

# **Laporan Ujian Akhir Semester Praktik Pemrograman Front End**



**Disusun oleh :**

**Riki Adhi Nugroho  
V3422060  
Kelas C**

**Dosen pengampu :**

**Eko Harry Pratisto, S.T., M.Info.Tech., Ph.D.  
NIP. 1981112420130201**

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS SEKKOLAH VOKASI  
UNIVERSITAS SEBELAS MARET  
2023**

## Aplikasi CRUD dan User Role menggunakan Vue dan Laravel

### 1. Laravel sebagai **Praktikum\_Vue\_Laravel\_CRUD2**

#### a. Membuat Project Laravel baru

```
PS C:\xampp\htdocs\laravel_FrontEnd> laravel new Praktikum_Vue_Laravel_CRUD2

  L A R A V E L

Would you like to install a starter kit?:
[none] No starter kit
[breeze] Laravel Breeze
[jetstream] Laravel Jetstream
> none

Which testing framework do you prefer?:
[0] PHPUnit
[1] Pest
> 0

Creating a "laravel/laravel" project at "./Praktikum_Vue_Laravel_CRUD2"
Installing laravel/laravel (v10.2.8)
- Installing laravel/laravel (v10.2.8): Extracting archive
Created project in C:\xampp\htdocs\laravel_FrontEnd\Praktikum_Vue_Laravel_CRUD2
> @php -r "file_exists('.env') || copy('.env.example', '.env');"
Loading composer repositories with package information
Updating dependencies
```

Masuk ke folder yang telah dibuat (**Praktikum\_Vue\_Laravel\_CRUD2**)

Pastikan Laravel telah terinstal dengan baik dengan perintah di bawah ini dan buka <http://localhost:8000> pada browser.

```
PS C:\xampp\htdocs\laravel_FrontEnd> cd .\Praktikum_Vue_Laravel_CRUD2\
PS C:\xampp\htdocs\laravel_FrontEnd\Praktikum_Vue_Laravel_CRUD2> php artisan serve

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

Press Ctrl+C to stop the server
```

#### b. Setting database

Hubungkan database dari Latihan sebelumnya dengan folder project didalam file .env. Sesuaikan dengan konfigurasi pada sistem Anda.

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=vue-laravel
DB_USERNAME=root
DB_PASSWORD=root
```

c. Membuat file Model Siswa

Buat model dan migration untuk Siswa dengan menjalankan script :

```
PS C:\xampp\htdocs\laravel_FrontEnd\Praktikum_Vue_Laravel_CRUD2> php artisan make:model Siswa -m

INFO Model [C:\xampp\htdocs\laravel_FrontEnd\Praktikum_Vue_Laravel_CRUD2\app\Models\Siswa.php]
created successfully.

INFO Migration [C:\xampp\htdocs\laravel_FrontEnd\Praktikum_Vue_Laravel_CRUD2\database\migrations\2023_11_14_015031_create_siswas_table.php] created successfully.

PS C:\xampp\htdocs\laravel_FrontEnd\Praktikum_Vue_Laravel_CRUD2> |
```

Dua file akan terbentuk yaitu:

- File model “app/Models/Siswa.php”
  - File migration “../database/migrations/....create\_siswas\_table.php”

Buka file migration pada “../database/migrations/....create\_siswas\_table.php” dan modifikasi pada blok fungsi up :

```
public function up()
{
    Schema::create('siswas', function (Blueprint $table) {
        $table->id();
        $table->text('nama')->nullable();
        $table->text('alamat')->nullable();
        $table->text('kota')->nullable();
        $table->text('provinsi')->nullable();
        $table->text('email')->nullable();
        $table->timestamps();
    });
}
```

Modifikasi juga file “app/Models/Siswa.php” pada bagian Berikut :

```
class Siswa extends Model
{
    use HasFactory;

    protected $table = 'siswas';
    protected $fillable = [
        'nama', 'alamat', 'kota', 'provinsi', 'email'
    ];
}
```

Protected properti dengan nama \$fillable memungkinkan field yang ditambahkan pada property tersebut dapat melakukan manipulasi data.

d. Membuat data dummy untuk table siswas

1. Download package Faker dengan menjalankan script :

```
PS C:\xampp\htdocs\laravel\Praktikum_Vue_Laravel_CRUD> composer require fakerphp/faker --dev
./composer.json has been updated
Running composer update fakerphp/faker
Loading composer repositories with package information
Updating dependencies
Nothing to modify in lock file
Writing lock file
Installing dependencies from lock file (including require-dev)
Nothing to install, update or remove
Generating optimized autoload files
> Illuminate\Foundation\ComposerScripts::postAutoloadDump
> @php artisan package:discover --ansi

 INFO  Discovering packages.

laravel/sail ..... DONE
laravel/sanctum ..... DONE
laravel/tinker ..... DONE
nesbot/carbon ..... DONE
nunomaduro/collision ..... DONE
nunomaduro/termwind ..... DONE
spatie/laravel-ignition ..... DONE

82 packages you are using are looking for funding.
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.
Using version ^1.23 for fakerphp/faker
PS C:\xampp\htdocs\laravel\Praktikum_Vue_Laravel_CRUD>
```

Nama folder berbeda karena saya sudah pernah menginstall data dummy pada folder sebelumnya, tetapi masih menggunakan database yang sama.

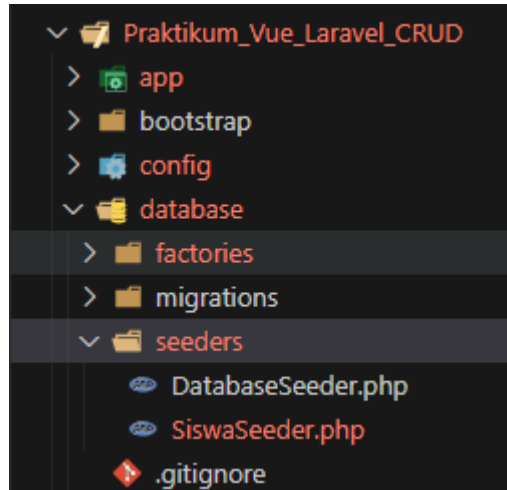
2. Membuat seeding class dengan nama SiswaSeeder dengan menjalankan script :

```
PS C:\xampp\htdocs\laravel\Praktikum_Vue_Laravel_CRUD> php artisan make:seeder SiswaSeeder

 INFO  Seeder [C:\xampp\htdocs\laravel\Praktikum_Vue_Laravel_CRUD\database\seeders\SiswaSeeder.php] created successfully.

PS C:\xampp\htdocs\laravel\Praktikum_Vue_Laravel_CRUD>
```

3. Akan terbentuk file “app\database\seeders\SiswaSeeder.php”



4. Membuat Factory file

Tambahkan extension “laravel artisan”

pada VS code Tekan tombol Ctrl + Shift + P

Cari Artisan: Make Factory dan

tekan Enter Factory Name isikan

SiswaFactory

Pilih Yes untuk menghubungkan

dengan model Ketik Siswa

(Alternatif lain membuat Factory file dengan script: php artisan make:factory SiswaFactory)

```
PS C:\xampp\htdocs\laravel_FrontEnd\Praktikum_Vue_Laravel_CRUD2\backend> php artisan make:factory Siswa
INFO Factory [C:\xampp\htdocs\laravel_FrontEnd\Praktikum_Vue_Laravel_CRUD2\backend\database\factories\SiswaFactory.php]
● ccessfully.
```

5. Menambahkan data pada file factory

Pertama diubah dulu standarisasi negara menjadi Indonesia Buka file

“app/config/app.php”

Gunakan pencarian untuk mencari kata kunci Faker Ubah baris kode, menjadi

```
'faker_locale' => env('FAKER_LOCALE', 'en_US'),
```

6. Kemudian pada file env tambahkan pada baris terakhir :

```
60
61  FAKER_LOCALE=id_ID
62
```

FAKER LOCALE merupakan sebuah variable global

7. Modifikasi file database\factories\SiswaFactory.php pada bagian :

```
public function definition()  
{  
    return [  
        'nama' => $this->faker->name(),  
        'alamat' => $this->faker->streetAddress(),  
        'kota' => $this->faker->city(),  
        'provinsi' => $this->faker->state(),  
        'email' => $this->faker->email(),  
    ];  
}
```

8. Modifikasi file database\seeders\SiswaSeeder.php :

```
1  <?php  
2  
3  namespace Database\Seeders;  
4  
5  use Illuminate\Database\Console\Seeds\WithoutModelEvents;  
6  use Illuminate\Database\Seeder;  
7  use App\Models\Siswa;  
8  
9  class SiswaSeeder extends Seeder  
10 {  
11     /**  
12      * Run the database seeds.  
13      */  
14     public function run()  
15     {  
16         Siswa::factory(20)->create();  
17     }  
18 }
```

- e. Membuat API

Membuat SiswaController dengan menjalankan script :

```
PS C:\xampp\htdocs\laravel_FrontEnd\Praktikum_Vue_Laravel_CRUD2> php artisan make:controller SiswaController --resource  
  
INFO Controller [C:\xampp\htdocs\laravel_FrontEnd\Praktikum_Vue_Laravel_CRUD2\app\Http\Controllers\SiswaController.  
php] created successfully.  
  
PS C:\xampp\htdocs\laravel_FrontEnd\Praktikum_Vue_Laravel_CRUD2> |
```

Buka dan edit file “../app/Http/Controllers/SiswaController”

Di dalam class SiswaCintroller terdapat 5 function/method, yaitu index, store, update, destroy.

```

1  <?php
2
3  namespace App\Http\Controllers;
4
5  use App\Models\Siswa;
6  use Illuminate\Http\Request;
7  use Illuminate\Support\Facades\Validator;
8
9  class SiswaController extends Controller
10 {
11     /**
12      * Display a listing of the resource.
13      */
14     public function index()
15     {
16         //get data from table siswas
17         $posts = Siswa::orderBy('nama', 'asc')->get();
18         //make response JSON
19         return response()->json([
20             'success' => true,
21             'message' => 'List Data Siswa',
22             'data' => $posts
23         ], 200);
24     }
25

```

```

26     /**
27      * Show the form for creating a new resource.
28      */
29     public function create()
30     {
31         //
32     }
33
34     /**
35      * Store a newly created resource in storage.
36      */
37     public function store(Request $request)
38     {
39         //set validation
40         $validator = Validator::make($request->all(), [
41             'nama' => 'required',
42             'alamat' => 'required',
43             'kota' => 'required',
44             'provinsi' => 'required',
45             'email' => 'required',
46         ]);
47         //response error validation
48         if ($validator->fails()) {
49             return response()->json($validator->errors(), 400);
50         }

```

```

51 //save to database
52 $post = Siswa::create([
53     'nama' => $request->nama,
54     'alamat' => $request->alamat,
55     'kota' => $request->kota,
56     'provinsi' => $request->provinsi,
57     'email' => $request->email
58 ]);
59 //success save to database
60 if ($post) {
61     return response()->json([
62         'success' => true,
63         'message' => 'Input berhasil',
64         'data' => $post
65     ], 201);
66 }
67 //failed save to database
68 return response()->json([
69     'success' => false,
70     'message' => 'Input gagal',
71 ], 409);
72 }
73
74 /**
75  * Display the specified resource.

```

```

76 */
77 public function show($id)
78 {
79     //find post by ID
80     $post = Siswa::findOrFail($id);
81     //make response JSON
82     return response()->json([
83         'success' => true,
84         'message' => 'Detail Data Siswa',
85         'data' => $post
86     ], 200);
87 }
88
89 /**
90  * Show the form for editing the specified resource.
91  */
92 public function edit(string $id)
93 {
94     //
95 }
96
97 /**
98  * Update the specified resource in storage.
99  */

```



```

100 public function update(Request $request, Siswa $post)
101 {
102     //set validation
103     $validator = Validator::make($request->all(), [
104         'nama' => 'required',
105         'alamat' => 'required',
106         'kota' => 'required',
107         'provinsi' => 'required',
108         'email' => 'required',
109     ]);
110     //response error validation
111     if ($validator->fails()) {
112         return response()->json($validator->errors(), 400);
113     }
114     //find siswa by ID
115     $post = Siswa::findOrFail($post->id);
116     if ($post) {
117         //update siswa
118         $post->update([
119             'nama' => $request->nama,
120             'alamat' => $request->alamat,
121             'kota' => $request->kota,
122             'provinsi' => $request->provinsi,
123             'email' => $request->email
124         ]);
125         return response()->json([
126             'success' => true,
127             'message' => 'Data siswa berhasil diupdate',
128             'data' => $post
129         ], 200);
130     }

```

```

131     //data siswa not found
132     return response()->json([
133         'success' => false,
134         'message' => 'Data tidak ditemukan',
135     ], 404);
136 }
137
138 /**
139  * Remove the specified resource from storage.
140  */
141 //destroy
142 public function destroy($id)
143 {
144     //find siswa by ID
145     $post = Siswa::findOrFail($id);
146     if ($post) {
147         //delete siswa
148         $post->delete();
149         return response()->json([
150             'success' => true,
151             'message' => 'Data siswa berhasil dihapus',
152         ], 200);
153     }
154     //data siswa not found
155     return response()->json([
156         'success' => false,
157         'message' => 'Data tidak ditemukan',
158     ], 404);
159 }
160 }

```

f. Membuat Route API

Buka file "routes/api.php dan ubah menjadi:

```

1  <?php
2
3  use Illuminate\Http\Request;
4  use Illuminate\Support\Facades\Route;
5
6  /*
7  |
8  | API Routes
9  |
10 |
11 | Here is where you can register API routes for your application. These
12 | routes are loaded by the RouteServiceProvider and all of them will
13 | be assigned to the "api" middleware group. Make something great!
14 |
15 */
16
17 Route::middleware('auth:sanctum')->get('/user', function (Request
18     $request) {
19     return $request->user();
20 });
21
22 Route::resource('/siswa', SiswaController::class);
23

```

Dengan menggunakan route jenis resource, route akan otomatis dibuatkan (index, store, update, destroy).

Untuk melihat hasil route jenis resource, jalankan perintah Berikut :

```

PS C:\xampp\htdocs\laravel_FrontEnd\Praktikum_Vue_Laravel_CRUD2> php artisan route:list

GET|HEAD      / .....
POST          _ignition/execute-solution ignition.executeSolution > Spatie\Larav...
GET|HEAD      _ignition/health-check ignition.healthCheck > Spatie\LaravelIgniti...
POST          _ignition/update-config ignition.updateConfig > Spatie\LaravelIgni...
GET|HEAD      api/siswa ..... siswa.index > SiswaController@index
POST          api/siswa ..... siswa.store > SiswaController@store
GET|HEAD      api/siswa/create ..... siswa.create > SiswaController@create
GET|HEAD      api/siswa/{siswa} ..... siswa.show > SiswaController@show
PUT|PATCH    api/siswa/{siswa} ..... siswa.update > SiswaController@update
DELETE        api/siswa/{siswa} ..... siswa.destroy > SiswaController@destroy
GET|HEAD      api/siswa/{siswa}/edit ..... siswa.edit > SiswaController@edit
GET|HEAD      api/user .....
GET|HEAD      sanctum/csrf-cookie sanctum.csrf-cookie > Laravel\Sanctum > CsrfCo...

Showing [13] routes

PS C:\xampp\htdocs\laravel_FrontEnd\Praktikum_Vue_Laravel_CRUD2>

```

g. Pengujian API index

Pengujian API dapat menggunakan Postman (<https://www.postman.com/>). Download dan install aplikasi.

**POST** <http://127.0.0.1:8000/api/siswa> **Send**

Params Authorization Headers (9) **Body** Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL

Key	Value	Description
nama	Clark Kent	
alamat	Jl. Cepat dapat	
kota	Metropolis	
provinsi	Yogyakarta	
email	clark.kent@gmail.com	

Body Cookies Headers (10) Test Results Status: 201 Created Time: 718 ms Size: 580 B Save as Example

Pretty Raw Preview Visualize JSON

```

2  "success": true,
3  "message": "Input berhasil",
4  "data": {
5    "nama": "Clark Kent",
6    "alamat": "Jl. Cepat dapat",
7    "kota": "Metropolis",
8    "provinsi": "Yogyakarta",
9    "email": "clark.kent@gmail.com",
10   "updated_at": "2023-11-14T04:06:09.000000Z",
11   "created_at": "2023-11-14T04:06:09.000000Z",

```

**GET** <http://127.0.0.1:8000/api/siswa/1> **Send**

Params Authorization Headers (7) **Body** Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL

This request does not have a body

Body Cookies Headers (10) Test Results Status: 200 OK Time: 14.86 s Size: 616 B Save as Example

Pretty Raw Preview Visualize JSON

```

1  {
2    "success": true,
3    "message": "Detail Data Siswa",
4    "data": {
5      "id": 1,
6      "nama": "Darimin Uwais",
7      "alamat": "Psr. Yos Sudarso No. 49",
8      "kota": "Administrasi Jakarta Utara",
9      "provinsi": "Sulawesi Tenggara",
10     "email": "pratama.endah@gmail.co.id",
11     "created_at": "2023-11-07T02:27:03.000000Z",
12     "updated_at": "2023-11-07T02:27:03.000000Z",
13   }
14 }

```

**DELETE** <http://127.0.0.1:8000/api/siswa/22> **Send**

Params Authorization Headers (7) **Body** Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL

This request does not have a body

Body Cookies Headers (10) Test Results Status: 200 OK Time: 477 ms Size: 363 B Save as Example

Pretty Raw Preview Visualize JSON

```

1  {
2    "success": true,
3    "message": "Data siswa berhasil dihapus"
4  }

```

## 2. VueJS sebagai Frontend

### a. Buat project Vue baru

Pada latihan ini kita akan menginstal Vue. Install Node JS (<https://nodejs.org/>)

Kembali ke folder utama (misal: laravel-vue-2)

Install Vue CLI

```
PS C:\xampp\htdocs\laravel_FrontEnd\Praktikum_Vue_Laravel_CRUD2> npm install -g @vue/cli
npm WARN deprecated source-map-url@0.4.1: See https://github.com/lydell/source-map-url#deprecated
npm WARN deprecated urix@0.1.0: Please see https://github.com/lydell/urix#deprecated
npm WARN deprecated resolve-url@0.2.1: https://github.com/lydell/resolve-url#deprecated
npm WARN deprecated @babel/plugin-proposal-class-properties@7.18.6: This proposal has been merged to the ECMAScript standard and thus this plugin is no longer maintained. Please use @babel/plugin-transform-class-properties instead.
npm WARN deprecated @babel/plugin-proposal-nullish-coalescing-operator@7.18.6: This proposal has been merged to the ECMAScript standard and thus this plugin is no longer maintained. Please use @babel/plugin-transform-nullish-coalescing-operator instead.
npm WARN deprecated source-map-resolve@0.5.3: See https://github.com/lydell/source-map-resolve#deprecated
```

Pastikan Anda berada di folder utama (laravel-vue-2 )

Jalankan script berikut untuk menginstal Vue pada folder bernama frontend

```
PS C:\xampp\htdocs\laravel_FrontEnd\Praktikum_Vue_Laravel_CRUD2> vue create frontend
? Your connection to the default npm registry seems to be slow.
  Use https://registry.npmmirror.com for faster installation? Yes

Vue CLI v5.0.8
? Please pick a preset: Default ([Vue 3] babel, eslint)
```

Ketika muncul pilihan seperti dibawah ini, untuk sementara pilih pilihan pertama

```
Vue CLI v5.0.8
? Please pick a preset: (Use arrow keys)
> Default ([Vue 3] babel, eslint)
  Default ([Vue 2] babel, eslint)
  Manually select features
```

Buka file package.json, pada bagian eslintConfig ubah pada bagian rules menjadi :

```
"eslintConfig": {
  "root": true,
  "env": {
    "node": true
  },
  "extends": [
    "plugin:vue/vue3-essential",
    "eslint:recommended"
  ],
  "parserOptions": {
    "parser": "@babel/eslint-parser"
  },
  "rules": {
    "vue/multi-word-component-names": "off"
  }
},
```

Ini akan mengeliminasi potensi masalah penamaan file Vue.

Setelah instalasi berhasil, coba jalankan project Vue.

Masuk ke folder frontend dan jalankan perintah :

```
PS C:\xampp\htdocs\laravel_FrontEnd\Praktikum_Vue_Laravel_CRUD2\frontend> npm run serve

> frontend@0.1.0 serve
> vue-cli-service serve

[INFO] Starting development server...

[DONE] Compiled successfully in 13965ms

App running at:
- Local: http://localhost:8080/
- Network: http://10.109.3.173:8080/

Note that the development build is not optimized.
```

Buka browser dan masuk ke <http://localhost:8080/>

- b. Instalasi Vue Router, Axios, Bootstrap dan library JS lainnya

Jalankan script Berikut :

```
PS C:\xampp\htdocs\laravel_FrontEnd\Praktikum_Vue_Laravel_CRUD2\frontend> npm install vue-router@next

up to date, audited 985 packages in 8s

4 packages are looking for funding
  run `npm fund` for details

4 moderate severity vulnerabilities

To address all issues (including breaking changes), run:
  npm audit fix --force

Run `npm audit` for details.

PS C:\xampp\htdocs\laravel_FrontEnd\Praktikum_Vue_Laravel_CRUD2\frontend> npm install axios

up to date, audited 985 packages in 6s

4 packages are looking for funding
  run `npm fund` for details

4 moderate severity vulnerabilities

To address all issues (including breaking changes), run:
  npm audit fix --force

Run `npm audit` for details.

PS C:\xampp\htdocs\laravel_FrontEnd\Praktikum_Vue_Laravel_CRUD2\frontend> npm install bootstrap popper.js jquery

up to date, audited 985 packages in 6s

4 packages are looking for funding
  run `npm fund` for details

4 moderate severity vulnerabilities

To address all issues (including breaking changes), run:
  npm audit fix --force

Run `npm audit` for details.
```

Library Vue Router digunakan untuk membuat navigasi dalam aplikasi Vue.

Library Axios digunakan untuk melakukan HTTP request ke server untuk melakukan manipulasi data.

Nantinya akan digunakan untuk mengelola endpoint API dari Laravel.

Library Bootstrap, Popper, dan jquery digunakan untuk memudahkan membangun tampilan.

Lakukan konfigurasi Bootstrap di VueJs dengan memodifikasi file "src/main.js"

```
1 import { createApp } from 'vue';
2 import App from './App.vue';
3 import "bootstrap/dist/css/bootstrap.css";
4 import "jquery/dist/jquery.min";
5 import "popper.js/dist/popper.min";
6 import "bootstrap/dist/js/bootstrap.min";
7
8 createApp(App).mount('#app');
9
10
```

c. Membuat Views

Pada tahap ini akan dibuat view untuk menampilkan template di Vue.

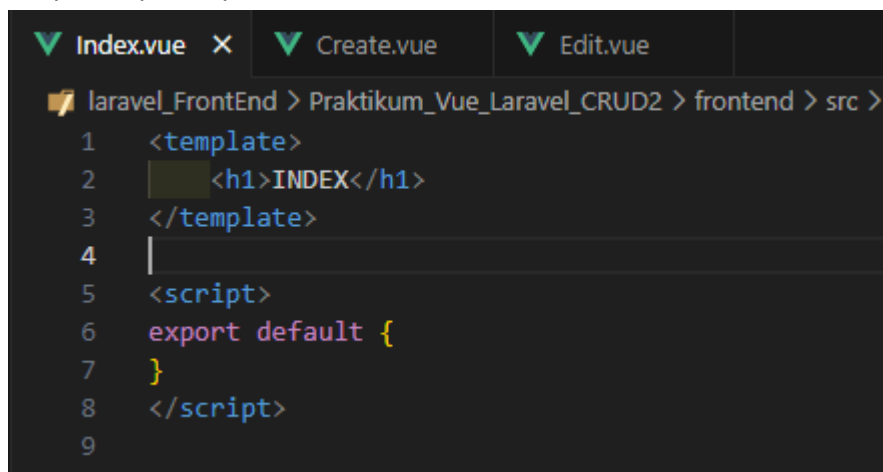
Buat sebuah folder baru bernama views didalam folder src.

Di dalam folder views buat sebuah folder dengan nama siswa.

Di dalam folder siswa buat 3 file baru: Index.vue, Create.vue dan Edit.vue

Untuk sementara ketiga file baru tersebut ditampilkan template sederhana :

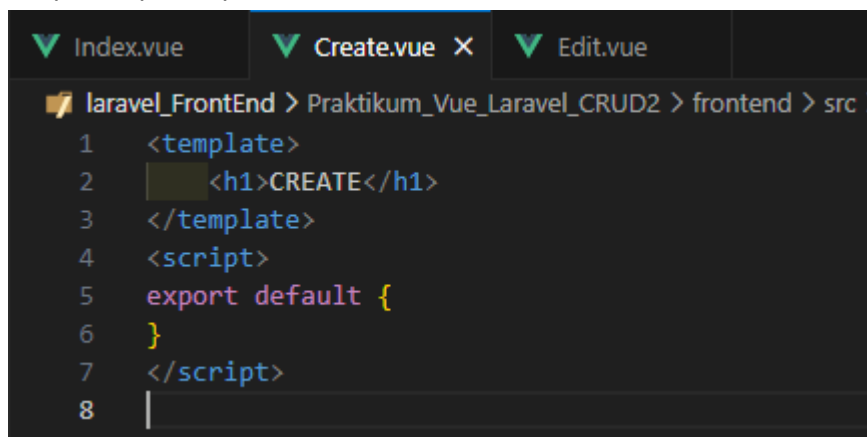
- src\views\siswa\Index.vue



The screenshot shows the VS Code editor with three tabs: Index.vue, Create.vue, and Edit.vue. The Index.vue tab is active, displaying the following code:

```
1 <template>
2   <h1>INDEX</h1>
3 </template>
4
5 <script>
6   export default {
7   }
8 </script>
9
```

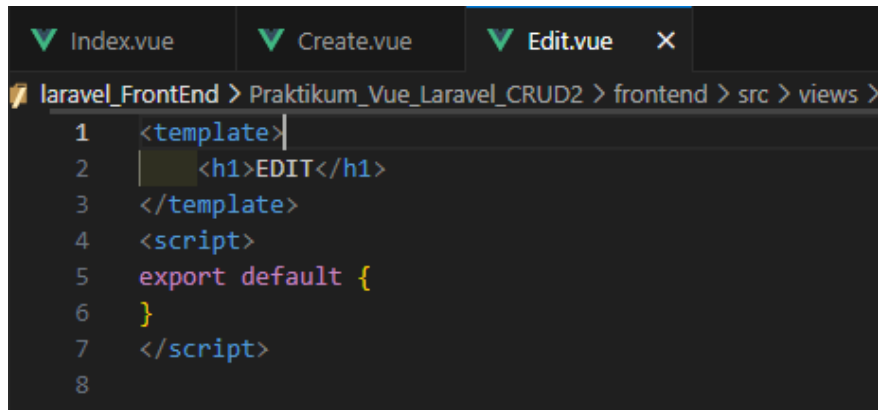
- src\views\siswa\Create.vue



The screenshot shows the VS Code editor with three tabs: Index.vue, Create.vue, and Edit.vue. The Create.vue tab is active, displaying the following code:

```
1 <template>
2   <h1>CREATE</h1>
3 </template>
4 <script>
5   export default {
6   }
7 </script>
8
```

- src\views\siswa\Edit.vue



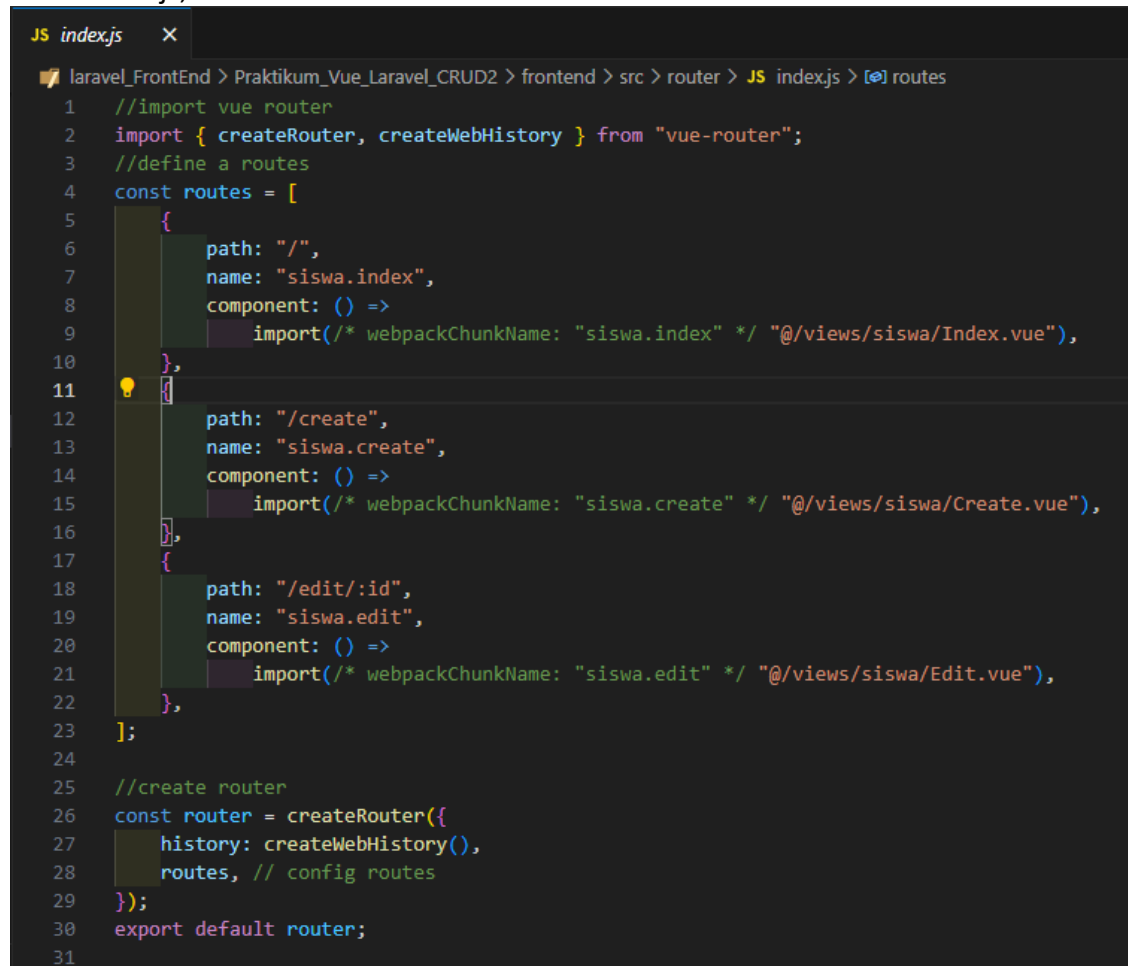
```
1 <template>
2   <h1>EDIT</h1>
3 </template>
4 <script>
5   export default {
6   }
7 </script>
8
```

d. Membuat Route

Buat sebuah folder baru dengan nama router dalam folder src.

Dalam folder router, buat sebuah file dengan nama index.js.

Pada index.js, masukkan kode Berikut :



```
JS index.js X
laravel_FrontEnd > Praktikum_Vue_Laravel_CRUD2 > frontend > src > router > JS index.js > routes
1 //import vue router
2 import { createRouter, createWebHistory } from "vue-router";
3 //define a routes
4 const routes = [
5   {
6     path: "/",
7     name: "siswa.index",
8     component: () =>
9       import(/* webpackChunkName: "siswa.index" */ "@views/siswa/Index.vue"),
10   },
11   {
12     path: "/create",
13     name: "siswa.create",
14     component: () =>
15       import(/* webpackChunkName: "siswa.create" */ "@views/siswa/Create.vue"),
16   },
17   {
18     path: "/edit/:id",
19     name: "siswa.edit",
20     component: () =>
21       import(/* webpackChunkName: "siswa.edit" */ "@views/siswa/Edit.vue"),
22   },
23 ];
24
25 //create router
26 const router = createRouter({
27   history: createWebHistory(),
28   routes, // config routes
29 });
30 export default router;
31
```

Router kemudian perlu diregister agar dapat diakses secara global dalam project Vue. Modifikasi "src/main.js" :

```
JS main.js X
laravel_FrontEnd > Praktikum_Vue_Laravel_CRUD2 > frontend > src > JS main.js > ...
1 import { createApp } from 'vue';
2 import App from './App.vue';
3
4 //import router
5 import router from './router'
6
7 //import Bootstrap, Popper, jQuery
8 import "bootstrap/dist/css/bootstrap.css";
9 import "jquery/dist/jquery.min";
10 import "popper.js/dist/popper.min";
11 import "bootstrap/dist/js/bootstrap.min";
12
13 // createApp(App).mount('#app');
14
15 const app = createApp(App)
16
17 //use vue router
18 app.use(router)
19 app.mount('#app')
20
```

Untuk menampilkan Vue Router dalam main component, modifikasi file "src/App.vue" menjadi :

```
App.vue X
laravel_FrontEnd > Praktikum_Vue_Laravel_CRUD2 > frontend > src > App.vue > {} template
1 <template>
2   <router-view></router-view>
3 </template>
4 <script>
5   export default {
6   }
7 </script>
8 <style>
9   #app {
10     font-family: Avenir, Helvetica, Arial, sans-serif;
11     -webkit-font-smoothing: antialiased;
12     -moz-osx-font-smoothing: grayscale;
13     text-align: center;
14     color: #2c3e50;
15     margin-top: 60px;
16   }
17 </style>
18
```



Jalankan script :

```
C:\xampp\htdocs\laravel_FrontEnd\Praktikum_Vue_Laravel_CRUD2\frontend>npm run serve

> frontend@0.1.0 serve
> vue-cli-service serve

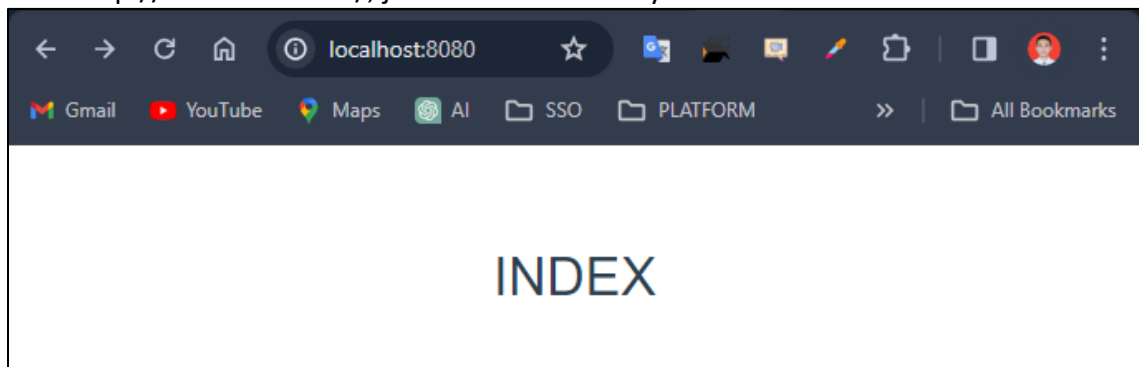
INFO Starting development server...

DONE Compiled successfully in 9265ms 12:47:08

App running at:
- Local: http://localhost:8080/
- Network: http://10.109.3.173:8080/

Note that the development build is not optimized.
To create a production build, run npm run build.
```

Buka <http://localhost:8080/>, jika lancar maka hanya akan terlihat tulisan INDEX.



e. Menampilkan data

Modifikasi file "src\views\siswa\Index.vue" menjadi :

```
Index.vue X
laravel_FrontEnd > Praktikum_Vue_Laravel_CRUD2 > frontend > src > views > siswa > Index.vue > {} style
1  <template>
2    <div class="container mt-5">
3      <div class="row">
4        <div class="col-md-12">
5          <div class="card border-0 rounded shadow">
6            <div class="card-body">
7              <h4>DATA SISWA</h4>
8
9            <hr>
10
11           <router-link :to="{ name: 'siswa.create' }" class="btn btn-md btn-
12             success">TAMBAH SISWA</router-link>
13
14           <table class="table table-striped table-bordered mt-4">
15             <thead class="thead-dark">
16               <tr>
17                 <th scope="col">Nama</th>
18                 <th scope="col">Alamat</th>
19                 <th scope="col">Kota</th>
20                 <th scope="col">Provinsi</th>
21                 <th scope="col">E-mail</th>
22                 <th scope="col">Aksi</th>
23
24               </tr>
25             </thead>
26             <tbody>
27
28               <tr v-for="(siswa, index) in posts" :key="index">
29                 <td>{{ siswa.nama }}</td>
30                 <td>{{ siswa.alamat }}</td>
31                 <td>{{ siswa.kota }}</td>
32                 <td>{{ siswa.provinsi }}</td>
33                 <td>{{ siswa.email }}</td>
34                 <td class="text-center">
35
36                   <router-link :to="{
37                     name: 'siswa.edit', params: {
38                       id:
39                         siswa.id
40                     }
41                   }" class="btn btn-sm btn-primary mr-1">Edit</router-link>
42
43                   <button class="btn btn-sm btn-danger ml-1">Delete</button>
44                 </td>
45               </tr>
46             </tbody>
47           </table>
48         </div>
49       </div>
50     </div>
51   </div>
52 </div>
53 </template>
54 <script>
55   import axios from 'axios'
56   import { onMounted, ref } from 'vue'
57   export default {
58     Complexity is 5 Everything is cool!
59     setup() {
60       //reactive state
61       let posts = ref([])
```

```

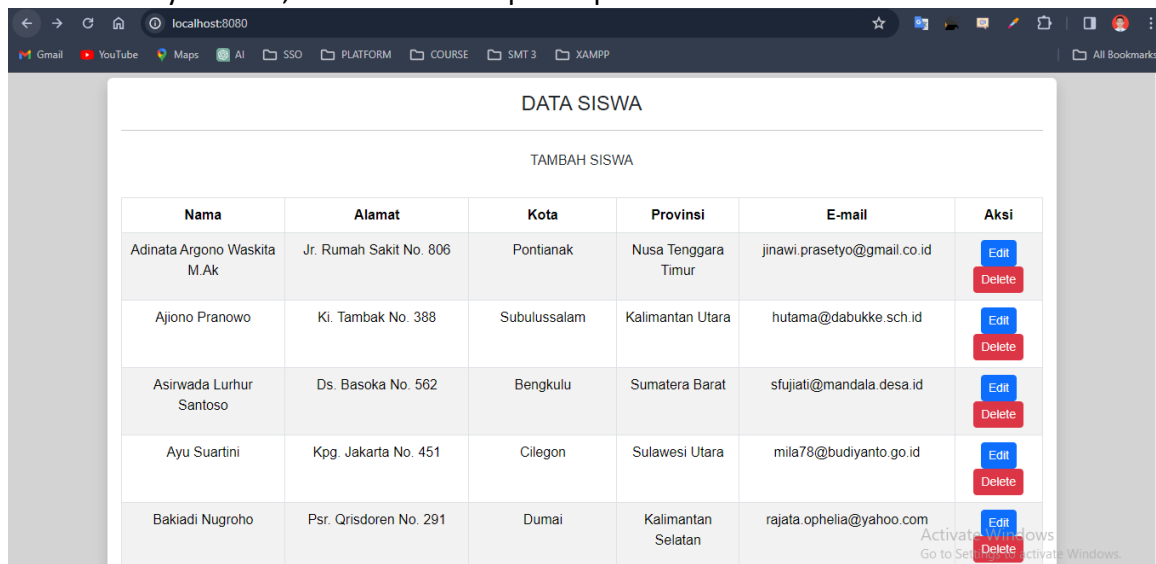
61 //mounted
    Complexity is 3 Everything is cool!
62 onMounted(() => {
63     //get API from Laravel Backend
64     axios.get('http://localhost:8000/api/siswa')
65     .then(response => {
66         //assign state posts with response data
67         posts.value = response.data.data
68     }).catch(error => {
69         console.log(error.response.data)
70     })
71 })
72 //return
73 return {
74     posts
75 }
76 }
77 }
78 </script>
79 <style>
80 body {
81     background: lightgray;
82 }
83 </style>
84
85

```

Jalankan “npm run serve” pada frontend dan “php artisan serve” pada **Praktikum\_Vue\_Laravel\_CRUD2**.

Jalankan pada browser <http://localhost:8080/>

Jika semuanya lancar, data akan ditampilkan pada browser :



DATA SISWA					
TAMBAH SISWA					
Nama	Alamat	Kota	Provinsi	E-mail	Aksi
Adinata Argono Waskita M.Ak	Jr. Rumah Sakit No. 806	Pontianak	Nusa Tenggara Timur	jinawi.prasetyo@gmail.co.id	<a href="#">Edit</a> <a href="#">Delete</a>
Ajiono Pranowo	Ki. Tambak No. 388	Subulussalam	Kalimantan Utara	hutama@dabukke.sch.id	<a href="#">Edit</a> <a href="#">Delete</a>
Asirwada Lurhur Santoso	Ds. Basoka No. 562	Bengkulu	Sumatera Barat	sfujiati@mandala.desa.id	<a href="#">Edit</a> <a href="#">Delete</a>
Ayu Suartini	Kpg. Jakarta No. 451	Cilegon	Sulawesi Utara	mila78@budiyanto.go.id	<a href="#">Edit</a> <a href="#">Delete</a>
Bakiadi Nugroho	Psr. Qrisdoren No. 291	Dumai	Kalimantan Selatan	rajata.ophelia@yahoo.com	<a href="#">Edit</a> <a href="#">Delete</a>

f. Input data

Modifikasi file “src\views\siswa\Create.vue” :

```

1 <template>
2 <div class="container mt-5">
3 <div class="row">
4 <div class="col md-12">
5 <div class="card border-0 rounded shadow">
6 <div class="card-body">
7 <h4>TAMBAH SISWA</h4>
8
9 <hr>
10
11 <form @submit.prevent="store">
12 <div class="form-group">
13 <label for="title" class="font-weight bold">Nama Siswa</label>
14
15 <input type="text" class="form-control" v-model="post.nama" placeholder="Masukkan nama siswa">
16
17 <!-- validation -->
18
19 <div v-if="validation.nama" class="mt-2 alert alert-danger">
20
21 {{ validation.nama[0] }}
22 </div>
23 </div>
24
25 <div class="form-group">
26
27 <label for="title" class="font-weight bold">Alamat</label>
28 <input type="text" class="form-control" v-model="post.alamat" placeholder="Nama jalan dan nomor rumah">
29
30 <!-- validation -->
31
32 <div v-if="validation.alamat" class="mt-2 alert alert-danger">
33
34 {{ validation.alamat[0] }}
35 </div>
36 </div>
37
38 <div class="form-group">
39
40 <label for="title" class="font-weight bold">Kota</label>
41
42 <input type="text" class="form-control" v-model="post.kota" placeholder="Masukkan nama kota tempat tinggal">
43
44 <!-- validation -->
45
46 <div v-if="validation.kota" class="mt-2 alert alert-danger">
47
48 {{ validation.kota[0] }}
49 </div>
50 </div>
51
52 <div class="form-group">
53
54 <label for="title" class="font-weight-
55 bold">Provinsi</label>
56
57 <input type="text" class="form-control" v-model="post.provinsi" placeholder="Masukkan nama provinsi tempat tinggal">
58
59 <!-- validation -->
60
61 <div v-if="validation.provinsi" class="mt-2 alert alert-danger">
62
63 {{ validation.provinsi[0] }}
64 </div>
65 </div>
66
67 <div class="form-group">
68
69 <label for="title" class="font-weight bold">E-mail</label>
70
71 <input type="text" class="form-control" v-model="post.email" placeholder="Masukkan alamat email siswa">
72
73 <!-- validation -->
74
75 <div v-if="validation.email" class="mt-2 alert alert-danger">
76
77 {{ validation.email[0] }}
78 </div>
79 </div>
80
81 <button type="submit" class="btn btn-primary">SIMPAN</button>
82
83 </form>
84 </div>
85 </div>
86 </div>
87 </div>
88 </div>
89 </template>
90 <script>
91 import { reactive, ref } from 'vue'
92 import { useRouter } from 'vue-router'
93 import axios from 'axios'
94 export default {
95   setup() {
96     //initial state siswa
97     const post = reactive({
98       nama: '',
99       alamat: '',
100       kota: '',
101       provinsi: '',
102       email: ''
103     })
104     //state validation
105     const validation = ref({})
106     //vue router
107     const router = useRouter()
108     //method store
109     function store() {
110       let nama = post.nama
111       let alamat = post.alamat
112       let kota = post.kota
113       let provinsi = post.provinsi
114       let email = post.email
115       axios.post('http://localhost:8000/api/siswa', {
116         nama: nama,
117         alamat: alamat,
118         kota: kota,
119         provinsi: provinsi,
120         email: email
121       }).then(() => {
122         //redirect ke siswa index
123         router.push({
124           name: 'siswa.index'
125         })
126       }).catch(error => {
127         //assign state validation with error
128         validation.value = error.response.data
129       })
130     }
131     //return
132     return {
133       post,
134       validation,
135       router,
136       store
137     }
138   }
139 }
140 </script>
141 <style>
142 body {
143   background: lightgray;
144 }
145 </style>
146

```

## TAMBAH SISWA

Nama Siswa

Masukkan nama siswa

Alamat

Nama jalan dan nomor rumah

Kota

Masukkan nama kota tempat tinggal

Provinsi

Masukkan nama provinsi tempat tinggal

E-mail

Masukkan alamat email siswa

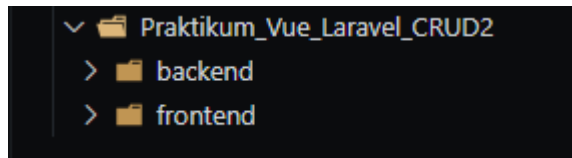
SIMPAN

A. Dari aplikasi yang telah dikembangkan sebelumnya, tambahkan fitur berikut pada aplikasi :

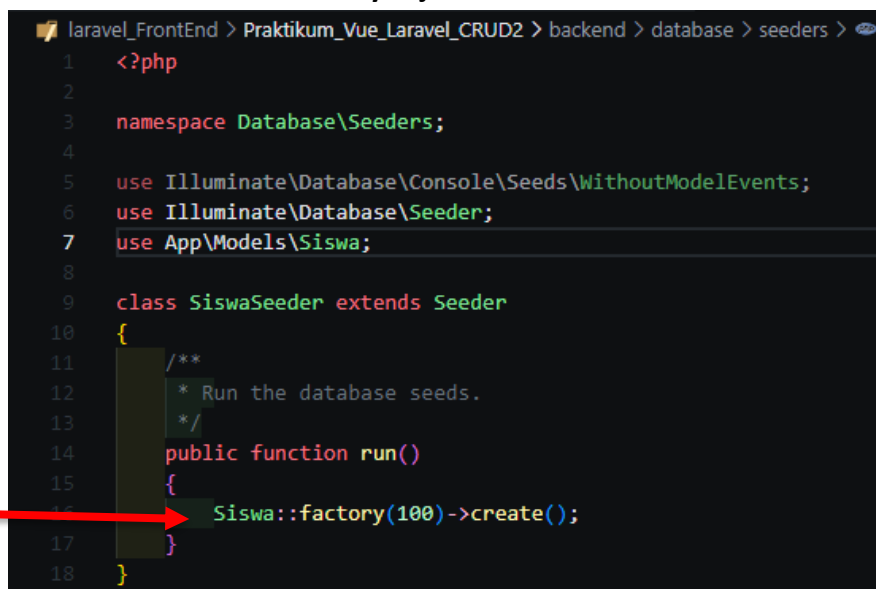
1. **Pagination.** Menampilkan list data menggunakan navigasi nomor halaman. Pastikan terdapat record data minimal 100. Tiap halaman menampilkan 15 data.

a) Pisahkan folder

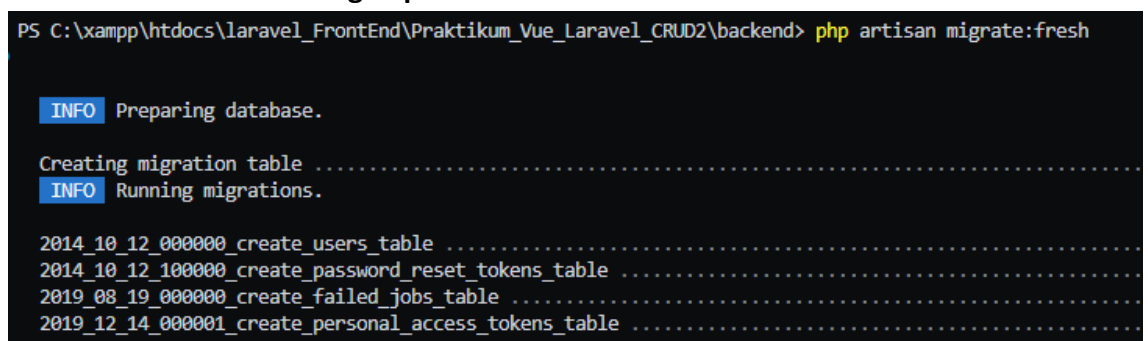
Maka saya membuat folder baru dengan nama backend, yang berisi file Laravel, dan Dan membuat folder frontend, yang berisi file vue.js



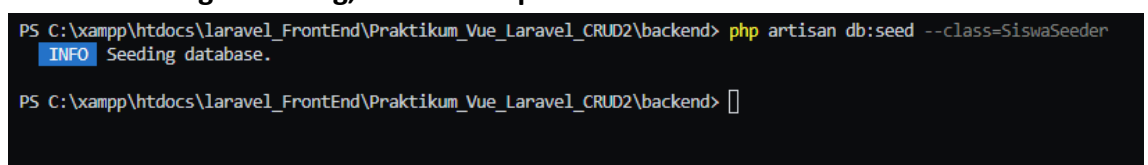
- b) Pada bagian factory(20), ubah menjadi factory(100) dimana untuk menambahkan data dummy sejumlah 100 data :



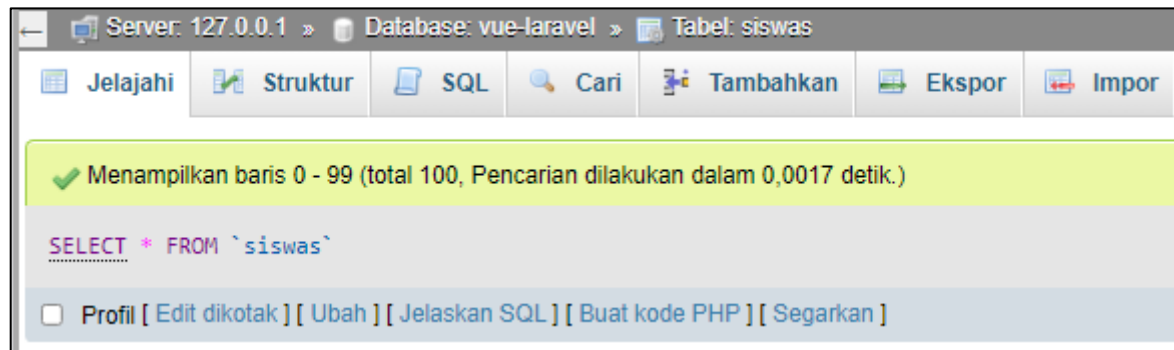
- c) Merefresh data table dengan perintah :



- d) Melakukan migrate ulang, untuk memparbarui data



- e) Setelah berhasil, maka kita akan mendapatkan data dummy siswa sebanyak 100 data.



- f) Untuk dapat menampilkan 15 data menggunakan navigasi nomor halaman, disini saya memodifikasi pada bagian :

backend/app/Http/Controllers/SiswaController.php

```
public function index()
{
    //get data from table siswas
    $posts = Siswa::paginate(15);
    //make response JSON
    return response()->json([
        'success' => true,
        'message' => 'List Data Siswa',
        'data' => $posts
    ], 200);
}
```

Bagian `$posts = Siswa::paginate(15);` untuk mengatur paginasi dengan 15 data per halaman

- g) Lalu pada bagian backend/routes/web.php perlu menambahkan rute :

```
21
22 Route::get('/siswa', 'SiswaController@index');
23
```

- h) Pada frontend/src/views/siswa/Index.vue menambahkan script Berikut untuk tampilan data siswa dan navigasi halaman, saya meletakkan setelah `</table>`

```
1 <nav aria-label="Page navigation">
2   <ul class="pagination">
3     <li
4       class="page-item"
5       :class="{ disabled: posts.current_page === 1 }"
6     >
7       <button
8         @click="fetchData(posts.current_page - 1)"
9         class="page-link"
10        aria-label="Previous"
11      >
12        <span aria-hidden="true">&laquo;</span>
13      </button>
14    </li>
15    <li
16      class="page-item"
17      v-for="page in posts.last_page"
18      :key="page"
19      :class="{ active: page === posts.current_page }"
20    >
21      <button @click="fetchData(page)" class="page-link">
22        {{ page }}
23      </button>
24    </li>
25    <li
26      class="page-item"
27      :class="{ disabled: posts.current_page === posts.last_page }"
28    >
29      <button
30        @click="fetchData(posts.current_page + 1)"
31        class="page-link"
32        aria-label="Next"
33      >
34        <span aria-hidden="true">&raquo;</span>
35      </button>
36    </li>
37  </ul>
38 </nav>
```

- Menggunakan class-binding (`:class="{ disabled: posts.current_page === 1 }"`) untuk menentukan apakah tombol Previous atau Next harus dinonaktifkan jika sudah berada di halaman pertama atau terakhir.
- Menggunakan `@click` untuk menangani pemanggilan fungsi `fetchData(page)` ketika tombol halaman ditekan.



- Script (@click="fetchData(page)": Ketika tombol halaman ditekan, fungsi fetchData(page) akan dipanggil untuk memuat data dari halaman yang dipilih. Fungsi ini mengambil data siswa dari backend berdasarkan nomor halaman yang dipilih.

i) Potongan script di bawah untuk menyimpan data yang diterima dari backend dan informasi halaman saat melakukan navigasi.

```
1 export default {
2   setup() {
3     let posts = ref({
4       data: [],
5       current_page: 1,
6       last_page: 1,
7     });
```

```
1 const fetchData = (page) => {
2   axios
3     .get(`http://localhost:8000/api/siswa?page=${page}`)
4     .then((response) => {
5       posts.value = response.data.data;
6     })
7     .catch((error) => {
8       console.log(error.response.data);
9     });
10  };
11
12  onMounted(() => {
13    fetchData(1);
14  });
15
16  const deleteSiswa = (siswaId) => {
17    if (confirm("Apakah Anda yakin ingin menghapus siswa ini?")) {
18      axios
19        .delete(`http://localhost:8000/api/siswa/${siswaId}`)
20        .then(() => {
21          fetchData(posts.value.current_page);
22        })
23        .catch((error) => {
24          console.log(error.response.data);
25        });
26    }
27  };
28  };
29
30  return {
31    posts,
32    deleteSiswa,
33    fetchData,
34  };
```

- j) Lalu saya juga menambahkan table nomor untuk memudahkan tampilan data siswa

```
<table class="table table-striped table-bordered mt-4">
  <thead class="thead-dark">
    <tr>
      <th scope="col">No</th>
      <th scope="col">Nama</th>
      <th scope="col">Alamat</th>
      <th scope="col">Kota</th>
      <th scope="col">Provinsi</th>
      <th scope="col">E-mail</th>
      <th scope="col">Aksi</th>
    </tr>
  </thead>
  <tbody>
    <tr v-for="(siswa, index) in posts.data" :key="index">
      <td>{{ siswa.id }}</td>
      <td>{{ siswa.nama }}</td>
      <td>{{ siswa.alamat }}</td>
      <td>{{ siswa.kota }}</td>
      <td>{{ siswa.provinsi }}</td>
      <td>{{ siswa.email }}</td>
      <td class="text-center">
```

- k) Berikut adalah hasil tampilannya :

DATA SISWA						
<div>TAMBAH SISWA</div>						
No	Nama	Alamat	Kota	Provinsi	E-mail	Aksi
91	Taya Doyle	1516 Ethelyn Ferry Apt. 840	Lake Marleyberg	Texas	shaylee.krajcik@hotmail.com	<a href="#">Edit</a> <a href="#">Delete</a>
92	Nola Terry	769 Kshlerin Glen	North Kelly	Nebraska	howard11@mayer.com	<a href="#">Edit</a> <a href="#">Delete</a>
93	Kaylee Bechtelar	869 Ellsworth Island Suite 724	East Selenaview	Alaska	medhurst.bernadette@gmail.com	<a href="#">Edit</a> <a href="#">Delete</a>
94	Mireya Tremblay	91018 Vena Alley Apt. 838	Hudson-ton	North Dakota	rosenbaum.ben@wisozk.com	<a href="#">Edit</a> <a href="#">Delete</a>
95	Lexi Kessler	773 Tyrique Loaf	Lake Modestobury	North Carolina	zackery.hayes@yahoo.com	<a href="#">Edit</a> <a href="#">Delete</a>
96	Mark Mills	76164 Gibson Ways	West Rhea	Montana	qlubowitz@gmail.com	<a href="#">Edit</a> <a href="#">Delete</a>
97	Mrs. Ofelia Jakubowski	2799 Volkman Corner	Abernathys-hire	Texas	yripping@oconner.net	<a href="#">Edit</a> <a href="#">Delete</a>
98	Isaac Gislason	525 Grady Ports	Britneyton	Alabama	loyal19@yahoo.com	<a href="#">Edit</a> <a href="#">Delete</a>
99	Damon Russel MD	64340 Rolfs-on Knoll	North Paula	Colorado	skiles.reginald@murphy.biz	<a href="#">Edit</a> <a href="#">Delete</a>
100	Gina Reynolds II	22796 Audra Fall	New Archibald	New Mexico	gkoss@keebler.com	<a href="#">Edit</a> <a href="#">Delete</a>

B. Autentikasi. Ketika aplikasi dijalankan akan muncul halaman login. User dapat login atau membuat akun baru menggunakan username dan password. Terdapat 2 tipe user, admin dan general user. Admin dapat modifikasi data (CRUD), general user hanya dapat melihat daftar.

1. Untuk tahap kedua ini, langkah pertama yang saya buat adalah menyempurnakan CRUD siswa dari praktikum sebelumnya. Dalam praktikum sebelumnya, telah berhasil menambahkan data tetapi tidak untuk edit dan hapus data.

## 2. Frontend/src/views/siswa/Edit.vue

```
1 <template>
2   <div class="container mt-5">
3     <div class="row">
4       <div class="col-md-12">
5         <div class="card border-0 rounded shadow">
6           <div class="card-body">
7             <h4>EDIT SISWA</h4>
8             <hr />
9
10            <form @submit.prevent="update">
11              <div class="form-group">
12                <label for="nama" class="font-weight-bold">Nama Siswa</label>
13                <input type="text" class="form-control" v-model="post.nama" placeholder="Masukkan nama siswa" />
14              </div>
15
16              <div class="form-group">
17                <label for="alamat" class="font-weight-bold">Alamat</label>
18                <input type="text" class="form-control" v-model="post.alamat" placeholder="Nama jalan dan nomor rumah" />
19              </div>
20
21              <div class="form-group">
22                <label for="kota" class="font-weight-bold">Kota</label>
23                <input type="text" class="form-control" v-model="post.kota"
24                  placeholder="Masukkan nama kota tempat tinggal" />
25              </div>
26
27              <div class="form-group">
28                <label for="provinsi" class="font-weight-bold">Provinsi</label>
29                <input type="text" class="form-control" v-model="post.provinsi"
30                  placeholder="Masukkan nama provinsi tempat tinggal" />
31              </div>
32
33              <div class="form-group">
34                <label for="email" class="font-weight-bold">E-mail</label>
35                <input type="text" class="form-control" v-model="post.email" placeholder="Masukkan alamat email siswa" />
36              </div>
37
38              <button type="button" class="btn btn-primary" @click="update">UPDATE</button>
39            </form>
40          </div>
41        </div>
42      </div>
43    </div>
44  </div>
45 </template>
46
```

```

1
2 <script>
3 import { reactive, onMounted } from "vue";
4 import { useRouter, useRoute } from "vue-router";
5 import axios from "axios";
6
7 export default {
8   setup() {
9     const route = useRoute();
10    const postId = route.params.id;
11    const post = reactive({
12      nama: "",
13      alamat: "",
14      kota: "",
15      provinsi: "",
16      email: "",
17    });
18
19    const router = useRouter();
20
21    const fetchData = async () => {
22      try {
23        const { data } = await axios.get(`http://localhost:8000/api/siswa/${postId}`);
24        // Populate the 'post' reactive object with retrieved data
25        Object.assign(post, data.data);
26      } catch (error) {
27        console.error("Error fetching data:", error);
28      }
29    };
30
31    const update = async () => {
32      try {
33        await axios.put(`http://localhost:8000/api/siswa/${postId}`, {
34          nama: post.nama,
35          alamat: post.alamat,
36          kota: post.kota,
37          provinsi: post.provinsi,
38          email: post.email,
39        });
40
41        // Fetch updated data
42        fetchData();
43
44        // Redirect to another route
45        router.push({ name: "siswa.siswa" });
46      } catch (error) {
47        console.error("Error updating data:", error);
48      }
49    };
50
51    // Fetch data on component mount
52    onMounted(() => {
53      fetchData();
54    });
55
56    return {
57      post,
58      router,
59      update,
60    };
61  },
62 };
63 </script>
64

```

Yang perlu diperhatikan adalah pada bagian setup yaitu untuk menempatkan logika utama komponen.

```
1  const route = useRoute();
2  const postId = route.params.id;
3  const post = reactive({
4    nama: "",
5    alamat: "",
6    kota: "",
7    provinsi: "",
8    email: "",
9  });
```

Script di atas untuk mendapatkan nilai 'id' dari parameter route. 'reactive' digunakan untuk membuat objek 'post' yang akan menyimpan data siswa.

```
1  const router = useRouter();
2
3  const fetchData = async () => {
4    try {
5      const { data } = await axios.get(`http://localhost:8000/api/siswa/${postId}`);
6      // Populate the 'post' reactive object with retrieved data
7      Object.assign(post, data.data);
8    } catch (error) {
9      console.error("Error fetching data:", error);
10   }
11  };
```

Script di atas untuk mengirim permintaan 'PUT' ke API untuk memperbarui data siswa. Lalu setelah berhasil, akan diarahkan kembali ke route siswa.siswa atau tampilan awal siswa dengan data yang telah diperbarui.

3. Pastikan sudah terdapat rute yang sesuai di router/index.js

```
33  {
34    path: "/edit/:id",
35    name: "siswa.edit",
36    component: () =>
37      import(/* webpackChunkName: "siswa.edit" */ "@/views/siswa/Edit.vue"),
38  },
39  ];
```

4. Coba jalankan `npm run serve` untuk melihat tampilan edit :

EDIT SISWA

Nama Siswa  
Humaira Ilsa Uyainah M.Ak

Alamat  
Ki. Bagonwoto No. 667

Kota  
Serangaa

Provinsi  
Gorontalo

E-mail  
galiono93@yahoo.co.id

UPDATE

5. Selanjutnya membuat login-register. Buka folder backend dan buat file login dan register menggunakan perintah di cmd seperti gambar di bawah :

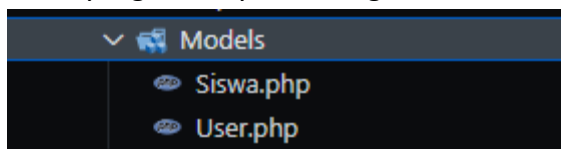
```
PS C:\xampp\htdocs\laravel_FrontEnd\Praktikum_Vue_Laravel_CRUD2\backend> php artisan make:controller RegisterController
INFO Controller [C:\xampp\htdocs\laravel_FrontEnd\Praktikum_Vue_Laravel_CRUD2\backend\app\Http\Controllers\RegisterController.php] created successfully.

PS C:\xampp\htdocs\laravel_FrontEnd\Praktikum_Vue_Laravel_CRUD2\backend> php artisan make:controller LoginController
INFO Controller [C:\xampp\htdocs\laravel_FrontEnd\Praktikum_Vue_Laravel_CRUD2\backend\app\Http\Controllers\LoginController.php] created successfully.

PS C:\xampp\htdocs\laravel_FrontEnd\Praktikum_Vue_Laravel_CRUD2\backend>
```

Terlihat di atas bahwa RegisterController dan LoginController telah berhasil dibuat dalam folder controller.

6. Dari praktikum sebelumnya kita telah menjalankan perintah “`php artisan migrate`” dengan ini table di dalam database sudah terbuat dan mencakup table users yang nantinya akan digunakan untuk login register.



Tabel	Tindakan
<input type="checkbox"/> failed_jobs	★ Jelajahi Struktur Cari Tambahkan Kosongkan Hapus
<input type="checkbox"/> migrations	★ Jelajahi Struktur Cari Tambahkan Kosongkan Hapus
<input type="checkbox"/> password_reset_tokens	★ Jelajahi Struktur Cari Tambahkan Kosongkan Hapus
<input type="checkbox"/> personal_access_tokens	★ Jelajahi Struktur Cari Tambahkan Kosongkan Hapus
<input type="checkbox"/> siswas	★ Jelajahi Struktur Cari Tambahkan Kosongkan Hapus
<input type="checkbox"/> users	★ Jelajahi Struktur Cari Tambahkan Kosongkan Hapus
6 tabel	Jumlah

7. Menambahkan script ke dalam file RegisterController.php

```

1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6  use App\Models\User;
7  use Illuminate\Support\Facades\Auth;
8  use Illuminate\Support\Facades\Hash;
9
10 class RegisterController extends Controller
11 {
12     // Method untuk menangani proses registrasi
13     public function store(Request $request)
14     {
15         // Mengambil semua data dari request
16         $input = $request->all();
17
18         // Membuat sebuah user baru dalam database
19         User::create([
20             'name' => $input['name'],
21             'email' => $input['email'],
22             'password' => Hash::make($input['password'])
23         ]);
24
25         // Memberikan respons JSON jika registrasi berhasil
26         return response()->json([
27             'status' => true,
28             'message' => "Registration Success"
29         ]);
30     }
31 }
32

```

- Controller berada dalam namespace App\Http\Controllers.
- Menggunakan beberapa class dari Laravel seperti Request, User, Auth, dan Hash.
- Membuat user baru di database menggunakan model User.
- Password user di-hash menggunakan fungsi Hash::make() sebelum disimpan ke database.
- Mengembalikan respon JSON yang menyatakan bahwa registrasi berhasil.

8. Menambahkan script berikut ke dalam file LoginController.php

```

1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6  use App\Models\User;
7  use Illuminate\Support\Facades\Auth;
8  use Hash;
9
10 class LoginController extends Controller
11 {
12     // Method untuk menangani proses autentikasi login
13     public function check(Request $request)
14     {
15         // Validasi data yang diterima dari request
16         $credentials = $request->validate([
17             'email' => ['required', 'email'],
18             'password' => ['required'],
19         ]);
20
21         // Mencoba melakukan autentikasi menggunakan metode attempt() dari Auth
22         if (Auth::attempt($credentials)) {
23             // Jika autentikasi berhasil, kirim respons JSON dengan status true, pesan sukses, dan role pengguna
24             return response()->json([
25                 'status' => true,
26                 'message' => "Success",
27                 'role' => Auth::user()->role,
28             ]);
29         }
30
31         // Jika autentikasi gagal, kirim respons JSON dengan status false dan pesan kegagalan
32         return response()->json([
33             'status' => false,
34             'message' => "Fail"
35         ]);
36     }
37
38     public function getuser()
39     {
40         // Ambil data pengguna
41         $users = Auth::user();
42
43         return response()->json([
44             'status' => true,
45             'message' => "Success",
46             'data' => $users
47         ]);
48     }
49 }
50

```

- Melakukan validasi terhadap data yang diterima dari request (email dan password).
- Menggunakan metode attempt() dari Auth untuk mencoba melakukan autentikasi.
- Jika autentikasi berhasil, mengembalikan respons JSON dengan status true dan pesan sukses.
- Jika autentikasi gagal, mengembalikan respons JSON dengan status false dan pesan kegagalan.

## 9. Membuat rute untuk mengelola permintaan url. Semua permintaan dikelola oleh

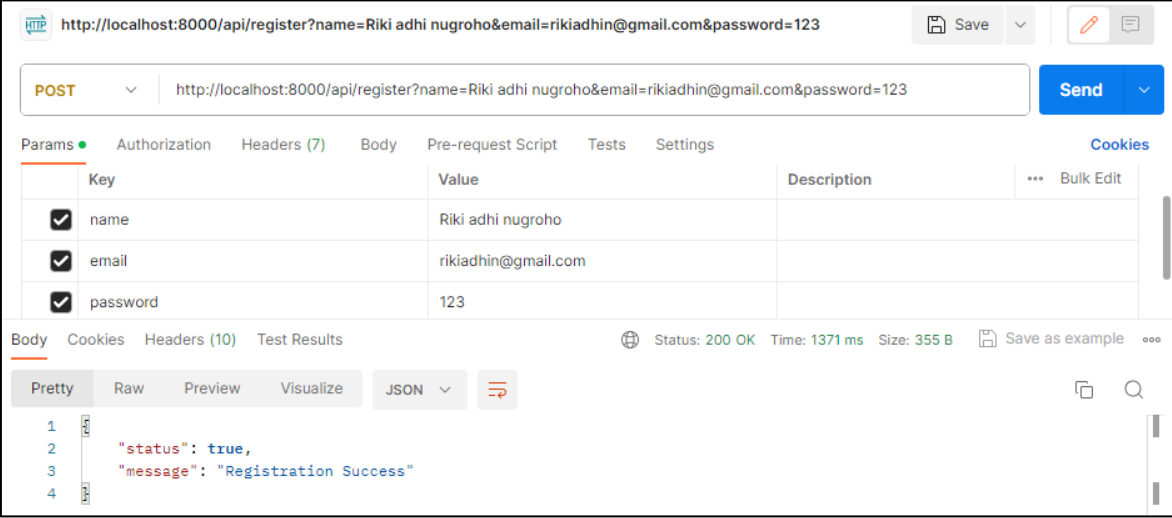


api.php

```
1
2 Route::middleware('auth:sanctum')->get('/user', function (Request $request) {
3     return $request->user();
4 });
5
6 Route::resource('/siswa', SiswaController::class);
7
8 Route::post('/register', [RegisterController::class, 'store']);
9
10 Route::post('/login', [LoginController::class, 'check']);
11 Route::get('/login', [LoginController::class, 'getuser']);
12
13 Route::put('siswa/{id}', 'SiswaController@update');
14
```

10. Selanjutnya melakukan pengujian API, disini saya menggunakan Postman.

➤ Berikut untuk menguji register.



Postman interface showing a successful POST request to the API endpoint `http://localhost:8000/api/register?name=Riki adhi nugroho&email=rikiadhin@gmail.com&password=123`. The request body contains the following parameters:

Key	Value
name	Riki adhi nugroho
email	rikiadhin@gmail.com
password	123

The response status is 200 OK, with a time of 1371 ms and a size of 355 B. The response body is a JSON object:

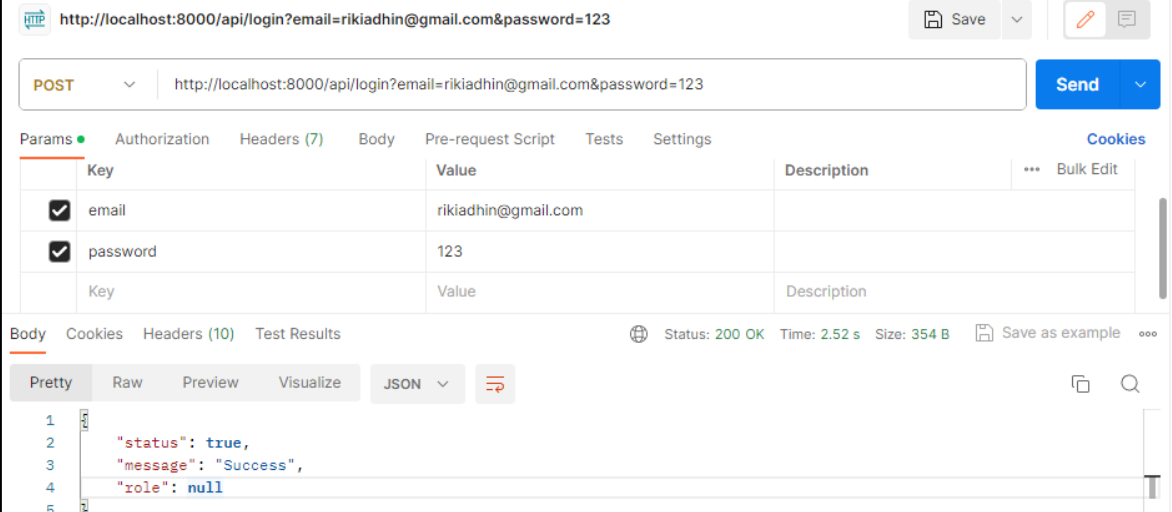
```
{
  "status": true,
  "message": "Registration Success"
}
```

Below the response, a table shows the user record stored in the database:

id	name	email	email_verified_at	password	remember_token	created_at	updated_at
1	adhi nugroho	rikiadhin@gmail.com	NULL	\$2y\$12\$J6CF0Pss1ki5Wh3d.Tcy4uBC8JdSoieG5QFyGqS8.S...	NULL	2023-12-12 03:34:54	2023-12-12 03:34:54

➤ Data berhasil masuk ke database. Selanjutnya menguji login dengan

**password benar.**

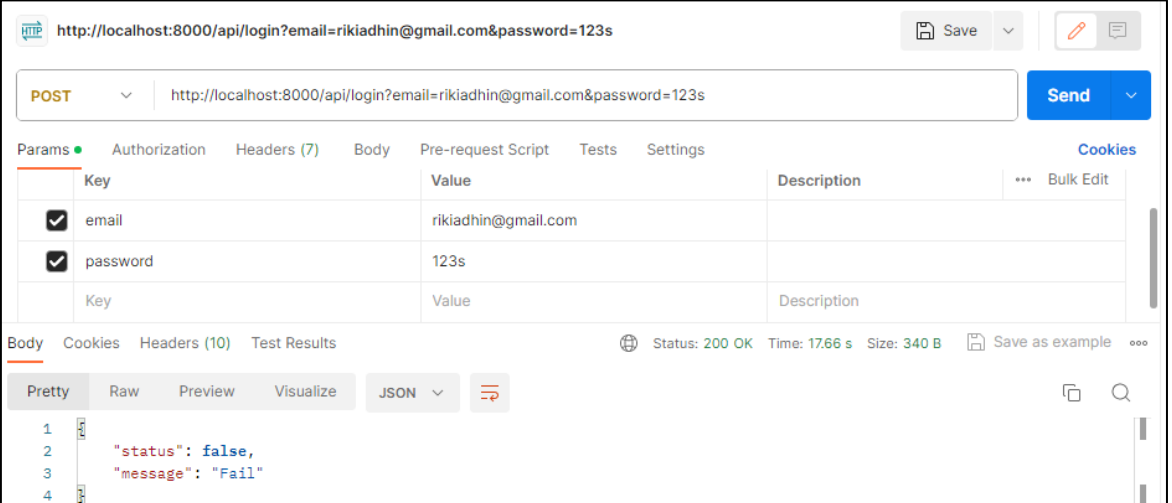


The screenshot shows a REST client interface with a POST request to `http://localhost:8000/api/login?email=rikiadhin@gmail.com&password=123`. The request parameters are `email=rikiadhin@gmail.com` and `password=123`. The response status is 200 OK, and the body is a JSON object: `{ "status": true, "message": "Success", "role": null }`.

Key	Value	Description
email	rikiadhin@gmail.com	
password	123	

```
1 {
2   "status": true,
3   "message": "Success",
4   "role": null
5 }
```

- Data berhasil masuk ke database. Selanjutnya menguji login dengan **password salah.**



The screenshot shows a REST client interface with a POST request to `http://localhost:8000/api/login?email=rikiadhin@gmail.com&password=123s`. The request parameters are `email=rikiadhin@gmail.com` and `password=123s`. The response status is 200 OK, and the body is a JSON object: `{ "status": false, "message": "Fail" }`.

Key	Value	Description
email	rikiadhin@gmail.com	
password	123s	

```
1 {
2   "status": false,
3   "message": "Fail"
4 }
```

11. Pada bagian frontend membuat tampilan login terlebih dahulu untuk halaman awal yang ditampilkan. Saya menambahkan folder 'auth' di dalam folder 'views'

untuk meletakkan file Login.vue dan Register.vue.

➤ **Login.vue**

```
laravel_FrontEnd > Praktikum_Vue_Laravel_CRUD2 > frontend > src > views > validation > Login.vue > ...
1  <template>
2    <div class="container mt-5">
3      <div class="row justify-content-center">
4        <div class="col-md-6">
5          <div class="card border-0 rounded shadow">
6            <div class="card-body">
7              <h2 class="text-center">Login</h2>
8            </div>
9            <form @submit.prevent="LoginData">
10             <div class="form-group">
11               <label>Email</label>
12               <input
13                 type="email"
14                 v-model="siswa.email"
15                 class="form-control"
16                 placeholder="Email" />
17             </div>
18             <div class="form-group">
19               <label>Password</label>
```

```
laravel_FrontEnd > Praktikum_Vue_Laravel_CRUD2 > frontend > src > views > validation > Login.vue > ...
20             <label>Password</label>
21             <input
22               type="password"
23               v-model="siswa.password"
24               class="form-control"
25               placeholder="Password" />
26             </div>
27             <button type="submit" class="btn btn-primary btn-block mt-3">
28               Login
29             </button>
30           </form>
31         </div>
32       </div>
33       <p class="mt-3 text-center">
34         Belum punya akun?
35         <router-link to="/register">Daftar di sini</router-link>
36       </p>
37     </div>
38   </div>
39 </div>
40 </div>
41 </div>
42 </template>
```

```

laravel_FrontEnd > Praktikum_Vue_Laravel_CRUD2 > frontend > src > views > validation > Login.vue > ...
44 <script>
45 import axios from "axios";
46
47 export default {
48   name: "Login",
49   data() {
50     return {
51       siswa: {
52         email: "",
53         password: "",
54       },
55     };
56   },
57   methods: {
58     LoginData() {
59       axios.post("http://127.0.0.1:8000/api/login", this.siswa)
60       .then(({ data }) => {
61         try {
62           // Login berhasil
63           if (data.status) {
64             // Berhasil login, tentukan alur berdasarkan role
65             if (data.role === "admin") {
66               console.log(data)
67               // Pengguna dengan peran admin, arahkan ke halaman siswa
68               this.$router.push({ name: "siswa.siswa" });

```

```

laravel_FrontEnd > Praktikum_Vue_Laravel_CRUD2 > frontend > src > views > validation > Login.vue > ...
66       console.log(data)
67       // Pengguna dengan peran admin, arahkan ke halaman siswa
68       this.$router.push({ name: "siswa.siswa" });
69     } else if (data.role === "user") {
70       // Pengguna dengan peran user, arahkan ke halaman user
71       this.$router.push({ name: "siswa.user" });
72     } else {
73       // Jika peran tidak sesuai, mungkin ada kesalahan
74       alert("Invalid role. Please contact administrator.");
75     }
76   } else {
77     // Jika status login adalah false, tampilkan pesan kegagalan
78     alert("Login failed. Please check your credentials.");
79   }
80 } catch (err) {
81   alert("Error, please try again");
82 }
83 })
84 .catch((error) => {
85   console.error("Error during login:", error);
86   alert("Error during login. Please try again.");
87 });
88 },
89 },
90 };
91 </script>

```

## ➤ Register.vue

```
Register.vue > {} template > div.container.mt-5 > div.row.justify-content-center > div.col-md-6 > div.card.border-0.rounded.shadow > div.card-body >
1 <template>
2   <div class="container mt-5">
3     <div class="row justify-content-center">
4       <div class="col-md-6">
5         <div class="card border-0 rounded shadow">
6           <div class="card-body">
7             <h2 class="text-center mb-4">Register</h2>
8
9             <form @submit.prevent="saveData">
10              <div class="form-group">
11                <label for="name">Name</label>
12                <input type="text" v-model="siswa.name" id="name" class="form-control" placeholder="Enter your name" />
13              </div>
14
15              <div class="form-group">
16                <label for="email">Email</label>
17                <input type="email" v-model="siswa.email" id="email" class="form-control"
18                  placeholder="Enter your email" />
19              </div>
20
```

```
Register.vue > {} template > div.container.mt-5 > div.row.justify-content-center > div.col-md-6 > div.card.border-0.rounded.shadow >
21      <div class="form-group">
22        <label for="password">Password</label>
23        <input type="password" v-model="siswa.password" id="password" class="form-control"
24          placeholder="Enter your password" />
25      </div>
26
27      <button type="submit" class="btn btn-success btn-block mt-3">Register</button>
28    </form>
29  </div>
30 </div>
31 </div>
32 </div>
33 </div>
34 </template>
```

```
laravel_FrontEnd > Praktikum_Vue_Laravel_CRUD2 > frontend > src > views > validation > Register.vue >
37 <script>
38   import axios from "axios";
39
40   export default {
41     name: "Register",
42     data() {
43       return {
44         result: {},
45         siswa: {
46           name: "",
47           email: "",
48           password: "",
49         },
50       };
51     },
52     created() { },
53     mounted() {
54       console.log("mounted() called.....");
55     },
56   };
57 }
```

```

58   methods: {
      Complexity is 4 Everything is cool!
59     saveData() {
60       axios
61       .post("http://127.0.0.1:8000/api/register", this.siswa)
      Complexity is 3 Everything is cool!
62       .then(({ data }) => {
63         console.log(data);
64         try {
65           alert("Registrasi telah berhasil di lakukan");
66           console.log(data)
67           this.$router.push({ name: "auth.login" });
68         } catch (err) {
69           alert("failed");
70         }
71       });
72     },
73   },
74 };
75 </script>

```

12. Menambahkan route. Semua rute dikelola di vuejs dengan di dalam folder router/index.js

```

5   {
6     path: "/",
7     name: "validation.login",
8     component: () => import("@/views/validation/Login.vue"),
9   },
10  {
11    path: "/register",
12    name: "validation.register",
13    component: () => import("@/views/validation/Register.vue"),
14  },

```

13. Berikut adalah tampilan dari login dan register :

The image shows a login form with the following elements:

- Title:** Login
- Email Input:** A text input field with the placeholder text "Email".
- Password Input:** A text input field with the placeholder text "Password".
- Login Button:** A blue button with the text "Login".
- Registration Link:** A link that says "Belum punya akun? [Daftar di sini](#)".

# Register

Name

Enter your name

Email

Enter your email

Password

Enter your password

Register

14. Selanjutnya membuat role untuk admin dan user. Menambahkan kolom rule pada backend/database/migrations/create\_users\_table.php

```
12 public function up(): void
13 {
14     Schema::create('users', function (Blueprint $table) {
15         $table->id();
16         $table->string('name');
17         $table->string('email')->unique();
18         $table->timestamp('email_verified_at')->nullable();
19         $table->string('password');
20         $table->string('role')->default('user');
21         $table->rememberToken();
22         $table->timestamps();
23     });
24 }
```

Lalu jalankan perintah php artisan migrate

```
PS C:\xampp\htdocs\laravel_FrontEnd\Praktikum_Vue_Laravel_CRUD2\backend> php artisan migrate
```

15. Dan terlihat di database bahwa kolom role berhasil ditambahkan.

#	Nama	Jenis	Penyortiran	Atribut	Tak Terilai	Bawaan
<input type="checkbox"/> 1	id	bigint(20)		UNSIGNED	Tidak	Tidak ada
<input type="checkbox"/> 2	name	varchar(255)	utf8mb4_unicode_ci		Tidak	Tidak ada
<input type="checkbox"/> 3	email	varchar(255)	utf8mb4_unicode_ci		Tidak	Tidak ada
<input type="checkbox"/> 4	email_verified_at	timestamp			Ya	NULL
<input type="checkbox"/> 5	password	varchar(255)	utf8mb4_unicode_ci		Tidak	Tidak ada
<input type="checkbox"/> 6	role	varchar(255)	utf8mb4_unicode_ci		Tidak	user
<input type="checkbox"/> 7	remember_token	varchar(100)	utf8mb4_unicode_ci		Ya	NULL
<input type="checkbox"/> 8	created_at	timestamp			Ya	NULL
<input type="checkbox"/> 9	updated_at	timestamp			Ya	NULL

16. Lalu membuat file user.vue di folder frontend. Pada dasarnya code sama dengan siswa.vue tetapi karena user tidak dapat melakukan perubahan dan hanya dapat melihat data, maka disini hilangkan akses edit, tambah, dan hapus. Hanya menampilkan data.

```

laravel_FrontEnd > Praktikum_Vue_Laravel_CRUD2 > frontend > src > views > siswa > User.vue > {}
1  <template>
2  <div class="container mt-5">
3    <div class="row">
4      <div class="col-md-12">
5        <div class="card border-0 rounded shadow">
6          <div class="card-body">
7            <h4>DATA SISWA</h4>
8
9            <hr />
10
11           <router-link
12             :to="{ name: 'siswa.create' }"
13             class="btn btn-md btn-success"
14           >
15             TAMBAH SISWA
16           </router-link>
17
18           <table class="table table-striped table-bordered mt-4">
19             <thead class="thead-dark">
20               <tr>
21                 <th scope="col">No</th>
22                 <th scope="col">Nama</th>
23                 <th scope="col">Alamat</th>
24                 <th scope="col">Kota</th>
25                 <th scope="col">Provinsi</th>
26                 <th scope="col">E-mail</th>
27               </tr>
28             </thead>

```



```

laravel_FrontEnd > Praktikum_Vue_Laravel_CRUD2 > frontend > src > views > siswa > User.vue > {}
29      <tbody>
30      <tr v-for="(siswa, index) in posts.data" :key="index">
31      <td>{{ siswa.id }}</td>
32      <td>{{ siswa.nama }}</td>
33      <td>{{ siswa.alamat }}</td>
34      <td>{{ siswa.kota }}</td>
35      <td>{{ siswa.provinsi }}</td>
36      <td>{{ siswa.email }}</td>
37      </tr>
38      </tbody>
39      </table>
40
41      <!-- Menampilkan navigasi halaman -->
42      <nav aria-label="Page navigation">
43      <ul class="pagination">
44      <li
45      class="page-item"
46      :class="{ disabled: posts.current_page === 1 }"
47      >
48      <button
49      @click="fetchData(posts.current_page - 1)"
50      class="page-link"
51      aria-label="Previous"
52      >
53      <span aria-hidden="true">&laquo;</span>
54      </button>
55      </li>

```

```

laravel_FrontEnd > Praktikum_Vue_Laravel_CRUD2 > frontend > src > views > siswa > User.vue > {} template
56      <li
57      class="page-item"
58      v-for="page in posts.last_page"
59      :key="page"
60      :class="{ active: page === posts.current_page }"
61      >
62      <button @click="fetchData(page)" class="page-link">
63      {{ page }}
64      </button>
65      </li>
66      <li
67      class="page-item"
68      :class="{ disabled: posts.current_page === posts.last_page }"
69      >
70      <button
71      @click="fetchData(posts.current_page + 1)"
72      class="page-link"
73      aria-label="Next"
74      >
75      <span aria-hidden="true">&raquo;</span>
76      </button>
77      </li>
78      </ul>
79      </nav>
80      </div>
81      </div>
82      </div>
83      </div>
84      </div>
85      </template>

```

```

laravel_FrontEnd > Praktikum_Vue_Laravel_CRUD2 > frontend > src > views > siswa >
87 <script>
88 import axios from "axios";
89 import { onMounted, ref } from "vue";
90
91 export default {
92   Complexity is 10 It's time to do something...
93   setup() {
94     let posts = ref({
95       data: [],
96       current_page: 1,
97       last_page: 1,
98     });
99
100     Complexity is 3 Everything is cool!
101     const fetchData = (page) => {
102       axios
103       .get(`http://localhost:8000/api/siswa?page=${page}`)
104       .then((response) => {
105         posts.value = response.data.data;
106       })
107       .catch((error) => {
108         console.log(error.response.data);
109       });
110     };
111
112     onMounted(() => {
113       fetchData(1);
114     });

```

```

laravel_FrontEnd > Praktikum_Vue_Laravel_CRUD2 > frontend > src > views > siswa > User.vue >
Complexity is 4 Everything is cool!
114 const deleteSiswa = (siswaId) => {
115   if (confirm("Apakah Anda yakin ingin menghapus siswa ini?")) {
116     axios
117     .delete(`http://localhost:8000/api/siswa/${siswaId}`)
118     .then(() => {
119       fetchData(posts.value.current_page);
120     })
121
122     .catch((error) => {
123       console.log(error.response.data);
124     });
125   }
126 };
127
128 return {
129   posts,
130   deleteSiswa,
131   fetchData,
132 };
133 },
134 };
135 </script>
136
137 <style>
138 body {
139   background: lightgray;
140 }
141 </style>

```

## 17. Frontend/src/views/auth/Login.vue

```
laravel_FrontEnd > Praktikum_Vue_Laravel_CRUD2 > frontend > src > views > validation > Login.vue > ...
44 <script>
45 import axios from "axios";
46
47 export default {
48   name: "Login",
49   data() {
50     return {
51       siswa: {
52         email: "",
53         password: "",
54       },
55     };
56   },
57   methods: {
58     LoginData() {
59       axios.post("http://127.0.0.1:8000/api/login", this.siswa)
60       .then(({ data }) => {
61         try {
62           // Login berhasil
63           if (data.status) {
64             // Berhasil login, tentukan alur berdasarkan role
65             if (data.role === "admin") {
66               console.log(data)
67               // Pengguna dengan peran admin, arahkan ke halaman siswa
68               this.$router.push({ name: "siswa.siswa" });
```

```
66 console.log(data)
67 // Pengguna dengan peran admin, arahkan ke halaman siswa
68 this.$router.push({ name: "siswa.siswa" });
69 } else if (data.role === "user") {
70   // Pengguna dengan peran user, arahkan ke halaman user
71   this.$router.push({ name: "siswa.user" });
72 } else {
73   // Jika peran tidak sesuai, mungkin ada kesalahan
74   alert("Invalid role. Please contact administrator.");
75 }
76 } else {
77   // Jika status login adalah false, tampilkan pesan kegagalan
78   alert("Login failed. Please check your credentials.");
79 }
80 } catch (err) {
81   alert("Error, please try again");
82 }
83 })
84 .catch((error) => {
85   console.error("Error during login:", error);
86   alert("Error during login. Please try again.");
87 });
88 },
89 },
90 };
91 </script>
```

18. Lalu di backend menambahkan pada  
backend/app/Http/Controllers/LoginController.php

```
if (Auth::attempt($credentials)) {  
    // Jika autentikasi berhasil, kirim respons JSON dengan  
    return response()->json([  
        'status' => true,  
        'message' => "Success",  
        'role' => Auth::user()->role,  
    ]);  
}
```

## C. Testing

3	Kaylah Green DDS	5778 Missouri Grove	Gerardochester	Ohio	jordi.will@hotmail.com
4	Laurine Steuber IV	933 Armstrong Lodge	Makenzieton	Ohio	torp.ona@boehm.com
5	Dr. Verna Lockman DVM	8674 Kertzmann Alley Suite 104	Port Xander	Rhode Island	freddy.mcdermott@yahoo.com
6	Albert Zemlak DDS	234 Kuhn Path	Cristianfurt	New York	javier25@kiehn.net
7	Dock Shanahan PhD	165 Lou Manor	Port Yoshikostad	Michigan	roslyn64@moen.com
8	Laila Hamill	94138 Dach Flats	North Telly	Kentucky	zachery28@bins.com
9	Collin Orn	390 Crawford Lane	Leonorview	Illinois	ritchie.annabell@hahn.biz
10	Kendra Gerhold	7732 Ariane Glens	Trantowton	Pennsylvania	ruecker.pearl@gmail.com
11	Mr. Aric Dach MD	3929 Jace Brooks	East Emilliotown	Indiana	sanford.cronin@gmail.com
12	Mr. Cullen Mertz DDS	37575 Rippin Drives Suite 990	Wildermanmouth	South Dakota	rupton@bahringer.net
13	Titus Hand	767 Lemke Rapids Apt. 990	South Elbert	Virginia	blake.lesch@hotmail.com
14	Kobe Windler	935 Fritsch Mountains	Port Nelsonside	Arkansas	tbrakus@botsford.org
15	Van Green	15722 Kulas Vista	Vancemouth	Arkansas	audra45@mclaughlin.com
16	Raymond Kohler	5640 Shields Island	Bryonland	North Dakota	rpredovic@gmail.com

« 1 2 3 4 5 6 7 »

Activate Windows  
Go to Settings to activate Windows

## Register

Name

Email

Password

### Data user tersedia :

id	name	email	email_verified_at	password	role	remember_token
3	admin	admin@gmail.com	2023-12-12 14:32:26	\$2y\$10\$BXHySAJ6m8CY3Veu7JSFgela7DhR/PqZDYOSA7cnrLb...	admin	NULL
4	user	user@gmail.com	2023-12-12 14:35:10	\$2y\$10\$Kj3Kv.imS2r2h8aQepb36euTDcoP0zlfqhcopJ/P1Yx...	user	NULL

### Login sebagai admin :

Email : admin@gmail.com

Password : admin

## Login

Email

Password

Login

Belum punya akun? [Daftar di sini](#)

DATA SISWA						
<button>TAMBAH SISWA</button>						
No	Nama	Alamat	Kota	Provinsi	E-mail	Aksi
2	Dr. Lexie Mann	4177 Gerlach Inlet Suite 656	Ernestinestad	Connecticut	triston.shields@turcotte.com	<button>Edit</button> <button>Delete</button>
3	Kaylah Green DDS	5778 Missouri Grove	Gerardochester	Ohio	jordi.will@hotmail.com	<button>Edit</button> <button>Delete</button>
4	Laurine Steuber IV	933 Armstrong Lodge	Makenzieleton	Ohio	torp.ona@boehm.com	<button>Edit</button> <button>Delete</button>
5	Dr. Verna Lockman DVM	8674 Kertzmann Alley Suite 104	Port Xander	Rhode Island	freddy.mcdermott@yahoo.com	<button>Edit</button> <button>Delete</button>
6	Albert Zemlak DDS	234 Kuhn Path	Cristianfurt	New York	javier25@kiehn.net	<button>Edit</button> <button>Delete</button>
7	Dock Shanahan PhD	165 Lou Manor	Port Yoshikostad	Michigan	roslyn64@moen.com	<button>Edit</button> <button>Delete</button>
8	Laila Hamill	94138 Dach Flats	North Telly	Kentucky	zachery28@bins.com	<button>Edit</button> <button>Delete</button>
9	Collin Orn	390 Crawford Lane	Leonorview	Illinois	ritchie.annabell@hahn.biz	<button>Edit</button> <button>Delete</button>
10	Kendra Gerhold	7732 Ariane Glens	Trantowton	Pennsylvania	ruecker.pearl@gmail.com	<button>Edit</button> <button>Delete</button>

**Login sebagai user :**

Email : user@gmail.com

Password : user

## Login

Email

Password

Login

Belum punya akun? [Daftar di sini](#)

DATA SISWA					
TAMBAH SISWA					
No	Nama	Alamat	Kota	Provinsi	E-mail
2	Dr. Lexie Mann	4177 Gerlach Inlet Suite 656	Ernestinestad	Connecticut	triston.shields@turcotte.com
3	Kaylah Green DDS	5778 Missouri Grove	Gerardocheester	Ohio	jordi.will@hotmail.com
4	Laurine Steuber IV	933 Armstrong Lodge	Makenzieton	Ohio	torp.ona@boehm.com
5	Dr. Verna Lockman DVM	8674 Kertzmann Alley Suite 104	Port Xander	Rhode Island	freddy.mcdermott@yahoo.com
6	Albert Zemlak DDS	234 Kuhn Path	Cristianfurt	New York	javier25@kiehn.net
7	Dock Shanahan PhD	165 Lou Manor	Port Yoshikostad	Michigan	roslyn64@moen.com
8	Laila Hamill	94138 Dach Flats	North Telly	Kentucky	zachery28@bins.com
9	Collin Orn	390 Crawford Lane	Leonorview	Illinois	ritchie.annabell@hahn.biz
10	Kendra Gerhold	7732 Ariane Glens	Trantowton	Pennsylvania	ruecker.pearl@gmail.com
11	Mr. Aric Dach MD	3929 Jace Brooks	East Emiliotown	Indiana	sanford.cronin@gmail.com
12	Mr. Cullen Metz DDS	37575 Rippin Drives Suite 990	Wildermanmouth	South Dakota	rutton@bahringer.net