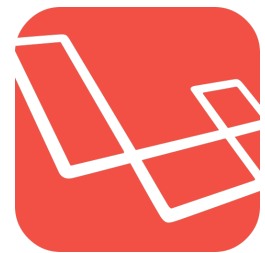


Modul 11

Laravel & Vue



+ **Axios** +



PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ATMA JAYA YOGYAKARTA
SEMESTER GASAL 2020/2021

Yang harus dipersiapkan

1. Internet
2. Text editor (disarankan VSCode)
3. Web browser
4. GD modul 7
5. GD modul 9

Pengertian

▪ Axios

Beberapa framework memiliki built-in HTTP APIs. Angular 2 memiliki http module, JQuery memiliki \$.ajax, hingga Vue 2.0, Vue.js memiliki vue-resource. pada Vue 2.0, developer memutuskan untuk membuat build-in http client modul, namun sampai sekarang masih kalah pamor third-party libraries. Alternatif yang paling sering direkomendasikan adalah Axios.

Axios merupakan http client library yang populer. default axios adalah menggunakan promises yang berjalan pada client maupun server (cocok untuk fetching data selama server-side ngerender). Axios juga cukup enak digunakan di vue. selain itu, karena menggunakan promises, kita dapat mengkombinasikan itu dengan async/await untuk membuat makin simple.

▪ Laravel Database seeding

Seeding pada laravel adalah sebuah fitur untuk mengisi data pada database dengan data dummy atau data testing.

Karena pada saat kita melakukan develop aplikasi atau sistem informasi, pasti kita memerlukan beberapa data sebagai contoh untuk diolah pada aplikasi kita.

Misalnya pada guided kali ini nanti kita akan memerlukan akun untuk masuk kedalam sistem, untuk prakteknya langsung saja ke koding guided.

Persiapan Guided

Pada guided ini, kita akan menyambungkan antara guided 7 (sebagai backend) dengan guided 9 (sebagai front end).

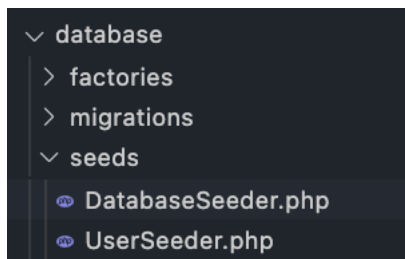
Pertama kita setting backendnya terlebih dahulu, buka folder guided 7 (REST web service laravel), pastikan database sudah ada di phpmyadmin dan terkoneksi dengan project laravel, jika lupa step stepnya bisa dicek kembali guided modul 7. Jika sudah dipastikan konek.

Maka buka terminal bisa lewat view-> terminal bisa juga dengan menekan "ctrl+shift+`",

Ketikkan command berikut pada terminal.

```
| | -> php artisan make:seeder UserSeeder
Seeder created successfully.
```

Maka akan terdapat file baru seperti ini



Buka UserSeeder.php lalu isikan dengan code dibawah ini.

```
1  <?php
2
3  use Illuminate\Database\Seeder;
4
5  class UserSeeder extends Seeder
6  {
7      /**
8       * Run the database seeds.
9       *
10      * @return void
11      */
12      public function run()
13      {
14          DB::table('users')
15              →insert([
16              'name' => 'Bang Jago', //isi dengan nama kalian
17              'email' => 'bangjago@admin.com',
18              'password' => '$2b$10$bFgZ1KRr6bn8E02SDyohuKkUIdpYbd7f5E/aLUh0fIQKhALZD/BG', //bangjago123
19              'created_at' => Carbon\Carbon::now(),
20              'updated_at' => Carbon\Carbon::now()
21          ]);
22      }
23  }
24
```

Untuk passwordnya bisa generate dari <https://passwordhashing.com/BCrypt> , jika sudah maka jalankan command

```
||-> php artisan db:seed --class=UserSeeder
Database seeding completed successfully.
```

Bisa dicek diphpmymadmin pada tabel users jika sudah terisi maka bisa lanjut ketahap berikutnya

Showing rows 0 - 0 (1 total, Query took 0.0014 seconds.)

SELECT * FROM `users`

Profiling [Edit inline] [Edit] [Exp]

Show all | Number of rows: 25 | Filter rows: Search this table

+ Options

	id	name	email	email_verified_at	password
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2	Bang Jago	bangjago@admin.com	NULL	\$2b\$10\$JbFgZ1KRr6bn8EO2SDyohuKkUidpYbd7f5E/aLUhOfI...

Lalu run laravelnya dengan cara menjalankan perintah “php artisan serve”.

```
||-> php artisan serve
Laravel development server started: http://127.0.0.1:8000
[Fri Nov 13 16:08:20 2020] PHP 7.4.5 Development Server (http://127.0.0.1:8000) started
```

Jika berhasil tampilan akan seperti ini.

GET menggunakan postman, “127.0.0.1:8000” itu tergantung dengan apa yang kalian dapat ketika menjalankan dengan perintah “php artisan serve” yaa.

GET http://127.0.0.1:8000/api/product/ Send

Jika responsnya seperti ini, berarti sudah oke. lanjut

```
1 {
2   "message": "Unauthenticated."
3 }
```

Oke untuk persiapan backend(laravel) sudah aman, selanjutnya kita akan setting frontendnya, langkah pertama buka kembali GD modul 9(Vue Cli & Webpack) kalian pada window baru, jika sudah ikuti langkah langkah berikut:

Install axios dengan menggunakan npm, buka terminal baru, lalu jalankan

```
||-> npm install axios
```

Agar axios yang telah kita install dapat digunakan maka perlu menambahkan code dibawah ini pada file main.js.

```
5   import axios from 'axios'
6
7   Vue.config.productionTip = false
8
9   Vue.prototype.$http = axios;
10  Vue.prototype.$api = 'http://127.0.0.1:8000/api';
11
```

Line ke-10 itu disesuaikan dengan url api kalian.

Lalu tambahkan route untuk login, Ubahlah file router.js dengan code dibawah ini.

```
12  const router = new VueRouter({
13    mode: "history",
14    routes: [{
15      path: "/",
16      component: importComponent('DashboardLayout'),
17      children: [
18        //Dashboard
19        {
20          path: "/dashboard",
21          name: "Dahboard",
22          meta: {title: 'Dashboard'},
23          component: importComponent('Dashboard'),
24        },
25        //Products
26        {
27          path: "/products",
28          name: "Products",
29          meta: {title: 'Products'},
30          component: importComponent('DataMaster/Products'),
31        },
32      ],
33    },
34    //Login
35    {
36      path: "/login",
37      name: "login",
38      meta: {title: 'Login'},
39      component: importComponent('Login'),
40    },
41    {
42      path: '*',
43      redirect: '/'
44    },
45  ],
46  });
47  //Mengset judul
48  router.beforeEach((to, from, next) => {
49    document.title = to.meta.title
50    next()
51  });
52
53  export default router;
```

Disini kita akan menambahkan tambilan navbar pada DashboardLayout gd 9 dengan cara mengganti isi DashboardLayout.vue dengan code berikut

```

1. <!-- @format -->
2.
3. <template>
4.   <div class="dashboard">
5.     <v-navigation-drawer v-model="drawer" class="fullheight" width="256" app>
6.       <v-list-item>
7.         <v-list-item-content>
8.           <v-list-item-title class="title"> Nama Kalian </v-list-item-title>
9.           <v-list-item-subtitle> NPM </v-list-item-subtitle>
10.        </v-list-item-content>
11.      </v-list-item>
12.
13.    <v-divider></v-divider>
14.
15.    <v-list dense nav>
16.      <v-list-item
17.        v-for="item in items"
18.        :key="item.title"
19.        link
20.        tag="router-link"
21.        :to="item.to"
22.      >
23.        <v-list-item-content>
24.          <v-list-item-title>{{ item.title }}</v-list-item-title>
25.        </v-list-item-content>
26.      </v-list-item>
27.    </v-list>
28.  </v-navigation-drawer>
29.  <v-app-bar app fixed height="75px">
30.    <v-app-bar-nav-icon @click.stop="drawer = !drawer"></v-app-bar-nav-icon>
31.    <VSpacer />
32.    <v-toolbar-items>
33.      <v-btn text router><v-icon>mdi-power</v-icon></v-btn>
34.    </v-toolbar-items>
35.  </v-app-bar>
36.  <div class="fullheight pa-5">
37.    <router-view></router-view>
38.  </div>
39. </div>
40. </template>
41. <script>
42. export default {
43.   name: "Dashboard",
44.   data() {
45.     return {
46.       drawer: true,
47.       items: [
48.         { title: "Dashboard", to: "/dashboard" },
49.         { title: "Product", to: "/products" },
50.       ],
51.     };
52.   },
53. };
54. </script>
55.
56. <style scoped>
57. .fullheight {
58.   min-height: 100vh !important;
59. }
60. .router {
61.   text-decoration: none;

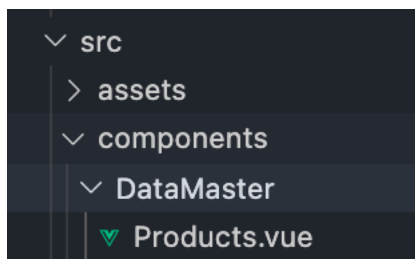
```

```

62.   color: black;
63. }
64. </style>

```

Nah karena disini kita akan fetch api dari guided 7 yang tidak lain tidak bukan adalah data product, maka kita harus membuat tampilan untuk menampung respons dari data product dan kita tidak membutuhkan tampilan todo dari guided 9, dengan cara merubah List.vue dalam folder TodoList menjadi seperti gambar dibawah / bisa juga hapus TodoList/List.vue dan buat baru DataMaster/Products.vue



Isikan dengan code dibawah ini.

```

1.  <!-- @format -->
2.
3.  <template>
4.    <v-main class="list">
5.      <h3 class="text-h3 font-weight-medium mb-5"> Products </h3>
6.
7.      <v-card>
8.        <v-card-title>
9.          <v-text-field
10.            v-model="search"
11.            append-icon="mdi-magnify"
12.            label="Search"
13.            single-line
14.            hide-details
15.          ></v-text-field>
16.          <v-spacer></v-spacer>
17.          <v-btn color="success" dark @click="dialog = true">
18.            Tambah
19.          </v-btn>
20.        </v-card-title>
21.        <v-data-table :headers="headers" :items="products" :search="search">
22.          <template v-slot:[`item.actions`]="{ item }">
23.            <v-btn small class="mr-2" @click="editHandler(item)">
24.              edit
25.            </v-btn>
26.            <v-btn small @click="deleteHandler(item.id)">
27.              delete
28.            </v-btn>
29.          </template>

```



```

30.     </v-data-table>
31. </v-card>
32.
33. <v-dialog v-model="dialog" persistent max-width="600px">
34.   <v-card>
35.     <v-card-title>
36.       <span class="headline">{{ formTitle }} Produk</span>
37.     </v-card-title>
38.     <v-card-text>
39.       <v-container>
40.         <v-text-field
41.           v-model="form.nama_produk"
42.           label="Nama produk"
43.           required
44.         ></v-text-field>
45.
46.         <v-text-field
47.           v-model="form.satuan"
48.           label="Satuan"
49.           required
50.         ></v-text-field>
51.
52.         <v-text-field
53.           v-model="form.harga_jual"
54.           label="Harga jual"
55.           required
56.         ></v-text-field>
57.
58.         <v-text-field
59.           v-model="form.stok"
60.           label="Stok"
61.           required
62.         ></v-text-field>
63.       </v-container>
64.     </v-card-text>
65.     <v-card-actions>
66.       <v-spacer></v-spacer>
67.       <v-btn color="blue darken-1" text @click="cancel">
68.         Cancel
69.       </v-btn>
70.       <v-btn color="blue darken-1" text @click="setForm">
71.         Save
72.       </v-btn>
73.     </v-card-actions>
74.   </v-card>
75. </v-dialog>
76.
77. <v-dialog v-model="dialogConfirm" persistent max-width="400px">
78.   <v-card>
79.     <v-card-title>
80.       <span class="headline">warning!</span>
81.     </v-card-title>
82.     <v-card-text>
83.       Anda yakin ingin menghapus produk ini?
84.     </v-card-text>
85.     <v-card-actions>
86.       <v-spacer></v-spacer>
87.       <v-btn color="blue darken-1" text @click="dialogConfirm = false">
88.         Cancel
89.       </v-btn>
90.       <v-btn color="blue darken-1" text @click="deleteData">

```

```

91.         Delete
92.     </v-btn>
93. </v-card-actions>
94. </v-card>
95. </v-dialog>
96.
97. <v-snackbar v-model="snackbar" :color="color" timeout="2000" bottom>
98.     {{error_message}}
99. </v-snackbar>
100. </v-main>
101. </template>
102. <script>
103. export default {
104.     name: "List",
105.     data() {
106.         return {
107.             inputType: 'Tambah',
108.             load: false,
109.             snackbar: false,
110.             error_message: '',
111.             color: '',
112.             search: null,
113.             dialog: false,
114.             dialogConfirm: false,
115.             headers: [
116.                 { text: "Nama produk",
117.                   align: "start",
118.                   sortable: true,
119.                   value: "nama_produk" },
120.                 { text: "Satuan", value: "satuan" },
121.                 { text: "Harga jual", value: "harga_jual" },
122.                 { text: "stok", value: "stok" },
123.                 { text: "Actions", value: "actions" },
124.             ],
125.             product: new FormData,
126.             products: [],
127.             form: {
128.                 nama_produk: null,
129.                 satuan: null,
130.                 harga_jual: null,
131.                 stok: null,
132.             },
133.             deleteId: '',
134.             editId: ''
135.         };
136.     },
137.     methods: {
138.         setForm() {
139.             if (this.inputType === 'Tambah') {
140.                 this.save()
141.             } else {
142.                 this.update()
143.             }
144.         },
145.         //read data product
146.         readData() {
147.             var url = this.$api + '/product'
148.             this.$http.get(url, {
149.                 headers: {
150.                     'Authorization': 'Bearer ' + localStorage.getItem('token')
151.                 }

```

```

152.         }).then(response => {
153.             this.products = response.data.data
154.         })
155.     },
156.     //simpan data produk
157.     save() {
158.         this.product.append('nama_produk', this.form.nama_produk);
159.         this.product.append('satuan', this.form.satuan);
160.         this.product.append('harga_jual', this.form.harga_jual);
161.         this.product.append('stok', this.form.stok);
162.
163.         var url = this.$api + '/product/'
164.         this.load = true
165.         this.$http.post(url, this.product, {
166.             headers: {
167.                 'Authorization': 'Bearer ' + localStorage.getItem('token')
168.             }
169.         }).then(response => {
170.             this.error_message=response.data.message;
171.             this.color="green"
172.             this.snackbar=true;
173.             this.load = false;
174.             this.close();
175.             this.readData(); //mengambil data
176.             this.resetForm();
177.         }).catch(error => {
178.             this.error_message=error.response.data.message;
179.             this.color="red"
180.             this.snackbar=true;
181.             this.load = false;
182.         })
183.     },
184.     //ubah data produk
185.     update() {
186.         let newData = {
187.             nama_produk: this.form.nama_produk,
188.             satuan: this.form.satuan,
189.             harga_jual: this.form.harga_jual,
190.             stok: this.form.stok
191.         }
192.         var url = this.$api + '/product/' + this.editId;
193.         this.load = true
194.         this.$http.put(url, newData, {
195.             headers: {
196.                 'Authorization': 'Bearer ' + localStorage.getItem('token')
197.             }
198.         }).then(response => {
199.             this.error_message=response.data.message;
200.             this.color="green"
201.             this.snackbar=true;
202.             this.load = false;
203.             this.close();
204.             this.readData(); //mengambil data
205.             this.resetForm();
206.             this.inputType = 'Tambah';
207.         }).catch(error => {
208.             this.error_message=error.response.data.message;
209.             this.color="red"
210.             this.snackbar=true;
211.             this.load = false;
212.         })

```

```

213.     },
214.     //hapus data produk
215.     deleteData() {
216.         //mengahapus data
217.         var url = this.$api + '/product/' + this.deleteId;
218.         //data dihapus berdasarkan id
219.         this.$http.delete(url, {
220.             headers: {
221.                 'Authorization': 'Bearer ' + localStorage.getItem('token')
222.             }
223.         }).then(response => {
224.             this.error_message=response.data.message;
225.             this.color="green"
226.             this.snackbar=true;
227.             this.load = false;
228.             this.close();
229.             this.readData(); //mengambil data
230.             this.resetForm();
231.             this.inputType = 'Tambah';
232.         }).catch(error => {
233.             this.error_message=error.response.data.message;
234.             this.color="red"
235.             this.snackbar=true;
236.             this.load = false;
237.         })
238.     },
239.     editHandler(item){
240.         this.inputType = 'Ubah';
241.         this.editId = item.id;
242.         this.form.nama_produk = item.nama_produk;
243.         this.form.satuan = item.satuan;
244.         this.form.stok = item.stok;
245.         this.form.harga_jual = item.harga_jual;
246.         this.dialog = true;
247.     },
248.     deleteHandler(id){
249.         this.deleteId = id;
250.         this.dialogConfirm = true;
251.     },
252.     close() {
253.         this.dialog = false
254.         this.inputType = 'Tambah';
255.     },
256.     cancel() {
257.         this.resetForm();
258.         this.readData();
259.         this.dialog = false;
260.         this.inputType = 'Tambah';
261.     },
262.     resetForm() {
263.         this.form = {
264.             nama_produk: null,
265.             satuan: null,
266.             harga_jual: null,
267.             stok: null,
268.         };
269.     },
270. },
271. computed: {
272.     formTitle() {
273.         return this.inputType

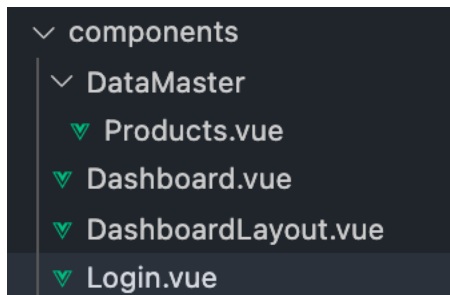
```

```

274.         },
275.     },
276.     mounted() {
277.         this.readData();
278.     },
279. };
280. </script>

```

Lalu kita akan membuat tampilan login untuk masuk kesistem, buat file baru lagi dengan nama Login.vue.



Lalu isikan dengan code ini

```

1. <!-- @format -->
2.
3. <template>
4.   <main>
5.     <v-container fluid fill-height class="posisinya">
6.       <v-layout flex align-center justify-center>
7.         <v-flex xs12 sm6 elevation-6>
8.           <v-toolbar class="grey darken-3">
9.             <v-toolbar-title class="grey--text">
10.              <h1>Met dteng y bossque, xixixi</h1>
11.            </v-toolbar-title>
12.          </v-toolbar>
13.          <v-card>
14.            <v-card-text class="pt-4">
15.              <div>
16.                <v-form v-model="valid" ref="form">
17.                  <v-text-field label="E-mail" v-
18.                    model="email" :rules="emailRules"
19.                      required></v-text-field>
20.                  <v-text-field label="Password" v-
21.                    model="password" type="password" min="8"
22.                      :rules="passwordRules" counter required></v-
23.                    text-field>
24.                  <v-layout justify-end>
25.                    <v-btn class="mr-2" @click="submit"
26.                      :class="{ 'grey darken-1 white--
27.                        text' : valid, disabled: !valid }">Go
28.                  </v-btn>
29.                    <v-btn @click="clear"
30.                      class="grey darken-3 white--text">Clear
31.                  </v-btn>
32.                </v-layout>
33.              </div>
34.            </v-card-text>
35.          </v-card>
36.        </v-flex>
37.      </v-layout>
38.    </v-container>
39.  </main>
40. </template>
41.
42. <script>
43.   export default {
44.     data() {
45.       return {
46.         valid: true,
47.         email: '',
48.         password: '',
49.         emailRules: {
50.           required: true,
51.           email: true
52.         },
53.         passwordRules: {
54.           required: true,
55.           min: 8
56.         }
57.       };
58.     },
59.     methods: {
60.       submit() {
61.         // TODO: Implement login logic
62.       },
63.       clear() {
64.         this.valid = true;
65.         this.email = '';
66.         this.password = '';
67.       }
68.     }
69.   };
70. </script>

```

```

29.                 </v-form>
30.             </div>
31.         </v-card-text>
32.     </v-card>
33.     <v-snackbar v-
model="snackbar" :color="color" timeout="2000" bottom>
34.         {{error_message}}
35.     </v-snackbar>
36. </v-flex>
37. </v-layout>
38. </v-container>
39. </main>
40. </template>
41.
42. <style>
43.     @import url("https://fonts.googleapis.com/css?family=Jolly%20Lodger");
44.     .grey--text {
45.         font-family: "Jolly Lodger";
46.     }
47.     .posisinya {
48.         position: absolute;
49.         top: 20px;
50.         left: 0px;
51.         right: 0px;
52.     }
53. </style>
54.
55. <script>
56.     export default {
57.         name: "Login",
58.         data() {
59.             return {
60.                 load: false,
61.                 snackbar: false,
62.                 error_message: '',
63.                 color: '',
64.                 valid: false,
65.                 password: '',
66.                 passwordRules: [
67.                     (v) => !!v || 'Password tidak boleh kosong :(',
68.                 ],
69.                 email: '',
70.                 emailRules: [
71.                     (v) => !!v || 'E-mail tidak boleh kosong :(',
72.                 ],
73.             };
74.         },
75.         methods: {
76.             submit() {
77.                 if (this.$refs.form.validate()) { //cek apakah data yang akan dikirim sudah
valid
78.                     this.load = true
79.                     this.$http.post(this.$api + '/login', {
80.                         email: this.email,
81.                         password: this.password
82.                     }).then(response => {
83.                         localStorage.setItem('id', response.data.user.id); //menyim
pan id user yang sedang login
84.                         localStorage.setItem('token', response.data.access_token);
//menyimpan auth token
85.                         this.error_message=response.data.message;

```

```

86.         this.color="green"
87.         this.snackbar=true;
88.         this.load = false;
89.         this.clear();
90.         this.$router.push({
91.             name: 'Products'
92.         })
93.     }).catch(error => {
94.         this.error_message=error.response.data.message;
95.         this.color="red"
96.         this.snackbar=true;
97.         localStorage.removeItem('token')
98.         this.load = false
99.     })
100.    }
101. },
102. clear() {
103.     this.$refs.form.reset() //Clear form login
104. }
105. },
106. };
107. </script>

```

Jika sekiranya sudah diikuti dengan benar maka jalankan command berikut untuk menjalankan project vuenya.

```
|l-> npm run serve
```


Lalu buka url yang muncul /login, dalam kasus ini “http://localhost:8080/login”

```
DONE Compiled successfully in 3875ms
```

```
App running at:
```

```
- Local:   http://localhost:8080/
- Network: http://10.200.9.87:8080/
```

Jika berhasil dengan mulus maka tampilan akan menjadi seperti ini.



Met dteng y bassque, xxxxi

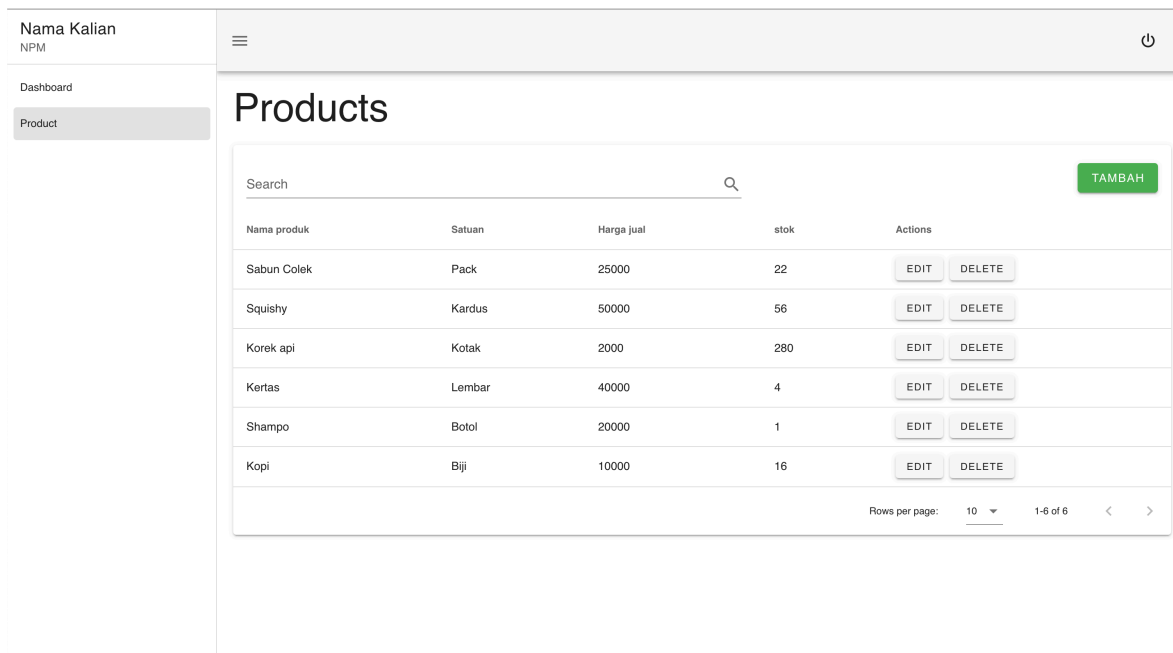
E-mail

Password

GO CLEAR

Lalu untuk masuk kedalam sistem masukkan email dan password dari seeder yang telah dilakukan diawal, lalu klik “GO”,

Jika berhasil maka akan masuk ke halaman ini.



Nama Kalian
NPM

Dashboard

Product

Products

Search

TAMBAH

Nama produk	Satuan	Harga jual	stok	Actions
Sabun Colek	Pack	25000	22	EDIT DELETE
Squishy	Kardus	50000	56	EDIT DELETE
Korek api	Kotak	2000	280	EDIT DELETE
Kertas	Lembar	40000	4	EDIT DELETE
Shampo	Botol	20000	1	EDIT DELETE
Kopi	Biji	10000	16	EDIT DELETE

Rows per page: 10 1-6 of 6

Jangan lupa mengganti nama dan npm :D

Silahkan cobalah CRUDSnya jika lancar maka siap untuk dikumpulkan :D, selamat bereksplor ria.