

VERSION 1.2
DESEMBER 13, 2024



PEMROGRAMAN WEB

FRONTEND DAN FRONTEND FRAMEWORK

DISUSUN OLEH:
OGYA ADYATMA PUTRA
GERALDI NATHAN TOMMY SAPUTRA

DIAUDIT OLEH:
AMINUDIN, S.KOM., M.CS

LAB. INFORMATIKA
UNIVERSITAS MUHAMMADIYAH MALANG

PEMROGRAMAN WEB

PERSIAPAN MATERI

- <https://nodejs.org/>
- <https://vuejs.org/>
- <https://vite.dev/>

TUJUAN

Praktikan diharapkan dapat menjalankan framework frontend dan melakukan integrasi dengan backend pada materi sebelumnya pada modul 5.

TARGET MODUL

1. Praktikan mampu memahami konsep frontend framework
2. Praktikan mampu melakukan integrasi backend kedalam frontend
3. Praktikan mampu melakukan CRUD kedalam frontend

PERSIAPAN SOFTWARE/APLIKASI

Hardware

- Laptop / PC
- Koneksi Internet

Software

- Text editor (Visual Studio Code, Sublime, Atom, Notepad++, dll)
- Laragon / XAMPP
- Composer
- Nodejs
- NPM

Tambahan (Extension Visual Studio Code) (Opsional)

- HTML CSS Support by Ecmel
- Prettier – Code Formatter by Prettier

MATERI

APA ITU FRONTEND FRAMEWORK?

Frontend framework adalah kerangka kerja yang membantu pengembang situs web dalam membangun User Interface (UI) dengan cepat dan efisien. Penggunaan Framework memungkinkan Developer untuk mengatur tampilan dan fungsionalitas UI, seperti layout, komponen UI, CSS, responsivitas, transisi, dan animasi.

APA ITU VUE.JS?

VueJS adalah sebuah framework JavaScript untuk membangun aplikasi web atau interface situs web agar lebih interaktif. VueJS dapat digunakan untuk membangun aplikasi seperti halaman web, aplikasi seluler, dan aplikasi desktop.

VuesJS menyediakan berbagai macam fitur yang tentunya dapat membantu para developer untuk mempermudah pengembangan website. Fitur-fitur tersebut antara lain sebagai berikut.

1. Components
2. Computed properties
3. Data binding
4. Directives
5. Event handling
6. Dan masih banyak lagi

APA ITU NODEJS?

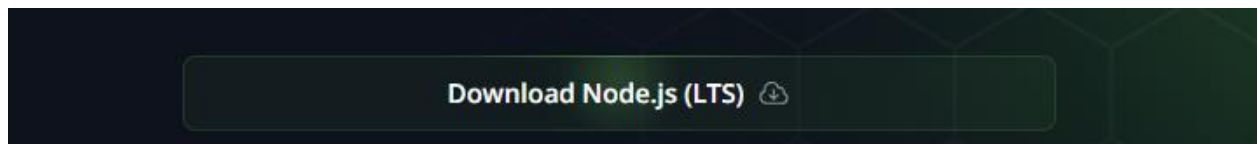
Node.js adalah sebuah platform yang dibuat khusus untuk membantu mengembangkan aplikasi berbasis web. Namun, Node.js bukanlah bahasa pemrograman baru, melainkan sebuah *runtime* atau *interpreter* untuk menjalankan bahasa pemrograman JavaScript untuk pengembangan back-end.

Node.js dibangun dengan mesin JavaScript V8 Google sehingga memiliki performa yang tinggi. Node.js juga memiliki library sendiri. Dengan begitu, pengguna tidak perlu menggunakan server web NGINX atau Apache.

TUTORIAL INSTALLASI NODEJS

Langkah 1 – Buka website Nodejs

<https://nodejs.org/> kemudian untuk windows dapat melakukan instalasi nodejs dengan mendownload installer dengan format .msi

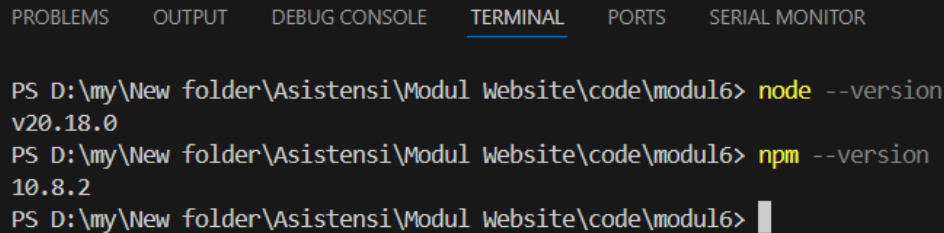


Langkah 2 – Periksa instalasi nodejs

Jalankan kedua perintah dibawah pada cmd atau terminal

```
node --version
```

```
npm --version
```



```

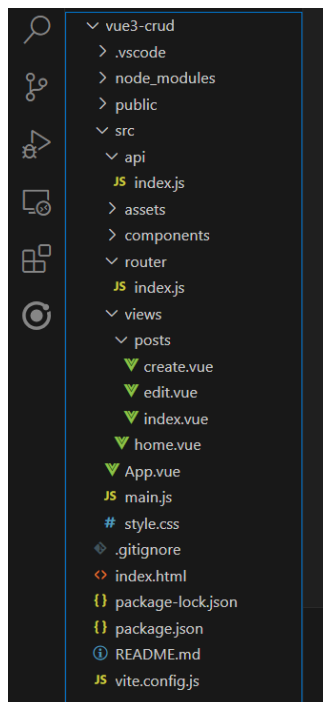
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  SERIAL MONITOR

PS D:\my\New folder\Asistensi\Modul Website\code\modul6> node --version
v20.18.0
PS D:\my\New folder\Asistensi\Modul Website\code\modul6> npm --version
10.8.2
PS D:\my\New folder\Asistensi\Modul Website\code\modul6> 

```

Jika output sudah seperti contoh diatas maka proses instalasi nodejs dan npm sudah berhasil.

TUTORIAL VUE.JS 3 DENGAN VITE



Disamping ini merupakan susunan directory dan file akhir pada modul codelab kali ini, pastikan untuk susunan sudah sesuai dengan contoh agar dapat berjalan dengan baik.

Langkah 1 – Instalasi template

Buatlah sebuah folder untuk project kali ini, kemudian buka folder tersebut dengan vscode, dan jalankan perintah dibawah pada terminal.

```
npm create vite@4.2.0 codelab -- --template vue
```

Jika berhasil, maka akan terbuat sebuah folder dengan nama codelab. Kemudian buka folder diatas dengan command dibawah ini.

```
cd codelab
```

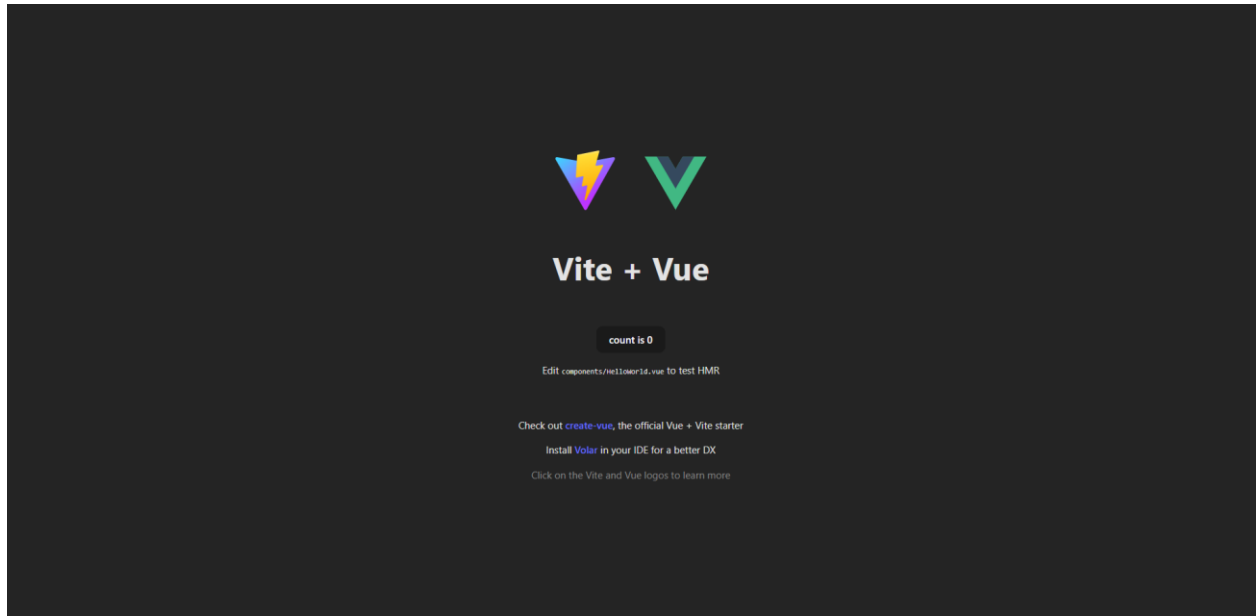
Setelah masuk kedalam directory codelab, kemudian jalankan command dibawah ini

```
npm install
```

Untuk menjalankan vue, sekarang jalankan command dibawah ini

```
npm run dev
```

Jika berhasil maka akan membuka localhost dengan port 5173 seperti dibawah ini



Langkah 2 – Membuat router

Untuk dapat membuat router pada vue, diperlukan installasi library router. Untuk mendapatkan library tersebut, dapat menjalankan command dibawah ini

```
npm install vue-router@4.1.6
```

Jika berhasil maka akan muncul tampilan dibawah ini

```
PS D:\my\New folder\Asistensi\Modul Website\code\modul6\vue3-crud> npm install vue-router@4.1.6
added 2 packages, and audited 31 packages in 1s

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

found 0 vulnerabilities
```

Langkah 3 – Integrasi Bootstrap pada vue

Disini kita akan menggunakan bootstrap pada project vue kali ini. Untuk mempermudah komponen, kita akan menggunakan bootstrap untuk membantu pengerjaan pada sisi komponen css. Silakan buka file **index.html** kemudian paste kan kode dibawah ini.

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <link rel="icon" href="/favicon.ico">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Vue 3 + Laravel 10 - Modul 6</title>
```

```

        <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/css/bootstrap.min.css" rel="stylesheet">
    </style>
    body { background-color: lightgray; font-family: 'Quicksand', sans-serif }
  </style>
</head>
<body>
  <div id="app"></div>
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/js/bootstrap.bundle.min.js"></script>
  <script type="module" src="/src/main.js"></script>
</body>
</html>

```

Langkah 4 – Membuat views

Setelah sebelumnya membuat router, maka selanjutnya adalah membuat views. Untuk dapat membuat views, buat folder baru dengan nama **views** di dalam folder **src**, lalu buat file dengan nama **home.vue** pada folder **src**. Isikan kode dibawah ini pada file **home.vue**. Sehingga path nya menjadi **/src/views/home.vue**.

```

<script setup>

</script>

<template>
  <div class="p-5 mb-4 bg-light rounded-3">
    <div class="container-fluid py-5">
      <h1 class="display-5 fw-bold">VUE 3 (VITE) + LARAVEL</h1>
      <p class="col-md-8 fs-4">Frontend dengan vue – codelab modul 6</p>
    </div>
  </div>
</template>

```

Langkah 5 – Membuat view Posts Index

Selanjutnya, buat folder baru dengan nama **posts** didalam folder **/src/views**, kemudian di dalam folder **posts** tersebut buat file dengan nama **index.vue**, sehingga menjadi **/src/views/posts/index.vue**.

```

<script setup>

</script>

<template>
  <div class="container mt-5 mb-5">
    <div class="row">
      <div class="col-md-12">

```

```

        <div class="card border-0 rounded shadow">
          <div class="card-body">
            HALAMAN POSTS INDEX
          </div>
        </div>
      </div>
    </div>
  </div>
</template>

```

Dengan begitu, untuk halaman index Post akan menampilkan kode html diatas.

Langkah 6 – Membuat view Post Create

Selanjutnya didalam folder **/src/views/posts**, buat file dengan nama **create.vue**, kemudian isikan kode dibawah ini. Sehingga pathnya menjadi **/src/views/posts/create.vue**.

```

<script setup>

</script>

<template>
  <div class="container mt-5 mb-5">
    <div class="row">
      <div class="col-md-12">
        <div class="card border-0 rounded shadow">
          <div class="card-body">
            HALAMAN POST CREATE
          </div>
        </div>
      </div>
    </div>
  </div>
</template>

```

Dengan begitu, pada halaman create maka akan menampilkan kode html diatas.

Langkah 7 – Membuat view Post Edit

Selanjutnya didalam folder **/src/views/posts**, buat file dengan nama **edit.vue**, kemudian isikan kode dibawah ini. Sehingga pathnya menjadi **/src/views/posts/edit.vue**.

```

<script setup>

</script>

<template>
  <div class="container mt-5 mb-5">

```

```

<div class="row">
  <div class="col-md-12">
    <div class="card border-0 rounded shadow">
      <div class="card-body">
        HALAMAN POST EDIT
      </div>
    </div>
  </div>
</div>
</div>
</template>

```

Langkah 8 – Membuat konfigurasi router

Setelah views berhasil dibuat, maka langkah selanjutnya adalah membuat konfigurasi router. Silakan buat folder baru dengan nama **/src/router** dengan file **index.js** didalamnya. Kemudian, masukkan kode dibawah ini:

```

//import vue router
import { createRouter, createWebHistory } from 'vue-router'

//define a routes
const routes = [
  {
    path: '/',
    name: 'home',
    component: () => import( /* webpackChunkName: "home" */
'../views/home.vue')
  },
  {
    path: '/posts',
    name: 'posts.index',
    component: () => import( /* webpackChunkName: "index" */
'../views/posts/index.vue')
  },
  {
    path: '/create',
    name: 'posts.create',
    component: () => import( /* webpackChunkName: "create" */
'../views/posts/create.vue')
  },
  {
    path: '/edit/:id',
    name: 'posts.edit',
    component: () => import( /* webpackChunkName: "edit" */
'../views/posts/edit.vue')
  }
]

```



```

    }
  ]

  //create router
  const router = createRouter({
    history: createWebHistory(),
    routes // <-- routes,
  })

  export default router

```

Dari penambahan kode diatas, dilakukan import **createRouter** dan **createWebHistory** dari Vue Router.

```

//import vue router
import { createRouter, createWebHistory } from 'vue-router'

```

Setelahnya, buat deklarasi router seperti contoh diatas pada kode dibawah ini

```

//define a routes
const routes = [
  {
    path: '/',
    name: 'home',
    component: () => import( /* webpackChunkName: "home" */ '../views/home.vue')
  },
]

```

Pada pembuatan router sederhana, kurang lebih terdapat 3 object, yaitu:

1. Path: digunakan untuk Alamat / url dari route.
2. Name: digunakan untuk penamaan route, hal ini dapat mempermudah ketika pemanggilan route kedepannya.
3. Component: merupakan lokasi file view yang dipanggil saat route diakses atau dijalankan.

Langkah 9 – Register Route

Agar route bisa diakses dari seluruh aplikasi vue, maka diperlukan register route yang mana melakukan konfigurasi secara global. Untuk melakukan hal tersebut, perlu meletakkan file **main.js**.

Silakan buka file **/src/main.js** terlebih dahulu, kemudian ganti kode sebelumnya menjadi dibawah ini:

```

//import createApp from Vue
import { createApp } from 'vue';

//import component App
import App from './App.vue';

//import config router
import router from './router'

```

```
//create App Vue
const app = createApp(App);

//gunakan "router" di Vue dengan plugin "use"
app.use(router);

app.mount('#app');
```

Pada kode diatas, dilakukan import konfigurasi router nya terlebih dahulu. Setelah itu, letakkan **createApp** dari vue di dalam variable yang bernama **app**. Ini bertujuan agar dapat lebih mudah dalam menggunakannya. Untuk melakukan register, dapat menggunakan keyword **use**.

Langkah 10 – Konfigurasi router View

Agar view dapat dirender pada halaman browser, maka perlu memanggil komponen dari **Vue Router** yang bernama `<router-view></router-view>`. Silakan buka file `/src/App.vue`, kemudian ganti kode menjadi dibawah ini:

```
<template>
  <div>
    <nav class="navbar navbar-expand-lg bg-dark" data-bs-theme="dark">
      <div class="container">
        <router-link :to="{ name: 'home' }" class="navbar-brand">HOME</router-link>
        <button
          class="navbar-toggler"
          type="button"
          data-bs-toggle="collapse"
          data-bs-target="#navbarSupportedContent"
          aria-controls="navbarSupportedContent"
          aria-expanded="false"
          aria-label="Toggle navigation"
        >
          <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarSupportedContent">
          <ul class="navbar-nav me-auto mb-2 mb-lg-0">
            <li class="nav-item">
              <router-link :to="{ name: 'posts.index' }" class="nav-link active"
                aria-current="page">POSTS</router-link>
            </li>
          </ul>
          <ul class="navbar-nav ms-auto mb-2 mb-lg-0" role="search">
```

```

        <a
          href="#"
          target="_blank"
          class="btn btn-success"
        >LOGIN</a>
      >
    </ul>
  </div>
</div>
</nav>

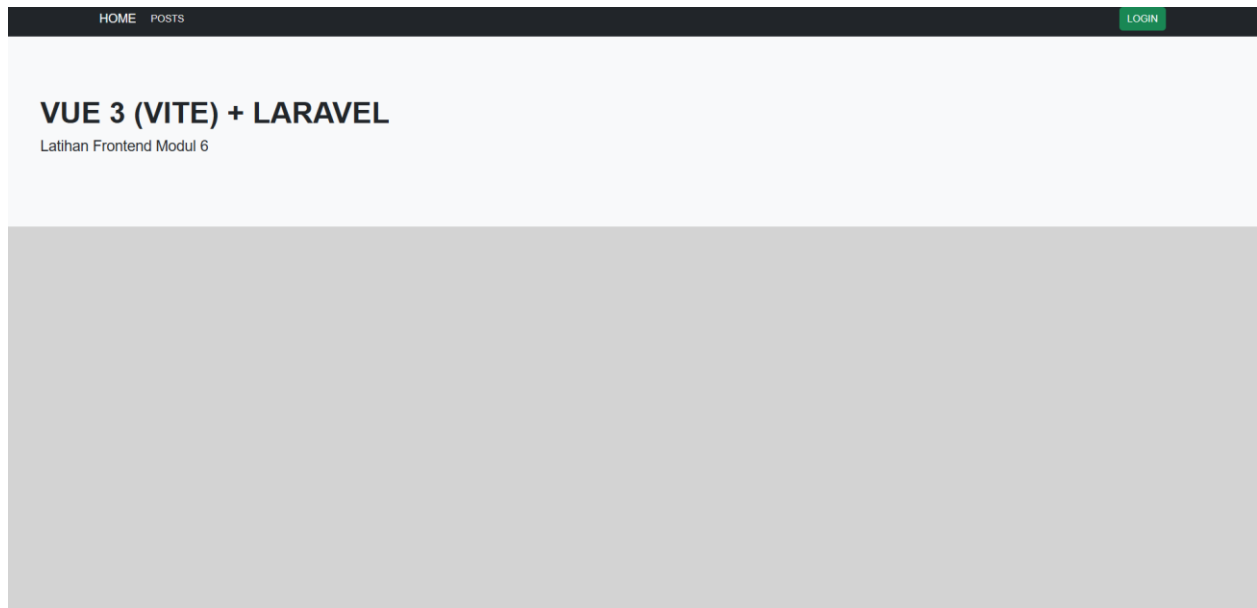
<!-- render router view -->
<router-view></router-view>

</div>
</template>

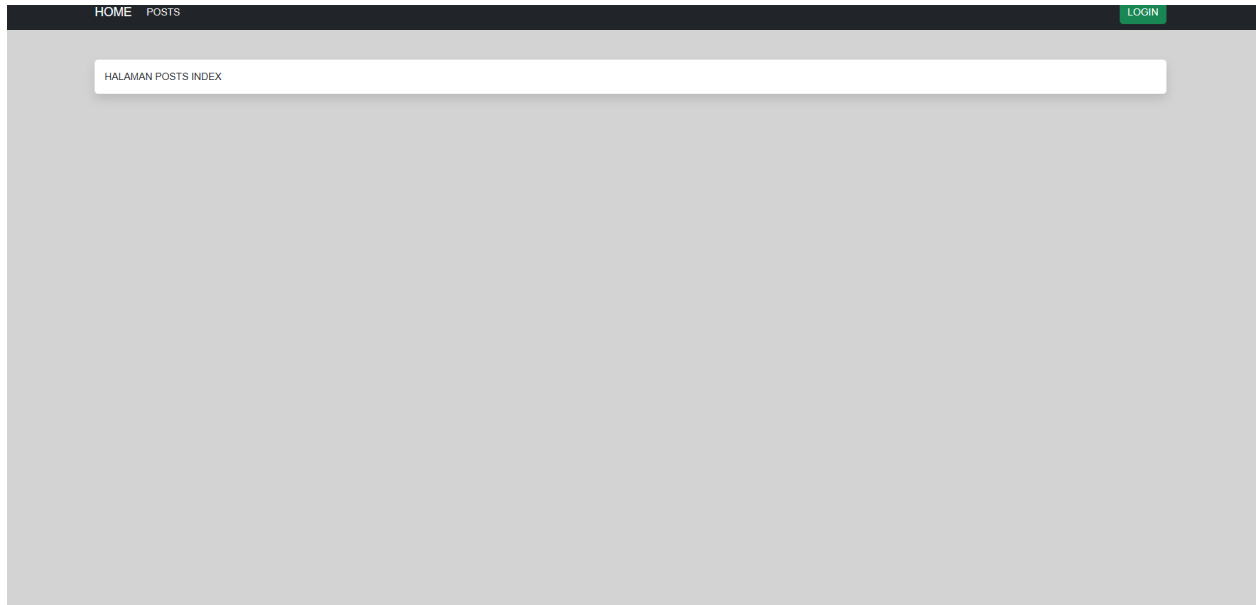
```

Langkah 11 – Uji Coba Project

Sekarang buka Kembali halaman: <http://localhost:5173/> dan periksa apakah tampilan sudah berubah sesuai dengan dibawah ini



Dan jika menekan menu **Posts** maka akan diarahkan ke halaman lain dengan path url <http://localhost:5173/posts> dan menampilkan halaman dibawah ini



Langkah 12 – Instalasi Axios

Axios digunakan untuk melakukan berbagai macam proses HTTP seperti **GET, POST, PATCH, dan DELETE**. Silakan jalankan command dibawah ini pada terminal didalam project codelab modul 6 ini.

```
npm install axios@1.3.4
```

Langkah 13 – Konfigurasi Endpoint API

Agar tidak perlu menuliskan endpoint api berulang ulang, maka dapat menuliskan konfigurasi dari endpoint API. Silakan buat folder bernama **api** di dalam folder **/src/** dan didalam folder api, buat file **index.js** dengan kode dibawah ini:

```
//import axios
import axios from 'axios';

const Api = axios.create({
  //samakan dengan url api Laravel modul 5
  baseURL: 'http://localhost:8000'
})

export default Api
```

Dari kode diatas, dilakukan konfigurasi untuk baseURL sesuai dengan url backend Laravel pada modul 5 minggu lalu.

Langkah 14 – Menampilkan data Rest API

Setelah axios dan endpoint API berhasil terinstall, maka dapat melakukan proses menampilkan data dari **Rest API** di dalam project vue. Silakan buka **/src/views/posts/index.vue**, kemudian ubah semua kodenya menjadi dibawah ini:

<https://pastebin.com/SxJsr2SL>

Dari kode diatas, ditambahkan kode import **ref** dan hook **onMounted** dari **Vue**.

```
//import ref and onMounted
import { ref, onMounted } from 'vue';
```

Setelah itu, masukkan konfigurasi **api**

```
//import api
import api from '../api';
```

Lalu lakukan fetch data ke backend menggunakan endpoint **/api/posts** dan method yang digunakan adalah **GET**.

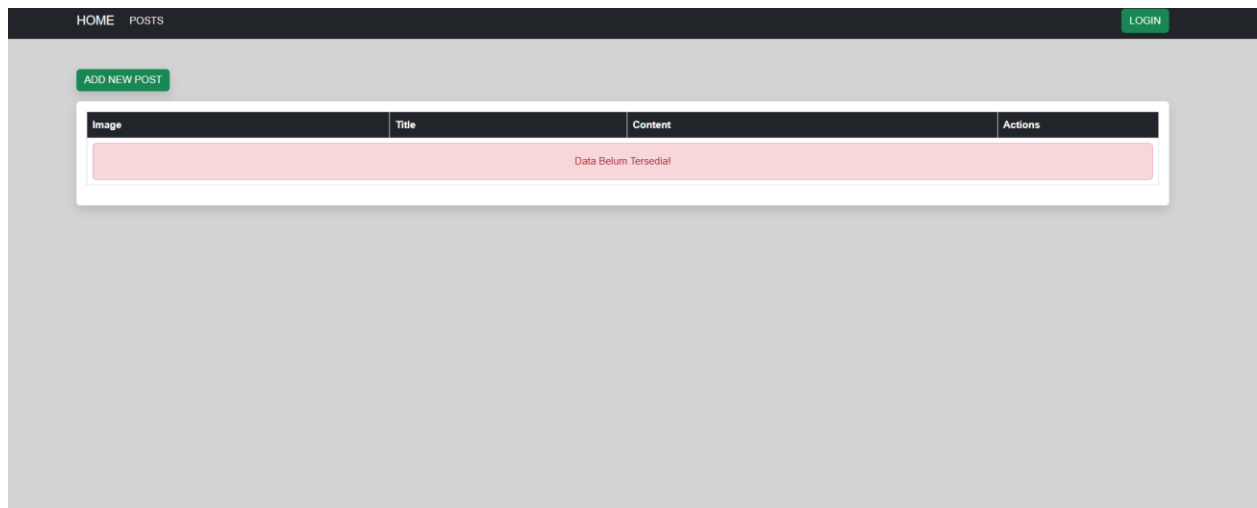
```
//fetch data
await api.get('/api/posts')
```

Jika berhasil melakukan fetch data, selanjutnya masukkan response data dari API kedalam state **posts**.

```
//set response data to state "posts"
posts.value = response.data.data.data
```

Langkah 15 – Uji Coba Menampilkan data Posts

Silakan buka kembali halaman web <http://localhost:5173/posts>, jika tidak terdapat error maka seharusnya akan tampil halaman seperti dibawah ini.



Langkah 16 – Inser dan Upload Data

Silakan buka kembali file `/src/views/posts/create.vue`, kemudian ubah semua kodenya menjadi dibawah ini:

<https://pastebin.com/ZeW8tLGG>

Dari perubahan kode diatas, ditambahkan import **ref** dari vue.

```
//import ref
import { ref } from "vue";
```

Kemudian import **useRouter** dari **Vue Router**. Nantinya, akan digunakan untuk melakukan navigasi ke route tertentu.

```
//import router
import { useRouter } from 'vue-router';
```

Karena akan berhubungan dengan **Rest API**, maka diperlukan konfigurasi **api** yang sudah dibuat sebelumnya.

```
//import api
import api from "../../api";
```

Langkah 17 – Uji Coba POST API

Tambahkan data pada field yang tersedia di <http://localhost:5173/create> kemudian jika berhasil maka data akan masuk kedalam database dan tersimpan.

Langkah 18 – Edit dan Update Data Post

Silakan buka kembali file `/src/views/posts/edit.vue`, kemudian ubah semua kodenya menjadi dibawah ini:

<https://pastebin.com/kD8E2yna>

Dari perubahan kode diatas, ditambahkan inisialisasi **FormData**

```
//init formData
let formData = new FormData();
```

Setelah itu, masukkan data ke dalam **FormData** menggunakan **append**

```
//assign state value to formData
formData.append("image", image.value);
formData.append("title", title.value);
formData.append("content", content.value);
formData.append("_method", "PATCH");
```

Jika datanya berhasil diupdate, maka akan redirect atau mengarahkan ke dalam route `/posts`. Jika terjadi error pada proses update data, maka akan menampilkan error response didalam state **errors**.

Langkah 18 – Uji Coba Update Post

Silakan klik button **EDIT** di salah satu data yang dimiliki, jika berhasil maka akan diarahkan ke halaman lain pada `/edit/{id}`. Kemudian, silakan update isinya sesuai dengan keinginan dan jika sudah klik button **Update**. Maka data baru akan tersimpan kedalam database dan dapat kembali dilihat pada halaman <http://localhost:5173/posts>.

Langkah 19 – Menambahkan Method Delete Post

Silakan buka kembali file `/src/views/posts/index.vue`, kemudian ubah semua kode menjadi dibawah ini:

<https://pastebin.com/BCTfU04y>

Pada kode diatas, ditambahkan method baru dimana akan berjalan ketika menekan tombol **DELETE**.

```
<td class="text-center">
  <router-link :to="{ name: 'posts.edit', params:{id: post.id} }" class="btn btn-sm btn-primary
    rounded-sm shadow border-0 me-2">EDIT</router-link>
  <button @click.prevent="deletePost(post.id)" class="btn btn-sm btn-danger rounded-sm shadow
    border-0">DELETE</button>
</td>
```

Karena pada button tersebut, diberikan event `@click` yang mana akan mengarah ke method `deletePost` dengan mengirimkan parameter `post id`.

```
//method deletePost
const deletePost = async (id) => {
  //delete post with API
  await api.delete(`/api/posts/${id}`)
  .then(() => {
    //call method "fetchDataPosts"
    fetchDataPosts();
  })
};
```

Dalam method diatas, dilakukan proses delete data menggunakan **Rest API** ke endpoint `/api/posts/:id` dengan method **DELETE**.

```
//delete post with API
await api.delete(`/api/posts/${id}`)
.then(() => {
```

Langkah 20 – Uji Coba API Delete

Silakan buka kembali halaman <http://localhost:5173/posts> kemudian klik button **DELETE**, jika berhasil maka data akan terhapus dari backend dan database

CODELAB

CODELAB 1

Coba step instalasi Nodejs dan Vue sesuai dengan contoh diatas.

Note: Segala jenis pertanyaan bisa disampaikan pada pekan materi. Tidak ada pertanyaan disaat demo praktikum kecuali untuk teknis penilaian.

TUGAS

TUGAS 1

Dari tema yang sudah ditulis pada spreadsheet sebelumnya, buatlah website dengan fitur CRUD menggunakan tampilan dan implementasikan pada **kedua table** yang sudah dibuat pada modul 5 sebelumnya.

(tidak ada penggunaan CRUD pada postman lagi)

Contoh:

localhost:/product/create

localhost:/category/edit

Note: Wajib menggunakan vue sesuai dengan tutorial diatas, dan diperbolehkan jika ingin tidak menggunakan bootstrap pada tugas kali ini.

KRITERIA & DETAIL PENILAIAN

Kriteria Penilaian	Persentase Penilaian
Tugas 1	40%
Pemahaman	60%