

MODUL MEMBUAT CRUD MENGGUNAKAN LARAVEL

A. PENJELASAN CRUD

CRUD adalah sebuah kriteria dasar untuk memastikan sebuah framework atau API dapat mengikuti perkembangan teknologi. CRUD adalah singkatan dari Create, Read, Update, dan Delete. Keempat hal tersebut merupakan perintah dengan peran yang esensial dalam sebuah aplikasi database yang sifatnya relational.

Beberapa contoh DBMS (Database Management System) yang menggunakan CRUD sebagai fungsi utamanya yaitu Oracle Database, Microsoft SQL Server, MySQL, PostgreSQL, dan sebagainya.

B. FUNGSI CRUD

fungsi ini merupakan fondasi kerja sebuah database. Bayangkan saja kinerja sebuah database itu sebagai sebuah perpustakaan. Agar operasional perpustakaan dapat berjalan, kamu perlu sebuah sistem pendataan buku-buku yang ada di dalamnya.

C. LANGKAH-LANGKAH MEMBUAT CRUD MENGGUNAKAN LARAVEL V.9

MEMBUAT PROJECT LARAVEL

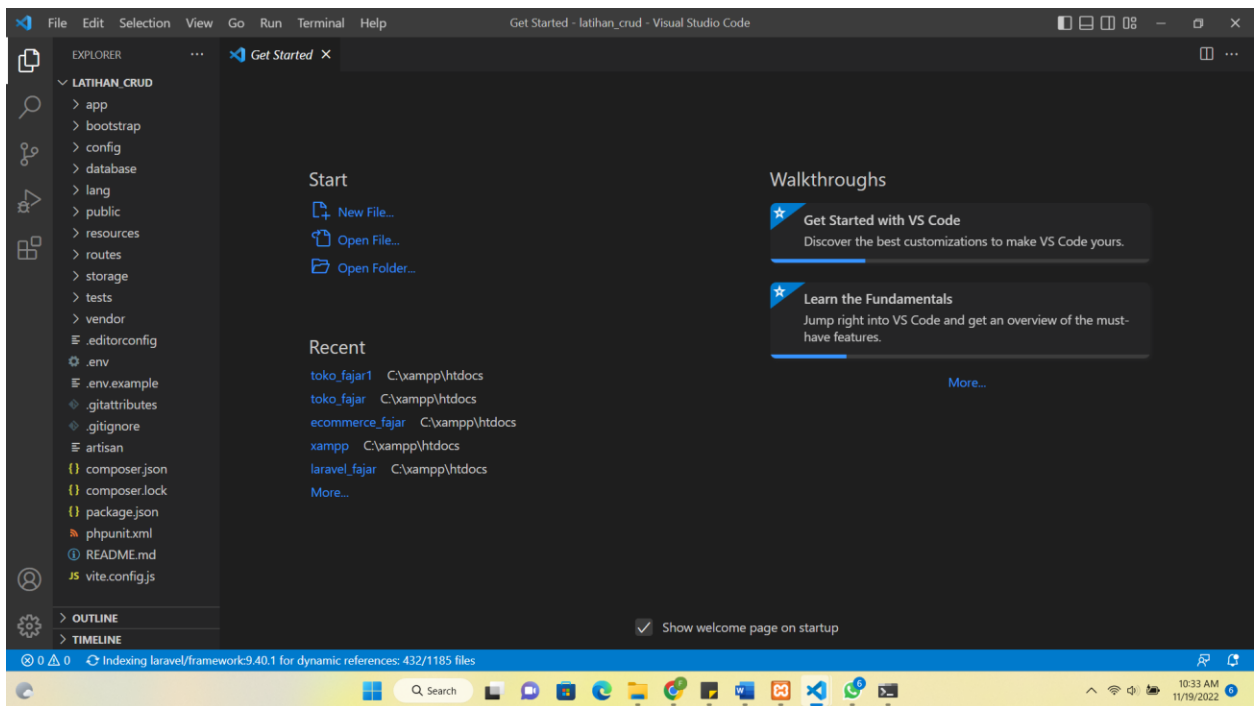
1. Install xampp dengan PHP versi 8.0, node.js dan Composer
2. Buka cmd dan install Laravel v.9 menggunakan composer dengan memasukan perintah "cd C:\xampp\htdocs"

```
C:\Users\ACER SWIFT 3>cd C:\xampp\htdocs  
C:\xampp\htdocs>|
```

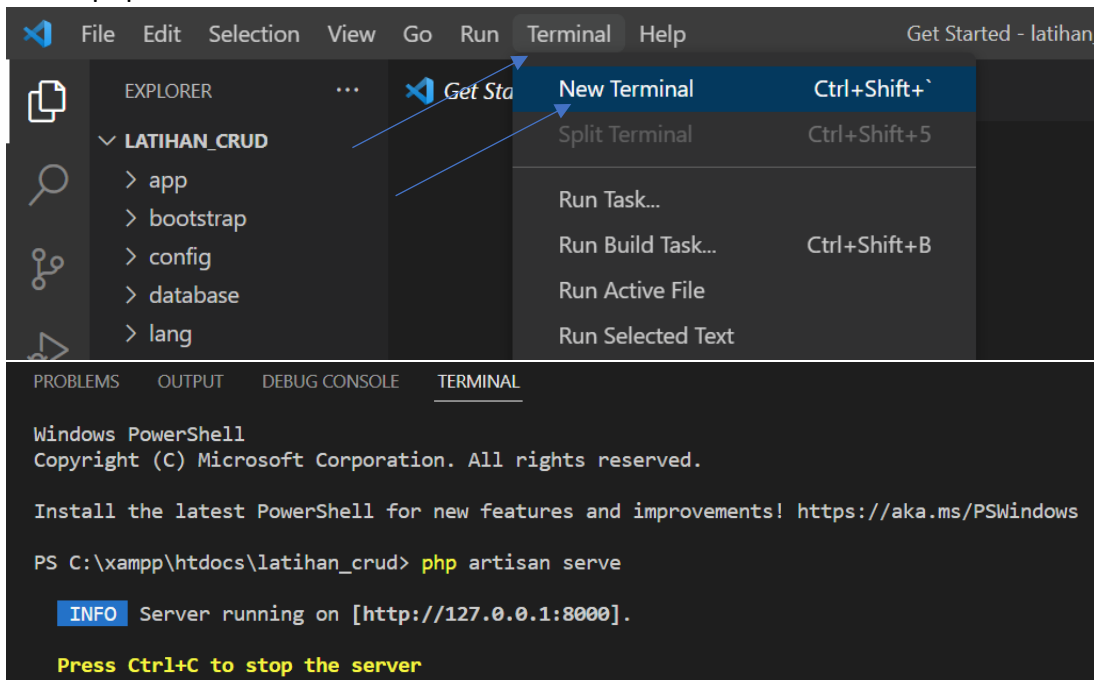
3. Lalu ketikan perintah ini "composer create-project --prefer-dist laravel/laravel latihan_crud"

```
C:\xampp\htdocs>composer create-project --prefer-dist laravel/laravel latihan_crud  
Creating a "laravel/laravel" project at "./latihan_crud"  
Info from https://repo.packagist.org: #StandWithUkraine  
Installing laravel/laravel (v9.3.11)  
- Installing laravel/laravel (v9.3.11): Extracting archive  
Created project in C:\xampp\htdocs\latihan_crud  
> @php -r "file_exists('.env') || copy('.env.example', '.env');"   
Loading composer repositories with package information  
Updating dependencies  
Lock file operations: 107 installs, 0 updates, 0 removals  
Use the `composer fund` command to find out more!  
> @php artisan vendor:publish --tag=laravel-assets --ansi --force  
  
INFO No publishable resources for tag [laravel-assets].  
  
No security vulnerability advisories found  
> @php artisan key:generate --ansi  
  
INFO Application key set successfully.
```

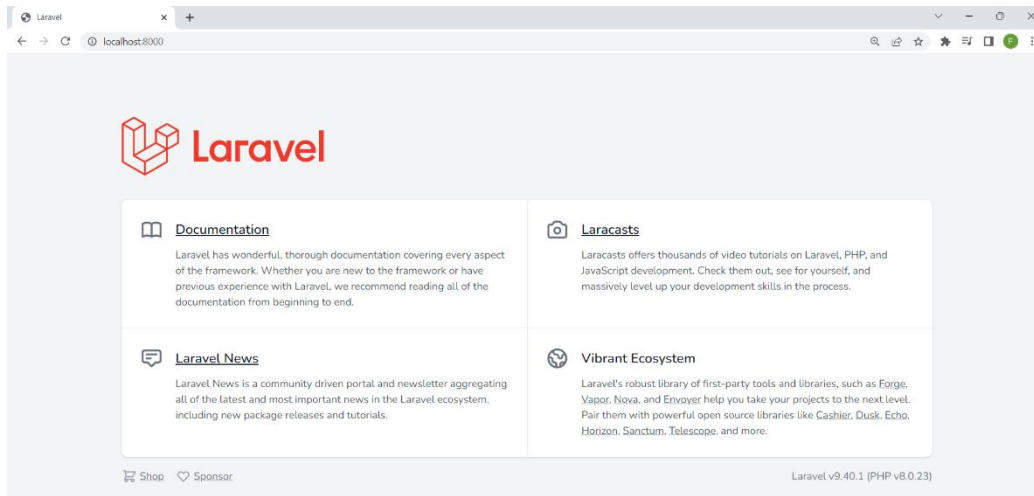
4. Setelah selesai, buka projectnya di visual studio code.



5. Ketik "php artisan serve" diterminal



6. Cek di google form atau di software browser lainnya, dengan menuliskan localhost:8000. Maka tampilannya akan seperti dibawah ini

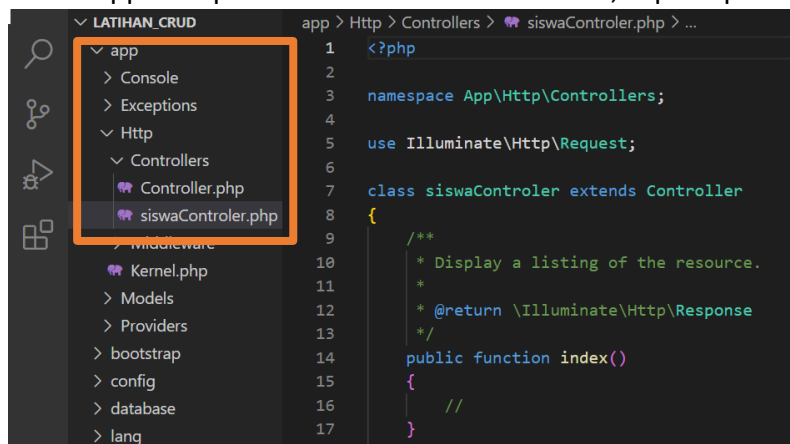


MEMBUAT CONTROLLER DAN ROUTE

7. Buka terminal lalu tuliskan perintah **"php artisan make:controller siswaController --resource"** tunggu sampai selesai

```
PS C:\xampp\htdocs\latihan_crud> php artisan make:controller siswaController --resource
INFO Controller [C:\xampp\htdocs\latihan_crud\app\Http\Controllers/siswaController.php] created successfully.
PS C:\xampp\htdocs\latihan_crud>
```

8. Setelah selesai membuat controller bernama siswaController, maka file tersebut letaknya ada di folder **app→http→contoller→siswaController**, seperti pada gambar.



9. Ketika kita membuka file **siswaController** tersebut maka di dalamnya ada function index, yang berfungsi untuk menampilkan semua data. Maka isi function index ini dengan seperti di bawah ini lalu save.

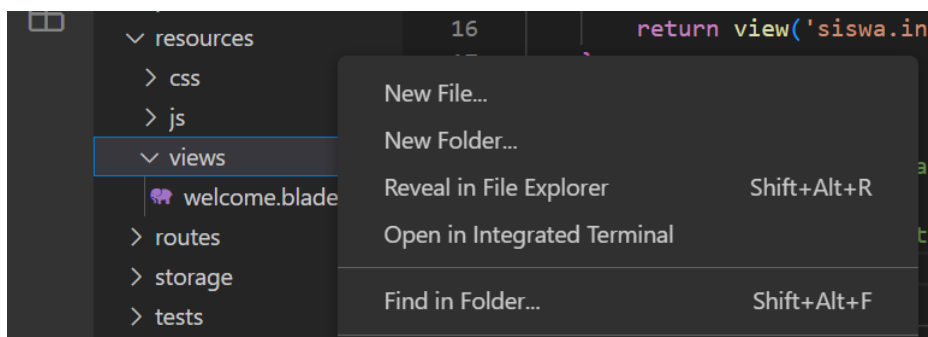
```

*
* @return \Illuminate\Http\Response
*/
public function index()
{
    return view('siswa.index');
}

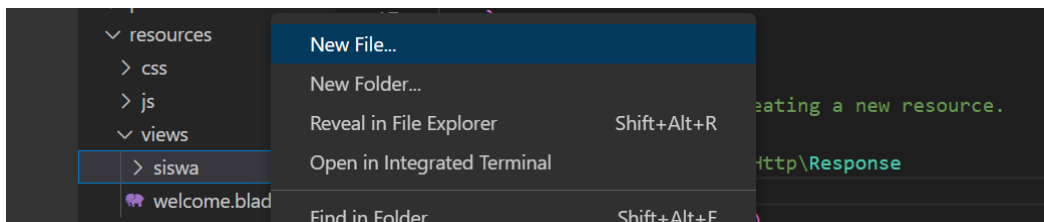
```

MEMBUAT BLADE HALAMAN DEPAN

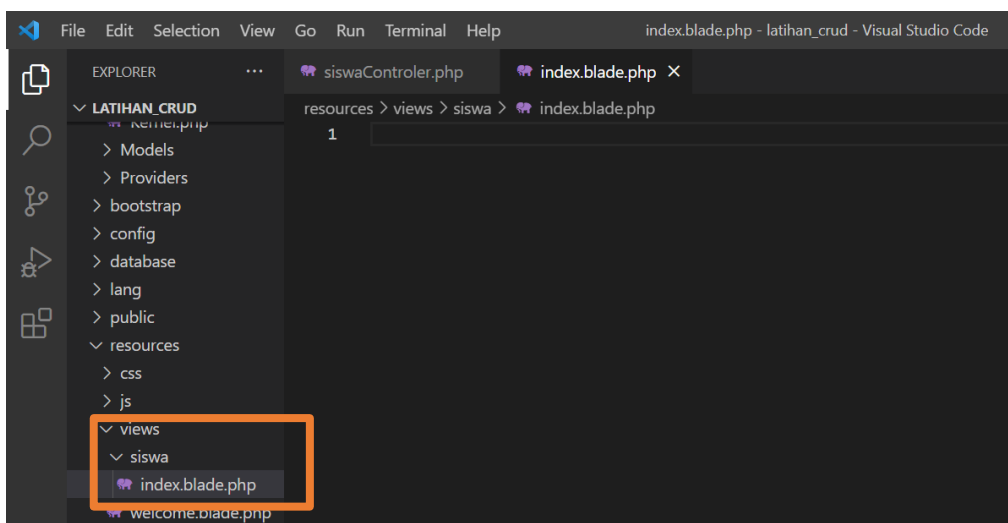
10. Lalu kita membuat folder dan file baru pada folder views. Kita membuat folder siswa dan membuat file bernama **index.blade.php** di dalamnya Seperti pada gambar di bawah ini.
11. klik kanan pada view, lalu klik New Folder, beri nama nama folder tersebut engan nama siswa



12. Setelah itu, klik kanan pada folder baru bernama siswa tadi, lalu pilih **New File** dan beri nama file baru tersebut dengan nama **index.blade.php**



13. Maka tampilannya akan seperti dibawah ini



14. Masukkan script pada file index tadi seperti di bawah ini, untuk membuat tampilan halaman index

```
Run Terminal Help • index.blade.php - latihan_crud1 - Visual Studio Code
siswaController.php Controller.php index.blade.php web.php
resources > views > siswa > index.blade.php > html > body.bg-light > main.container > form > div.my-3.p-3.bg-body.rounded.shadow-sm > div.mb-3.row
1 <!doctype html>
2 <html lang="en">
3 <head>
4 <meta charset="utf-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1">
6 <title>Data Siswa</title>
7 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/css/bootstrap.min.css"
8 rel="stylesheet" integrity="sha384-iYQeCZEFYfKjA/T2uDLTPkwGzCiq6soy8tYaI1GyVh/UjpbCx/TYkiZhlZB6+fzT" crossorigin="anonymous">
9 </head>
10
11 <!-- START FORM -->
12 <form action="" method="post">
13 <div class="my-3 p-3 bg-body rounded shadow-sm">
14 <div class="mb-3 row">
15 <label for="nim" class="col-sm-2 col-form-label">NIS</label>
16 <div class="col-sm-10">
17 <input type="number" class="form-control" name="nis" id="nis">
18 </div>
19 </div>
20 <div class="mb-3 row">
21 <label for="nama" class="col-sm-2 col-form-label">Nama</label>
22 <div class="col-sm-10">
23 <input type="text" class="form-control" name="nama" id="nama">
24 </div>
25 </div>
26 <div class="mb-3 row">
27 <label for="jurusan" class="col-sm-2 col-form-label">Jurusan</label>
28 <div class="col-sm-10">
29 <input type="text" class="form-control" name="jurusan" id="jurusan">
30 </div>
31 </div>
32 <div class="mb-3 row">
33 <label for="jurusan" class="col-sm-2 col-form-label"></label>
34 <div class="col-sm-10"><button type="submit" class="btn btn-primary" name="submit">SIMPAN</button></div>
35 </div>
36 </form>
37 </div>
38 <!-- AKHIR FORM -->
39
40 <!-- START DATA -->
41
42 <div class="my-3 p-3 bg-body rounded shadow-sm">
43 <!-- FORM PENCARIAN -->
44 <div class="pb-3">
45 <form class="d-flex" action="" method="get">
46 <input class="form-control me-1" type="search" name="katakunci"
47 value="{{ Request::get('katakunci') }}" placeholder="Masukkan kata kunci" aria-label="Search">
48 <button class="btn btn-secondary" type="submit">Cari</button>
49 </form>
50 </div>
51 </div>
52
53
```

```
Go Run Terminal Help • index.blade.php - latihan_crud1 - Visual Studio Code
siswaController.php Controller.php index.blade.php web.php
resources > views > siswa > index.blade.php > html > body.bg-light > script
51 <!-- TOMBOL TAMBAH DATA -->
52 <div class="pb-3">
53 | <a href="" class="btn btn-primary">+ Tambah Data</a>
54 </div>
55
56 <table class="table table-striped">
57 | <thead>
58 | | <tr>
59 | | | <th class="col-md-1">No</th>
60 | | | <th class="col-md-3">NIS</th>
61 | | | <th class="col-md-4">Nama</th>
62 | | | <th class="col-md-2">Jurusan</th>
63 | | | <th class="col-md-2">Aksi</th>
64 | | </tr>
65 | </thead>
66 | <tbody>
67 | | <tr>
68 | | | <td>1</td>
69 | | | <td>1001</td>
70 | | | <td>Fajar</td>
71 | | | <td>Rekayasa Perangkat Lunak</td>
72 | | | <td>
73 | | | | <a href="" class="btn btn-warning btn-sm">Edit</a>
74 | | | | <a href="" class="btn btn-danger btn-sm">Del</a>
75 | | | </td>
76 | | </tr>
77 | </tbody>
78 </table>
79
80 </div>
81 <!-- AKHIR DATA -->
82 </main>
83 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/js/bootstrap.bundle.min.js"
84 | integrity="sha384-u10knCvxywY5kfmNBILK2hRnQC3Pr17a+RTT6rIHI7NnikvZlHgTPOOmMi466C8" crossorigin="anonymous"></script>
85 </body>
86 </html>
```

15. Sekarang kita membuat route untuk menampilkan halaman index yang sudah kita buat, dengan cara buka folder routes→web.php dan buat route seperti di bawah ini.

```
routes
  api.php
  channels.php
  console.php
  web.php
  storage
16
17 Route::get('/', function () {
18 |   return view('welcome');
19 | });
20
21 Route::resource('siswa'. siswaController::class);
22
```

16. Buka kembali browser kita, dan masukan localhost:8000/siswa. Maka tampilannya akan seperti di bawah ini

Web application interface for student data management. The interface includes input fields for NIS, Nama, and Jurusan, a SIMPAN button, a search bar with a Cari button, a + Tambah Data button, and a table with one row of student data.

No	NIS	Nama	Jurusan	Aksi
1	1001	Fajar	Rekayasa Perangkat Lunak	<button>Edit</button> <button>Del</button>

MEMBUAT DATABASE, TABLE, MODEL DAN MIGRATION

17. Buatlah database di phpmyadmin dengan nama **crud_latihan**
18. Buka file **.env** dan ganti nama databasenya dengan **crud_latihan**

```

1 APP_NAME=Laravel
2 APP_ENV=local
3 APP_KEY=base64:30yd7qnP5XnT5nUsBt/Ut9N4bLOKLpve1SBkbPCEmMo=
4 APP_DEBUG=true
5 APP_URL=http://localhost
6
7 LOG_CHANNEL=stack
8 LOG_DEPRECATIONS_CHANNEL=null
9 LOG_LEVEL=debug
10
11 DB_CONNECTION=mysql
12 DB_HOST=127.0.0.1
13 DB_PORT=3306
14 DB_DATABASE=crud_latihan
15 DB_USERNAME=root
16 DB_PASSWORD=
17
18 BROADCAST_DRIVER=log
19 CACHE_DRIVER=file
20 FILESYSTEM_DISK=local
21 QUEUE_CONNECTION=sync
22 SESSION_DRIVER=file
23 SESSION_LIFETIME=120
24

```

19. Selanjutnya, kita akan membuat model dan juga migration sekaligus. Model digunakan untuk proses akses kita kepada data dan migration dilakukan untuk membuat skema database kita. Buka terminal dan tuliskan dalam terminal seperti di bawah ini.

```
PS C:\xampp\htdocs\latihan_crud1> php artisan make:model siswa -m
```

```
INFO Model [C:\xampp\htdocs\latihan_crud1\app\Models\siswa.php] created successfully.
```

```
INFO Created migration [2022_11_21_205412_create_siswas_table].
```

20. Buka file migrarion, database→migrations→2022_11_21_205412_create_siswas_table. Jika sudah dibuka makan hapus “s” pada kata siswas(nama table) menjadi siswa, lalu masukan script. Seperti pada gambar

SEBELUM

```
EXPLORER  ...  siswaController.php  index.blade.php  .env  2022_11_21_205412_create_siswas_table.php > ...

LATIHA...  database > migrations > 2022_11_21_205412_create_siswas_table.php > ...
├─ app
├─ bootstrap
├─ config
├─ database
├─ factories
├─ migrations
│   ├── 2014_10_12_00000...
│   ├── 2014_10_12_10000...
│   ├── 2019_08_19_00000...
│   ├── 2019_12_14_00000...
│   └── 2022_11_21_20541...
├─ seeders
├─ .gitignore
├─ lang
├─ public
├─ resources
│   ├── css
│   └─ js

11  *
12  * @return void
13  */
14  public function up()
15  {
16      Schema::create('siswas', function (Blueprint $table) {
17          $table->id();
18          $table->timestamps();
19      });
20  }
21
22  /**
23  * Reverse the migrations.
24  *
25  * @return void
26  */
27  public function down()
28  {
29      Schema::dropIfExists('siswas');
30  }
31  };
```


SESUDAH

```

1  <?php
2
3  use Illuminate\Database\Migrations\Migration;
4  use Illuminate\Database\Schema\Blueprint;
5  use Illuminate\Support\Facades\Schema;
6
7  return new class extends Migration
8  {
9      /**
10       * Run the migrations.
11       *
12       * @return void
13       */
14     public function up()
15     {
16         Schema::create('siswa', function (Blueprint $table) {
17             $table->integer('nis');
18             $table->unique('nis');
19             $table->string('nama');
20             $table->string('jurusan');
21         });
22     }

```

21. Lalu tulis **php artisan migrate** pada termnal, dan hasilnya akan seperti di bawah

PS C:\xampp\htdocs\latihan_crud1> **php** artisan migrate

INFO Preparing database.

Creating migration table 28ms **DONE**

INFO Running migrations.

2014_10_12_000000_create_users_table 34ms **DONE**

2014_10_12_100000_create_password_resets_table 29ms **DONE**

2019_08_19_000000_create_failed_jobs_table 33ms **DONE**

2019_12_14_000001_create_personal_access_tokens_table 45ms **DONE**

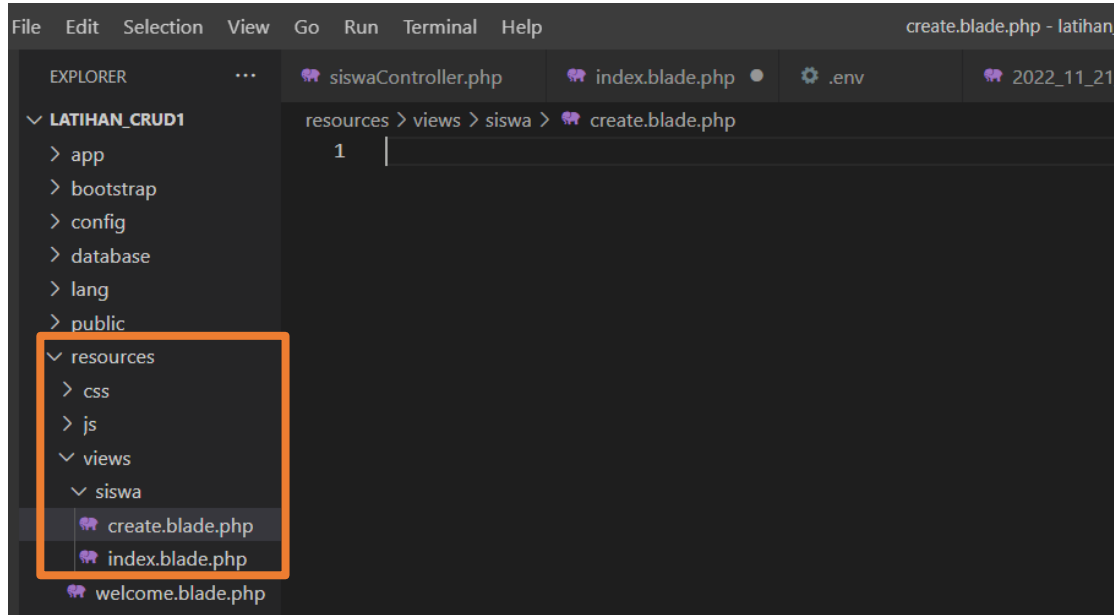
Server: 127.0.0.1 » Database: crud_latihan » Table: siswa

Table structure

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	nis	int(11)			No	None			Change Drop More
2	nama	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
3	jurusan	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More

PROSES CREATE DATA

22. Tambah file baru pada folder siswa dengan nama create.blade.php



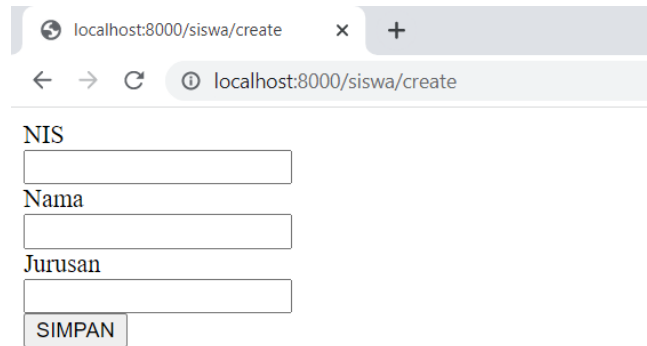
23. Untuk membuat create ini, kita **CUT** script yg ada di index dari mulai **START FORM** sampai **AKHIR FORM** dan letakan di file **create.blade.php**

```
11 <!-- START FORM -->
12 <form action="" method='post'>
13   <div class="my-3 p-3 bg-body rounded shadow-sm">
14     <div class="mb-3 row">
15       <label for="nim" class="col-sm-2 col-form-label">NIS</label>
16       <div class="col-sm-10">
17         <input type="number" class="form-control" name='nis' id="nis">
18       </div>
19     </div>
20     <div class="mb-3 row">
21       <label for="nama" class="col-sm-2 col-form-label">Nama</label>
22       <div class="col-sm-10">
23         <input type="text" class="form-control" name='nama' id="nama">
24       </div>
25     </div>
26     <div class="mb-3 row">
27       <label for="jurusan" class="col-sm-2 col-form-label">Jurusan</label>
28       <div class="col-sm-10">
29         <input type="text" class="form-control" name='jurusan' id="jurusan">
30       </div>
31     </div>
32     <div class="mb-3 row">
33       <label for="jurusan" class="col-sm-2 col-form-label"></label>
34       <div class="col-sm-10"><button type="submit" class="btn btn-primary" name="submit">SIMPAN</button></div>
35     </div>
36   </form>
37 </div>
38 <!-- AKHIR FORM -->
```

24. Lalu masukan script di bawah ini pada file **siswaController**.

```
public function create()
{
    return view('siswa.create');
}
```

25. Setelah selesai, kita coba buka browser lalu ketik localhost:8000/siswa/create dan reload halaman, maka tampilannya akan seperti ini



localhost:8000/siswa/create

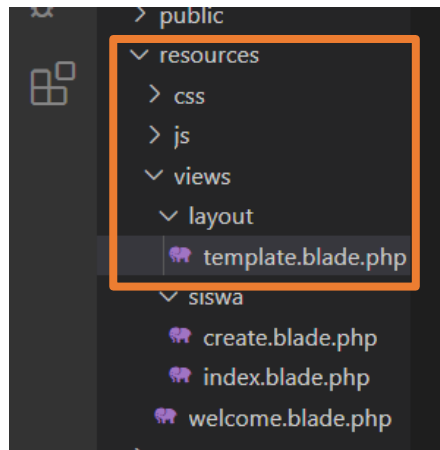
NIS

Nama

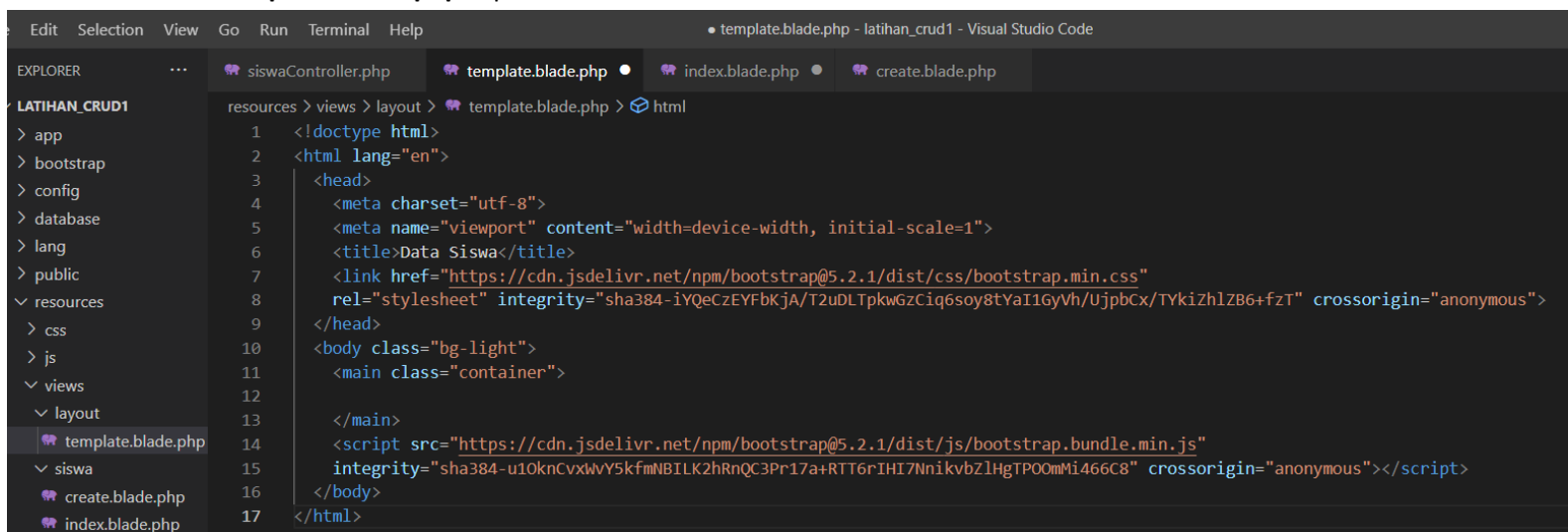
Jurusan

SIMPAN

26. Karena tampilan di atas kurang menarik, maka kita pakai template yg sudah ada di index dengan cara buat folder baru di view, beri nama **layout** dan tambah file baru dengan nama **template.blade.php** di dalam folder layout.



27. CUT pada bagian atas dan bagian paling bawah pada file **index.blade.php** dan masukan ke file **template.blade.php** seperti di bawah ini



```
1 <!doctype html>
2 <html lang="en">
3 <head>
4 <meta charset="utf-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1">
6 <title>Data Siswa</title>
7 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/css/bootstrap.min.css"
8 rel="stylesheet" integrity="sha384-iYQeCZEVfBKjA/T2uDLTPkwGzCiq6soy8tYaI1GyVh/UjpbCx/TYkiZhlZB6+fzT" crossorigin="anonymous">
9 </head>
10 <body class="bg-light">
11 <main class="container">
12
13 </main>
14 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/js/bootstrap.bundle.min.js"
15 integrity="sha384-u10knCvxwVY5kfmbNBILK2hRnQC3Pr17a+RTT6rIHI7NnikvZlHgTP00mMi466C8" crossorigin="anonymous"></script>
16 </body>
17 </html>
```

28. Lalu kita masukan **@extends('layout.template')** dibagian atas pada file index dan create untuk menggunakan template yg sudah kita buat

```

ces > views > siswa > index.blade.php > ...
@extends('layout.template')
<!-- START DATA -->
<div class="my-3 p-3 bg-body rounded shadow-sm">
  <!-- FORM PENCARIAN -->
  <div class="pb-3">
    <form class="d-flex" action="" method="get">
      <input class="form-control me-1" type="search">
      <button class="btn btn-secondary" type="submit">

```

29. Masukkan **@yield('konten')** ditengah-tengah main yg ada pada file **template.blade.php**

```

Terminal Help
template.blade.php - latihan_crud1 - Visual Studio Code

Controller.php  template.blade.php X  index.blade.php  create.blade.php

s > views > layout > template.blade.php > html > head
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Data Siswa</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/css/bootstrap.min.css"
    rel="stylesheet" integrity="sha384-iYQeCzEYFbKjA/T2uDLTPkwGzCiq6soy8tVaI1GyVh/UjpbCx/TYkiZhlZB6+fzT" crossorigin="anonymous">
  </head>
  <body class="bg-light">
    <main class="container">
      @yield('konten')
    </main>
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/js/bootstrap.bundle.min.js"
    integrity="sha384-u10knCvxwVY5kfMNBILK2hRnQC3Pr17a+RTT6rIH7NnikvbZlHgTP00mMi466C8" crossorigin="anonymous"></script>
  </body>
</html>

```

30. Lalu panggil konten tersebut pada file index dan create, dengan cara menambahkan **@section('konten')** di awal dan **@endsection** di akhir script seperti dibawah ini.

```

Terminal Help
index.blade.php - latihan_crud1 - Visual Studio Code

Controller.php  template.blade.php  index.blade.php X  create.blade.php

s > views > siswa > index.blade.php > ...
@extends('layout.template')
@section('konten')
  <!-- START DATA -->
  <div class="my-3 p-3 bg-body rounded shadow-sm">
    <!-- FORM PENCARIAN -->
    <div class="pb-3">
      <form class="d-flex" action="" method="get">
        <input class="form-control me-1" type="search" name="katakunci">
        <button class="btn btn-secondary" type="submit">Cari</button>
      </form>
    </div>
  </div>
</div>

```

```

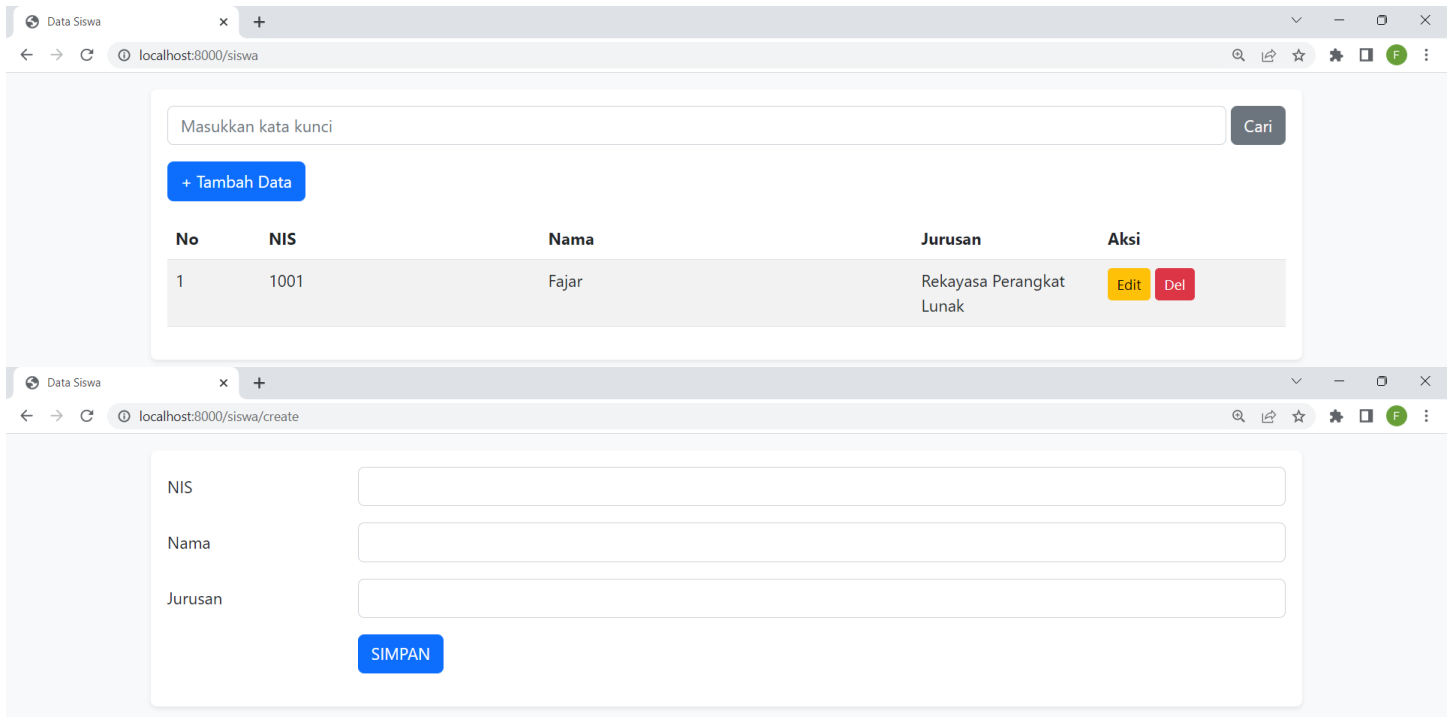
        <a href="" class="btn btn-warning btn-sm">Edit</a>
        <a href="" class="btn btn-danger btn-sm">Del</a>
      </td>
    </tr>
  </tbody>
</table>

</div>
<!-- AKHIR DATA -->
@endsection

```

31. Lakukan hal yang sama pada file **create.blade.php**

32. Maka tampilannya akan seperti dibawah ini



33. Sekarang kita membuat aksi post agar bisa mengarahkan ke siswaController, dengan cara mengisi `{{url('siswa')}}` pada form action yg kosong di file **create.blade.php** dan `@csrf` dibawahnya

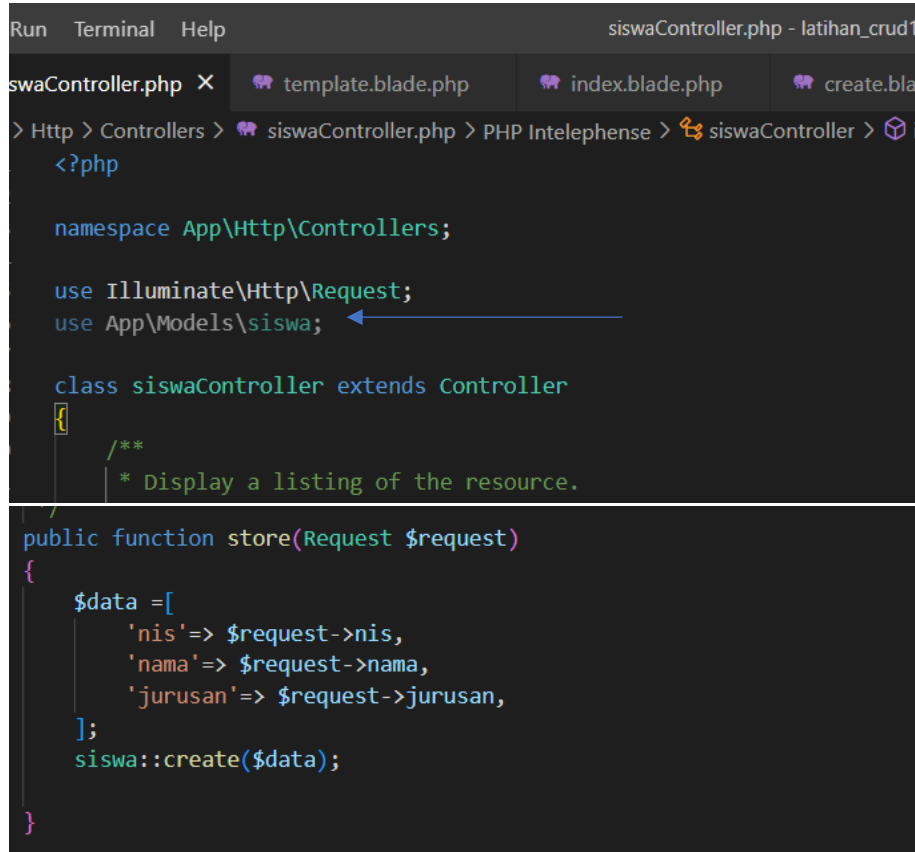
```

create.blade.php - latihan_crud1 - Visual Studio Code
views > siswa > create.blade.php > form
@extends('layout.template')
@section('konten')

<!-- START FORM -->
<form action="{{url('siswa')}}" method='post'>
  @csrf
  <div class="my-3 p-3 bg-body rounded shadow-sm">
    <div class="mb-3 row">
      <label for="nim" class="col-sm-2 col-form-label">NIS</label>
      <div class="col-sm-10">
        <input type="number" class="form-control" name='nim' id="nim">
      </div>
    </div>
  </div>

```

34. Sekarang kita buka siswaController tuliskan **use App\Models\siswa;** , dan masukan pada fungsi store seperti dibawah ini.



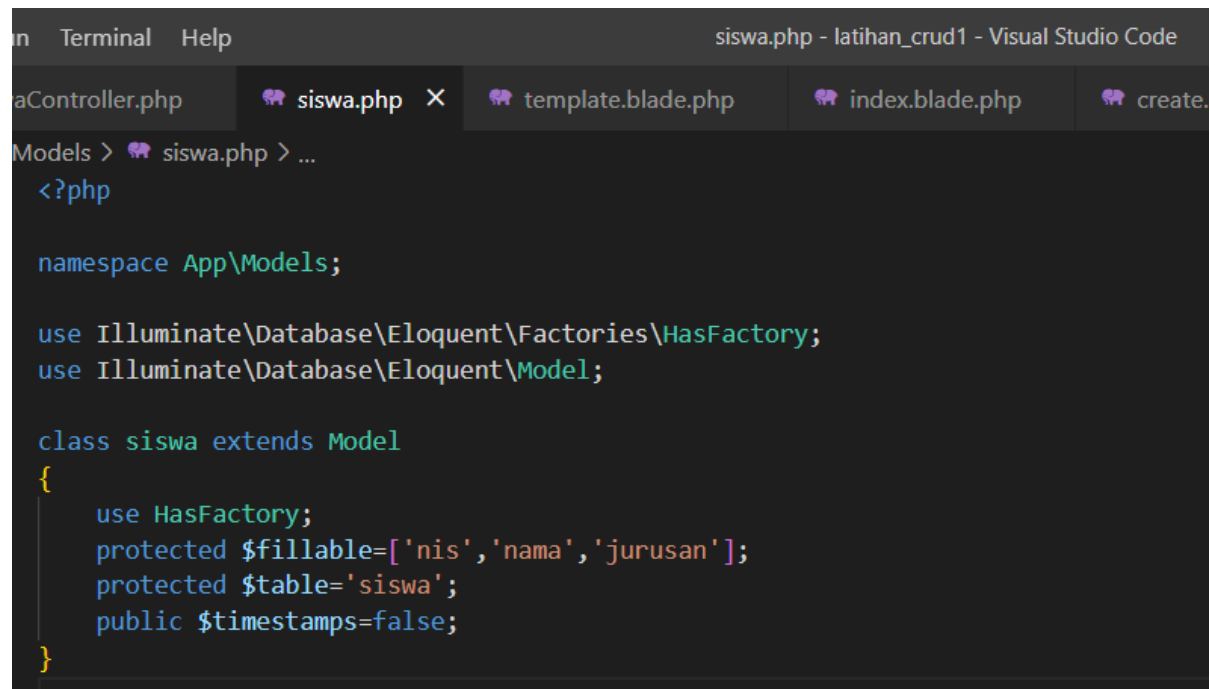
```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Models\siswa;

class siswaController extends Controller
{
    /**
     * Display a listing of the resource.
     */
    public function store(Request $request)
    {
        $data =[
            'nis'=> $request->nis,
            'nama'=> $request->nama,
            'jurusan'=> $request->jurusan,
        ];
        siswa::create($data);
    }
}
```

35. Selanjutnya buka file **siswa.php** yang ada di app→models→siswa.php, isikan scrip berikut



```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class siswa extends Model
{
    use HasFactory;
    protected $fillable=['nis','nama','jurusan'];
    protected $table='siswa';
    public $timestamps=false;
}
```

36. Setelah itu kita isi data siswanya lalu klik simpan, maka yg terjadi tampilan aka kosong, dan cek database, data yg kita isi masuk

Web browser window showing a form to create a new student record. The form has three input fields: NIS (1234), Nama (asdas), and Jurusan (asdas). A blue 'SIMPAN' button is at the bottom.

phpMyAdmin interface showing the 'siswa' table in the 'crud_latihan' database. The table contains two rows of data. A blue arrow points to the second row.

	nis	nama	jurusan
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	123	asdas	asdas
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1234	asdas	asdas

PROSES VALIDASI

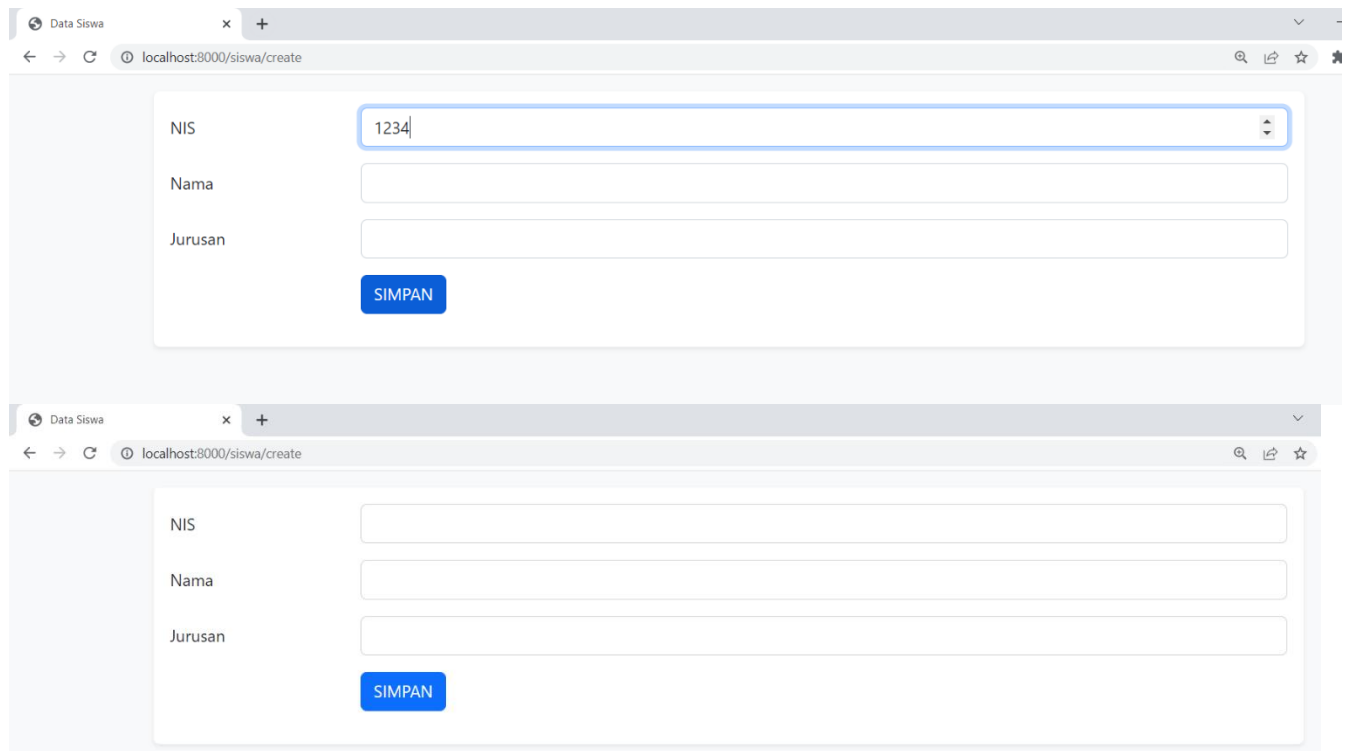
37. Untuk melakukan validasi, maka masukan script pada **siswaController** seperti di bawah ini

```

public function store(Request $request)
{
    $request->validate([
        'nis'=> 'required|numeric|unique:siswa,nis',
        'nama'=> 'required',
        'jurusan'=> 'required',
    ]);
    $data = [
        'nis'=> $request->nis,
        'nama'=> $request->nama,
        'jurusan'=> $request->jurusan,
    ];
    siswa::create($data);
}

```

38. Reload halaman create di browser, jika di klik simpan dan textfield ada yg kosong, maka akan kembali ke halaman create dan data tidak masuk ke database.



39. Untuk mengetahui errornya karena textfield kosong, maka kita tambahkan perintah pada file **create.blade.php** seperti dibawah ini.

```
waController.php • siswa.php • template.blade.php
resources > views > siswa > create.blade.php > div.pt-3 > div.alert.danger
@extends('layout.template')
@section('konten')

@if ($errors->any())
<div class="pt-3">
  <div class="alert alert-danger">
    <ul>
      @foreach ($errors->all() as $item)
        <li>{{$item}}</li>
      @endforeach
    </ul>
  </div>
</div>
@endif
```

40. Save, lalu coba lagi isi bagian NIS nya saja, maka hasilnya akan seperti pada gambar

Data Siswa

localhost:8000/siswa/create

- The nama field is required.
- The jurusan field is required.

NIS

Nama

Jurusan

SIMPAN

41. Karena untuk peringatannya masih default, maka kita ganti dengan peringatan sesuai yg kita inginkan. Buka file **siswaController** lalu tuliskan seperti di bawah ini.

```
Go Run Terminal Help siswaController.php - latihan_crud1 - Visual Studio Code
siswaController.php X siswa.php template.blade.php index.blade.php
app > Http > Controllers > siswaController.php > PHP Intelephense > siswaController >
34 * @return \Illuminate\Http\Response
35 */
36 public function store(Request $request)
37 {
38     $request->validate([
39         'nis'=> 'required|numeric|unique:siswa,nis',
40         'nama'=> 'required',
41         'jurusan'=> 'required',
42     ],[
43         'nis.required'=>'NIS harus diisi',
44         'nis.numeric'=>'NIS harus dalam angka',
45         'nis.unique'=>'NIS sudah ada',
46         'nama.required'=>'Nama harus diisi',
47         'jurusan.required'=>'Jurusan harus diisi',
48     ]);
49     $data =[
50         'nis'=> $request->nis,
51         'nama'=> $request->nama,
52         'jurusan'=> $request->jurusan,
53     ];
54     siswa::create($data);
```

42. Sekarang kita coba, jika nis nya sama, nama dan jurusan dikosongkan. Maka hasilnya.

Data Siswa

localhost:8000/siswa/create

- NIS sudah ada
- Nama harus diisi
- Jurusan harus diisi

NIS

Nama

Jurusan

SIMPAN

43. Agar kita tau apa saja yg sudah kita input kan dalam form create, maka kita perlu memasukan script dibawah ini pada file **siswaController** di function store.

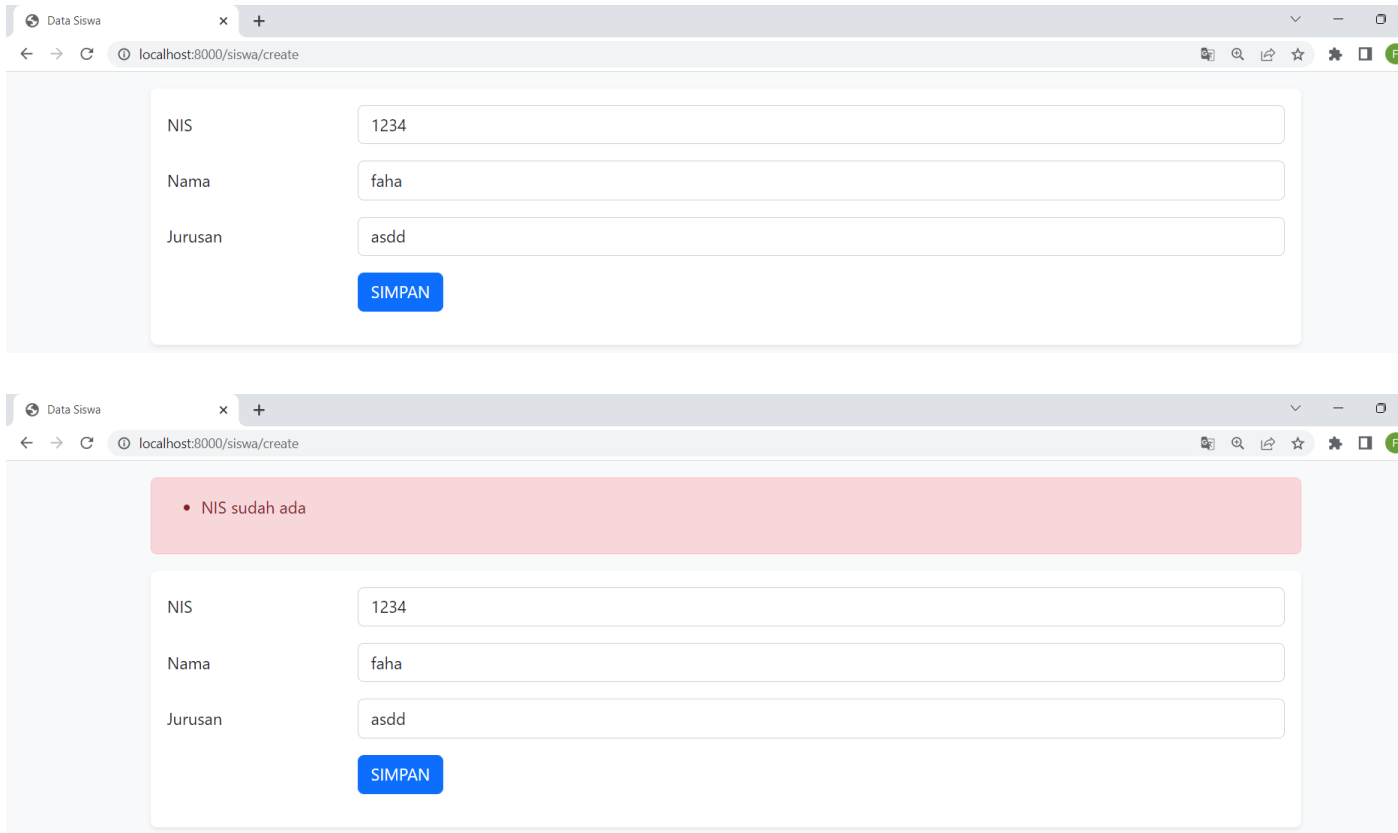
```
Controller.php X siswa.php template.blade.php ind
ttp > Controllers > siswaController.php > PHP Intelephense > siswaCont

* Store a newly created resource in storage.
*
* @param \Illuminate\Http\Request $request
* @return \Illuminate\Http\Response
*/
public function store(Request $request)
{
    Session::flash('nis', $request->nis);
    Session::flash('nama', $request->nama);
    Session::flash('jurusan', $request->jurusan);
}
```

44. Lalu buka file **create.blade.php** masukan **value="{{Session::get('nis')}}"** pada inputan nis seperti dibawah ini dan lakukan hal yg sama pada nama dan jurusan

```
<!-- START FORM -->
<form action='{{url('siswa')}}' method='post'>
    @csrf
    <div class="my-3 p-3 bg-body rounded shadow-sm">
        <div class="mb-3 row">
            <label for="nim" class="col-sm-2 col-form-label">NIS</label>
            <div class="col-sm-10">
                <input type="number" class="form-control" name='nis' value="{{Session::get('nis')}}" id="nis">
            </div>
        </div>
        <div class="mb-3 row">
            <label for="nama" class="col-sm-2 col-form-label">Nama</label>
            <div class="col-sm-10">
                <input type="text" class="form-control" name='nama' value="{{Session::get('nama')}}" id="nama">
            </div>
        </div>
        <div class="mb-3 row">
            <label for="jurusan" class="col-sm-2 col-form-label">Jurusan</label>
            <div class="col-sm-10">
                <input type="text" class="form-control" name='jurusan' value="{{Session::get('jurusan')}}" id="jurusan">
            </div>
        </div>
    </div>
</form>
```

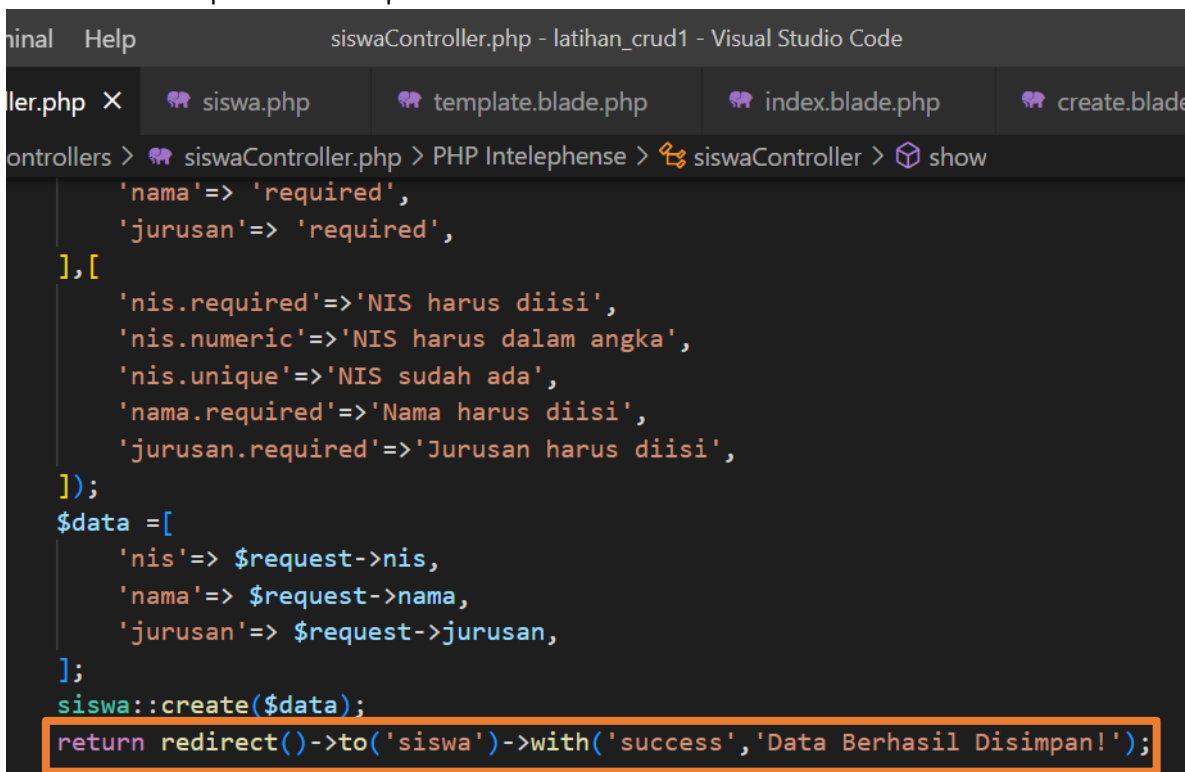
45. Kita coba reload lagi pada browser dan hasilnya akan seperti ini



The first screenshot shows a web browser window with the URL `localhost:8000/siswa/create`. The page contains a form with three input fields: NIS (value: 1234), Nama (value: faha), and Jurusan (value: asdd). Below the fields is a blue button labeled "SIMPAN".

The second screenshot shows the same browser window after a reload. A red error message "NIS sudah ada" is displayed above the form. The form fields and the "SIMPAN" button are still visible below the message.

46. Kita akan menambahkan pesan lagi jika data sudah masuk ke dalam database. Masuk ke file **siswaController** lalu masukan script dibawah ini pada **function store** .



```
siswaController.php - latihan_crud1 - Visual Studio Code

siswaController.php > PHP Intelephense > siswaController > show

    'nama'=> 'required',
    'jurusan'=> 'required',
], [
    'nis.required'=>'NIS harus diisi',
    'nis.numeric'=>'NIS harus dalam angka',
    'nis.unique'=>'NIS sudah ada',
    'nama.required'=>'Nama harus diisi',
    'jurusan.required'=>'Jurusan harus diisi',
]);
$data = [
    'nis'=> $request->nis,
    'nama'=> $request->nama,
    'jurusan'=> $request->jurusan,
];
siswa::create($data);
return redirect()->to('siswa')->with('success','Data Berhasil Disimpan!');
```

47. Kita arahkan ke **template.blade.php**. buka **template.blade.php** lalu masukan perintah di bawah ini

```
Terminal Help template.blade.php - latihan_crud1 - Visual Studio Code

Controller.php siswa.php template.blade.php X index.blade.p

s > views > layout > template.blade.php > html

<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-sc
    <title>Data Siswa</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist
    rel="stylesheet" integrity="sha384-iYQeCzEYFbKjA/T2uDLTpkwGzC
  </head>
  <body class="bg-light">
    <main class="container">
      @if (Session::has('success'))
        <div class="pt-3">
          <div class="alert alert-success">
            {{Session::get('success')}}
          </div>
        </div>
      @endif
      @yield('konten')
    </main>
```

48. Saat kita inputkan data dan klik simpan maka halamannya akan mengarahkan ke tampilan siswa, pemberitahuan berhasil ada bagian atas, dan data masuk ke database, seperti pada gambar

Data Siswa

localhost:8000/siswa/create

NIS: 10008

Nama: Azron

Jurusan: RPL

SIMPAN

wa

localhost:8000/siswa

Data Berhasil Disimpan!

Masukkan kata kunci **Cari**

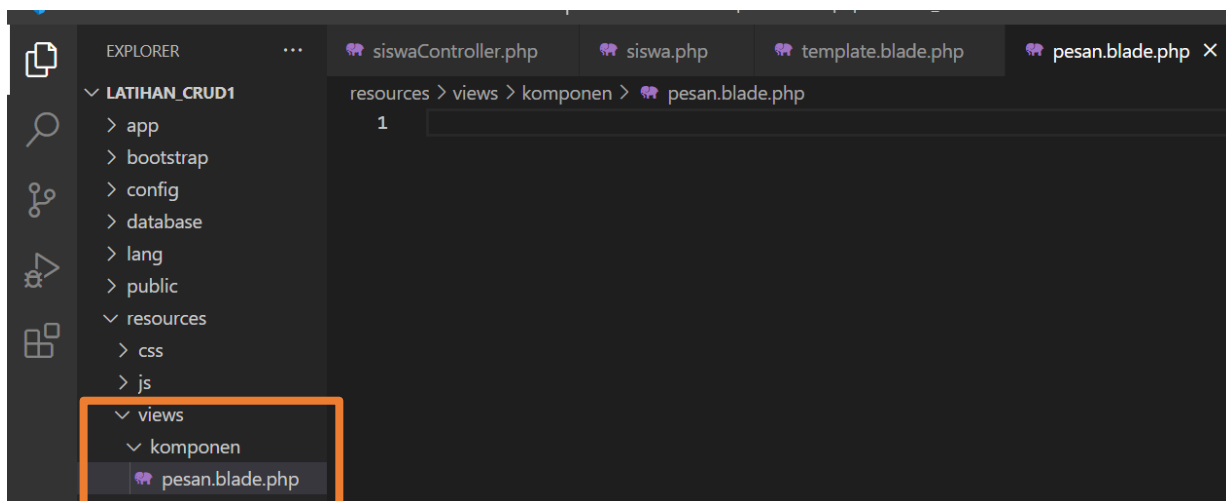
+ Tambah Data

No	NIS	Nama	Jurusan	Aksi
1	1001	Fajar	Rekayasa Perangkat Lunak	Edit Del

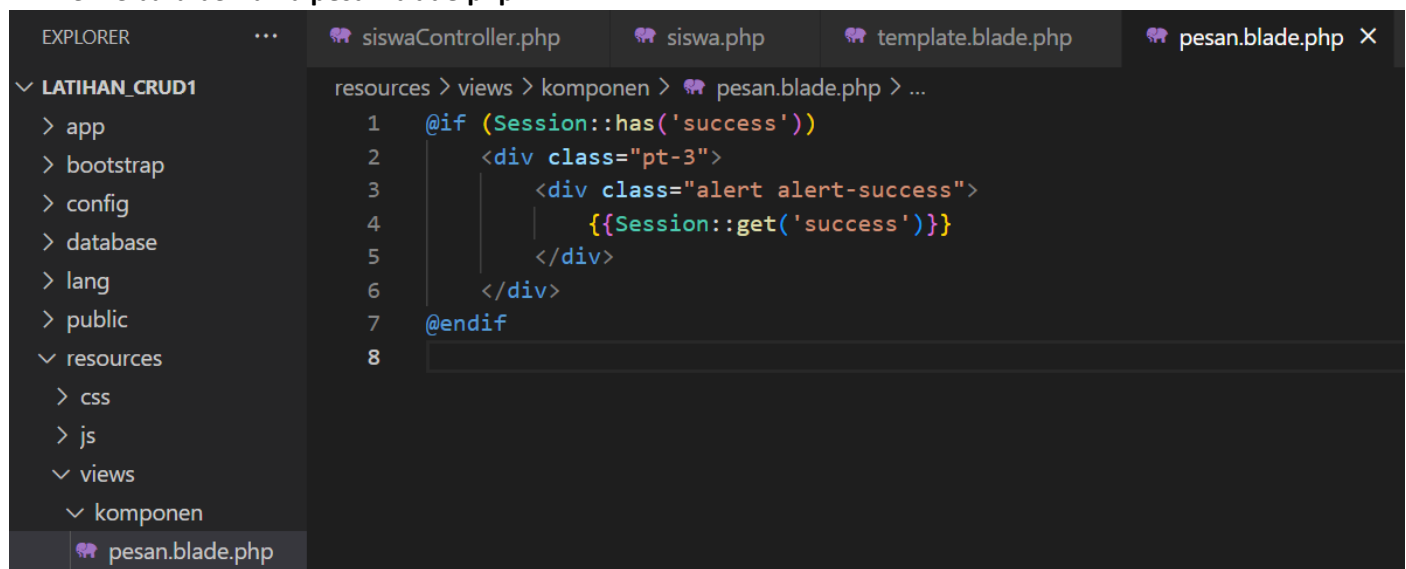
				nis	nama	jurusan
<input type="checkbox"/>				123	asdas	asdas
<input type="checkbox"/>				255	ghgfh	jhkjkhjh
<input type="checkbox"/>				1234	asdas	asdas
<input type="checkbox"/>				10008	Azron	RPL
<input type="checkbox"/>				12346	Sonia	Perkantoran

MENAMBAHKAN BLADE PESAN

49. Langkah pertama kita buat dulu folder baru bernama **komponen** di dalam folder views, lalu kita buat file baru di dalamnya bernama **pesan.blade.php**. Seperti pada gambar



50. Agar terlihat rapih, **@if (Session::has('success'))** sampai **@endif** pada file **template.blade.php** kita pindahkan ke file baru bernama **pesan.blade.php**

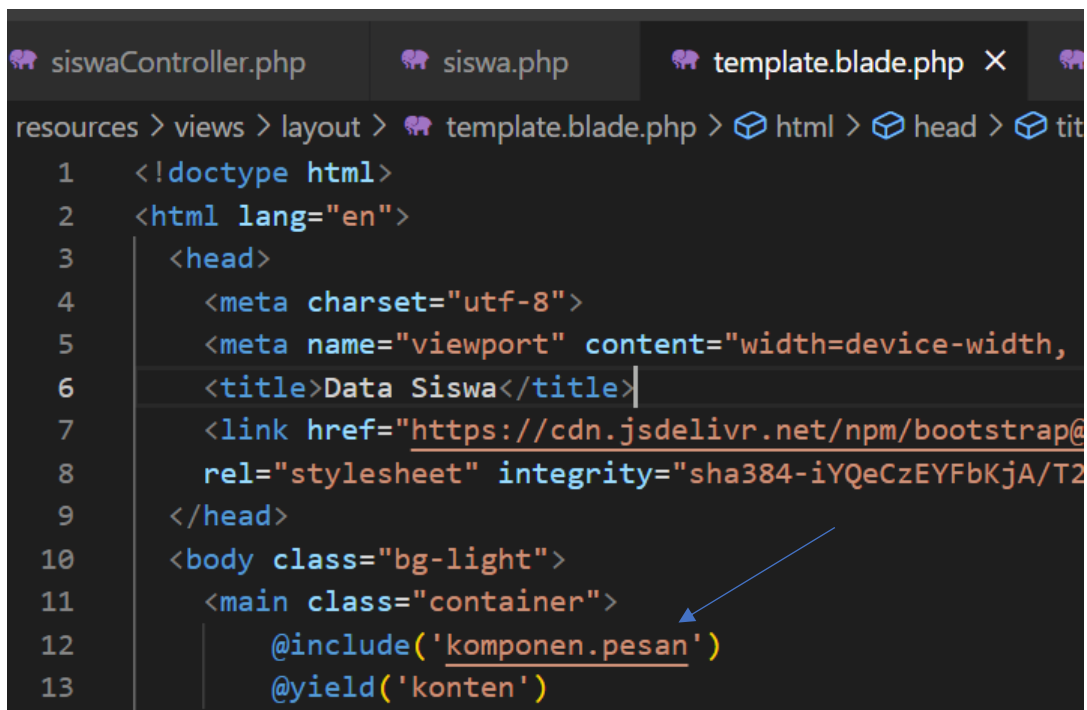


51. Lalu pindahkan juga `@if ($errors->any())` sampai `@endif` yang ada di dalam file **create.blade.php** ke dalam file **pesan.blade.php**



```
resources > views > komponen > pesan.blade.php > ...
1  @if (Session::has('success'))
2      <div class="pt-3">
3          <div class="alert alert-success">
4              {{Session::get('success')}}
5          </div>
6      </div>
7  @endif
8
9  @if ($errors->any())
10     <div class="pt-3">
11         <div class="alert alert-danger">
12             <ul>
13                 @foreach ($errors->all() as $item)
14                     <li>{{ $item }}</li>
15                 @endforeach
16             </ul>
17         </div>
18     </div>
19 @endif
```

52. Sekarang kita tinggal panggil **pesan.blade.php** ke dalam **template.blade.php**. cara memanggilnya seperti pada gambar



```
resources > views > layout > template.blade.php > html > head > tit
1  <!doctype html>
2  <html lang="en">
3      <head>
4          <meta charset="utf-8">
5          <meta name="viewport" content="width=device-width,
6          <title>Data Siswa</title>
7          <link href="https://cdn.jsdelivr.net/npm/bootstrap@
8          rel="stylesheet" integrity="sha384-iYQeCzEYFbKjA/T2
9      </head>
10     <body class="bg-light">
11         <main class="container">
12             @include('komponen.pesan')
13             @yield('konten')
```

53. Setelah itu kita coba hasilnya. Maka akan seperti pada gambar

• NIS sudah ada
• Nama harus diisi
• Jurusan harus diisi

NIS

Nama

Jurusan

SIMPAN

54. Agar tombol tambah data bisa berfungsi untuk mengarahkan ke halaman tambah data, maka masukan script dibawah ini

```

siswaController.php  siswa.php  template.blade.php  pesan.blade.php  index.blade.php X
resources > views > siswa > index.blade.php > div.my-3.p-3.bg-body.rounded.shadow-sm > div.pb-3 > form.d-flex
4      <!-- START DATA -->
5      <div class="my-3 p-3 bg-body rounded shadow-sm">
6          <!-- FORM PENCARIAN -->
7          <div class="pb-3">
8              <form class="d-flex" action="" method="get">
9                  <input class="form-control me-1" type="search" name="katakunci" value="{<
10                  <button class="btn btn-secondary" type="submit">Cari</button>
11              </form>
12          </div>
13
14          <!-- TOMBOL TAMBAH DATA -->
15          <div class="pb-3">
16              <a href="{{url('siswa/create')}}" class="btn btn-primary">+ Tambah Data</a>
17          </div>

```

MENAMPILKAN DATA

55. Sekarang kita buka file **siswaController.php** . Sesuai dengan apa yg akan kita tampilkan data dalam database. Maka kita masuk ke **function index**. Lalu masukan script di bawah ini

```

Go Run Terminal Help  siswaController.php - latihan_crud1 - Visual S
siswaController.php X  siswa.php  template.blade.php  pe
app > Http > Controllers > siswaController.php > PHP Intelephense > siswaCor
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6  use App\Models\siswa;
7  use Illuminate\Support\Facades\Session;
8
9  class siswaController extends Controller
10 {
11     /**
12      * Display a listing of the resource.
13      *
14      * @return \Illuminate\Http\Response
15      */
16     public function index()
17     {
18         $data=siswa::orderBy('nis','desc')->get();
19         return view('siswa.index')->with(['data',$data]);

```

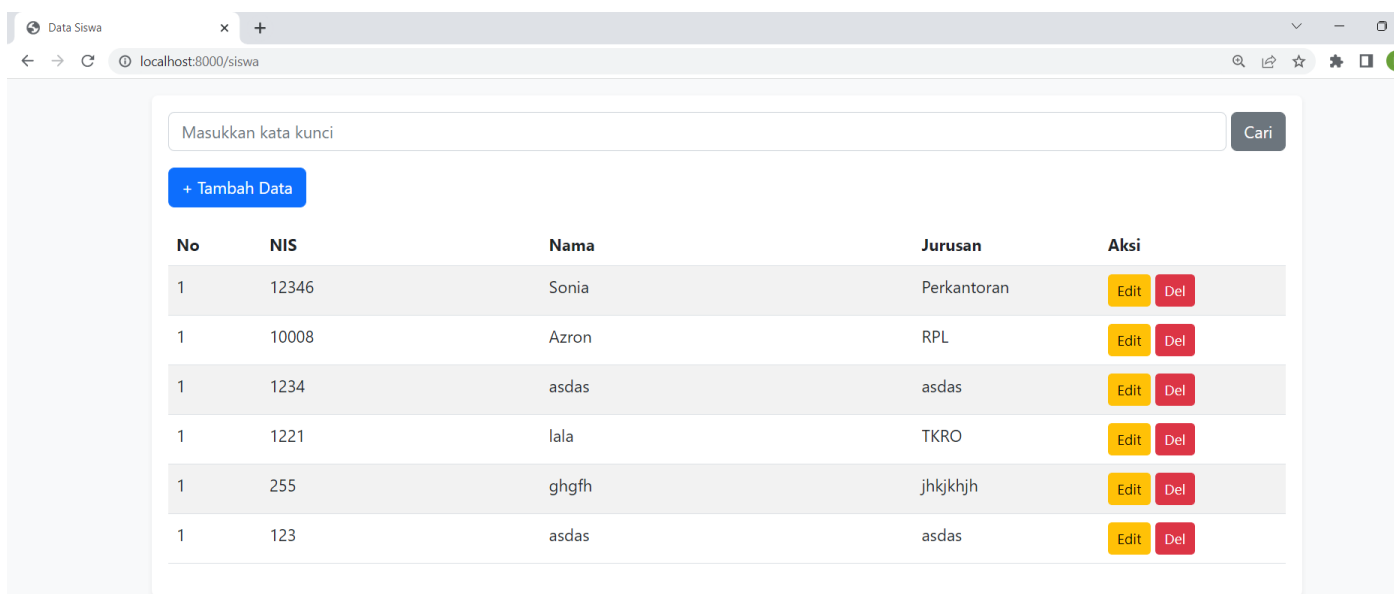
56. Buka file **index.blade.php** lalu kita ubah bagian dibawah ini

```
<tbody>
  <tr>
    <td>1</td>
    <td>1001</td>
    <td>Fajar</td>
    <td>Rekayasa Perangkat Lunak</td>
    <td>
```

Menjadi seperti di bawah ini

```
<tbody>
  @foreach ( $data as $item )
    <tr>
      <td>1</td>
      <td>{{ $item->nis }}</td>
      <td>{{ $item->nama }}</td>
      <td>{{ $item->jurusan }}</td>
      <td>
        <a href='' class="btn btn-warning btn-sm">Edit</a>
        <a href='' class="btn btn-danger btn-sm">Del</a>
      </td>
    </tr>
  @endforeach
</tbody>
```

57. Lalu kita coba reload browser kita, dan hasilnya akan seperti di bawah ini



No	NIS	Nama	Jurusan	Aksi
1	12346	Sonia	Perkantoran	<button>Edit</button> <button>Del</button>
1	10008	Azron	RPL	<button>Edit</button> <button>Del</button>
1	1234	asdas	asdas	<button>Edit</button> <button>Del</button>
1	1221	lala	TKRO	<button>Edit</button> <button>Del</button>
1	255	ghgfh	jhgkjh	<button>Edit</button> <button>Del</button>
1	123	asdas	asdas	<button>Edit</button> <button>Del</button>

MEMBUAT PAGINATION

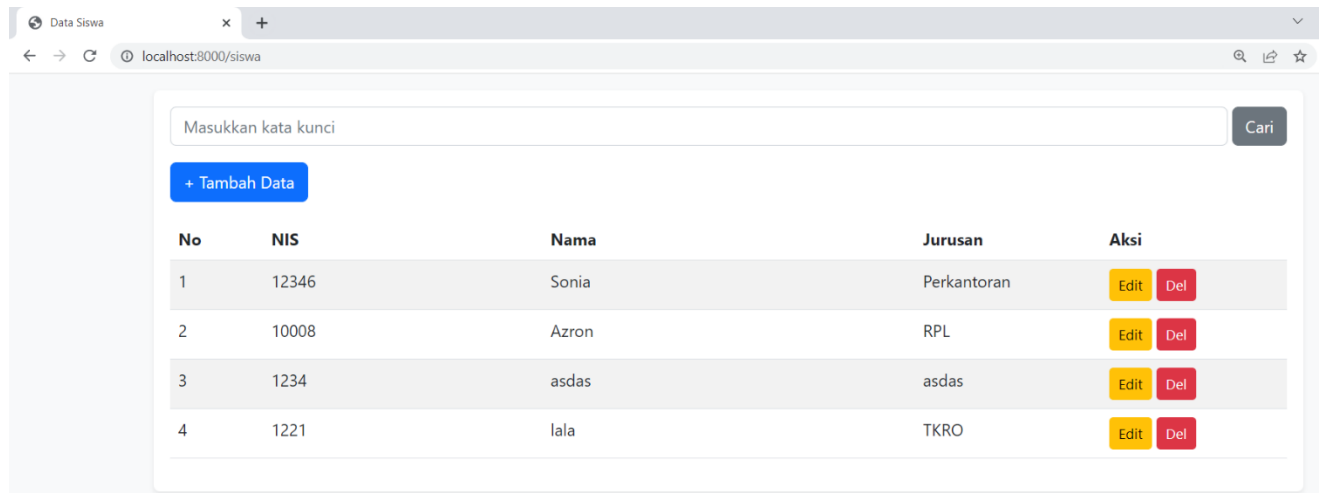
58. Masih dalam fil **index.blade.php** , kita tambahkan script seperti di bawah agar bagian no pada table bertambah

```
<tbody>
    <?php $i =$data->firstItem()?>
    @foreach ( $data as $item )
    <tr>
        <td>{{ $i }}</td>
        <td>{{ $item->nis }}</td>
        <td>{{ $item->nama }}</td>
        <td>{{ $item->jurusan }}</td>
        <td>
            <a href='' class="btn btn-warning btn-sm">Edit</a>
            <a href='' class="btn btn-danger btn-sm">Del</a>
        </td>
    </tr>
    <?php $i++?>
    @endforeach
</tbody>
```

59. Lalu kita ubah **get()** pada bagian function index di file **siswaController.php** menjadi seperti di bawah ini

```
Go Run Terminal Help siswaController.php - latihan_crud1 - Visual Studio Code
siswaController.php X siswa.php template.blade.php pesan.blade.p
app > Http > Controllers > siswaController.php > PHP Intelephense > siswaController >
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6 use App\Models\siswa;
7 use Illuminate\Support\Facades\Session;
8
9 class siswaController extends Controller
10 {
11     /**
12      * Display a listing of the resource.
13      *
14      * @return \Illuminate\Http\Response
15      */
16     public function index()
17     {
18         $data=siswa::orderBy('nis','desc')->paginate(4);
19         return view('siswa.index')->with('data',$data);
20     }
```

60. Maka hasilnya akan jadi seperti di bawah ini



Masukkan kata kunci

No	NIS	Nama	Jurusan	Aksi
1	12346	Sonia	Perkantoran	<input type="button" value="Edit"/> <input type="button" value="Del"/>
2	10008	Azron	RPL	<input type="button" value="Edit"/> <input type="button" value="Del"/>
3	1234	asdas	asdas	<input type="button" value="Edit"/> <input type="button" value="Del"/>
4	1221	lala	TKRO	<input type="button" value="Edit"/> <input type="button" value="Del"/>

61. Sekarang kita tambahkan tombol next dan back pada bagian bawah table. Masukan script di bawah ini ke dalam file **index.blade.php** pada bagian bawah, tepatnya ada di bawah **</table>**

```
</tbody>
</table>
{{ $data->links() }}
</div>
<!-- AKHIR DATA -->
@endsection
```

62. Masih di dalam file **index.blade.php** kita tekan **ctrl+p** lalu cari **AppServiceProvider.php** setelah itu masukan script dibawah ini

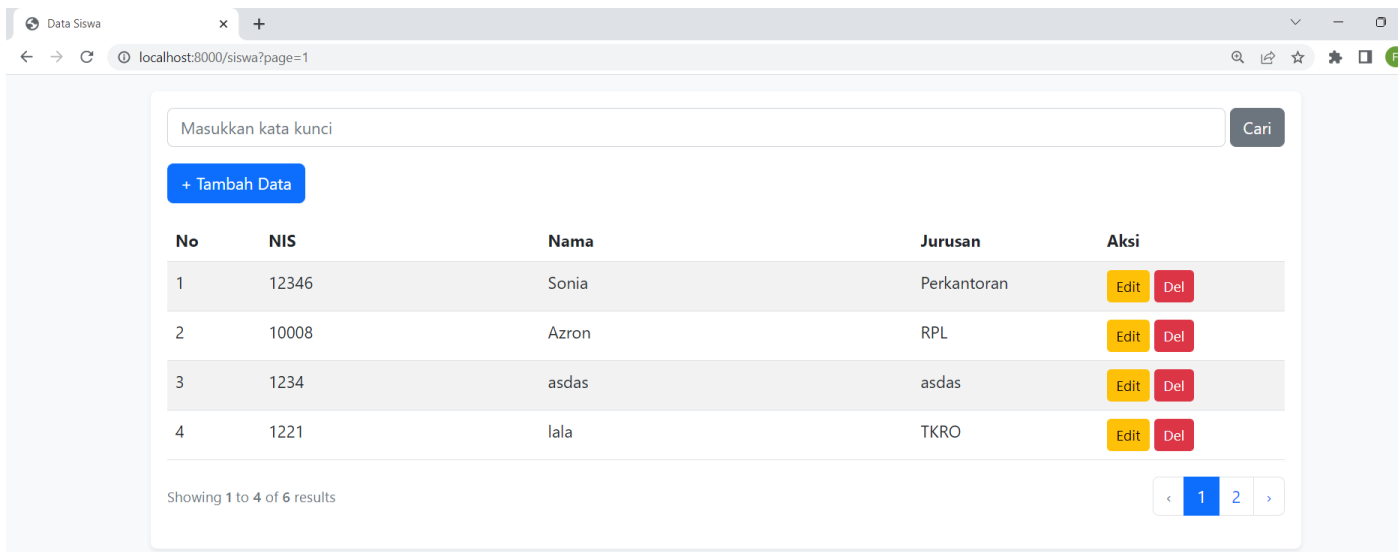
Pada bagian atas

```
use Illuminate\Pagination\Paginator;
```

Pada bagian **function boot**

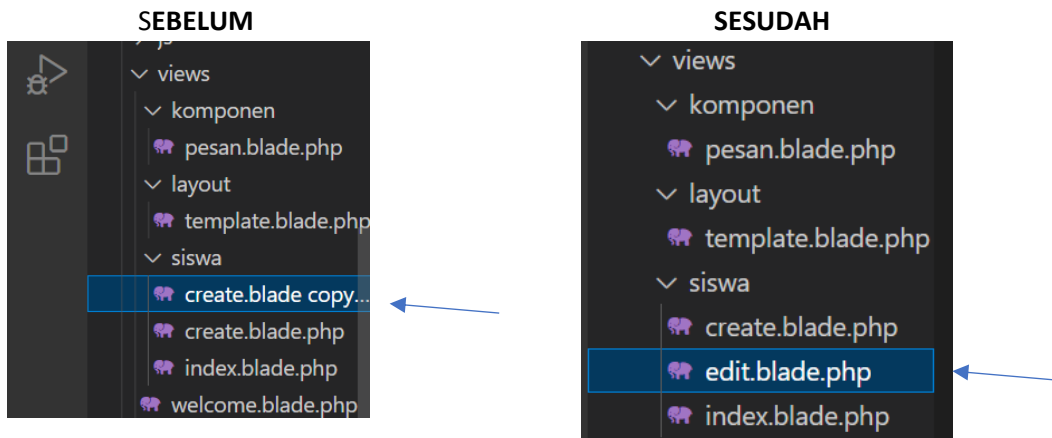
```
public function boot()
{
    Paginator::useBootstrapFive();
}
```

63. Lalu kita coba reload browser kita, maka hasilnya akan seperti ini



MEMBUAT FUNGSI EDIT

64. Kita buat dulu file edit dengan cara, copy file create.blade.php lalu paste, maka akan muncul file bernama **create.blade copy.php**. maka harus kita ganti dengan nama **edit.blade.php** seperti pada gambar dibawah



65. Masuk ke **siswaController.php** lalu masukan script di bawah ini pada function edit

```
public function edit($id)
{
    $data=siswa::where('nis',$id)->first();
    return view('siswa.edit')->with('data',$data);
}
```

66. Sekarang kita buka file **edit.blade.php** . karena pada saat edit data kita menginginkan untuk kolom nis tidak bisa di edit, maka ganti **value** nya seperti ini

```

<div class="mb-3 row">
    <label for="nim" class="col-sm-2 col-form-label">NIS</label>
    <div class="col-sm-10">
        {{{data->nis}}}
    </div>
</div>
<div class="mb-3 row">
    <label for="nama" class="col-sm-2 col-form-label">Nama</label>
    <div class="col-sm-10">
        <input type="text" class="form-control" name='nama' value="{{data->nama}}" id="nama">
    </div>
</div>
<div class="mb-3 row">
    <label for="jurusan" class="col-sm-2 col-form-label">Jurusan</label>
    <div class="col-sm-10">
        <input type="text" class="form-control" name='jurusan' value="{{data->jurusan}}"
        id="jurusan">
    </div>
</div>

```

67. Kita tambahkan tombol kembali pada halaman edit dan create dengan memasukan script di bawah ini

```

@extends('layout.template')
@section('konten')

<!-- START FORM -->
<form action='{{url('siswa')}}' method='post'>
    @csrf
    <div class="my-3 p-3 bg-body rounded shadow-sm">
        <a href="{{url('siswa')}}" class="btn btn-secondary"><< Kembali</a>

```

68. Kita tambahkan script di bawah ini pada file **index.blade.php** di tombol edit

```

<a href='{{url('siswa/'.$item->nis.'/edit')}}'
    class="btn btn-warning btn-sm">Edit</a>
<a href='' class="btn btn-danger btn-sm">Del</a>

```

69. Buka file **edit.blade.php** lalu tambahkan script seperti dibawah ini

```

@extends('layout.template')
@section('konten')

<!-- START FORM -->
<form action='{{url('siswa/'.$data->nis)}}' method='post'>
    @csrf
    @method('PUT')

```

70. Buka file **siswaController.php** lalu pada **function update** masukan script di bawah ini

```
public function update(Request $request, $id)
{
    $request->validate([
        'nama'=>'required',
        'jurusan'=>'required',
    ], [
        'nama.required'=>'Nama harus diisi!',
        'jurusan.required'=>'Jurusan harus diisi!',
    ]);
    $data = [
        'nama'=>$request->nama,
        'jurusan'=>$request->jurusan,
    ];

    siswa::where('nis',$id)->update($data);
    return redirect()->to('siswa')->with('success','Data berhasil diperbaharui');
```

71. Sekarang reload browsernya dan kita coba ganti maka hasilnya seperti di bawah ini

The first screenshot shows a web application interface with a search bar labeled 'Masukkan kata kunci' and a 'Cari' button. Below the search bar is a '+ Tambah Data' button. The main content is a table with the following columns: No, NIS, Nama, Jurusan, and Aksi. The table contains 4 rows of data:

No	NIS	Nama	Jurusan	Aksi
1	12346	Sonia1	Perkantoran	Edit Del
2	10008	Azron	RPL	Edit Del
3	1234	asdas	asdas	Edit Del
4	1221	lala	TKRO	Edit Del

Below the table, it says 'Showing 1 to 4 of 6 results' and there are pagination controls for 1 and 2.

The second screenshot shows the same interface after a successful update. A green message box at the top says 'Data berhasil diperbaharui'. The table now shows the first row updated to 'Sonia12'.

No	NIS	Nama	Jurusan	Aksi
1	12346	Sonia12	Perkantoran	Edit Del
2	10008	Azron	RPL	Edit Del
3	1234	asdas	asdas	Edit Del
4	1221	lala	TKRO	Edit Del

Below the table, it says 'Showing 1 to 4 of 6 results' and there are pagination controls for 1 and 2.

MEMBUAT FUNGSI DELETE

72. Untuk membuat fungsi delete, pada file **index.blade.php** kita ganti `<a href=` untuk delete jadi menggunakan `<form>` seperti pada gambar.

SEBELUM

```
<a href=' ' class="btn btn-danger btn-sm">Del</a>
```

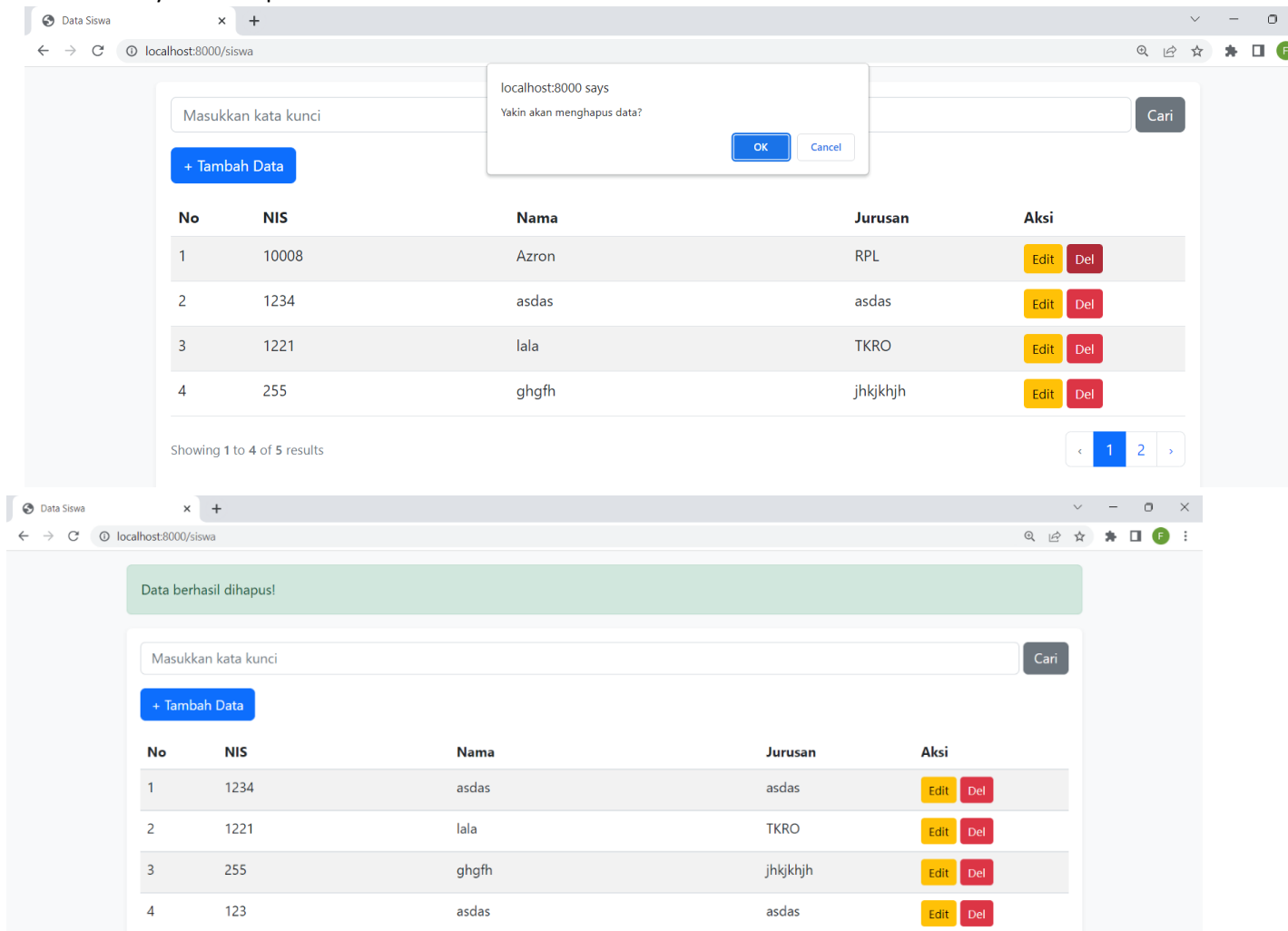
SESUDAH

```
<form onsubmit="return confirm ('Yakin akan menghapus data?')"
class='d-inline' action="{{ 'siswa/' . $item->nis }}" method="POST">
    @csrf
    @method('DELETE')
    <button type="submit" name="submit" class="btn btn-danger btn-sm">Del</button>
</form>
```

73. Lalu masukan script di bawah ini pada **function destroy** yg ada di file **siswaController.blade.php**

```
public function destroy($id)
{
    siswa::where('nis', $id)->delete();
    return redirect()->to('siswa')->with('success','Data berhasil dihapus!');
}
```

74. Maka hasilnya akan seperti ini



MEMBUAT FUNGSI PENCARIAN

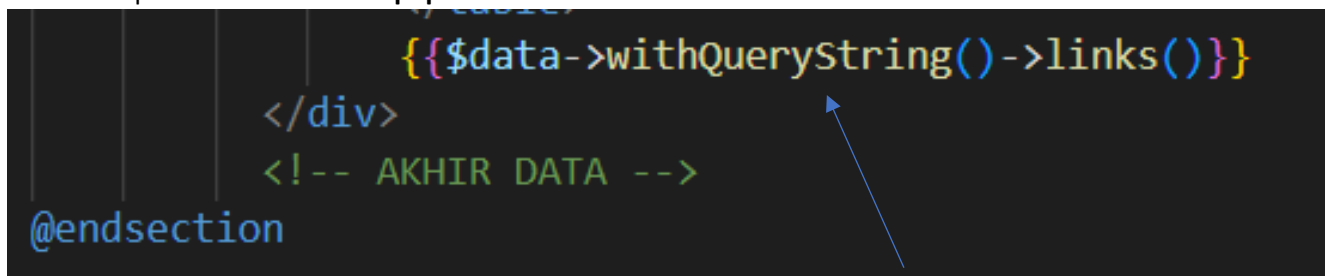
75. Sekarang kita buka halaman **index.blade.php** kita tambahkan action pada tombol cari, dengan menulis perintah di bawah ini



```
Controller.php  index.blade.php  create.blade.php  edit.blade.php
es > views > siswa > index.blade.php > div.my-3.p-3.bg-body.rounded.shadow-sm > div.pb-3
@extends('layout.template')

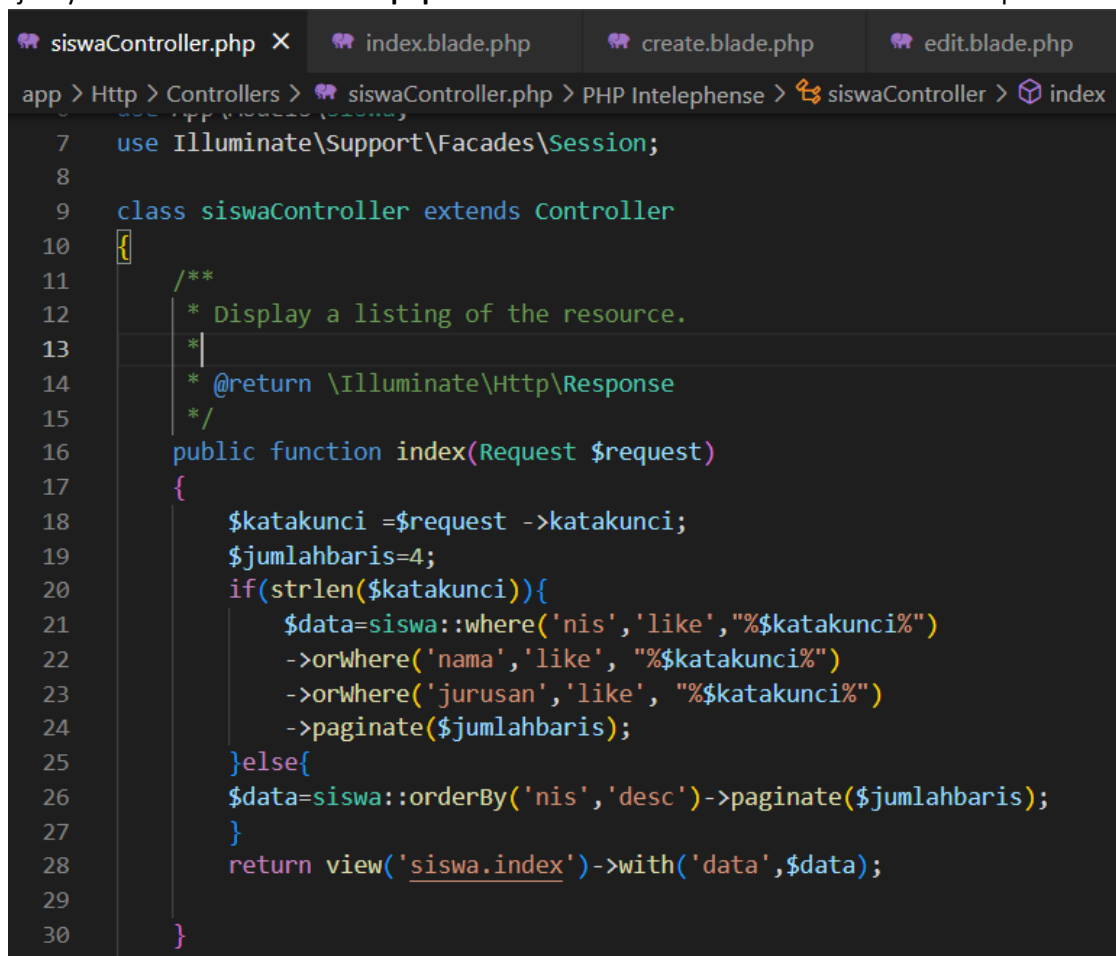
@section('konten')
    <!-- START DATA -->
    <div class="my-3 p-3 bg-body rounded shadow-sm">
        <!-- FORM PENCARIAN -->
        <div class="pb-3">
            <form class="d-flex" action="{{url('siswa')}}" method="get">
                <input class="form-control me-1" type="search" name="katakunci" value="{{ Request::get('katakunci') }}" placeholder="Masukkan
                <button class="btn btn-secondary" type="submit">Cari</button>
            </form>
        </div>
    </div>
    <!-- AKHIR DATA -->
@endsection
```

76. Masih pada file **index.blade.php** kita tambahkan



```
</div>
<!-- AKHIR DATA -->
{{ $data->withQueryString()->links() }}
@endsection
```

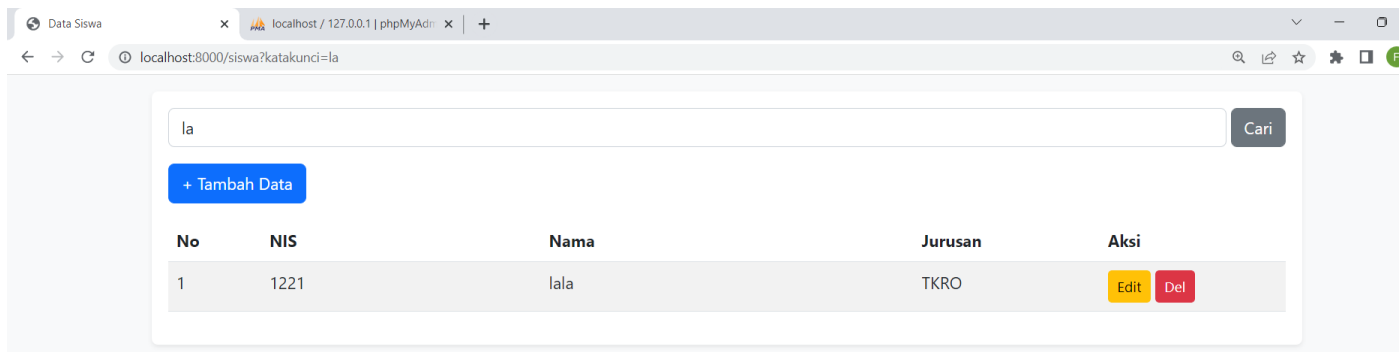
77. Selanjutnya buka file **siswaController.php** lalu masuk ke **function index** dan masukan script dibawah ini



```
siswaController.php  index.blade.php  create.blade.php  edit.blade.php
app > Http > Controllers > siswaController.php > PHP Intelephense > siswaController > index

7 use Illuminate\Support\Facades\Session;
8
9 class siswaController extends Controller
10 {
11     /**
12      * Display a listing of the resource.
13      *
14      * @return \Illuminate\Http\Response
15      */
16     public function index(Request $request)
17     {
18         $katakunci = $request->katakunci;
19         $jumlahbaris=4;
20         if(strlen($katakunci)){
21             $data=siswa::where('nis','like','"%$katakunci%"')
22                 ->orWhere('nama','like','"%$katakunci%"')
23                 ->orWhere('jurusan','like','"%$katakunci%"')
24                 ->paginate($jumlahbaris);
25         }else{
26             $data=siswa::orderBy('nis','desc')->paginate($jumlahbaris);
27         }
28         return view('siswa.index')->with('data',$data);
29     }
30 }
```

78. Sekarang kita coba hasilnya. Maka akan tampil seperti pada gambar



79. Untuk CRUD dasar sudah selesai.

D. TUGAS!

1. Buatlah tombol dan fungsi sorting Ascending dan Descending pada NIS, Nama, Jurusan
2. Buatlah fitur login dan register dihubungkan langsung ke form CRUD