

Adelia Neyoan

**Penjelasan singkat
tentang vue serta
fitur-fitur yang
terdapat pada vue.**



Penjelasan Tentang Vue JS

VueJS adalah sebuah framework JavaScript untuk membangun aplikasi web atau tampilan interface website agar terlihat lebih menarik. VueJS dapat digunakan dalam membangun sebuah aplikasi yang berbasis user interface.

Framework ini juga menawarkan berbagai fitur yang dapat membantu user dalam membangun aplikasi skalabel. Hal ini membuat VueJS populer di kalangan pengembang web yang mencari solusi yang mudah digunakan dan ringan untuk membangun aplikasi web yang berkualitas.



Variabel Reaktif

Variabel reaktif merujuk pada sebuah variabel yang dilacak oleh sistem reaktif Vue. Artinya ketika user merubah nilai variabel, maka secara otomatis akan memperbarui DOM (Document Object Model) tanpa harus menulis banyak kode untuk melakukan pembaruan.

Contoh:



```
<script setup>
import { ref } from 'vue'

// Membuat variabel reaktif
const message = ref('Hello Vue!')
```

Text Interpolation

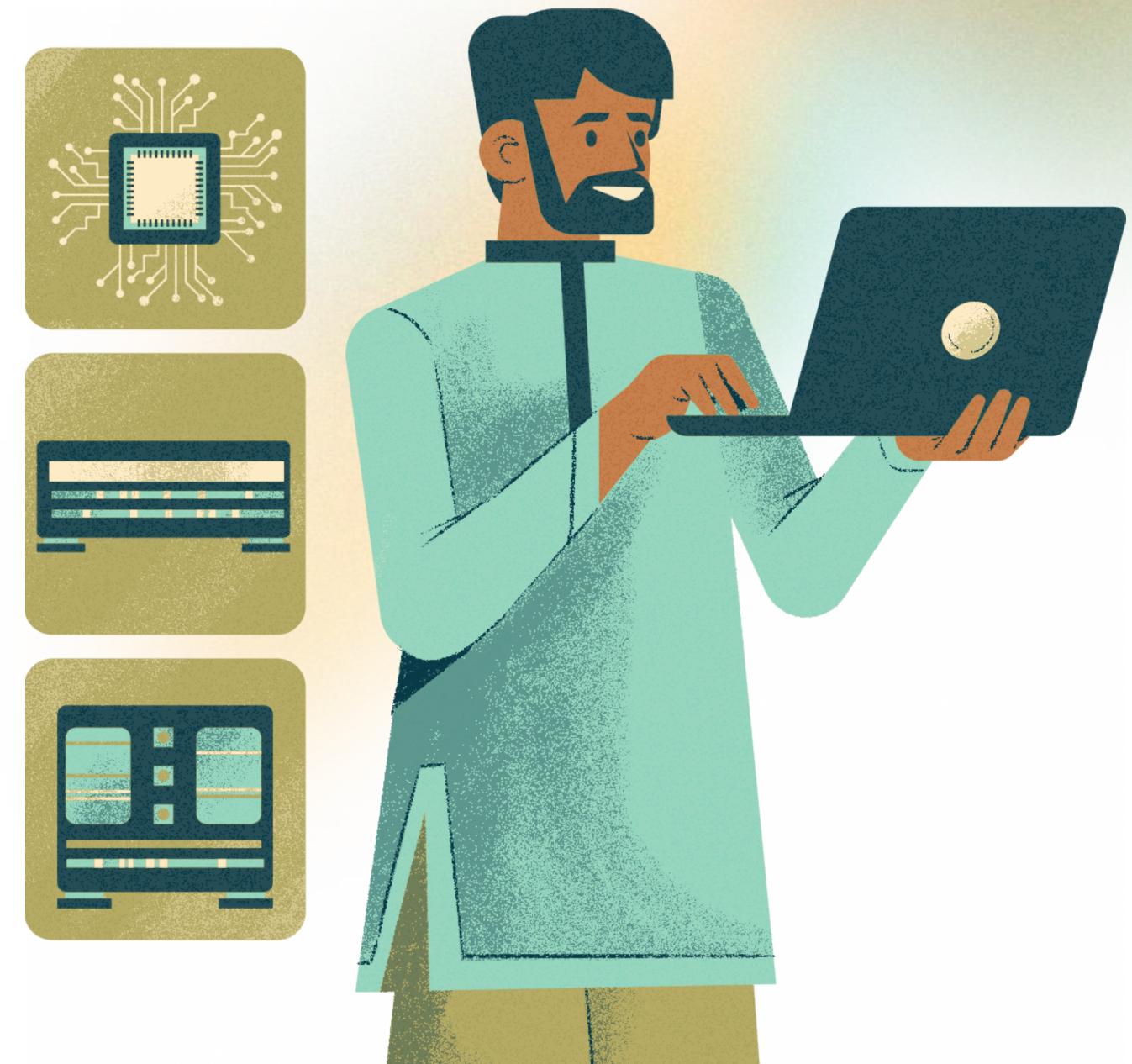
Text interpolation adalah cara untuk menyiipkan sebuah nilai ke dalam Javascript langsung ke dalam elemen HTML. Hal ini memungkinkan binding data secara otomatis antara data/model dengan tampilan.

Contoh:

```
<script setup>
  import { ref } from 'vue'

  const greeting = ref('Welcome to Vue!')
  const name = ref('John Doe')
</script>

<template>
  <h1>{{ greeting }} {{ name }}</h1>
</template>
```





Form Binding

Form Binding adalah fitur yang memungkinkan untuk menghubungkan elemen dalam data secara dinamis. Dengan menggunakan form binding, nilai akan otomatis terikat ke data model dalam vue, sehingga akan dapat dilihat setiap perubahannya.

```
<script setup>
import { ref } from 'vue'

// variabel reaktif untuk input form
const inputValue = ref('')

</script>

<template>
  <input v-model="inputValue" placeholder="Type something here" /> <!-- Menggunakan form binding -->
  <p>You typed: {{ inputValue }}</p>
</template>
```

Event Listener

Event listener digunakan untuk menangani interaksi dengan project seperti penggunaan klik, ketikan atau tindakan lainnya pada elemen HTML. Vue.js menyediakan cara yang mudah dan terstruktur dalam menambahkan event listener pada elemen DOM.

Contoh:



```
<script setup>
import { ref } from 'vue'
// Variabel reaktif untuk counter
const count = ref(0)
// Fungsi untuk increment counter
function increment() {
  count.value++
}
</script>
<template>
  <!-- Menggunakan v-on untuk mendengarkan event 'click' -->
  <button v-on:click="increment">Click me</button> <!-- Menggunakan event listener -->
  <p>Counter: {{ count }}</p>
</template>
```

Attribut Binding

Atribut binding memungkinkan untuk mengikat nilai ke atribut HTML. Ini sangat berguna karena lebih interaktif dan responsif untuk mengubah atribut HTML.

Contoh:

```
<script setup>
import { ref } from 'vue'
// Mendeklarasikan variabel reaktif untuk class tombol
const buttonClass = ref('normal-btn')
// Fungsi untuk mengubah class tombol ketika tombol diklik
const toggleButtonClass = () => {
  // Mengubah kelas tombol antara 'normal-btn' dan 'active-btn'
  buttonClass.value = buttonClass.value === 'normal-btn' ? 'active-btn' : 'normal-btn'
}
</script>
<template>
  <!-- Menggunakan v-bind untuk mengubah class tombol secara dinamis -->
  <button v-bind:class="buttonClass" @click="toggleButtonClass">Click Me</button>
</template>
```

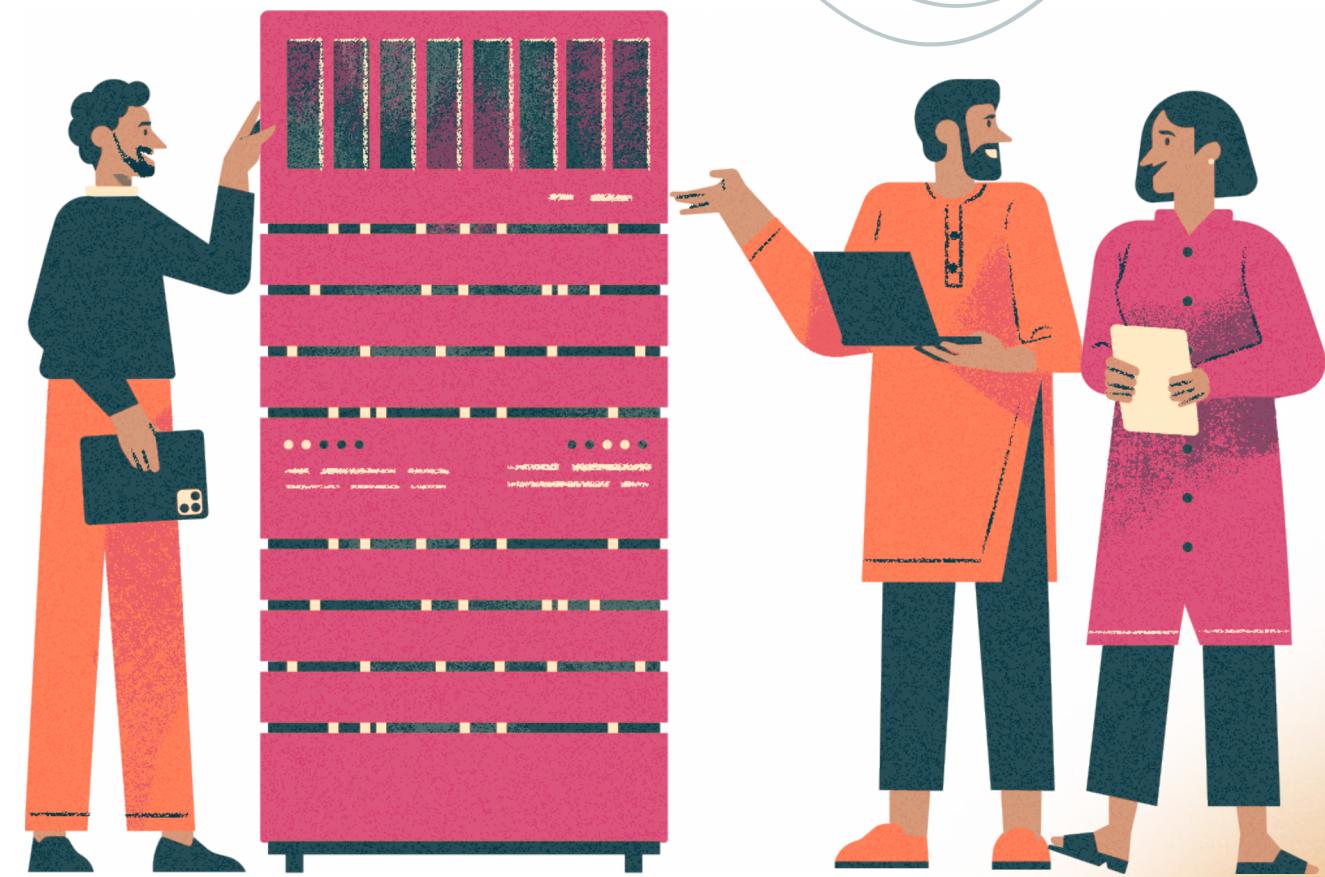


Conditional Rendering

Conditional rendering memungkinkan untuk menampilkan atau menyembunyikan sebuah elemen HTML berdasarkan kondisi tertentu. Fitur conditional rendering berguna dalam membuat antarmuka yang dinamis.

Contoh:

```
<script setup>
import { ref } from 'vue'
// Variabel reaktif untuk kondisi
const isVisible = ref(true)
</script>
<template>
  <button @click="isVisible = !isVisible">KLIK</button>
  <!-- Menggunakan conditional rendering -->
  <p v-if="isVisible">You can see me!</p>
  <p v-else>I'm hidden</p>
</template>
```





Thank You

Semoga penjelasan saya
mudah difahami