

Introduction to Vue.js

Disusun oleh :
M. Ferza Erlangga
233510555



Instalasi Node.Js

Vue.Js



Instalasi Node.Js



Node.Js diperlukan agar Javascript bisa dijalankan diluar browser

- 1. Pergi ke link yang tertera**
- 2. Download versi terbaru Node.Js**
- 3. Lalu Install Node.Js kedalam program files**

Link Download : **<https://nodejs.org/en/download>**



Membuat Vue Project Dengan Vite

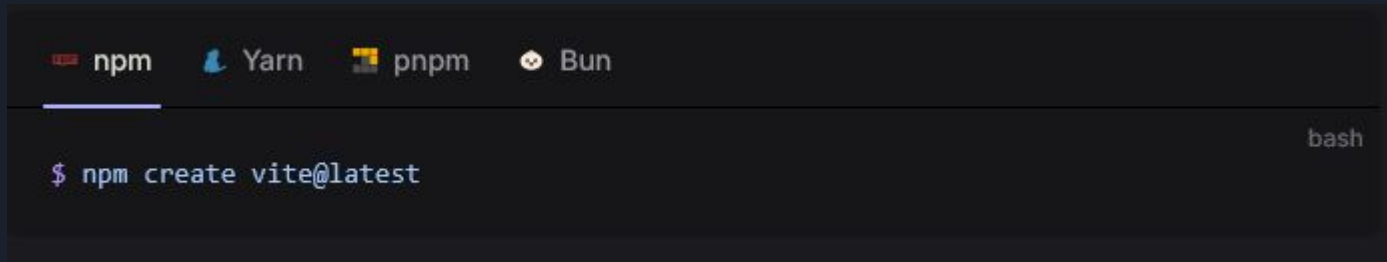


Vue.js



Membuat Vue Project Dengan Vite

Dalam Text Editor/Terminal anda, Jalankan Program Ini, lalu ikuti promptnya.

A terminal window with a dark background. At the top, there are four package manager icons: npm (red), Yarn (blue), pnpm (yellow), and Bun (purple). Below them, a horizontal line separates the header from the command area. The command '\$ npm create vite@latest' is entered in a light blue font. The word 'bash' is visible in the bottom right corner of the terminal window.

```
npm Yarn pnpm Bun  
$ npm create vite@latest
```

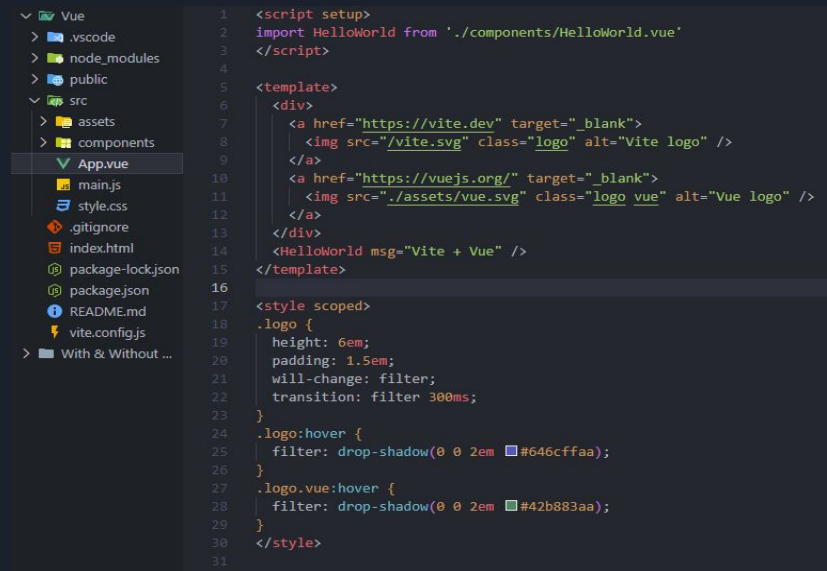
Gunakan framework Vue dan variant Javascript dalam promptnya.

Lalu akan muncul 3 perintah yang harus dimasukkan ke terminal, jalankan perintah tersebut

Setelah semua perintah sudah dijalankan, akan terdapat localhost yang bisa kita akses.

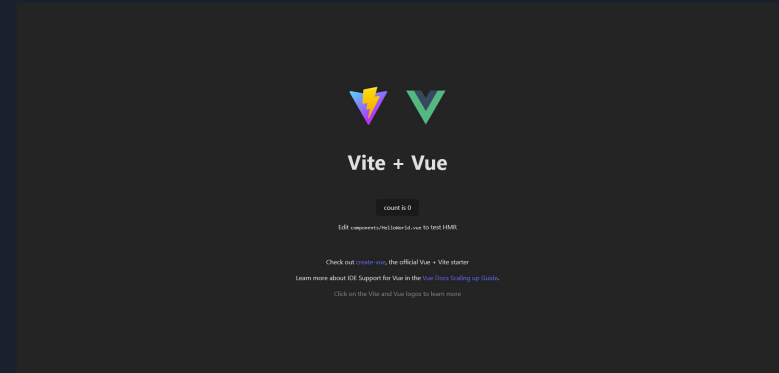
Pemograman Berbasis Komponen

Disini akan terbuat file yang kita install dari prompt sebelumnya, di bagian App.vue adalah tempat dimana kita menjalankan program kedalam localhost yang kita miliki



```
1 <script setup>
2 import HelloWorld from './components/HelloWorld.vue'
3 </script>
4
5 <template>
6   <div>
7     <a href="https://vite.dev" target="_blank">
8       
9     </a>
10    <a href="https://vuejs.org/" target="_blank">
11      
12    </a>
13  </div>
14  <HelloWorld msg="Vite + Vue" />
15 </template>
16
17 <style scoped>
18 .logo {
19   height: 6em;
20   padding: 1.5em;
21   will-change: filter;
22   transition: filter 300ms;
23 }
24 .logo:hover {
25   filter: drop-shadow(0 0 2em #646cffff);
26 }
27 .logo.vue:hover {
28   filter: drop-shadow(0 0 2em #42b883aa);
29 }
30 </style>
31
```

Jika kita klik localhost yang berada di terminal, maka kita akan dipindahkan ke localhost tersebut dengan tampilan seperti ini



Membuat Variabel Reaktif Dengan “ref”

Vue.js



Membuat Variabel Reaktif Dengan “ref”

Dalam Vue.js, ref adalah sebuah fungsi yang digunakan untuk membuat variabel reaktif.

Variabel reaktif adalah variabel yang ketika nilainya berubah, akan secara otomatis memperbarui tampilan (UI) yang bergantung pada variabel tersebut.

Sintaks :

```
import { ref } from 'vue';  
  
const myValue = ref(initialValue);
```


Menggunakan Syntax Text Interpolation

Vue.js



Menggunakan Syntax Text Interpolation

Text interpolation adalah cara untuk menyisipkan (interpolate) atau memanggil nilai dari variabel reaktif atau hasil ekspresi JavaScript ke dalam teks yang ditampilkan di HTML.

Di Vue.js, text interpolation dilakukan menggunakan sintaks `{{ }}` (double curly braces).

Contoh :

```
const text = ref('Hello World');  
  
</script>  
  
<template>  
  <div>{{ text }}</div>  
</template>
```

Menggunakan Syntax Form Binding

Vue.js



Menggunakan Syntax Form Binding

Form Binding dalam Vue.js adalah cara untuk mengikat (bind) data antara elemen form (seperti input, checkbox, select, dll.) dengan variabel reaktif di Vue.

Dengan form binding, kita dapat:

1. Mengambil nilai input pengguna dan menyimpannya ke dalam variabel reaktif.
2. Memperbarui tampilan secara otomatis ketika nilai input berubah.
3. Mengontrol nilai input dari kode JavaScript.

Form binding di Vue.js biasanya dilakukan menggunakan direktif v-model.

Menggunakan Syntax Form Binding

v-model adalah direktif Vue.js yang digunakan untuk membuat two-way data binding antara elemen form dan variabel reaktif. Artinya:

- Jika nilai input berubah, variabel reaktif akan diperbarui.
- Jika variabel reaktif berubah, nilai input juga akan diperbarui.

```
const binding = ref('');  
</script>
```

Contoh :

```
<template>  
  <div>binding = {{ binding }}</div>  
  <select v-model="binding">  
    <option disabled value="">please select one</option>  
    <option>A</option>  
    <option>B</option>  
    <option>C</option>  
  </select>
```

Menggunakan Syntax Event Listener

Vue.js



Menggunakan Syntax Event Listener

Event Listener adalah mekanisme dalam pemrograman yang memungkinkan kita "mendengarkan" (listen) event atau kejadian tertentu yang terjadi pada elemen HTML (seperti klik, input, hover, submit, dll.) dan menjalankan fungsi tertentu sebagai respons terhadap event tersebut.

Di Vue.js, event listener dapat ditambahkan menggunakan direktif v-on atau sintaks singkatnya @.

```
<element v-on:event="methodName"></element>  
<element @event="methodName"></element>
```

Menggunakan Attribute Binding

Vue.js



Menggunakan Attribute Binding

Attribute Binding dalam Vue.js adalah cara untuk mengikat (bind) nilai atribut HTML (seperti src, href, class, style, dll.) ke variabel reaktif atau ekspresi JavaScript. Dengan attribute binding, kita dapat mengubah atribut elemen HTML secara dinamis berdasarkan data di Vue.

Di Vue.js, attribute binding dilakukan menggunakan direktif v-bind atau sintaks singkatnya :

```
<div v-bind:attribute="dataProperty"></div>  
<div :attribute="dataProperty"></div>
```

Menggunakan Conditional Rendering

Vue.js



Menggunakan Conditional Rendering

Conditional Rendering dalam Vue.js adalah teknik untuk menampilkan atau menyembunyikan elemen HTML berdasarkan kondisi tertentu.

Dengan conditional rendering, kita dapat mengontrol tampilan elemen di halaman web secara dinamis, tergantung pada nilai variabel reaktif atau hasil ekspresi JavaScript.

Di Vue.js, conditional rendering biasanya dilakukan menggunakan direktif v-if, v-else, v-else-if, atau v-show.

Contoh :

```
<template>
  <div>
    <p v-if="status === 'loading'">Loading...</p>
    <p v-else-if="status === 'success'">Data Loaded!</p>
    <p v-else>Failed to load data.</p>
  </div>
</template>
```

Pemograman Berbasis Komponen

SELESAI

Vue.Js

