a href="/mahasiswa/hapus/{{ $p->id }}">Hapus</a>
32        </td>
33      </tr>
34    @endforeach
35  </table>
36
37
38
39 </body>
```

Penjelasan:

Kode diatas berfungsi untuk menampilkan data yang diambil dari tabel mahasiswa yang disimpan dalam variabel \$mahasiswa, dimana disini langsung ditampilkan menggunakan perulangan foreach.

8. Isi file **tambah.blade.php** dengan potongan kode berikut.

```
tambah.blade.php X
resources > views > tambah.blade.php
1  <head>
2
3  </head>
4
5  <body>
6
7      <h3>Data mahasiswa</h3>
8
9      <a href="/mahasiswa"> Kembali </a>
10
11     <br />
12     <br />
13
14     <form action="/mahasiswa/store" method="post">
15         {{ csrf_field() }}
16         Nim <input type="text" name="nim"> <br />
17         Nama <input type="text" name="nama"> <br />
18         Umur <input type="text" name="umur"> <br />
19         prodi <input type="text" name="prodi"> <br />
20         <input type="submit" value="Simpan data">
21     </form>
22
23 </body>
24
25 </html>
```

Penjelasan:

- Line 17: Action form diarahkan ke route ‘/mahasiswa/store’, karena kita ingin route inilah yang akan menangani pemrosesan data yang di input agar dapat diatur oleh controller.
- Line 28: Merupakan fitur keamanan untuk mencegah penginputan data dari luar aplikasi atau sistem.
- Line 19-23: Merupakan beberapa buah form inputan yaitu Nim, Nama, Umur dan Prodi sesuai dengan format tabel yang sudah dibuat di database sebelumnya.

9. Isi file **edit.blade.php** dengan potongan kode berikut.

```
edit.blade.php X
resources > views > edit.blade.php
1 <head>
2 </head>
3
4 <body>
5 <h3>Edit mahasiswa</h3>
6
7 <a href="/mahasiswa"> Kembali</a>
8
9 <br />
10 <br />
11
12 @foreach($mahasiswa as $p)
13 <form action="/mahasiswa/update" method="post">
14     {{ csrf_field() }}
15     <input type="hidden" name="id" value="{{ $p->id }}"> <br />
16     nim <input type="text" required="required" name="nim" value="{{ $p->nim }}"> <br />
17     Nama <input type="text" required="required" name="nama" value="{{ $p->nama }}"> <br />
18     Umur <input type="number" required="required" name="umur" value="{{ $p->umur }}"> <br />
19     prodi <textarea required="required" name="prodi">{{ $p->prodi }}</textarea> <br />
20     <input type="submit" value="Simpan Data">
21 </form>
22 @endforeach
23
24 </body>
25
26 </html>
```

Penjelasan:

Untuk menampilkan data mahasiswa yang ingin di edit ke dalam form edit digunakan perulangan foreach.

10. Setelah pembuatan controller dan view telah selesai, kemudian lakukan penambahan route pada file **web.php** yang berada pada folder **routes**, sehingga menjadi seperti berikut:

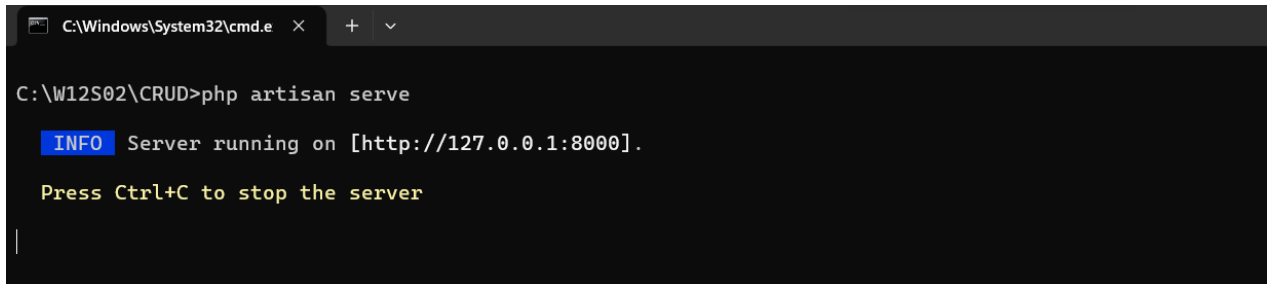
```
web.php X
routes > web.php
1 <?php
2
3 use App\Http\Controllers\MahasiswaController;
4 use Illuminate\Support\Facades\Route;
5
6 Route::get('/', function () {
7     return view('welcome');
8 });
9
10 Route::get('/mahasiswa', 'App\Http\Controllers\MahasiswaController@index');
11 Route::get('/mahasiswa/tambah', 'App\Http\Controllers\MahasiswaController@tambah');
12 Route::post('/mahasiswa/store', 'App\Http\Controllers\MahasiswaController@store');
13 Route::get('/mahasiswa/edit/{id}', 'App\Http\Controllers\MahasiswaController@edit');
14 Route::post('/mahasiswa/update', 'App\Http\Controllers\MahasiswaController@update');
15 Route::get('/mahasiswa/hapus/{id}', 'App\Http\Controllers\MahasiswaController@hapus');
16
```

Penjelasan:

- Line 21: Memerintahkan untuk menjalankan method index() pada controller MahasiswaController pada saat route '/mahasiswa' diakses.

- Line 22: Memerintahkan untuk menjalankan method `tambah()` pada controller MahasiswaController pada saat route `‘/mahasiswa/tambah’` diakses.
- Line 23: Memerintahkan untuk menjalankan method `store()` pada controller MahasiswaController pada saat route `‘/mahasiswa/store’` diakses. Pada route ini kita menggunakan method **post** karena kita mengirimkan data melalui form ke route `‘/mahasiswa/store’`
- Line 24: Memerintahkan untuk menjalankan method `edit()` pada controller MahasiswaController pada saat route `‘/mahasiswa/edit/id’` diakses.
- Line 25: Memerintahkan untuk menjalankan method `update()` pada controller MahasiswaController pada saat route `‘/mahasiswa/update’` diakses.
- Line 26: Memerintahkan untuk menjalankan method `hapus()` pada controller MahasiswaController pada saat route `‘/mahasiswa/hapus/id’` diakses.

11. Setelah semua telah selesai kemudian aktifkan web menggunakan command **php artisan serve**



```

C:\Windows\System32\cmd.e  x  +  v

C:\W12S02\CRUD>php artisan serve

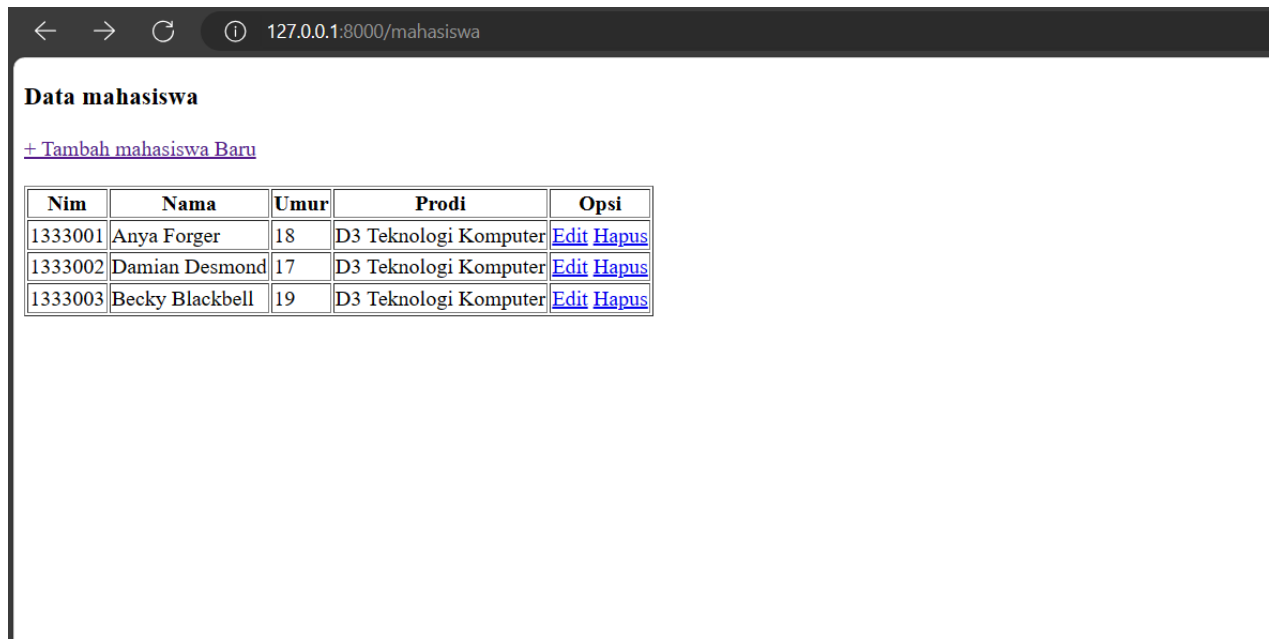
INFO Server running on [http://127.0.0.1:8000].

Press Ctrl+C to stop the server

|

```

12. Kemudian kita coba untuk membuka/mengakses website dengan 127.0.0.1:8000/mahasiswa atau localhost:8000/mahasiswa

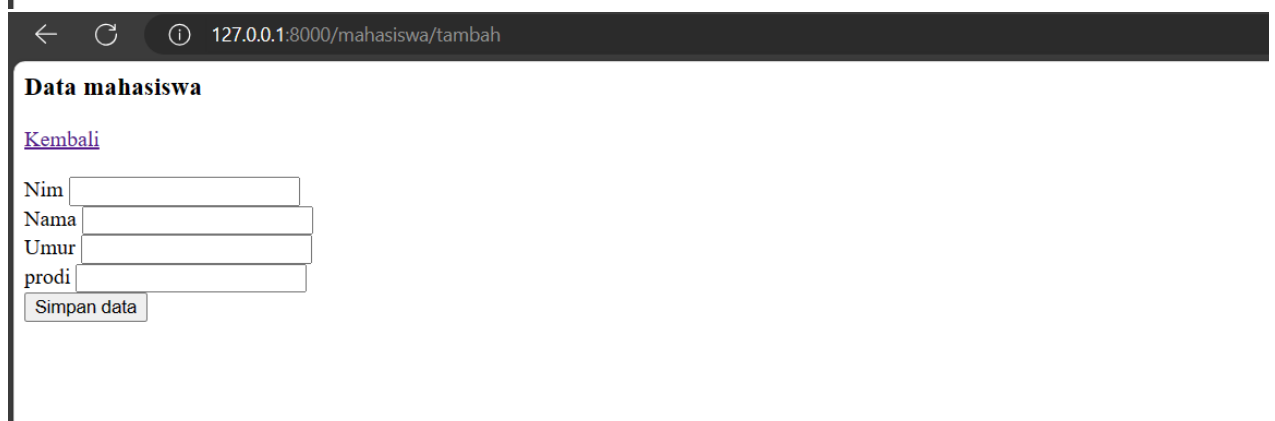


← → ↻ ⓘ 127.0.0.1:8000/mahasiswa

Data mahasiswa

[+ Tambah mahasiswa Baru](#)

Nim	Nama	Umur	Prodi	Opsi
1333001	Anya Forger	18	D3 Teknologi Komputer	Edit Hapus
1333002	Damian Desmond	17	D3 Teknologi Komputer	Edit Hapus
1333003	Becky Blackbell	19	D3 Teknologi Komputer	Edit Hapus



← ↻ ⓘ 127.0.0.1:8000/mahasiswa/tambah

Data mahasiswa

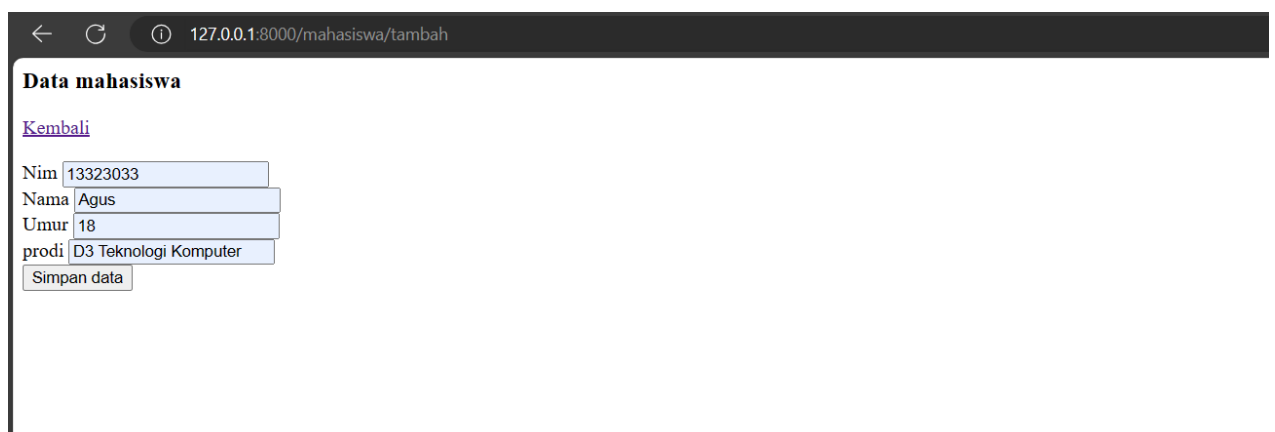
[Kembali](#)

Nim

Nama

Umur

prodi



← ↻ ⓘ 127.0.0.1:8000/mahasiswa/tambah

Data mahasiswa

[Kembali](#)

Nim

Nama

Umur

prodi

Data mahasiswa

[+ Tambah mahasiswa Baru](#)

Nim	Nama	Umur	Prodi	Opsi
1333001	Anyar Forger	18	D3 Teknologi Komputer	Edit Hapus
1333002	Damian Desmond	17	D3 Teknologi Komputer	Edit Hapus
1333003	Becky Blackbell	19	D3 Teknologi Komputer	Edit Hapus
13323033	Agus	18	D3 Teknologi Komputer	Edit Hapus