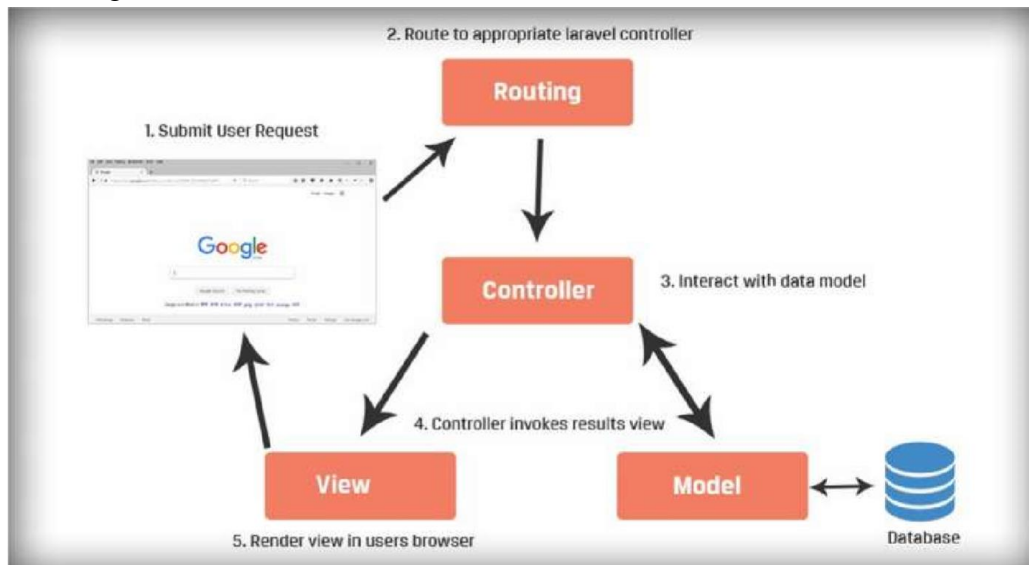


Passing Data dari Inputan

Konsep ini merupakan konsep yg sering digunakan dalam aplikasi, seperti yang kita ketahui MVC, juga dipakai dalam Laravel konsep alur proses nya bisa dilihat pada bagan dibawah ini, semua berawal dari setingan route.



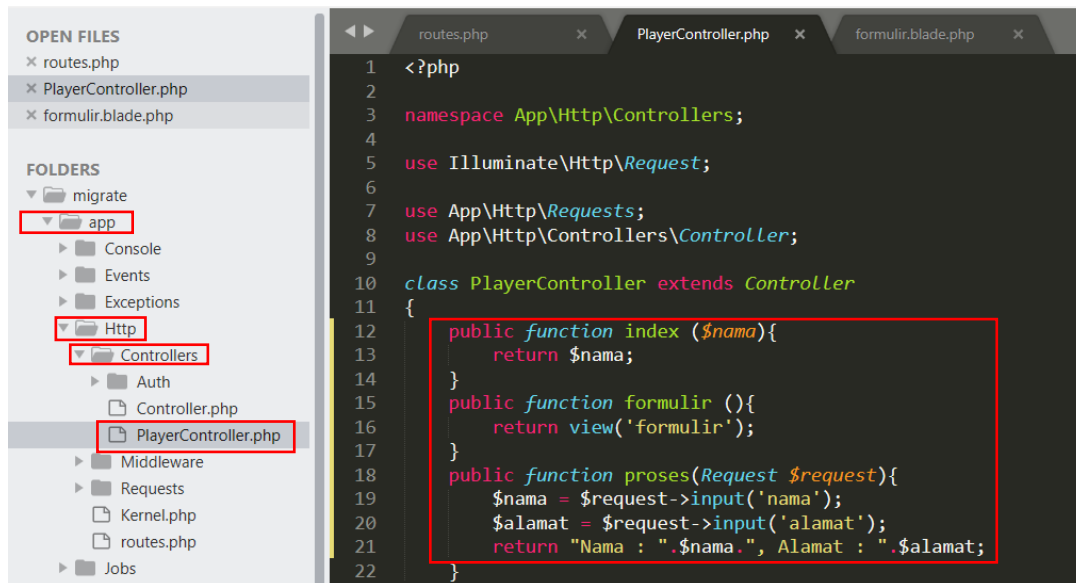
Dari alur proses diatas maka untuk percobaan kita kali ini yg perlu diseting adalah route nya, untuk itu tambahkan sebuah Route baru di project yang diberikan adapun kodingannya bisa dilihat pada sample dibawah ini:

```
18 Route::get('/formulir','PlayerController@formulir');  
19 Route::post('/formulir/proses','PlayerController@proses');
```

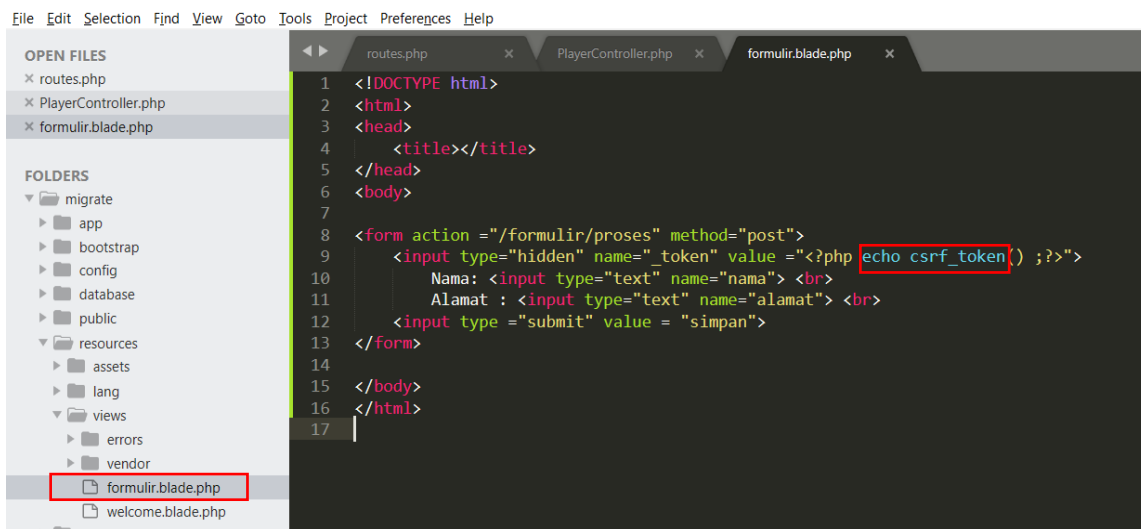
Dari script kita bisa lihat bahwa view nya adalah formulir dan controllernya adalah **PlayerController**. Kemudian dalam controller kita bisa lihat 2 fungsi yang dibutuhkan yaitu fungsi formulir dan proses. Jika sudah barulah pekerjaan kita lanjutkan dengan mengcreate sebuah Controller baru dengan menggunakan Git, yang kemaren sudah dilakukan instalasi, Controller yang dibuat tidak perlu lengkap untuk routenya karena dalam hal ini kita sendiri yang costumize controllernya sehingga perintah yang diketikkan dalam git bisa dilihat pada tampilan gambar dibawah ini:

```
Command Prompt  
Microsoft Windows [Version 10.0.18362.778]  
(c) 2019 Microsoft Corporation. All rights reserved.  
C:\Users\ROG>cd c:/xampp/htdocs/migrate  
c:\xampp\htdocs\migrate>php artisan make:controller PlayerController
```

Setelah proses controller berhasil kita buat dengan php artisan, lalu kita cek controller tersebut pada path foldernya **app/http/Controller/PlayerController**, tambahkan berapa fungsi yang menjadi penting karena dibutuhkan oleh proses route sebelumnya, yaitu fungsi formulir, dan fungsi proses. Seperti tampilan gambar dibawah ini:



Jika sudah itu artinya kita telah berhasil membuatnya dilanjutkan dengan mengcreate View nya, seperti yang kita ketahui bahwa di Laravel punya engine template sendiri yang kita sebut dengan Blade. Maka buatlah sebuah file baru dengan nama **formulir.blade.php** seperti tampilan gambar dibawah ini:



Yang menjadi nilai lebih dari laravel adalah dari segi kemanannya laravel sudah sediakan berapa fasilitas, sehingga sangat bagus, dalam hal ini kita coba menerapkan dengan penambahan sebuah file input bertipe hidden, yang valuenya diisi dengan **token csrf_token()**.

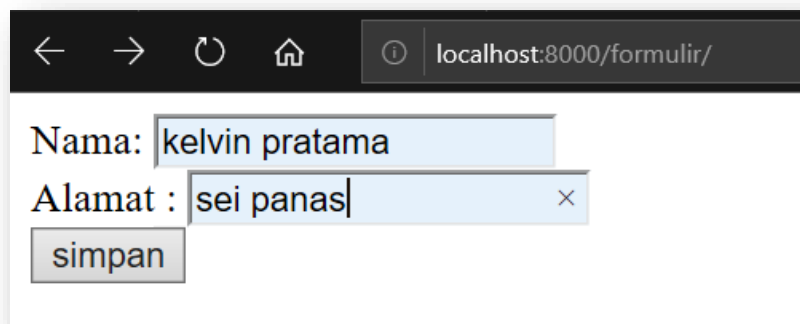
Kelvin Pratama
2018131037

Kemudian selanjutnya agar itu berfungsi ketikkan perintah di git sebagai berikut: **php artisan key:generate** jika success baru lakukan layanan laravel dengan perintah **php artisan serve** ingat semua perintah dilakukan pada folder project yang sudah dibangun sebelumnya. Seperti tampilan gambar dibawah ini

```
c:\xampp\htdocs\migrate>php artisan key:generate
Application key [IF6TgpG6lX7MHARSa2E8LaIMdeSUPch9] set successfully.

c:\xampp\htdocs\migrate>php artisan serve
Laravel development server started on http://localhost:8000/
```

Kemudian baru kita coba pengujian dengan mengetikkan **localhost:8000/formulir**, maka tampilan dari blade akan kita dapati seperti tampilan gambar dibawah ini:

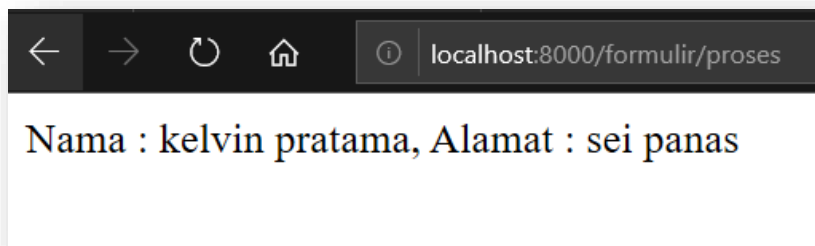


← → ↻ 🏠 ⓘ localhost:8000/formulir/

Nama: kelvin pratama

Alamat : sei panas ×

simpan



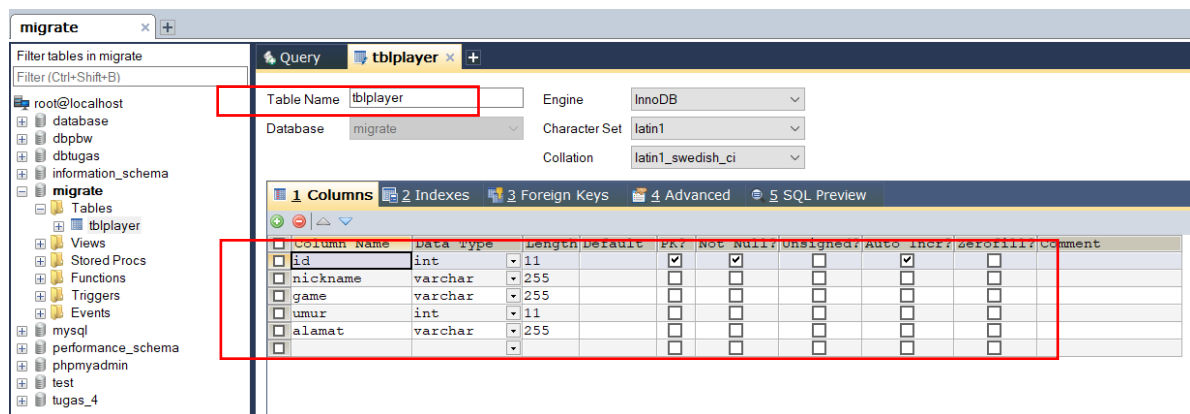
← → ↻ 🏠 ⓘ localhost:8000/formulir/proses

Nama : kelvin pratama, Alamat : sei panas

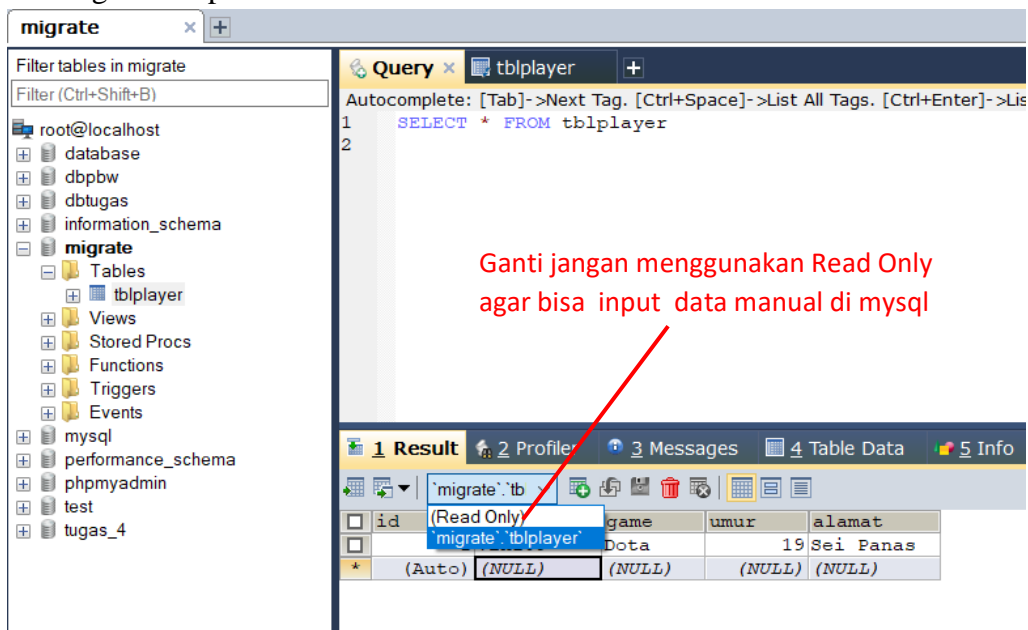
Menampilkan data dari Database

Untuk menampilkan data atau view merupakan sebuah basic yg perlu kita ketahui dalam konsep CRUD, dalam laravel hal ini bisa kita lakukan, salah satunya dengan mengikuti langkah langkah berikut:

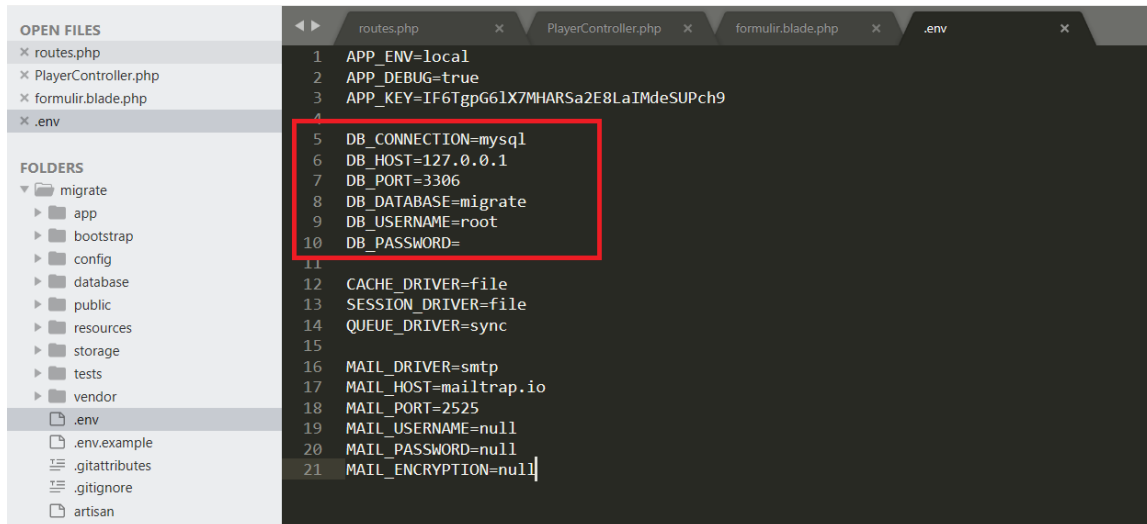
1. Buatlah sebuah database, di mysql jika sudah ada tidak perlu lagi dibuat tapi perlu dibuat sebuah table baru, untuk contoh kita buat **table tblplayer** lalu ketikkan field/kolom yang dibutuhkan dalam **tblplayer** ini yaitu: id, nickname, game, umur, alamat dengan ketentuan type data sebagai berikut: **Id**→**integer**, **nickname**→**varchar**, **game**→**varchar**, **umur**→**int**, **alamat**→**varchar**. Id merupakan primary key sehingga perlu dicentang primary key dan bikin auto increment.



2. Lalu tambahkan data secara manual, untuk table yang baru dibuat karena tugas kita saat ini adalah bagaimana proses view di laravel.



3. Jika sudah lanjutkan dengan proses setting ke databasenya. Seperti sebelumnya dijelaskan setingan database di laravel ada dua yg pertama **config/database.php** serta yang kedua di file **.env**. kali ini lakukan setingan di .env tambahkan informasi mengenai database yang anda buat didalam file **.env** nya

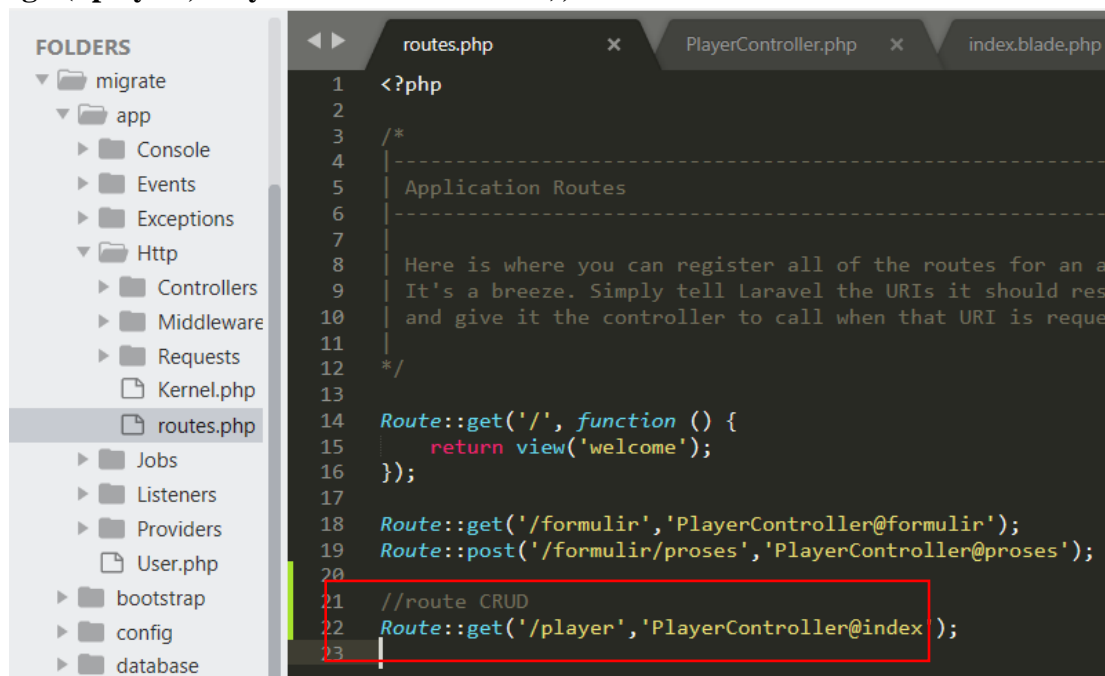


```
1 APP_ENV=local
2 APP_DEBUG=true
3 APP_KEY=IF6TgpG6lX7MHARsa2E8LaIMdeSUPch9
4
5 DB_CONNECTION=mysql
6 DB_HOST=127.0.0.1
7 DB_PORT=3306
8 DB_DATABASE=migrate
9 DB_USERNAME=root
10 DB_PASSWORD=
11
12 CACHE_DRIVER=file
13 SESSION_DRIVER=file
14 QUEUE_DRIVER=sync
15
16 MAIL_DRIVER=smtp
17 MAIL_HOST=mailtrap.io
18 MAIL_PORT=2525
19 MAIL_USERNAME=null
20 MAIL_PASSWORD=null
21 MAIL_ENCRYPTION=null
```

Lalu tambahkan Route baru untuk Project menampilkan data dari database ini seperti langkah berikut:


//route CRUD

Route::get('/player','PlayerController@index');



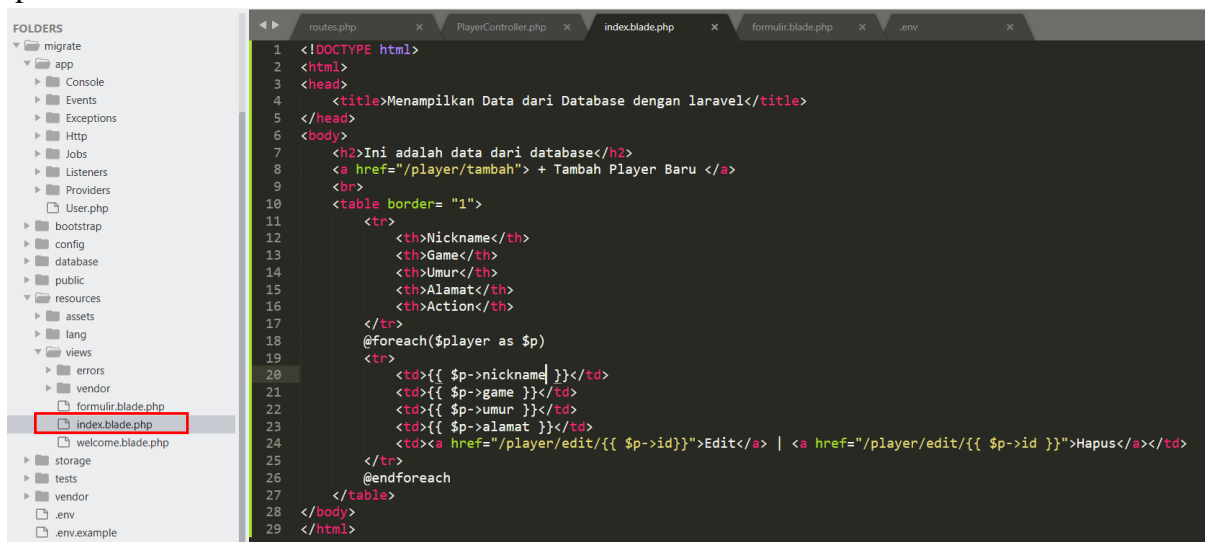
```
1 <?php
2
3 /*
4  |-----
5  | Application Routes
6  |-----
7  |
8  | Here is where you can register all of the routes for an ap
9  | It's a breeze. Simply tell Laravel the URIs it should resp
10 | and give it the controller to call when that URI is reques
11 |
12 | */
13
14 Route::get('/', function () {
15     return view('welcome');
16 });
17
18 Route::get('/formulir','PlayerController@formulir');
19 Route::post('/formulir/proses','PlayerController@proses');
20
21 //route CRUD
22 Route::get('/player','PlayerController@index ');
23
```

4. Selanjutnya lakukan penambahan atau modifikasi pada controller fungsi index nya seperti tampilan gambar dibawah ini: Note: catatan script percobaan sebelumnya di skip dulu karena kita makai controller yang sama tapi baru, yang ganti controller baru juga tidak masalah.



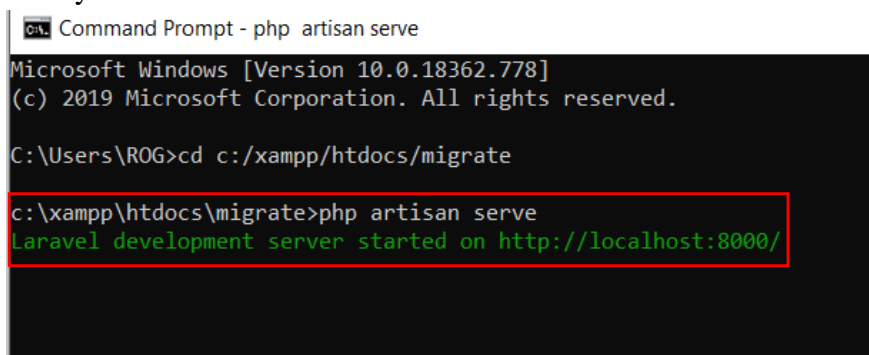
```
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6
7 use App\Http\Requests;
8 use App\Http\Controllers\Controller;
9 use DB;
10
11 class PlayerController extends Controller
12 {
13     public function index(){
14         //mengambil data dari tabel player
15         $player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

Jika sudah barulah view untuk tampilan kita buat, dari script bisa kita lihat bahwa view yang dibutuhkan adalah index, ini artinya kita buat file baru dengan nama **index.blade.php** seperti tampilan berikut:



```
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <title>Menampilkan Data dari Database dengan laravel</title>
5 </head>
6 <body>
7     <h2>Ini adalah data dari database</h2>
8     <a href="/player/tambah"> + Tambah Player Baru </a>
9     <br>
10    <table border= "1">
11        <tr>
12            <th>Nickname</th>
13            <th>Game</th>
14            <th>Umur</th>
15            <th>Alamat</th>
16            <th>Action</th>
17        </tr>
18        @foreach($player as $p)
19            <tr>
20                <td>{{ $p->nickname }}</td>
21                <td>{{ $p->game }}</td>
22                <td>{{ $p->umur }}</td>
23                <td>{{ $p->alamat }}</td>
24                <td><a href="/player/edit/{{ $p->id }}">Edit</a> | <a href="/player/edit/{{ $p->id }}">Hapus</a></td>
25            </tr>
26        @endforeach
27    </table>
28 </body>
29 </html>
```

Jika sudah baru dijalankan projectnya dengan perintah **php artisan serve** pada command git project kita sebelumnya



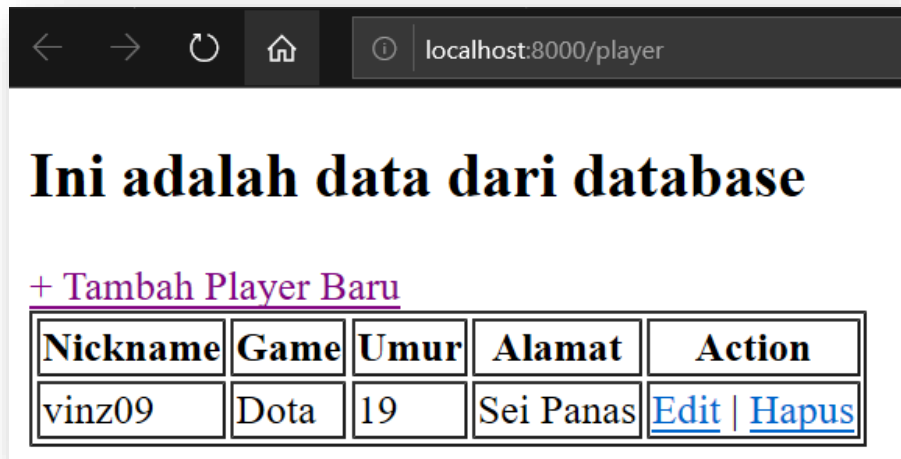
```
C:\> Command Prompt - php artisan serve
Microsoft Windows [Version 10.0.18362.778]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\ROG>cd c:/xampp/htdocs/migrate

c:\xampp\htdocs\migrate>php artisan serve
Laravel development server started on http://localhost:8000/
```

Kelvin Pratama
2018131037

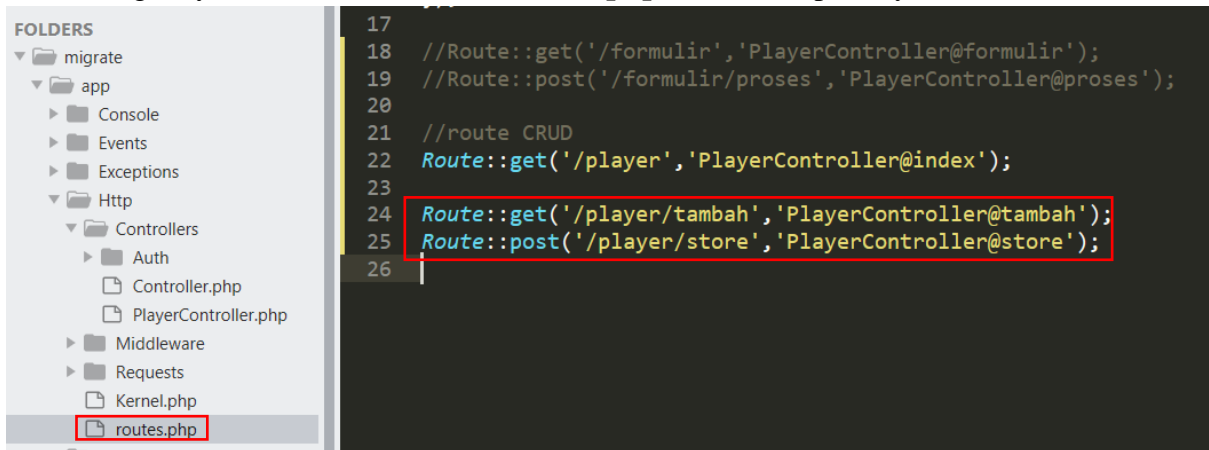
5. Jika selesai berate tugas kita menampilkan data dari database berhasil dilakukan sekarang buka di browser dengan url <http://localhost:8000/player>. Sehingga tampilan seperti tampilan berikut:



Menginput data Ke database

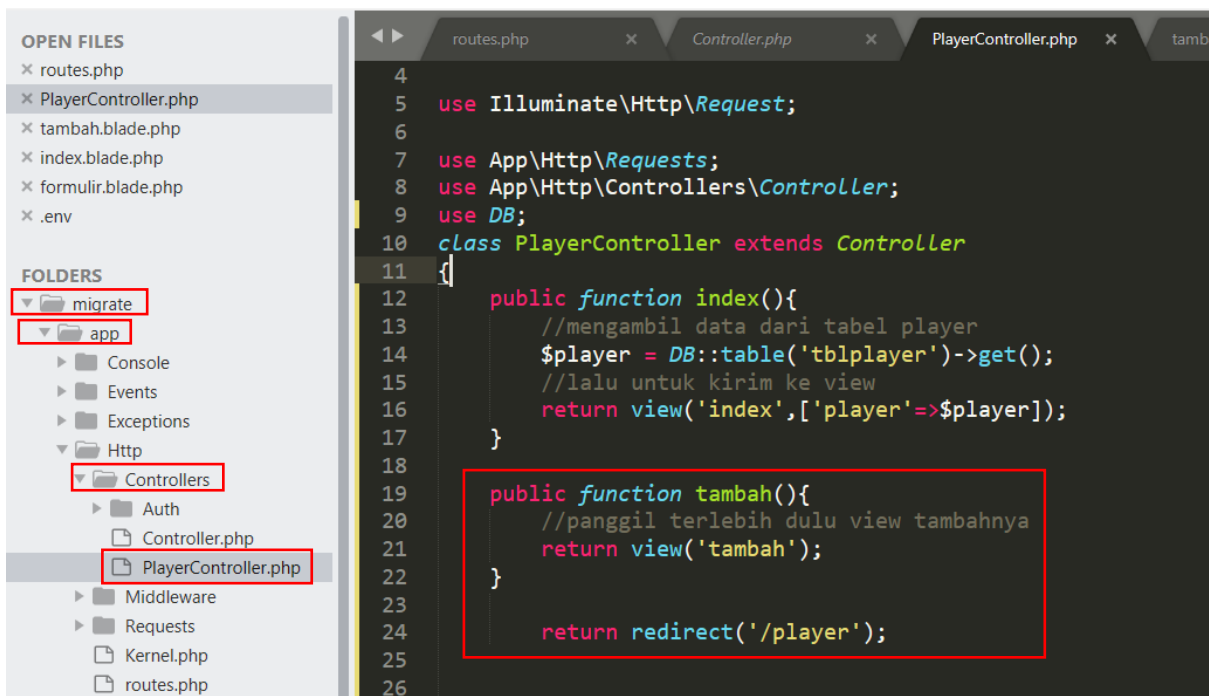
Selanjutnya setelah kita berhasil menampilkan data yang ada di database tahapan selanjutnya adalah proses menambahkan data ke dalam database yang ada, atau proses insert ke dalam database. Adapun langkah nya adalah sebagai berikut:

1. Buatlah sebuah route baru untuk proses tambah data, berikut list codingannya, list codingannya ini kita tambahkan di **routes.php** berikut tampilannya:




```
17
18 //Route::get('/formulir', 'PlayerController@formulir');
19 //Route::post('/formulir/proses', 'PlayerController@proses');
20
21 //route CRUD
22 Route::get('/player', 'PlayerController@index');
23
24 Route::get('/player/tambah', 'PlayerController@tambah');
25 Route::post('/player/store', 'PlayerController@store');
26
```

Selanjutnya lakukan modifikasi untuk controller **PlayerController**, seperti tampilan dibawah ini:



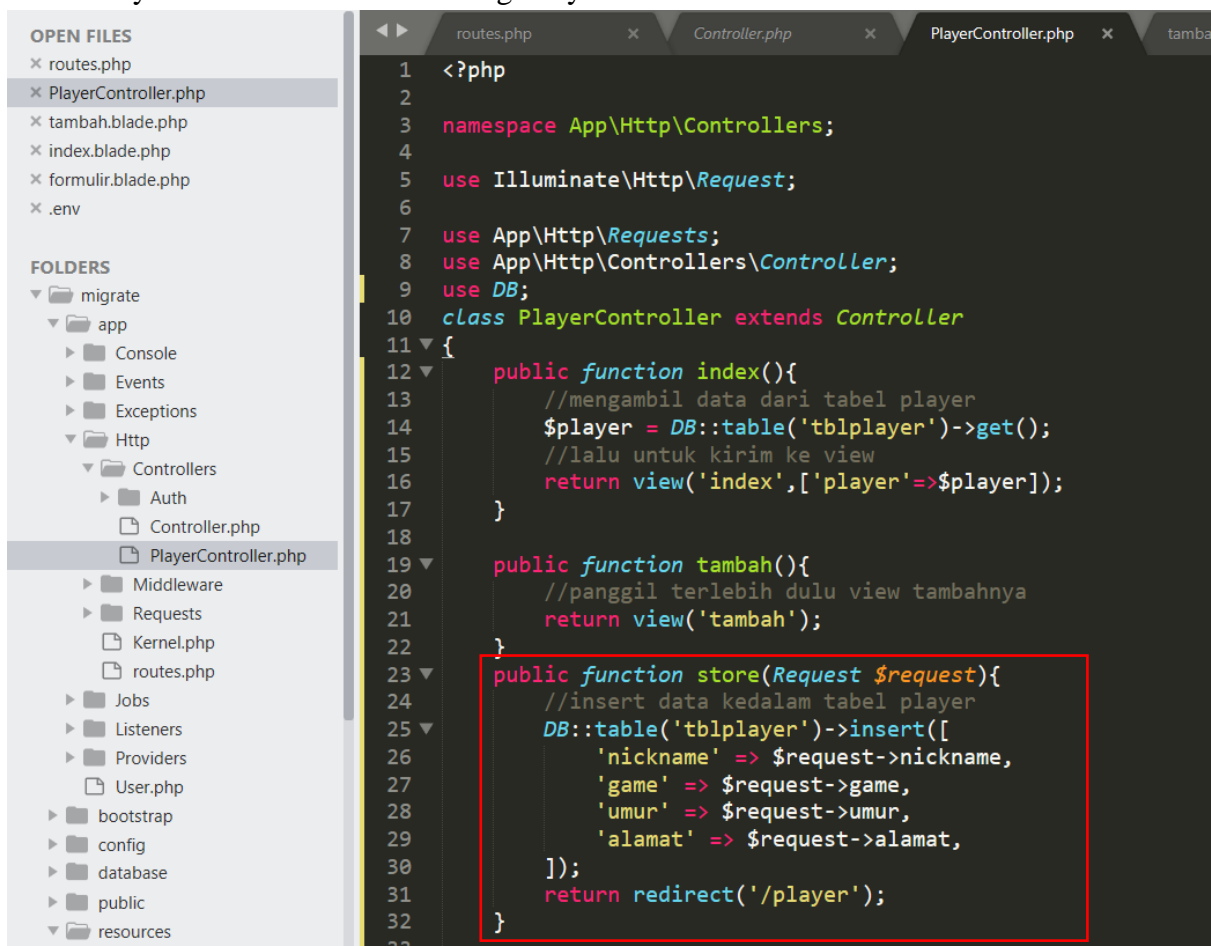
```
4
5 use Illuminate\Http\Request;
6
7 use App\Http\Requests;
8 use App\Http\Controllers\Controller;
9 use DB;
10 class PlayerController extends Controller
11 {
12     public function index(){
13         //mengambil data dari tabel player
14         $player = DB::table('tblplayer')->get();
15         //lalu untuk kirim ke view
16         return view('index', ['player'=>$player]);
17     }
18
19     public function tambah(){
20         //panggil terlebih dulu view tambahnya
21         return view('tambah');
22     }
23
24     return redirect('/player');
25
26
```


- Setelah controller ditambahkan proses selanjutnya adalah menambahkan view baru dengan membuat sebuah file blade yang kita namakan dengan **tambah.blade.php**, adapun codingannya bisa dilihat dari tampilan dibawah ini:



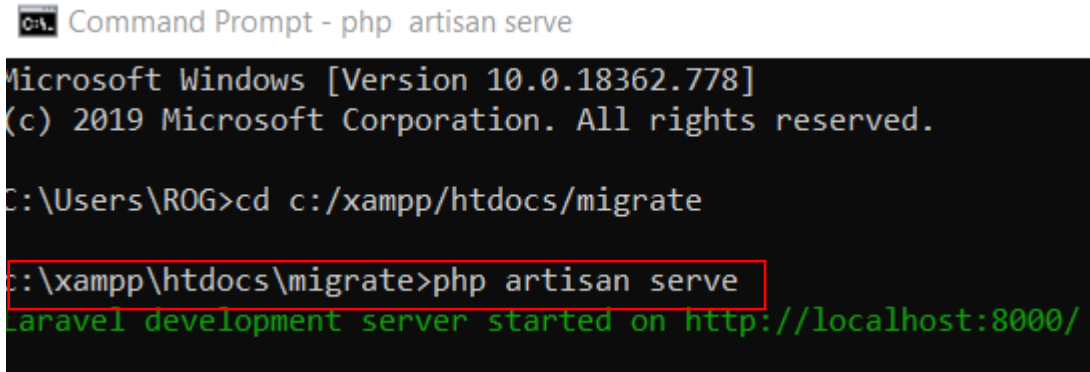
```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Menambah nama player baru kedalam Database</title>
5 </head>
6 <body>
7   <div class="container">
8     <h2>Input nama player baru</h2>
9     <a href="/player">Kembali</a>
10    <br>
11    <form action="/player/store" method="post">
12      {{ csrf_field() }}
13      Nickname <input type="text" name="nickname" required = "required"> <br>
14      Game <input type="text" name="game" required = "required"> <br>
15      umur <input type="text" name="umur" required = "required"> <br>
16      alamat <textarea name="alamat"></textarea> <br>
17      <input type="submit" value="Simpan Data">
18    </form>
19  </div>
20 </body>
21 </html>
```

- Setelah kita berhasil membuat file view tambah.blade.php barulah dilanjutkan dengan proses penambahan fungsi store pada controller, fungsi inilah yang bertugas untuk menambahkan datanya ke database berikut codingan nya



```
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6
7 use App\Http\Requests;
8 use App\Http\Controllers\Controller;
9 use DB;
10 class PlayerController extends Controller
11 {
12   public function index(){
13     //mengambil data dari tabel player
14     $player = DB::table('tblplayer')->get();
15     //lalu untuk kirim ke view
16     return view('index',['player'=>$player]);
17   }
18
19   public function tambah(){
20     //panggil terlebih dulu view tambahannya
21     return view('tambah');
22   }
23
24   public function store(Request $request){
25     //insert data kedalam tabel player
26     DB::table('tblplayer')->insert([
27       'nickname' => $request->nickname,
28       'game' => $request->game,
29       'umur' => $request->umur,
30       'alamat' => $request->alamat,
31     ]);
32     return redirect('/player');
33   }
34 }
```

Jika sudah baru dijalankan projectnya dengan perintah **php artisan serve** pada command git project kita sebelumnya



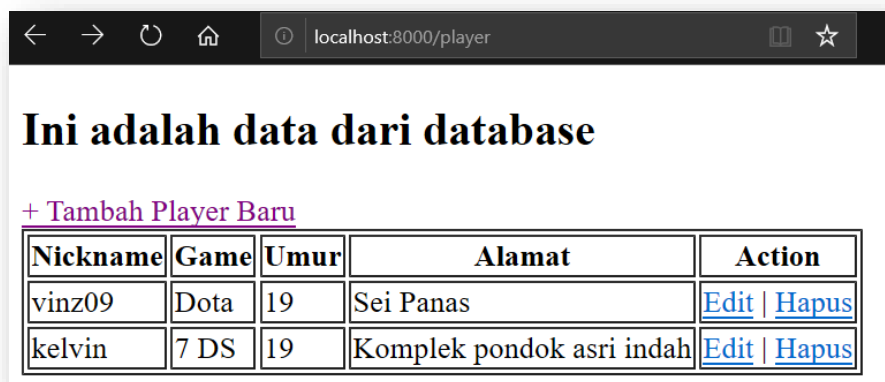
```
Command Prompt - php artisan serve

Microsoft Windows [Version 10.0.18362.778]
(c) 2019 Microsoft Corporation. All rights reserved.

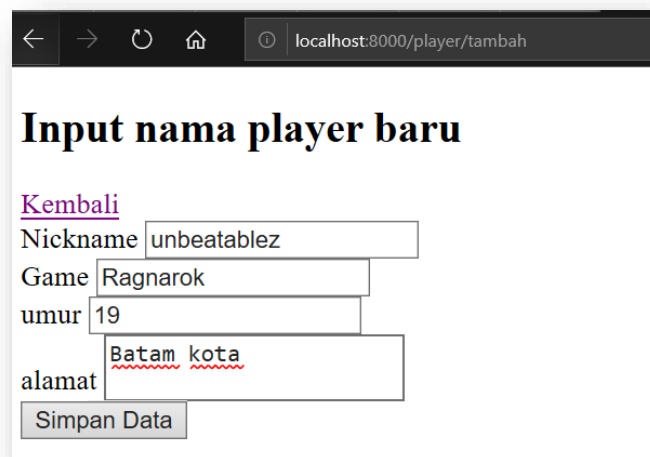
C:\Users\ROG>cd c:/xampp/htdocs/migrate

C:\xampp\htdocs\migrate>php artisan serve
Laravel development server started on http://localhost:8000/
```

4. Jika sudah baru kita coba jalan lagi dibrowser prosesnya, dengan cara buka url <http://localhost:8000/player> Seperti tampilan gambar dibawah ini:



Lalu pada klik pada bagian tambah player baru di atas sehingga browser akan memunculkan tampilan seperti berikut:



[Kembali](#)

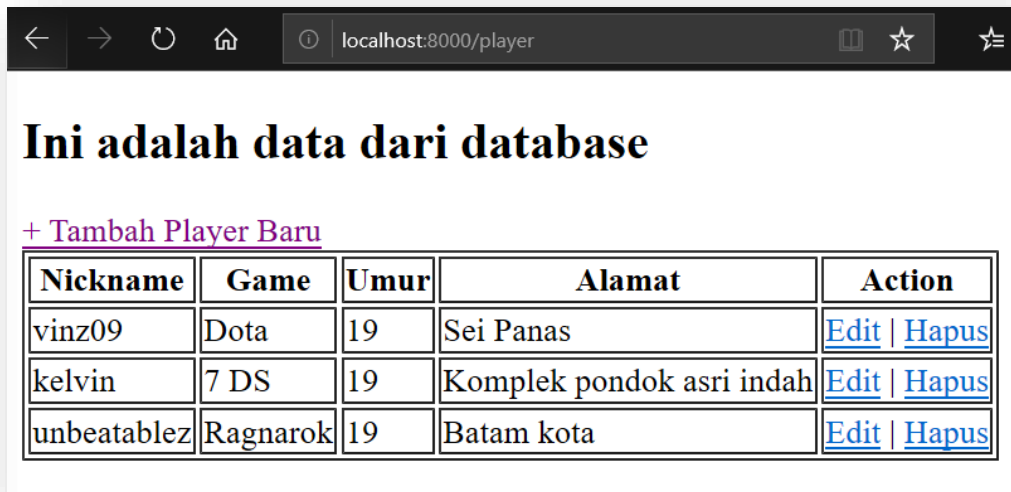
Nickname

Game

umur

alamat

5. Isilah form diatas dengan isian data setelah itu simpan data maka secara otomatis data akan ke insert ke dalam database dan list player baru akan ditampilkan juga seperti tampilan gambar dibawah ini:

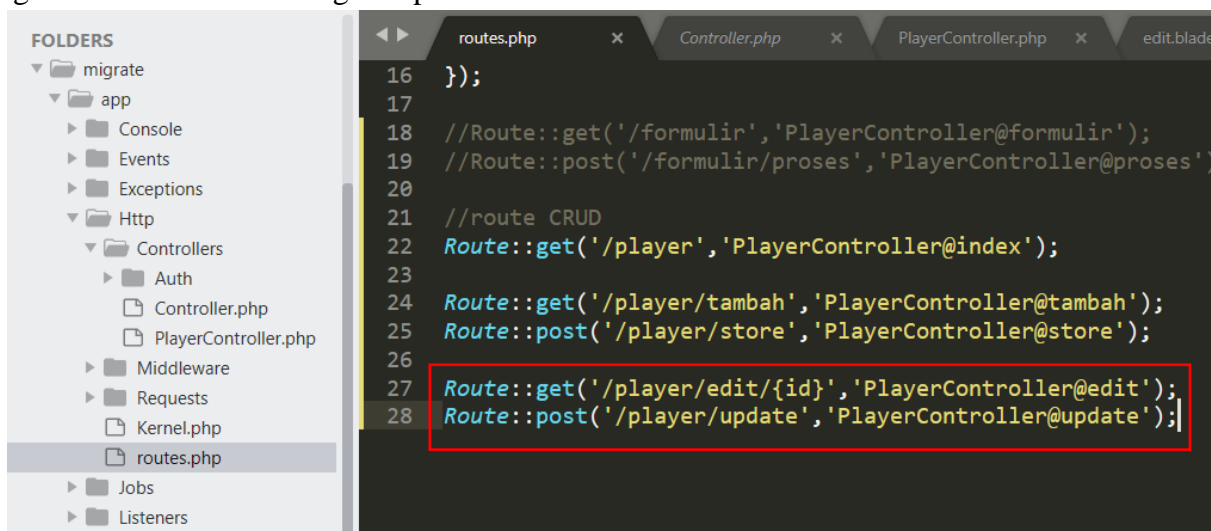


The screenshot shows a web browser window with the address bar displaying 'localhost:8000/player'. The page content includes a heading 'Ini adalah data dari database' and a link '+ Tambah Player Baru'. Below the link is a table with five columns: Nickname, Game, Umur, Alamat, and Action. The table contains three rows of data. Each row has 'Edit' and 'Hapus' links in the Action column.

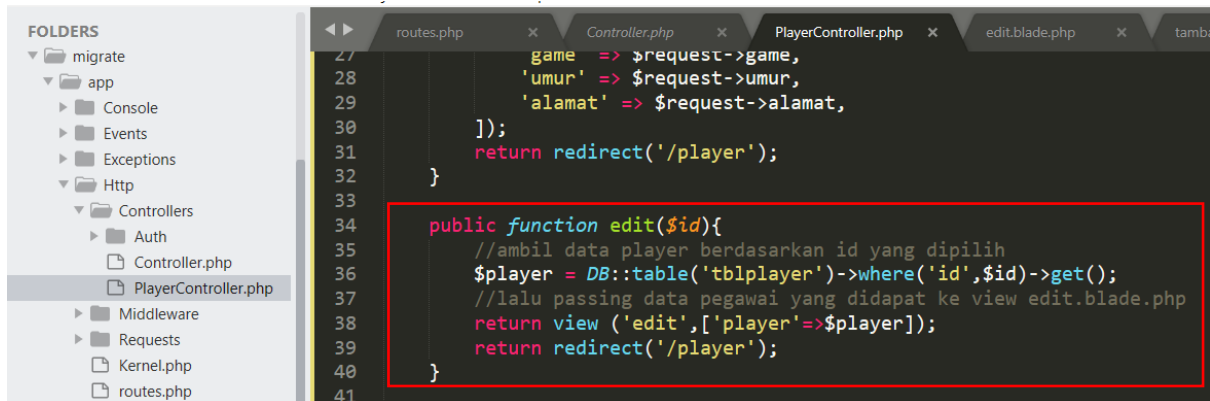
Nickname	Game	Umur	Alamat	Action
vinz09	Dota	19	Sei Panas	Edit Hapus
kelvin	7 DS	19	Komplek pondok asri indah	Edit Hapus
unbeatablez	Ragnarok	19	Batam kota	Edit Hapus

Edit data Mysql Dengan Laravel

Edit juga merupakan hal mendasar dari proses crud di laravel, langkah langkah untuk membentuk sebuah edit di laravel adalah sebagai berikut: Kembali lakukan perubahan di Route dengan menambahkan codingan seperti berikut:

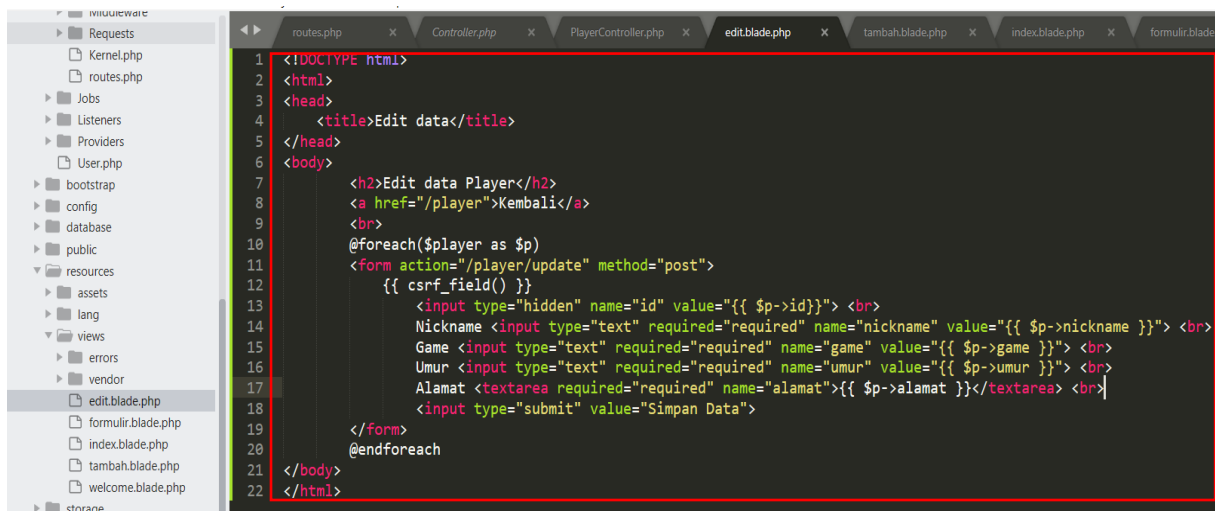


Setelah itu lakukan perubahan pada controller **PlayerController** seperti tampilan coding dibawah ini:



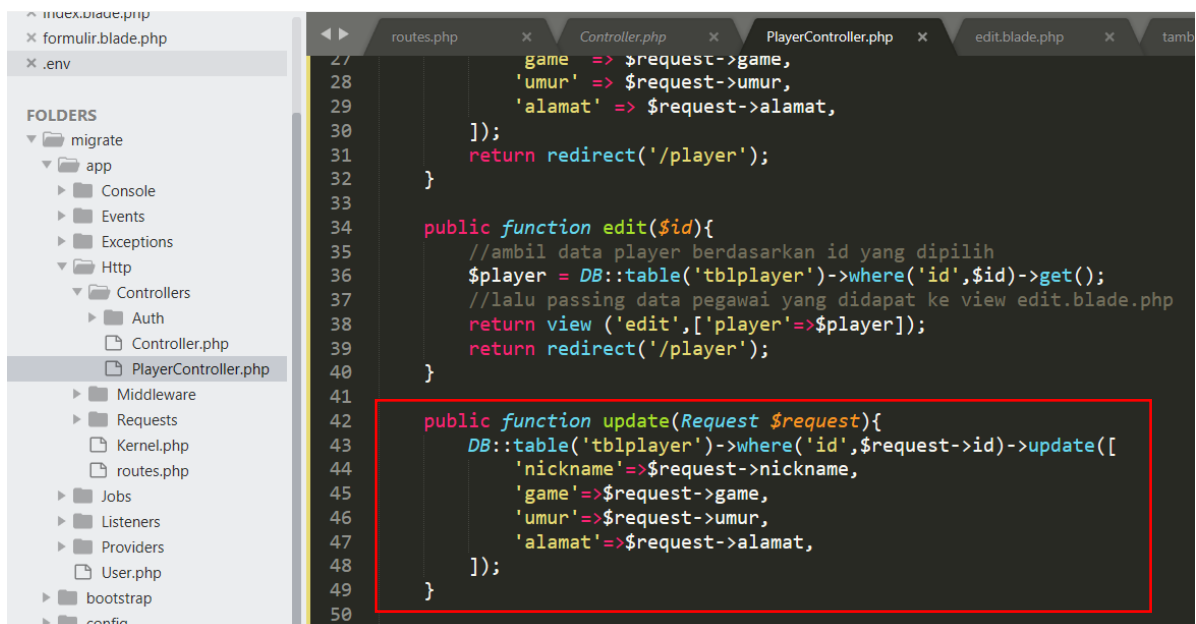
```
27     game => $request->game,
28     umur => $request->umur,
29     alamat => $request->alamat,
30 ];
31 return redirect('/player');
32 }
33
34 public function edit($id){
35     //ambil data player berdasarkan id yang dipilih
36     $player = DB::table('tblplayer')->where('id',$id)->get();
37     //lalu passing data pegawai yang didapat ke view edit.blade.php
38     return view ('edit',['player'=>$player]);
39     return redirect('/player');
40 }
41
```

Lanjutkan dengan proses membuat sebuah layout file edit, dengan menggunakan blade, yang kita namakan dengan **edit.blade.php** seperti tampilan dibawah ini:



```
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <title>Edit data</title>
5 </head>
6 <body>
7     <h2>Edit data Player</h2>
8     <a href="/player">Kembali</a>
9     <br>
10    @foreach($player as $p)
11        <form action="/player/update" method="post">
12            {{ csrf_field() }}
13            <input type="hidden" name="id" value="{{ $p->id }}"> <br>
14            Nickname <input type="text" required="required" name="nickname" value="{{ $p->nickname }}"> <br>
15            Game <input type="text" required="required" name="game" value="{{ $p->game }}"> <br>
16            Umur <input type="text" required="required" name="umur" value="{{ $p->umur }}"> <br>
17            Alamat <textarea required="required" name="alamat">{{ $p->alamat }}</textarea> <br>
18            <input type="submit" value="Simpan Data">
19        </form>
20    @endforeach
21 </body>
22 </html>
```

Lalu tambahkan codingan di controller untuk menjalankan fungsi update, seperti tampilan dibawah ini:



```
27     game => $request->game,
28     umur => $request->umur,
29     alamat => $request->alamat,
30 ];
31 return redirect('/player');
32 }
33
34 public function edit($id){
35     //ambil data player berdasarkan id yang dipilih
36     $player = DB::table('tblplayer')->where('id',$id)->get();
37     //lalu passing data pegawai yang didapat ke view edit.blade.php
38     return view ('edit',['player'=>$player]);
39     return redirect('/player');
40 }
41
42 public function update(Request $request){
43     DB::table('tblplayer')->where('id',$request->id)->update([
44         'nickname'=>$request->nickname,
45         'game'=>$request->game,
46         'umur'=>$request->umur,
47         'alamat'=>$request->alamat,
48     ]);
49 }
50
```

Kelvin Pratama
2018131037

Jika sudah baru jalankan lagi **php artisan serve** di git command

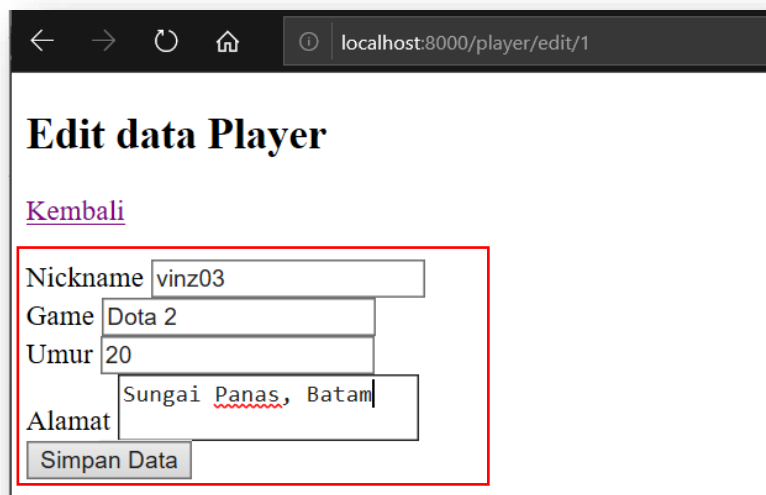
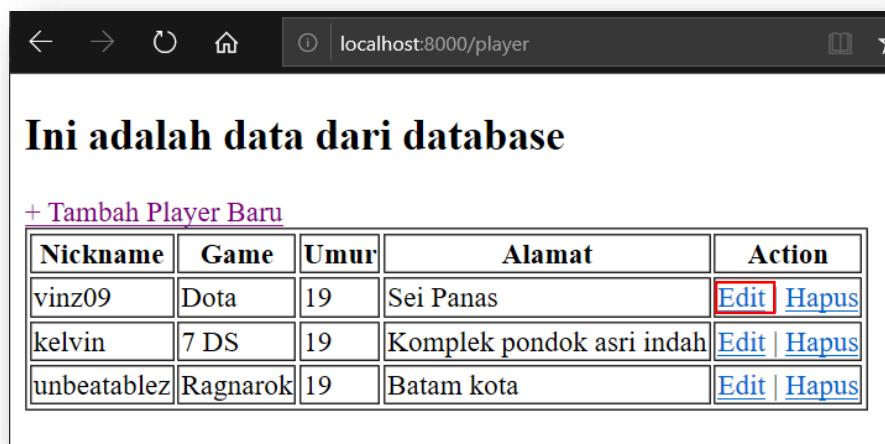
```
Command Prompt - php artisan serve

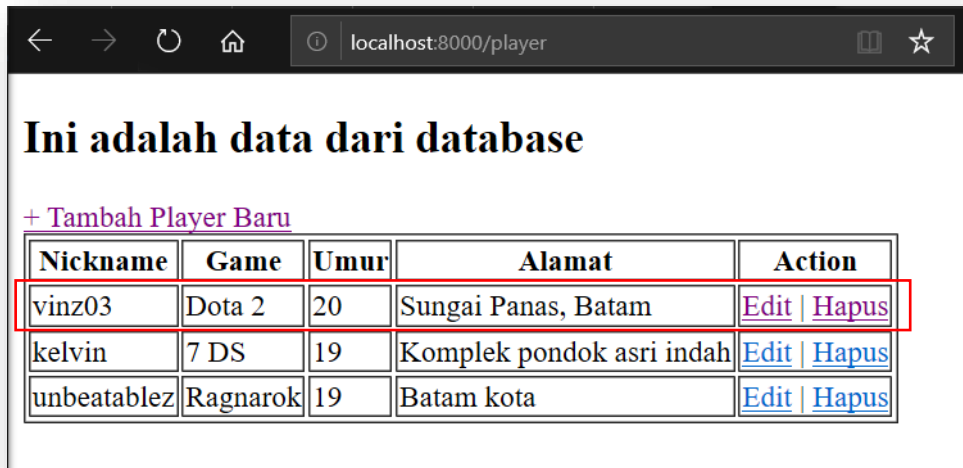
Microsoft Windows [Version 10.0.18362.778]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\ROG>cd c:/xampp/htdocs/migrate

C:\xampp\htdocs\migrate>php artisan serve
Laravel development server started on http://localhost:8000/
```

Lalu ujikan di browser dengan mengetikkan url: <http://localhost:8000/player> sehingga akan memunculkan tampilan sebagai berikut:



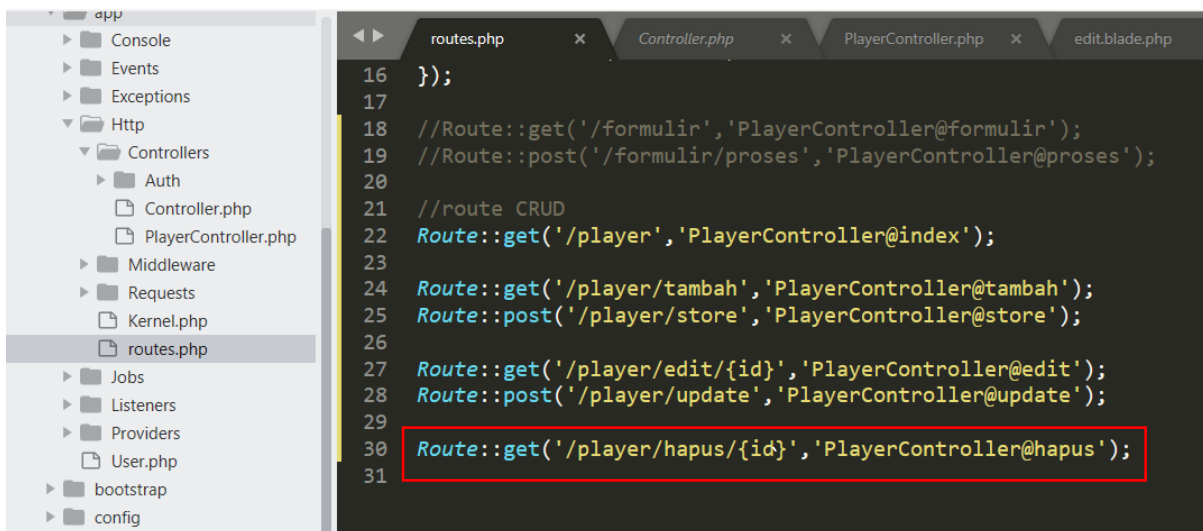


The screenshot shows a web browser window with the address bar at localhost:8000/player. The page title is "Ini adalah data dari database". Below the title is a link "+ Tambah Player Baru". A table displays player data with columns: Nickname, Game, Umur, Alamat, and Action. The first row is highlighted with a red border.

Nickname	Game	Umur	Alamat	Action
vinz03	Dota 2	20	Sungai Panas, Batam	Edit Hapus
kelvin	7 DS	19	Komplek pondok asri indah	Edit Hapus
unbeatablez	Ragnarok	19	Batam kota	Edit Hapus

Hapus data Mysql dengan Laravel

Delete juga bagian yang dasar dalam sebuah crud, berikut langkah langkah untuk melakukan proses delete di laravel. Langkah pertama adalah menambahkan route baru untuk mengatur Proses delete, dengan menambahkan codingan seperti tampilan dibawah ini:



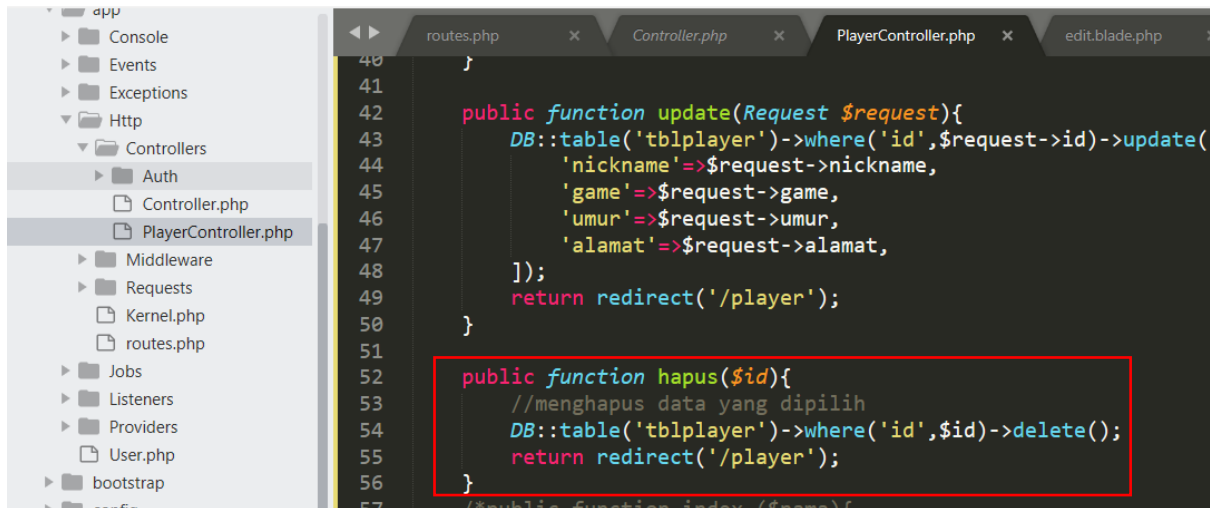
The screenshot shows a code editor with the file explorer on the left and the routes.php file open. The file explorer shows the following structure:

- app
 - Console
 - Events
 - Exceptions
 - Http
 - Controllers
 - Auth
 - Controller.php
 - PlayerController.php
 - Middleware
 - Requests
 - Kernel.php
 - routes.php
 - Jobs
 - Listeners
 - Providers
 - User.php
 - bootstrap
 - config

The routes.php file contains the following code:

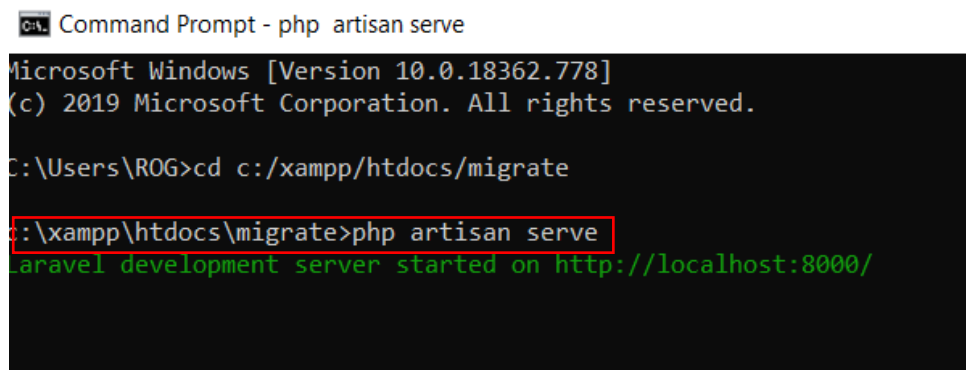
```
16 });
17
18 //Route::get('/formulir','PlayerController@formulir');
19 //Route::post('/formulir/proses','PlayerController@proses');
20
21 //route CRUD
22 Route::get('/player','PlayerController@index');
23
24 Route::get('/player/tambah','PlayerController@tambah');
25 Route::post('/player/store','PlayerController@store');
26
27 Route::get('/player/edit/{id}','PlayerController@edit');
28 Route::post('/player/update','PlayerController@update');
29
30 Route::get('/player/hapus/{id}','PlayerController@hapus');
31
```

Setelah Route untuk delete selesai dibuat, barulah kita menambahkan fungsi delete pada controller **PlayerController**, Seperti tampilan codingan dibawah ini:



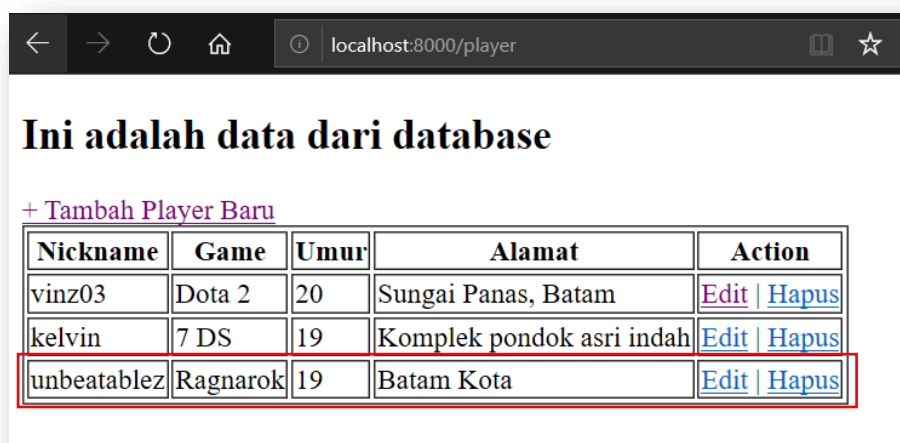
```
40  
41  
42 public function update(Request $request){  
43     DB::table('tblplayer')->where('id',$request->id)->update(  
44         'nickname'=>$request->nickname,  
45         'game'=>$request->game,  
46         'umur'=>$request->umur,  
47         'alamat'=>$request->alamat,  
48     );  
49     return redirect('/player');  
50 }  
51  
52 public function hapus($id){  
53     //menghapus data yang dipilih  
54     DB::table('tblplayer')->where('id',$id)->delete();  
55     return redirect('/player');  
56 }  
57  
58 /*public function index($nama){
```

Jika sudah baru jalankan lagi **php artisan serve** di git command



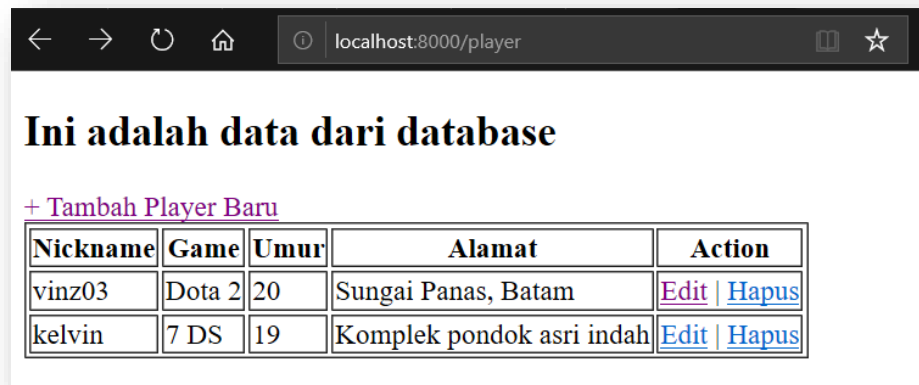
```
C:\> Command Prompt - php artisan serve  
  
Microsoft Windows [Version 10.0.18362.778]  
(c) 2019 Microsoft Corporation. All rights reserved.  
  
C:\Users\ROG>cd c:/xampp/htdocs/migrate  
  
C:\xampp\htdocs>php artisan serve  
Laravel development server started on http://localhost:8000/
```

Lalu buka di browser dengan url <http://localhost:8000/player>, sehingga tampilannya bisa dilihat pada tampilan berikut ini
Sebelum dihapus



Kelvin Pratama
2018131037

Setelah dihapus



TUGAS

Jika sudah berarti kita sudah berhasil melakukan proses crud dengan menggunakan Laravel, tugas Anda untuk membuat sebuah Crud dari awal dari proses **create table** dengan menggunakan **migrate**, dan menggunakan **bootstrap** untuk memperindah tampilan layoutnya.

1. Hasil dari create table database dengan **migrate**

The screenshot shows a database management interface. On the left, a tree view displays the database structure, with 'tblplayer' highlighted under the 'database' folder. The main panel shows the 'Query' editor with the SQL statement: `SELECT * FROM `tblplayer``. Below the query editor, the 'Result' tab displays the data from the 'tblplayer' table. The table has 8 columns: id, nickname, game, umur, jenis_kelamin, negara, created_at, and updated_at. The data is as follows:

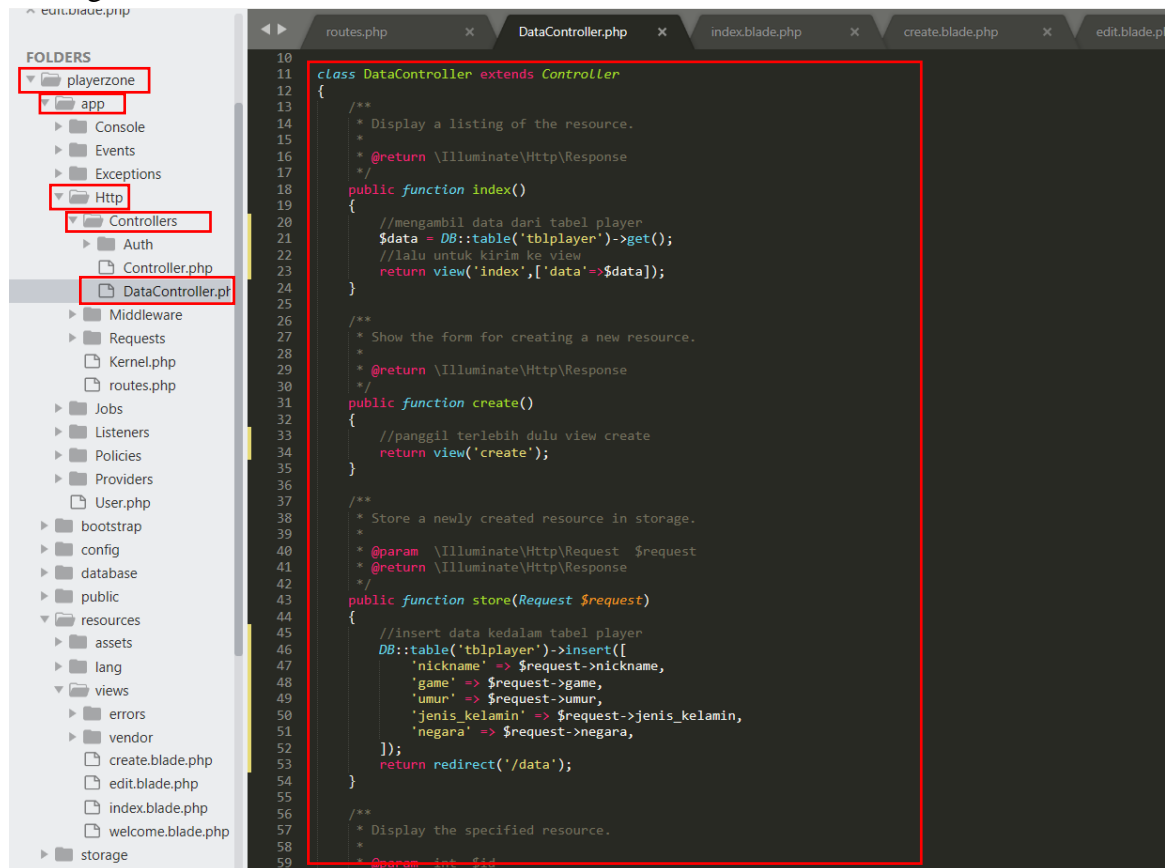
id	nickname	game	umur	jenis_kelamin	negara	created_at	updated_at
4	vinz03	Ragnarok	20	Laki-laki	Indonesia	0000-00-00 00:00:00	0000-00-00 00:00:00
5	Elizabeth	7 DS	19	Perempuan	Japan	0000-00-00 00:00:00	0000-00-00 00:00:00
6	Kelvinz09	Dota 2	19	Laki-laki	Indonesia	0000-00-00 00:00:00	0000-00-00 00:00:00

2. Codingan **route.php**

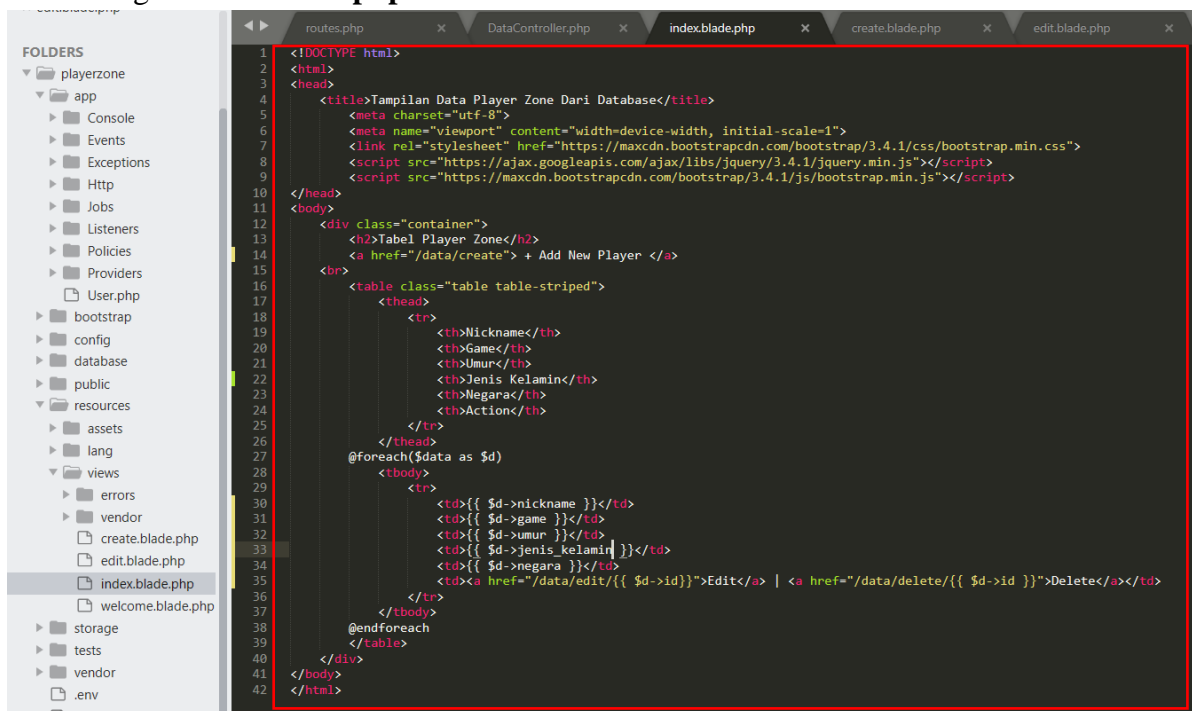
The screenshot shows a code editor with the 'routes.php' file open. The file contains the following code:

```
<?php
/*
 * Application Routes
 */
Here is where you can register all of the routes for an application.
It's a breeze. Simply tell Laravel the URIs it should respond to
and give it the controller to call when that URI is requested.
*/
Route::get('/', function () {
    return view('welcome');
});
Route::get('/data', 'DataController@index');
Route::get('/data/create', 'DataController@create');
Route::post('/data/store', 'DataController@store');
Route::get('/data/edit/{id}', 'DataController@edit');
Route::post('/data/update', 'DataController@update');
Route::get('/data/delete/{id}', 'DataController@delete');
```

3. Codingan DataController



4. Codingan index.blade.php



5. Codingan create.blade.php

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Penambahan Data Player Baru</title>
5 <meta charset="utf-8">
6 <meta name="viewport" content="width=device-width, initial-scale=1">
7 <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">
8 <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
9 <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>
10 </head>
11 <body>
12 <div class="container">
13 <h2>Create New Data Player</h2>
14 <a href="/data">Kembali</a>
15 <br>
16 <form action="/data/store" method="post">
17 <div class="form-group">
18 {{ csrf_field() }}
19 <label>Nickname: </label>
20 <input type="text" name="nickname" required="required" class="form-control"> <br>
21 <label>Game: </label>
22 <input type="text" name="game" required="required" class="form-control"> <br>
23 <label>Umur: </label>
24 <input type="text" name="umur" required="required" class="form-control"> <br>
25 <label>Jenis kelamin</label> <br>
26 <input class="form-check-input" type="radio" name="jenis_kelamin" value="Laki-laki" checked> Laki-Laki <br>
27 <input class="form-check-input" type="radio" name="jenis_kelamin" value="Perempuan"> Perempuan <br>
28 <label>Negara: </label>
29 <input type="text" name="negara" required="required" class="form-control"> <br>
30 <input type="submit" value="Save Data">
31 </div>
32 </form>
33 </div>
34 </body>
35 </html>
```

6. Codingan edit.blade.php

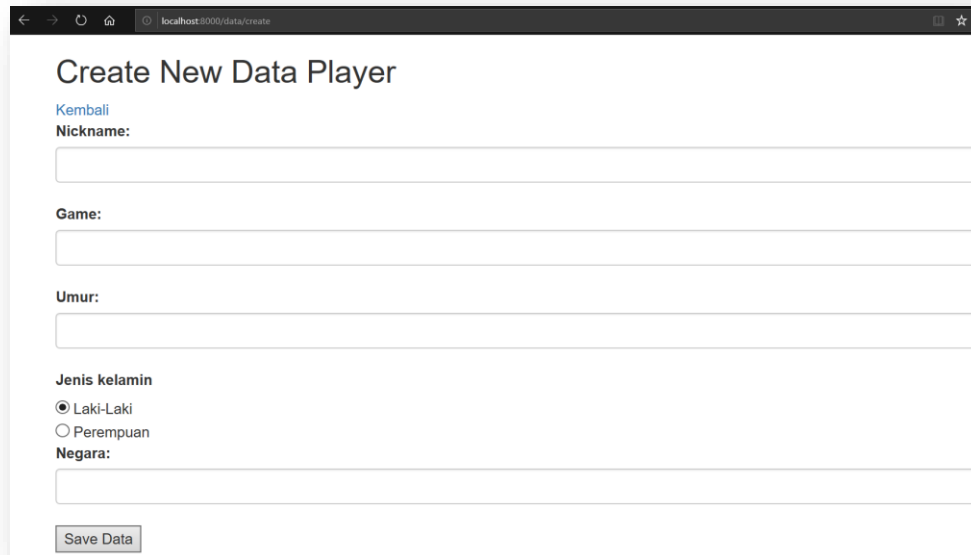
```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Pengeditan Data Player</title>
5 <meta charset="utf-8">
6 <meta name="viewport" content="width=device-width, initial-scale=1">
7 <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">
8 <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
9 <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>
10 </head>
11 <body>
12 <div class="container">
13 <h2>Edit Data Player</h2>
14 <a href="/data">Kembali</a>
15 <br>
16 @foreach($data as $d)
17 <form action="/data/update" method="post">
18 {{ csrf_field() }}
19 <div class="form-group">
20 <input type="hidden" name="id" value="{{ $d->id }}"> <br>
21 <label>Nickname: </label>
22 <input type="text" required="required" name="nickname" value="{{ $d->nickname }}" class="form-control"> <br>
23 <label>Game: </label>
24 <input type="text" required="required" name="game" value="{{ $d->game }}" class="form-control"> <br>
25 <label>Umur: </label>
26 <input type="text" required="required" name="umur" value="{{ $d->umur }}" class="form-control"> <br>
27 <label>Jenis kelamin</label> <br>
28 <input class="form-check-input" type="radio" required="required" name="jenis_kelamin" value="Laki-laki" checked> Laki-Laki <br>
29 <input class="form-check-input" type="radio" required="required" name="jenis_kelamin" value="Perempuan"> Perempuan <br>
30 <label>Negara: </label>
31 <input type="text" required="required" name="negara" value="{{ $d->negara }}" class="form-control"> <br>
32 <input type="submit" value="Save Data">
33 </div>
34 </form>
35 @endforeach
36 </div>
37 </body>
38 </html>
```

7. Tampilan Table Player Zone

localhost:8000/data

Tabel Player Zone					
+ Add New Player					
Nickname	Game	Umur	Jenis Kelamin	Negara	Action
vinz03	Ragnarok	20	Laki-laki	Indonesia	Edit Delete
Elizabeth	7 DS	19	Perempuan	Japan	Edit Delete
Kelvinz09	Dota 2	19	Laki-laki	Indonesia	Edit Delete

8. Tampilan **Create New Player**



A screenshot of a web browser showing a form titled "Create New Data Player". The browser's address bar displays "localhost:8000/data/create". The form includes a blue link "Kembali" at the top left. Below it are input fields for "Nickname:", "Game:", and "Umur:". There is a section for "Jenis kelamin" with radio buttons for "Laki-Laki" (selected) and "Perempuan". Below that is an input field for "Negara:". At the bottom left of the form is a "Save Data" button.

9. Tampilan **Edit Data Player**



A screenshot of a web browser showing a form titled "Edit Data Player". The browser's address bar displays "localhost:8000/data/edit/4". The form includes a blue link "Kembali" at the top left. Below it are input fields for "Nickname:" (containing "vinz03"), "Game:" (containing "Ragnarok"), and "Umur:" (containing "20"). There is a section for "Jenis kelamin" with radio buttons for "Laki-Laki" (selected) and "Perempuan". Below that is an input field for "Negara:" (containing "Indonesia"). At the bottom left of the form is a "Save Data" button.

Note: Untuk lebih jelasnya dapat dilihat pada codingan yang telah dibuat.