

Langkah-Langkah Membuat Proyek Vue 3 dari Awal Sampai Akhir

Untuk memenuhi permintaan Anda, saya akan menjelaskan langkah-langkah dari pembuatan proyek hingga menyelesaikan keempat fitur dengan Vue 3 menggunakan Vite.

1. Inisialisasi Proyek Vue 3

Jalankan perintah berikut di terminal untuk membuat proyek Vue 3 menggunakan Vite:

```
npm create vite@latest vue-api-test
cd vue-api-test
npm install
npm run dev
```

Proyek akan berjalan di localhost:5173 (default Vite).

2. Struktur Folder Proyek

Buat folder `components` di dalam `src` untuk menyimpan komponen-komponen yang diperlukan:

```
src/
├── components/
│   ├── UserList.vue
│   ├── UserForm.vue
│   ├── PostPagination.vue
│   └── LoginForm.vue
├── App.vue
└── main.ts
```

3. Komponen 1: Mengambil Data API (UserList.vue)

```
<template>
  <div>
    <p v-if="loading">Memuat data...</p>
    <p v-if="error">{{ error }}</p>
    <ul v-if="!loading && !error">
      <li v-for="user in users" :key="user.id">{{ user.name }}</li>
    </ul>
  </div>
</template>
```

```

<script lang="ts" setup>
import { ref, onMounted } from 'vue';
import axios from 'axios';

const users = ref([]);
const loading = ref(true);
const error = ref("");

onMounted(async () => {
  try {
    const response = await axios.get('https://jsonplaceholder.typicode.com/users');
    users.value = response.data;
  } catch (err) {
    error.value = 'Gagal mengambil data.';
  } finally {
    loading.value = false;
  }
});
</script>

```

4. Komponen 2: Form Input dan Pengiriman Data (UserForm.vue)

```

<template>
  <form @submit.prevent="submitForm">
    <label>Nama:</label>
    <input v-model="form.name" required />
    <label>Email:</label>
    <input v-model="form.email" type="email" required />
    <label>Pesan:</label>
    <textarea v-model="form.message" required></textarea>
    <button type="submit">Kirim</button>
    <p v-if="notification">{{ notification }}</p>
  </form>
</template>

<script lang="ts" setup>
import { ref } from 'vue';
import axios from 'axios';

```

```

const form = ref({
  name: "",
  email: "",
  message: ""
});
const notification = ref("");

const submitForm = async () => {
  try {
    await axios.post('https://jsonplaceholder.typicode.com/posts', form.value);
    notification.value = 'Pengiriman berhasil!';
  } catch {
    notification.value = 'Pengiriman gagal!';
  }
};
</script>

```

5. Komponen 3: Paginasi Data (PostPagination.vue)

```

<template>
  <div>
    <ul>
      <li v-for="post in paginatedPosts" :key="post.id">{{ post.title }}</li>
    </ul>
    <button @click="prevPage" :disabled="page === 1">Previous</button>
    <button @click="nextPage">Next</button>
  </div>
</template>

<script lang="ts" setup>
import { ref, computed, onMounted } from 'vue';
import axios from 'axios';

const posts = ref([]);
const page = ref(1);
const postsPerPage = 5;

```

```

const paginatedPosts = computed(() => {
  const start = (page.value - 1) * postsPerPage;
  return posts.value.slice(start, start + postsPerPage);
});

const fetchPosts = async () => {
  const response = await axios.get('https://jsonplaceholder.typicode.com/posts');
  posts.value = response.data;
};

const nextPage = () => page.value++;
const prevPage = () => page.value--;

onMounted(fetchPosts);
</script>

```

6. Komponen 4: Login dengan Token (LoginForm.vue)

```

<template>
  <form @submit.prevent="login">
    <label>Username:</label>
    <input v-model="username" required />
    <label>Password:</label>
    <input v-model="password" type="password" required />
    <button type="submit">Login</button>
    <p v-if="message">{{ message }}</p>
  </form>
</template>

<script lang="ts" setup>
import { ref } from 'vue';
import axios from 'axios';

const username = ref("");
const password = ref("");
const message = ref("");

const login = async () => {

```

```
try {
  const response = await axios.post('https://fictitious-api.com/login', {
    username: username.value,
    password: password.value
  });
  localStorage.setItem('token', response.data.token);
  message.value = 'Login Berhasil';
} catch {
  message.value = 'Login Gagal';
}
};
</script>
```

7. App.vue

Gabungkan semua komponen di App.vue.

```
<template>
  <div>
    <h1>Vue 3 Test App</h1>
    <UserList />
    <UserForm />
    <PostPagination />
    <LoginForm />
  </div>
</template>

<script lang="ts" setup>
import UserList from './components/UserList.vue';
import UserForm from './components/UserForm.vue';
import PostPagination from './components/PostPagination.vue';
import LoginForm from './components/LoginForm.vue';
</script>
```

8. Menjalankan Proyek

Jalankan proyek dengan perintah:

```
npm run dev
```

Akses di browser menggunakan <http://localhost:5173>.

Semua spesifikasi soal sudah dipenuhi dengan masing-masing komponen dan fungsionalitas yang sesuai. Anda bisa mengunggah proyek ini ke GitHub dan mengirimkan tautannya ke email HRD sesuai instruksi.