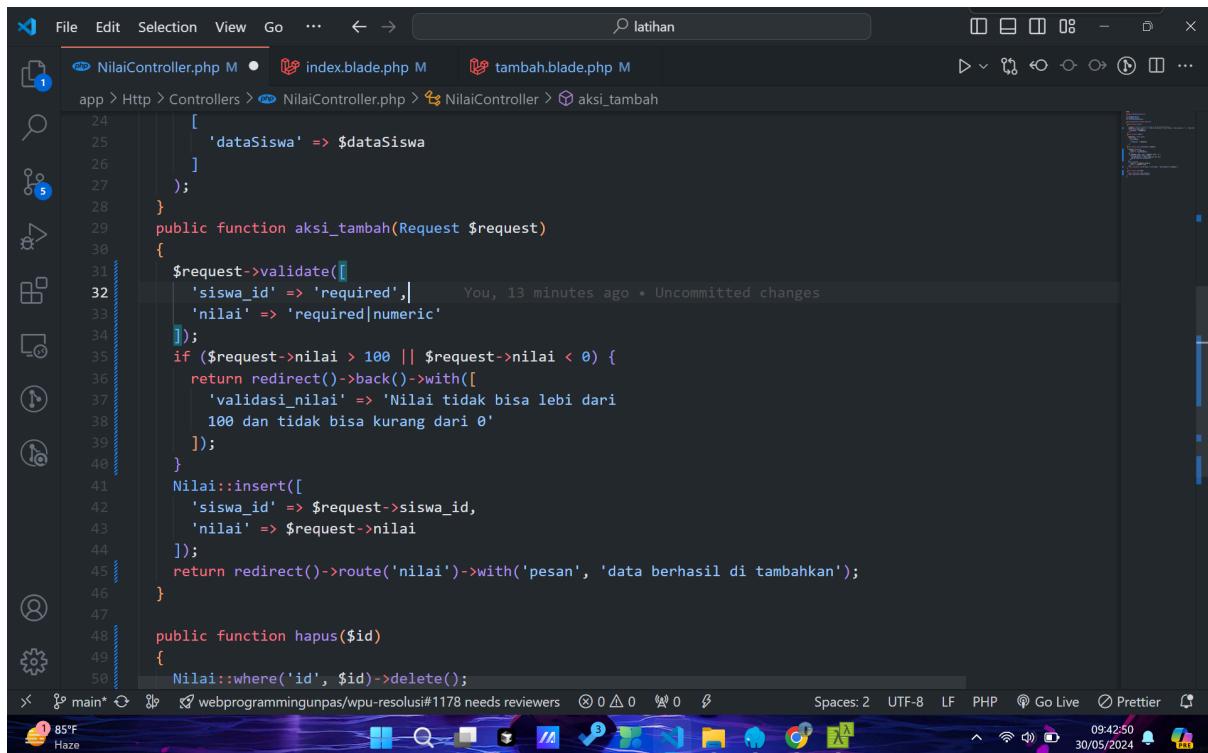


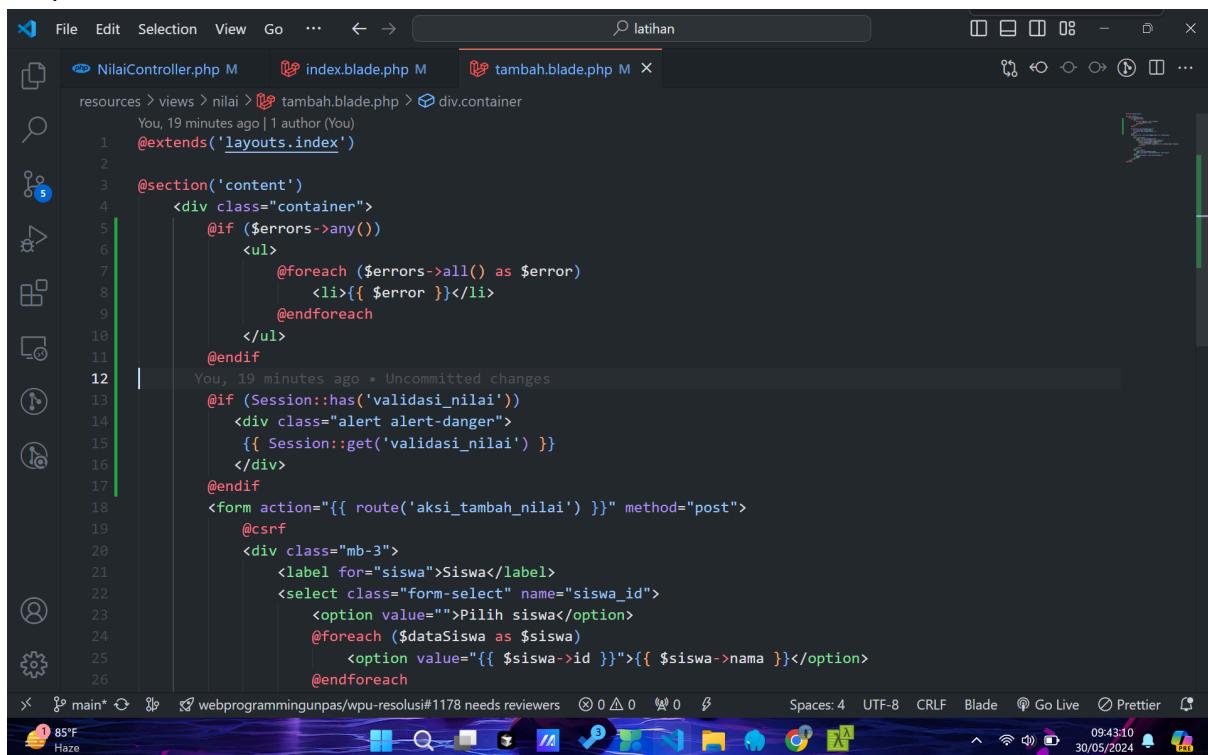
Membuat validasi tambah nilai

ubah fungsi tambah nilai jadi seperti ini



```
File Edit Selection View Go ... ← → latihan
NilaiController.php M index.blade.php M tambah.blade.php M
app > Http > Controllers > NilaiController.php > aksi_tambah
24     [
25         'dataSiswa' => $dataSiswa
26     ];
27 );
28 }
29 public function aksi_tambah(Request $request)
30 {
31     $request->validate([
32         'siswa_id' => 'required',
33         'nilai' => 'required|numeric'
34     ]);
35     if ($request->nilai > 100 || $request->nilai < 0) {
36         return redirect()->back()->with([
37             'validasi_nilai' => 'Nilai tidak bisa lebih dari
38             100 dan tidak bisa kurang dari 0'
39         ]);
40     }
41     Nilai::insert([
42         'siswa_id' => $request->siswa_id,
43         'nilai' => $request->nilai
44     ]);
45     return redirect()->route('nilai')->with('pesan', 'data berhasil ditambahkan');
46 }
47
48 public function hapus($id)
49 {
50     Nilai::where('id', $id)->delete();
51 }
```

tampilkan error validasi nilai ketika tambah data



```
File Edit Selection View Go ... latihan
NilaiController.php M index.blade.php M tambah.blade.php M
resources > views > nilai > tambah.blade.php > div.container
1 @extends('_layouts.index')
2
3 @section('content')
4     <div class="container">
5         @if ($errors->any())
6             <ul>
7                 @foreach ($errors->all() as $error)
8                     <li>{{ $error }}</li>
9                 @endforeach
10            </ul>
11        @endif
12        You, 19 minutes ago * Uncommitted changes
13        @if (Session::has('validasi_nilai'))
14            <div class="alert alert-danger">
15                {{ Session::get('validasi_nilai') }}
16            </div>
17        @endif
18        <form action="{{ route('aksi_tambah_nilai') }}" method="post">
19            @csrf
20            <div class="mb-3">
21                <label for="siswa">Siswa</label>
22                <select class="form-select" name="siswa_id">
23                    <option value="">Pilih siswa</option>
24                    @foreach ($dataSiswa as $siswa)
25                        <option value="{{ $siswa->id }}>{{ $siswa->nama }}</option>
26                    @endforeach
27                </select>
28            </div>
29        </form>
30    </div>
31
```

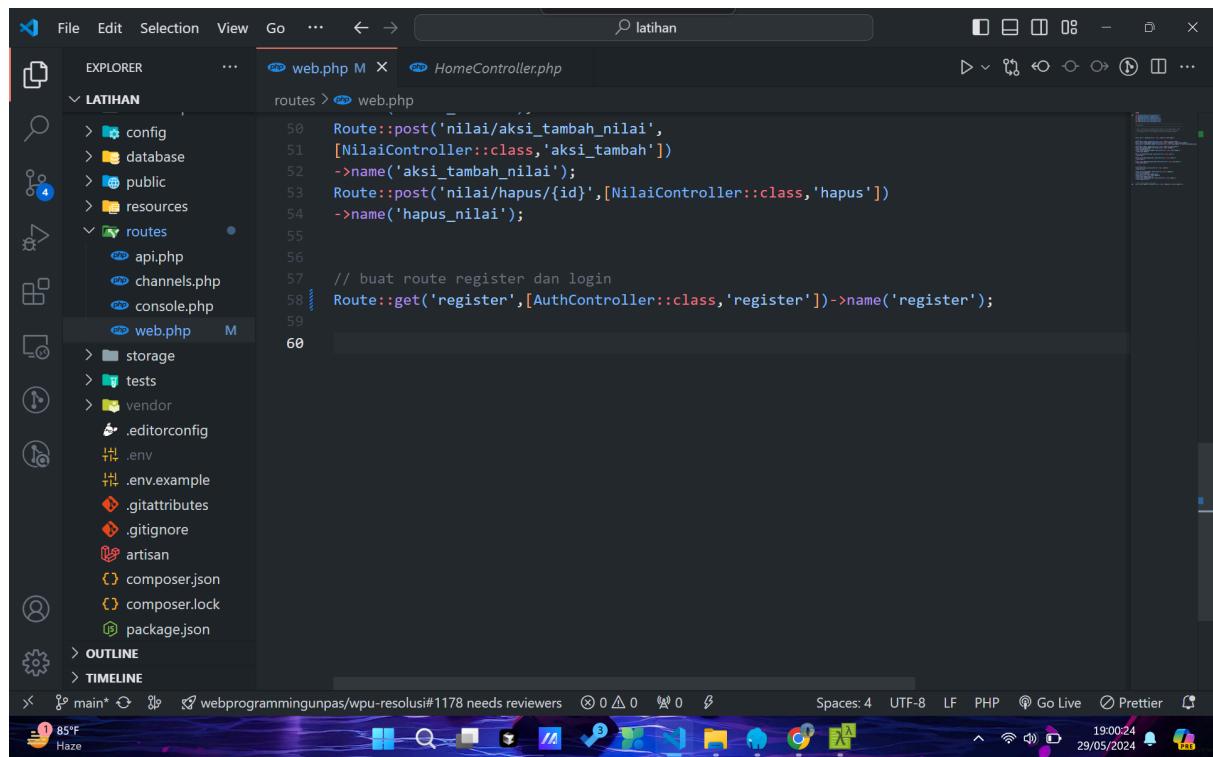
edit data nilai

buat authentication di laravel

pertama buat controller dan model untuk register dan login

buat controller nya dengan menjalankan **php artisan make:controller AuthController** dan **php artisan make:model Pengguna**

kemudian buat route register nya

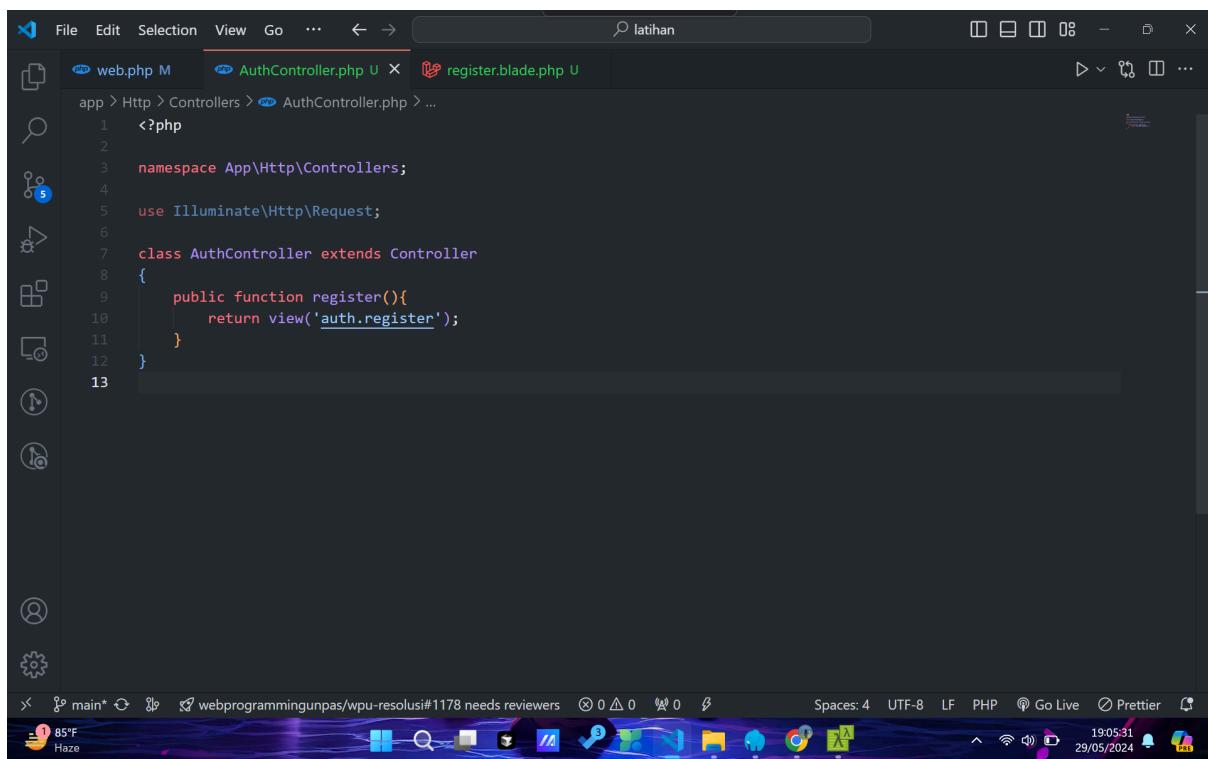


```
File Edit Selection View Go ... ← → latihan
EXPLORER routes > web.php
LATIHAN routes > web.php
routes > web.php
50 Route::post('nilai/aksi_tambah_nilai',
51 [NilaiController::class, 'aksi_tambah']);
52 ->name('aksi_tambah_nilai');
53 Route::post('nilai/hapus/{id}',[NilaiController::class,'hapus']);
54 ->name('hapus_nilai');
55
56
57 // buat route register dan login
58 Route::get('register',[AuthController::class,'register'])->name('register');
59
60
```

The screenshot shows a code editor with the routes/web.php file open. A new route for registration has been added at line 58:

```
Route::get('register',[AuthController::class,'register'])->name('register');
```

lalu buat fungsi/method untuk register



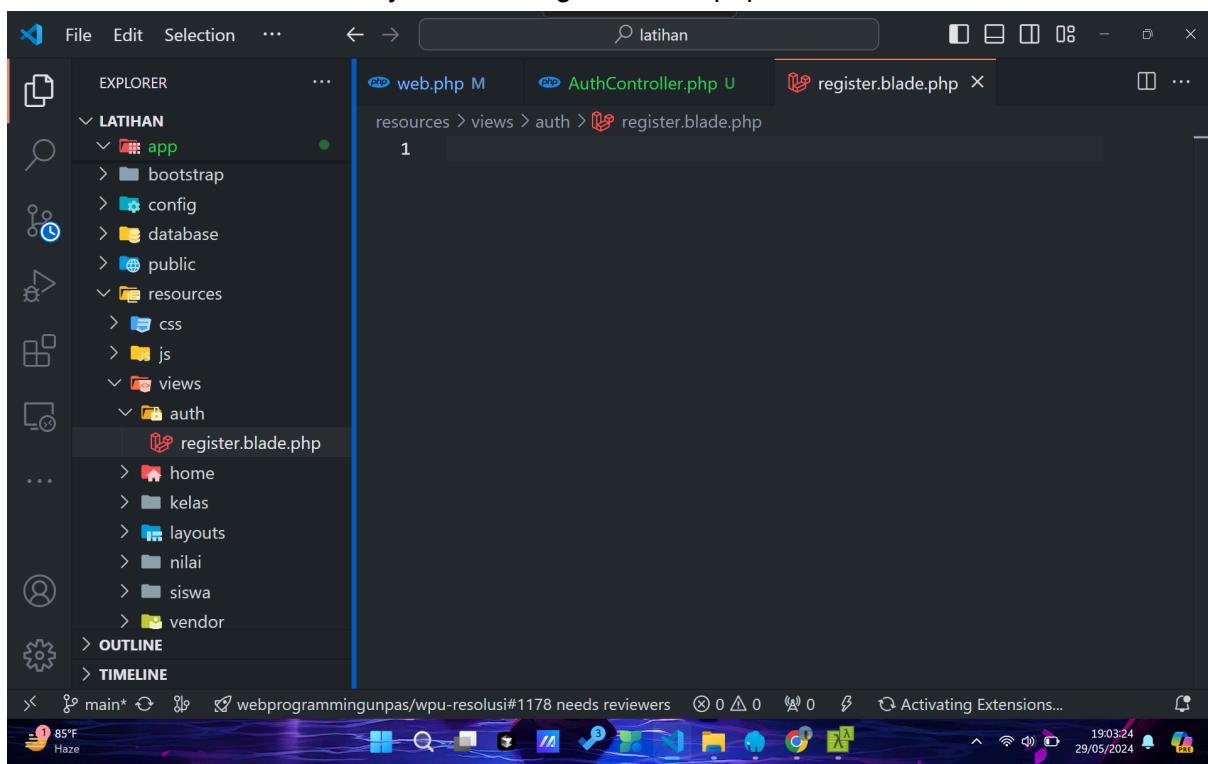
The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, ...
- Search Bar:** latihan
- Editor:** Three tabs are open: web.php (M), AuthController.php (U), and register.blade.php (U). The AuthController.php tab is active, displaying the following code:

```
<?php  
namespace App\Http\Controllers;  
use Illuminate\Http\Request;  
class AuthController extends Controller  
{  
    public function register()  
    {  
        return view('auth.register');  
    }  
}
```

The sidebar on the left contains icons for file operations like Open, Save, Find, and Refresh.

lalu buat folder auth didalamnya ada file register.blade.php



dan buat form html seperti ini dan arahkan action nya ke route **aksi_register**

The screenshot shows a code editor with three tabs open: `web.php`, `AuthController.php`, and `register.blade.php`. The `register.blade.php` tab contains the following Blade template code:

```
<head>
    <title>Bootstrap demo</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet"
        integrity="sha384-QWTKyjipFjTSv5wRl9Ferpk6YctnYmDr5pNyT2bRjXh0MhjY6hH+ALewIH" crossorigin="anonymous">
</head>
<body>
    <div class="container mt-5">
        <div class="row justify-content-center">
            <div class="col-lg-4">
                <h3>Register</h3>
                <form action="{{ route('aksi_register') }}" method="post">
                    @csrf
                    <div class="mb-3 row">
                        <label for="nama">Nama</label>
                        <div class="col-sm-10">
                            <input type="text" class="form-control" name="nama" placeholder="Masukan nama">
                        </div>
                    </div>
                    <div class="mb-3 row">
                        <label for="staticEmail">Email</label>
                        <div class="col-sm-10">
                            <input type="text" class="form-control" name="email" placeholder="Masukan email">
                        </div>
                    </div>
                    <div class="mb-3 row">
                        <label for="inputPassword">Password</label>
                        <div class="col-sm-10">
                            <input type="password" class="form-control" name="password" placeholder="masukan password">
                        </div>
                    </div>
                    <div class="row">
                        <button class="btn btn-primary">Register</button>
                    </div>
                </form>
            </div>
        </div>
    </div>
</body>
```

The status bar at the bottom shows the file is 18 lines long, 26 columns wide, and uses UTF-8 encoding. It also indicates there are 4 spaces, 0 tabs, and 0 line breaks.

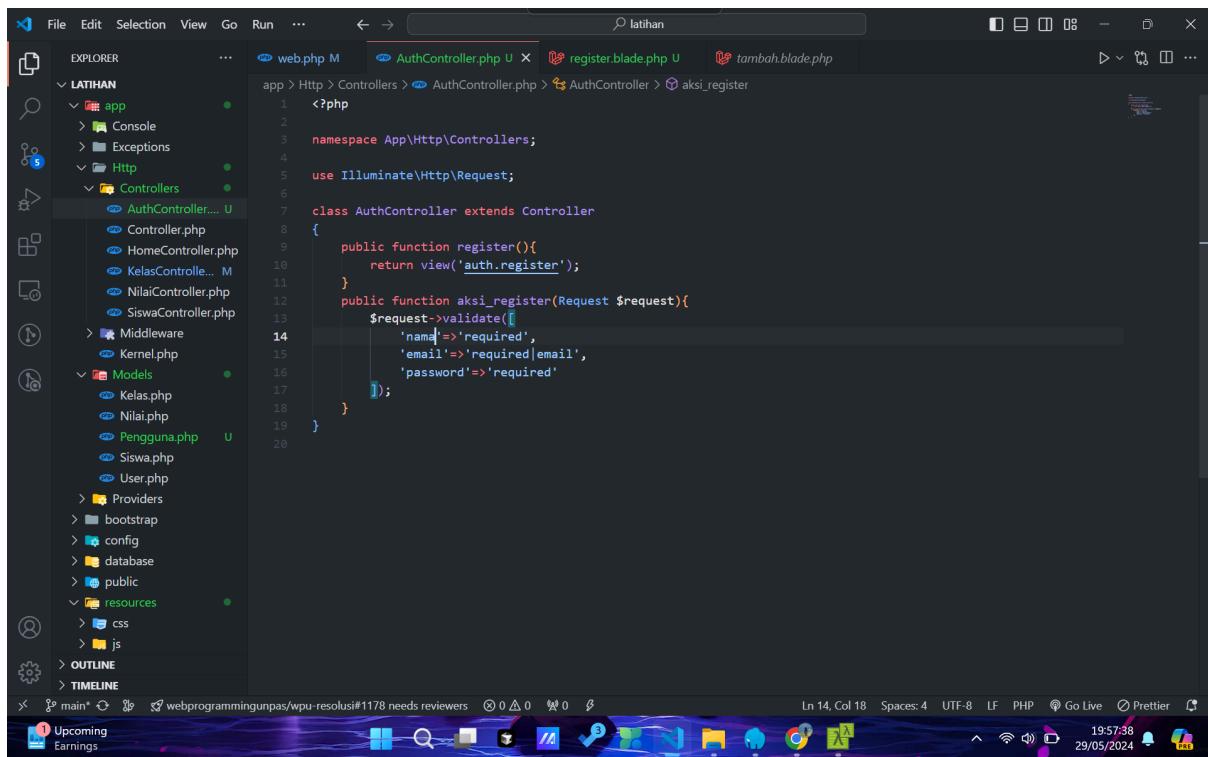
selanjutnya buat route untuk fungsi register nya

The screenshot shows a code editor with three tabs open: `web.php`, `AuthController.php`, and `register.blade.php`. The `web.php` tab contains the following route definition:

```
Route::get('register',[AuthController::class,'register'])->name('register');
Route::post('register',[AuthController::class,'aksi_register'])->name('aksi_register');
```

The status bar at the bottom shows the file is 59 lines long, 88 columns wide, and uses UTF-8 encoding. It also indicates there are 4 spaces, 0 tabs, and 0 line breaks.

kemudian tambahkan validasi untuk fungsi register nya



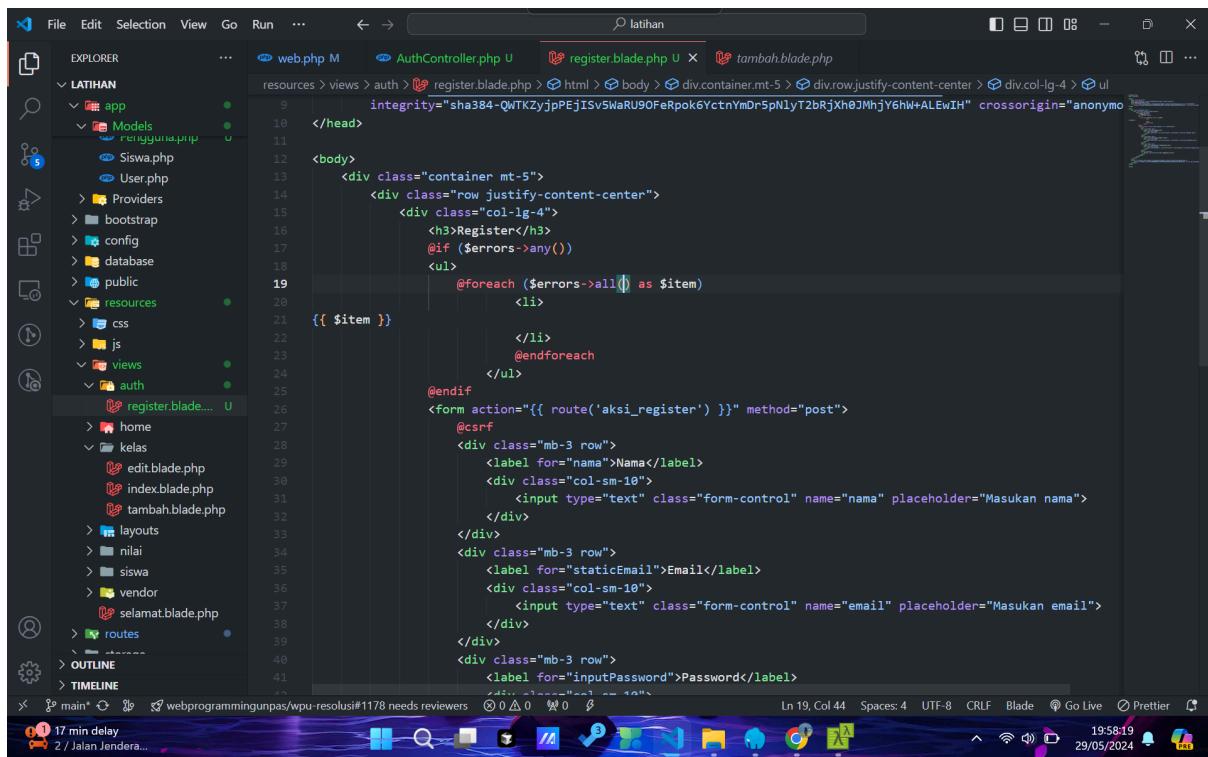
```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

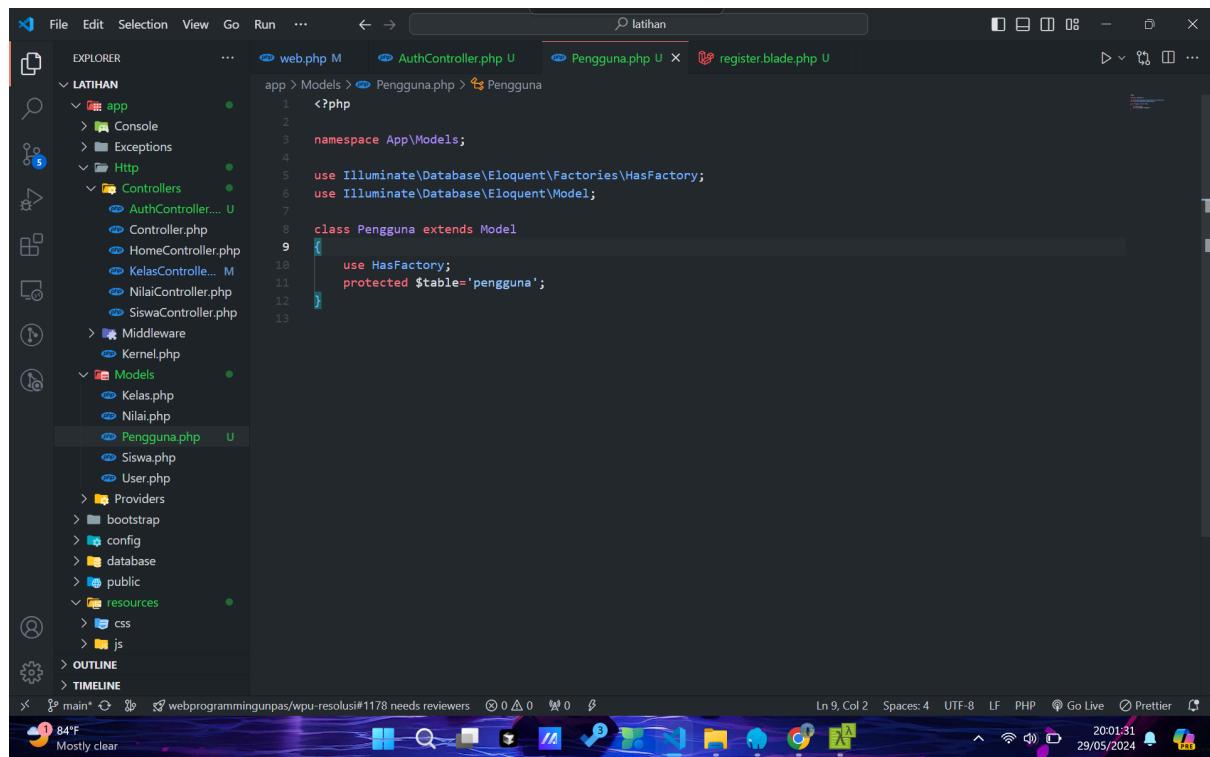
class AuthController extends Controller
{
    public function register(){
        return view('auth.register');
    }
    public function aksi_register(Request $request){
        $request->validate([
            'name'=>'required',
            'email'=>'required|email',
            'password'=>'required'
        ]);
    }
}
```

dan di form register nya tampilkan pesan validasinya



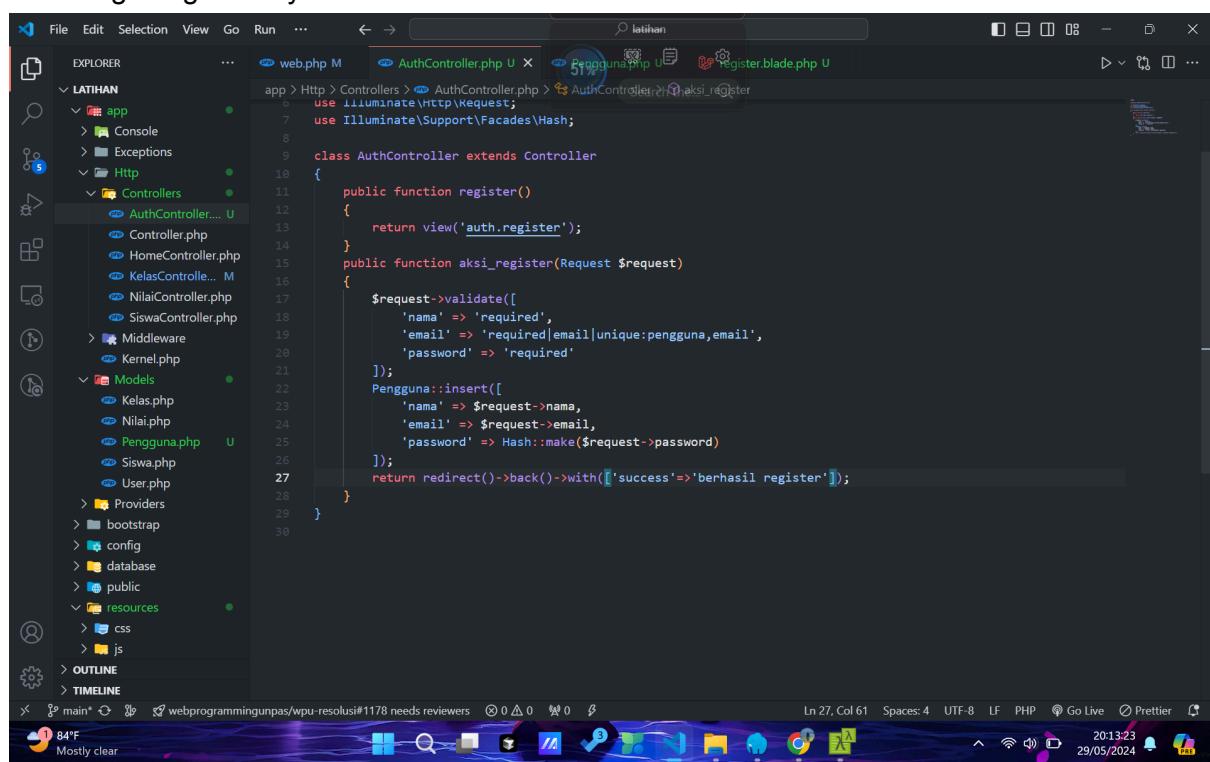
```
resources > views > auth > register.blade.php > html > body > div.container.mt-5 > div.row.justify-content-center > div.col-lg-4 > ul
integrity="sha384-QWTKZyjpEjISv5WaRU90FeRpok6YctnYmDr5pNlT2bRjXh0JMhjY6hw+ALEwIH" crossorigin="anonymous"
</head>
<body>
    <div class="container mt-5">
        <div class="row justify-content-center">
            <div class="col-lg-4">
                <h3>Register</h3>
                @if ($errors->any())
                    <ul>
                        @foreach ($errors->all() as $item)
                            <li>
                                {{ $item }}
                            </li>
                        @endforeach
                    </ul>
                @endif
                <form action="{{ route('aksi_register') }}" method="post">
                    @csrf
                    <div class="mb-3 row">
                        <label for="nama">Nama</label>
                        <div class="col-sm-10">
                            <input type="text" class="form-control" name="nama" placeholder="Masukan nama">
                        </div>
                    </div>
                    <div class="mb-3 row">
                        <label for="staticEmail">Email</label>
                        <div class="col-sm-10">
                            <input type="text" class="form-control" name="email" placeholder="Masukan email">
                        </div>
                    </div>
                    <div class="mb-3 row">
                        <label for="inputPassword">Password</label>
                    </div>
                </form>
            </div>
        </div>
    </div>
</body>
```

lalu buka models Pengguna lalu tetapkan models pengguna ke table pengguna



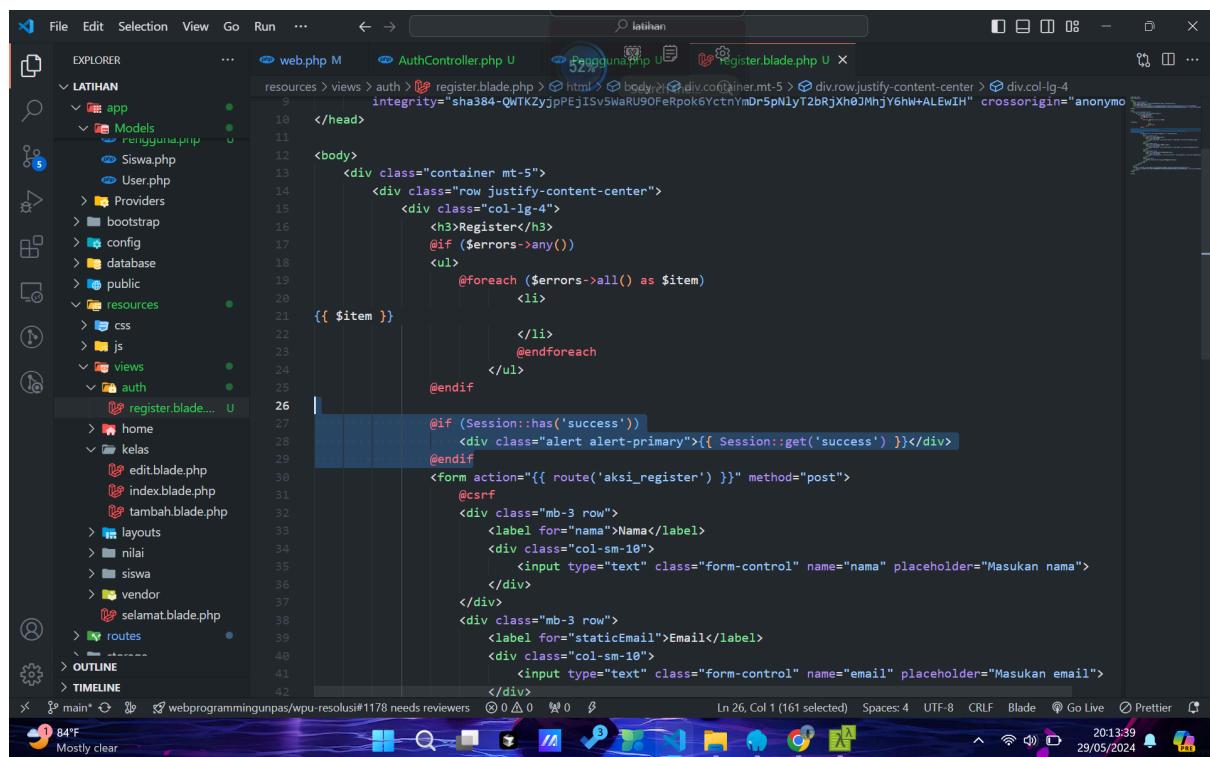
```
<?php  
namespace App\Models;  
  
use Illuminate\Database\Eloquent\Factories\HasFactory;  
use Illuminate\Database\Eloquent\Model;  
  
class Pengguna extends Model  
{  
    use HasFactory;  
    protected $table='pengguna';
```

buat fungsi register nya



```
use Illuminate\Http\Request;  
use Illuminate\Support\Facades\Hash;  
  
class AuthController extends Controller  
{  
    public function register()  
    {  
        return view('auth.register');  
    }  
    public function aksi_register(Request $request)  
    {  
        $request->validate([  
            'nama' => 'required',  
            'email' => 'required|email|unique:pengguna,email',  
            'password' => 'required'  
        ]);  
        Pengguna::insert([  
            'nama' => $request->nama,  
            'email' => $request->email,  
            'password' => Hash::make($request->password)  
        ]);  
        return redirect()->back()->with(['success'=>'berhasil register']);  
    }  
}
```

dan tampilkan pesan jika berhasil



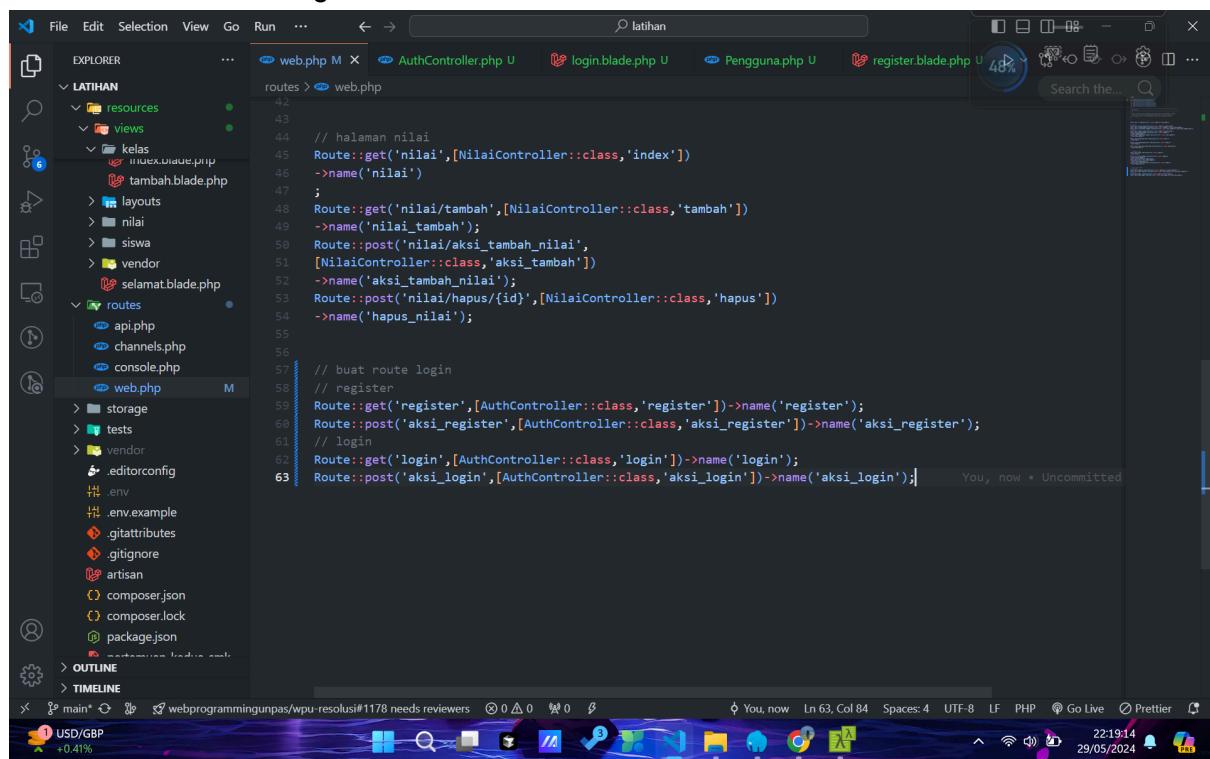
```
resources > views > auth > register.blade.php > html > body > div.container.mt-5 > div.row.justify-content-center > div.col-lg-4
    integrity="sha384-QWTKZyjpPEjISv5WaRU90FeRpok6YctnYmDr5pNlyT2bRjXh0JMJhjY6hw+ALEwIH" crossorigin="anonymous"

```

```
</head>
<body>
    <div class="container mt-5">
        <div class="row justify-content-center">
            <div class="col-lg-4">
                <h3>Register</h3>
                @if ($errors->any())
                    <ul>
                        @foreach ($errors->all() as $item)
                            <li>
                                {{ $item }}
                            </li>
                        @endforeach
                    </ul>
                @endif
                @if (Session::has('success'))
                    <div class="alert alert-primary">{{ Session::get('success') }}</div>
                @endif
                <form action="{{ route('aksi_register') }}" method="post">
                    @csrf
                    <div class="mb-3 row">
                        <label for="nama">Nama</label>
                        <div class="col-sm-10">
                            <input type="text" class="form-control" name="nama" placeholder="Masukan nama">
                        </div>
                    </div>
                    <div class="mb-3 row">
                        <label for="staticEmail">Email</label>
                        <div class="col-sm-10">
                            <input type="text" class="form-control" name="email" placeholder="Masukan email">
                        </div>
                    </div>
                </form>
            </div>
        </div>
    </div>

```

Membuat route untuk login



```
routes > web.php
42
43
44 // halaman nilai
45 Route::get('nilai',[NilaiController::class,'index'])
46 ->name('nilai')
47 ;
48 Route::get('nilai/tambah',[NilaiController::class,'tambah'])
49 ->name('nilai_tambah');
50 Route::post('nilai/aksi_tambah_nilai',
51 [NilaiController::class,'aksi_tambah'])
52 ->name('aksi_tambah_nilai');
53 Route::post('nilai/hapus/{id}',[NilaiController::class,'hapus'])
54 ->name('hapus_nilai');

55
56 // buat route login
57 // register
58 Route::get('register',[AuthController::class,'register'])->name('register');
59 Route::post('aksi_register',[AuthController::class,'aksi_register'])->name('aksi_register');
60 // login
61 Route::get('login',[AuthController::class,'login'])->name('login');
62 Route::post('aksi_login',[AuthController::class,'aksi_login'])->name('aksi_login');
```

lalu buat method login untuk menampilkan halaman login

```

14     }
15     public function aksi_register(Request $request)
16     {
17         $request->validate([
18             'nama' => 'required',
19             'email' => 'required|email|unique:pengguna,email',
20             'password' => 'required'
21         ]);
22         Pengguna::insert([
23             'nama' => $request->nama,
24             'email' => $request->email,
25             'password' => Hash::make($request->password)
26         ]);
27         return redirect()->back()->with(['success'=>'berhasil register']);
28     }
29
30     public function login()
31     {
32         return view('auth.login');
33     }
34

```

lalu buat file login.blade.php di dalam folder auth dan buat form untuk login

```

8         <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet"
9             integrity="sha384-QWTKzjpPEjISv5WaUR09FeRpk6YctnYmDr5pNlyT2bRjXh0JWhjY6hw+ALEwIH" crossorigin="anonymous"/>
10    </head>
11
12    <body>
13        <div class="container mt-5">
14            <div class="row justify-content-center">
15                <div class="col-lg-4">
16                    <h3>Login</h3>
17                    <form action="{{ route('aksi_login') }}" method="post">
18                        @csrf
19                        <div class="mb-3 row">
20                            <label for="staticEmail">Email</label>
21                            <div class="col-sm-10">
22                                <input type="text" class="form-control" name="email" placeholder="Masukan email">
23                            </div>
24                        </div>
25                        <div class="mb-3 row">
26                            <label for="inputPassword">Password</label>
27                            <div class="col-sm-10">
28                                <input type="password" class="form-control" name="password" placeholder="masukan pas">
29                            </div>
30                        </div>
31                        <div class="row">
32                            <button class="btn btn-primary">Login</button>
33                        </div>
34                    </form>
35                </div>
36            </div>
37        </div>
38        <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"
39             integrity="sha384-YvpcrYf0tY3lhB60NNkmXc5s9fDVZLESaAA55NDzOxy9GKcIdslK1eN7N6jIeHz" crossorigin="anonymous">
40        </script>

```

kemudian buat fungsi loginnya dengan method **aksi_login**

```

    $request->validate([
        'nama' => 'required',
        'email' => 'required|email|unique:pengguna,email',
        'password' => 'required'
    ]);
    Pengguna::insert([
        'nama' => $request->nama,
        'email' => $request->email,
        'password' => Hash::make($request->password)
    ]);
    return redirect()->back()->with(['success'=>'berhasil register']);
}

public function login(){
    return view('auth.login');
}

public function aksi_login(Request $request)
{
    $credentials = $request->validate([
        'email' => ['required', 'email'],
        'password' => ['required'],
    ]);

    if (Auth::attempt($credentials)) {
        $request->session()->regenerate();
        return redirect()->route('home');
    }
    return redirect()->back()->with(['failed'=>'email atau password salah!']);
}

```

Ln 43, Col 43 Spaces: 4 UTF-8 LF PHP Go Live Prettier

18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47

jika di coba akan muncul error seperti ini

Illuminate\Database\QueryException
SQLSTATE[42S02]: Base table or view not found: 1146 Table 'latihan_smk_ds.users' doesn't exist

```

SELECT * FROM `users` WHERE `email` = zae@gmail.com limit 1

```

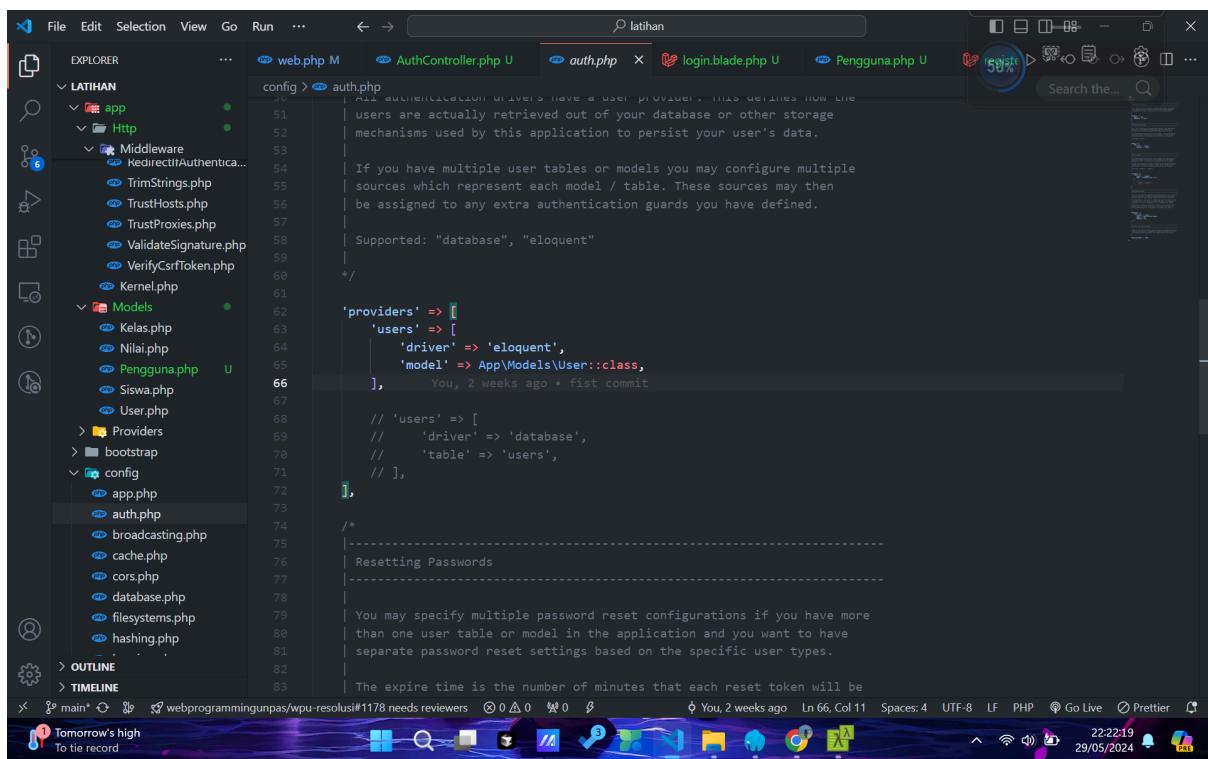
A table was not found
RUN MIGRATIONS
You might have forgotten to run your database migrations.
You can try to run your migrations using 'php artisan migrate'.
[Database: Running Migrations docs](#)

C:\laragon\www\smk-smk\latihan\app\Http\Controllers\AuthController.php:41

19 'nama' => 'required',
20 'email' => 'required|email|unique:pengguna,email',
21 'password' => 'required'
22];
23 Pengguna::insert([

ini karena secara default dari laravel ketika login akan mengarahkan ke model Users, sedangkan kita tidak mempunyai table user tapi kita menggunakan table pengguna.

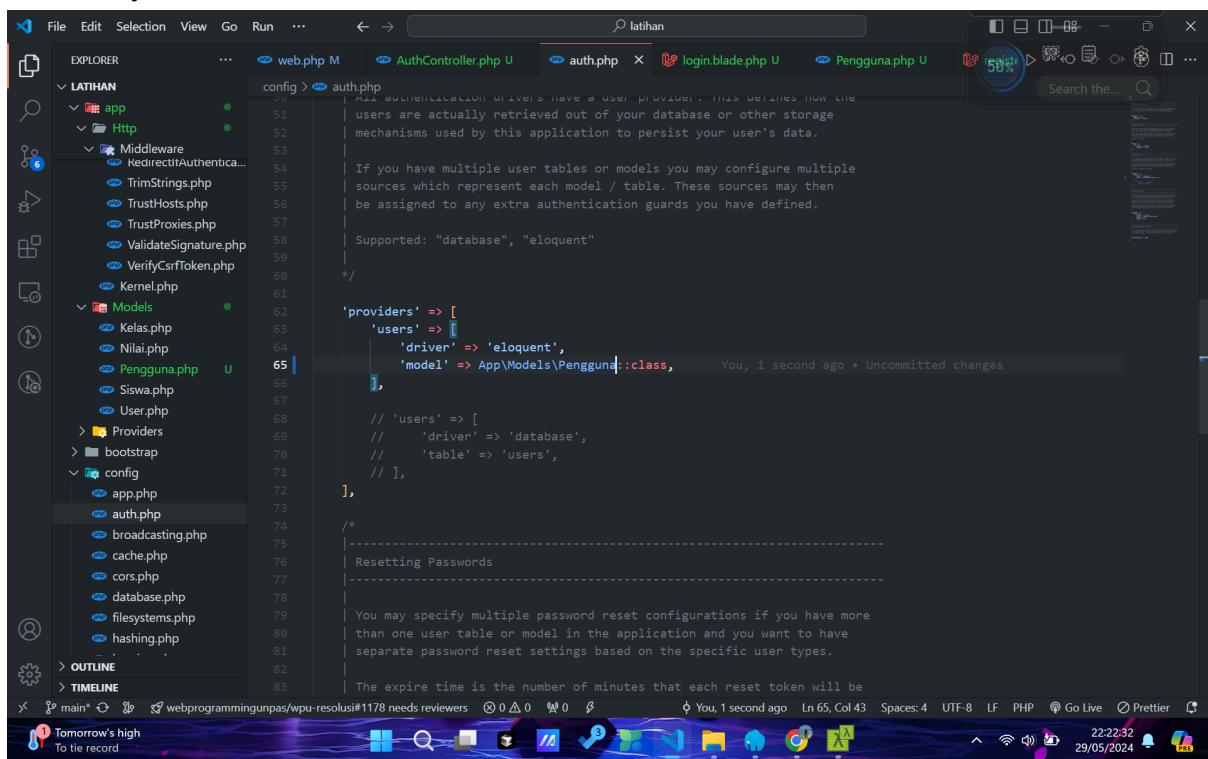
cara nya untuk mengubah Auth nya jadi memeriksa data ke table pengguna buka folder config/auth.php lalu cari script ini



```
config > auth.php
50 | All authentication drivers have a user provider. This defines how the
51 | users are actually retrieved out of your database or other storage
52 | mechanisms used by this application to persist your user's data.
53 |
54 | If you have multiple user tables or models you may configure multiple
55 | sources which represent each model / table. These sources may then
56 | be assigned to any extra authentication guards you have defined.
57 |
58 | Supported: "database", "eloquent"
59 |
60 */
61
62 'providers' => [
63   'users' => [
64     'driver' => 'eloquent',
65     'model' => App\Models\User::class,
66   ],
67   You, 2 weeks ago * first commit
68
69   // 'users' => [
70   //   'driver' => 'database',
71   //   'table' => 'users',
72   // ],
73 ],
74 /*
75 | -----
76 | Resetting Passwords
77 | -----
78 |
79 | You may specify multiple password reset configurations if you have more
80 | than one user table or model in the application and you want to have
81 | separate password reset settings based on the specific user types.
82 |
83 | The expire time is the number of minutes that each reset token will be
84 |
85 */

You, 2 weeks ago Ln 66, Col 11 Spaces: 4 UTF-8 LF PHP ⚡ Go Live ⚡ Prettier
```

ubah menjadi



```
config > auth.php
50 | All authentication drivers have a user provider. This defines how the
51 | users are actually retrieved out of your database or other storage
52 | mechanisms used by this application to persist your user's data.
53 |
54 | If you have multiple user tables or models you may configure multiple
55 | sources which represent each model / table. These sources may then
56 | be assigned to any extra authentication guards you have defined.
57 |
58 | Supported: "database", "eloquent"
59 |
60 */
61
62 'providers' => [
63   'users' => [
64     'driver' => 'eloquent',
65     'model' => App\Models\Pengguna::class,
66   ],
67   You, 1 second ago * Uncommitted changes
68
69   // 'users' => [
70   //   'driver' => 'database',
71   //   'table' => 'users',
72   // ],
73 ],
74 /*
75 | -----
76 | Resetting Passwords
77 | -----
78 |
79 | You may specify multiple password reset configurations if you have more
80 | than one user table or model in the application and you want to have
81 | separate password reset settings based on the specific user types.
82 |
83 | The expire time is the number of minutes that each reset token will be
84 |
85 */

You, 1 second ago Ln 65, Col 43 Spaces: 4 UTF-8 LF PHP ⚡ Go Live ⚡ Prettier
```

setelah itu ubah model Pengguna menjadi seperti ini

The screenshot shows a code editor interface with a dark theme. The left sidebar contains a file tree under the 'EXPLORER' tab, showing a project structure named 'LATIHAN'. The 'app' folder is expanded, revealing subfolders like 'Console', 'Exceptions', 'Http', 'Controllers', 'Middleware', and 'Models'. Within the 'Models' folder, files such as 'Kelas.php', 'Nilai.php', 'Pengguna.php', and 'Siswa.php' are listed. The main editor area displays a PHP file named 'Pengguna.php' with the following code:

```
<?php  
namespace App\Models;  
use Illuminate\Database\Eloquent\Factories\HasFactory;  
use Illuminate\Database\Eloquent\Model;  
use Illuminate\Foundation\Auth\User as Authenticatable;  
  
class Pengguna extends Authenticatable  
{  
    use HasFactory;  
    protected $table='pengguna';  
}
```

The status bar at the bottom provides information about the file: 'In 14, Col 1', 'Spaces: 4', 'UTF-8', 'LF', 'PHP', 'Go Live', and 'Prettier'. It also shows the current date and time: '29/05/2024' and '22:24:52'. A weather widget on the left indicates '83°F' and 'Partly cloudy'.