Listing 2.1

Contoh Penggunaan Vue Di HTML Menggunakan CDN.

$ npm init -y

$ npm install vue@next --savedev Contoh file HTML:

<div id=”app”></div>

<script src=”node\_modules/vue/dist/vue.global.js”></script>

// <1>

<script>

const { createApp } = Vue;

const App = { data() { return {

name: “Vue npm”,

};

},

template: `<h1>Hello {{ name }}!</h1>`,

};

createApp(App).mount(“#app”);

</script>

Listing 2.2

Contoh Penggunaan Vue Di HTML Menggunakan *npm*

Untuk menggunakan GUI, jalankan CLI dengan argumen **UI**:

$ vue ui

<div id=”app”>

Halo {{ name }}, detik berjalan: {{ counter }}

</div>

<script [src=”http://unpkg.com/vue@next”></script>](http://unpkg.com/vue%40next)

<script>

const { createApp } = Vue;

const App = { data() { return {

name: “Vue”,

counter: 0

};

},

mounted() { setInterval(() => { this.counter++

}, 1000)

}

};

createApp(App).mount(“#app”);

</script>

Listing 2.3

*Rendering* Deklaratif

Hasil jika ditampilkan di *browser*:

## Halo Vue, detik berjalan: 826

<div id=”app”>

<p>{{ pesan }}</p>

<button v-on:click=”reverseStr”>Balik String</button> // <1>

</div>

<script [src=”http://unpkg.com/vue@next”></script>](http://unpkg.com/vue%40next)

<script>

const { createApp } = Vue;

const App = { data() { return {

pesan: “Universitas Terbuka”,

};

},

methods: {

reverseStr() { // <2>

this.pesan = this.pesan.split(‘’).reverse().join(‘’)

}

}

};

createApp(App).mount(“#app”);

</script>

Listing 2.4

*Event Listener* Untuk Menangani Interaksi Dengan Pemakai Aplikasi

<div id=”app”>

<p v-if=’terlihat’>Universitas Terbuka<p>

<button v-on:click=”onOff”>{{ aksi }}</button>

</div>

// <1>

// <2>

<script [src=”http://unpkg.com/vue@next”></script>](http://unpkg.com/vue%40next)

<script>

const { createApp } = Vue;

const App = { data() { return {

terlihat: true,

aksi: ‘Sembunyikan’

};

},

methods: {

onOff() {

// <3>

if (this.aksi === ‘Sembunyikan’) { this.aksi = ‘Tampilkan’; this.terlihat = false;

} else {

this.aksi = ‘Sembunyikan’;

this.terlihat = true;

}

}

}

};

// <4>

createApp(App).mount(“#app”);

</script>

Listing 2.5

Direktif *if* Untuk *rendering* Kondisional

<div id=”app”>

<p>Universitas Terbuka<p>

<ol>

<li v-for=”mk in daftarMK” v-bind:key=”mk.text”>

{{ mk.text }}

// <1>

</li>

</ol>

</div>

<script [src=”http://unpkg.com/vue@next”></script>](http://unpkg.com/vue%40next)

<script>

const { createApp } = Vue;

const App = { data() { return {

daftarMK: [

{ text: ‘Algoritma Pemrograman’ },

{ text: ‘Pemrograman Perangkat Bergerak’ },

{ text: ‘Sistem Terdistribusi’ }

]

};

},

};

// <2>

createApp(App).mount(“#app”);

</script>

Listing 2.6

Perulangan Menggunakan Direktif *v-for*

<div id=”list-mk-app”>

<p>Universitas Terbuka<p>

<ol>

<daftar-mk

v-for=’mkDitawarkan in daftarMK’

v-bind:mk=’mkDitawarkan’

v-bind:key=’mkDitawarkan.id’>

</daftar-mk>

</ol>

</div>

// <1>

<script [src=”http://unpkg.com/vue@next”></script>](http://unpkg.com/vue%40next)

<script> const App = { data() {

return {

daftarMK: [

{ id: 0, text: ‘Algoritma Pemrograman’ },

{ id: 1, text: ‘Pemrograman Perangkat Bergerak’ },

{ id: 2, text: ‘Sistem Terdistribusi’ }

]

};

},

};

// <2>

const app = Vue.createApp(App)

app.component(‘daftar-mk’, { props: [‘mk’],

template: ‘<li>{{ mk.text }}</li>’

})

// <3>

// <4>

// <5>

app.mount(‘#list-mk-app’)

</script>

Listing 2.7 Komponen Dasar HTML

<div id=”list-mk-app”>

…

…

…

</div>

// <1>

<script src=”[http://unpkg.com/vue@next](http://unpkg.com/vue%40next)”></script>

// <2>

<script> const App = {

…

…

…

};

// <3>

const app = Vue.createApp(App)

// <4>

app.component(‘daftar-mk’, {

…

…

})

// <5>

app.mount(‘#list-mk-app’)

</script>

// <6>

Listing 2.8

Instan Aplikasi Dan Komponen Serta Keterkaitan Dengan Elemen DOM

…

…

…

const App = { data() { return {

name: “Vue”,

counter: 0

};

},

mounted() { setInterval(() => { this.counter++

}, 1000)

}

…

…

import { createApp } from ‘vue’ import App from ‘./App.vue’ import router from ‘./router’

createApp(App).use(router).mount(‘#app’)

Listing 2.9

*Routing* - *main.js*

import { createWebHistory, createRouter } from "vue-router"; import Awal from "@/components/Awal.vue";

import Tentang from "@/components/Tentang.vue";

const routes = [

{

path: "/",

name: "Awal", component: Awal,

},

{

path: "/tentang", name: "Tentang", component: Tentang,

},

];

const router = createRouter({ history: createWebHistory(), routes,

});

export default router;

Listing 2.10 Routing - router/index.js

<template>

<div id="nav">

<router-link to="/awal">Awal</router-link> |

<router-link to="/tentang">Tentang</router-link>

</div>

<router-view />

</template>

Listing 2.11

*Routing - App.vue*

<template>

<p>Halaman <b>Awal</b></p>

</template>

Listing 2.12

*Routing - components/Awal.vue*

<template>

<p>Halaman <b>Tentang</b></p>

</template>

Listing 2.13

*Routing - components/Tentang.vue*

<template>

<div id="nav">

<router-link :to="{ name: 'Awal' }">Awal</router-link> |

<router-link :to="{ name: 'Tentang' }">Tentang</router-link>

</div>

<router-view />

</template>

Listing 2.14

*Routing - named routes*

…

…

import Departemen from "@/components/Departemen.vue";

const routes = [

…

…

},

{

path: "/dept/:departemen", name: "Departemen", component: Departemen,

},

…

…

Listing 2.15

*Routing -* Segmen dinamis

<template>

<p>

Anda mengakses halaman departemen <b>{{ $route.params. departemen }}</b>

</p>

</template>

Listing 2.16

*Routing -* Cara Mengambil Parameter / Argumen

<template>

<span>Pesan: {{ pesan }}</span>

</template>

Listing 2.17

*Template*

<div id="app">

Halo {{ name }}

</div>

<script [src="http://unpkg.com/vue@next"></script>](http://unpkg.com/vue%40next)

<script>

const { createApp } = Vue;

const App = { data() { return {

name: "Vue"

};

},

};

createApp(App).mount("#app");

</script>

Listing 2.18

*Data Binding* Untuk *Property*

<div id="app">

<ol>

<li v-for="prg in proglang">

{{ prg }}

</li>

</ol>

</div>

<script [src="http://unpkg.com/vue@next"></script>](http://unpkg.com/vue%40next)

<script>

const { createApp } = Vue;

const App = { data() { return {

proglang: ['TypeScript','Rust','Go', 'Zig', 'JavaScript']

};

},

};

createApp(App).mount("#app");

</script>

<!-- hasil:

1. TypeScript
2. Rust
3. Go
4. Zig
5. JavaScript

-->

Listing 2.19

*Data Binding* Untuk *Array*

<div id="app">

<div v-if="angka > 80">Nilai A</div>

<div v-else-if="angka > 70">Nilai B</div>

<div v-else-if="angka > 40">Nilai C</div>

<div v-else-if="angka > 20">Nilai D</div>

<div v-else>Nilai E</div>

<div v-show="bagus">Bagus!</div>

</div>

<script [src="http://unpkg.com/vue@next"></script>](http://unpkg.com/vue%40next)

<script>

const { createApp } = Vue;

const App = { data() { return {

angka: 85, bagus: true

};

},

};

createApp(App).mount("#app");

</script>

Listing 2.20

*Data Binding* Kondisional

<style scoped>

.active {

background: cyan;

}

</style>

// <1>

<div id="app">

<div v-bind:class = "{ active:isactive }"><b>{{title}}</b></div>

<div v-bind:style = "{ color: activeColor, fontSize: fontSize + 'px' }">

<ol>

<li v-for="prg in proglang">

{{ prg }}

</li>

</ol>

</div>

</div>

<script [src="http://unpkg.com/vue@next"></script>](http://unpkg.com/vue%40next)

<script>

const { createApp } = Vue;

const App = { data() { return {

title: ‘Daftar Bahasa Pemrograman’, isactive: true,

activeColor: ‘blue’, fontSize :’22’,

proglang: [‘TypeScript’, ‘Rust’, ‘Go’, ‘Zig’, ‘JavaScript’]

};

},

};

createApp(App).mount(“#app”);

</script>

Listing 2.21

*Data Binding* Untuk *Class* Dan *Style*

<div id="app">

<p>Jumlah klik: {{ jumlahKlik }}</p>

<button @click="jumlahKlik += 1">Klik untuk menambah 1</button

</div>

<script [src="http://unpkg.com/vue@next"></script>](http://unpkg.com/vue%40next)

<script>

const { createApp } = Vue;

const App = { data() { return {

jumlahKlik: 0

};

},

};

createApp(App).mount("#app");

</script>

Listing 2.22 Membuat *Event Handler*

<div id="app">

<input v-model="angka1" placeholder="0" @change="hitung"> x

<input v-model="angka2" placeholder="0" @change="hitung">

<p>{{ angka1 }} x {{ angka2 }} = {{ hasil }}</p>

</div>

// <1>

<script [src="http://unpkg.com/vue@next"></script>](http://unpkg.com/vue%40next)

<script>

const { createApp } = Vue;

const App = { data() { return {

angka1: 0,

angka2: 0,

hasil: 0

};

},

methods: {

hitung() {

this.hasil = this.angka1 \* this.angka2

}

}

};

// <2>

createApp(App).mount("#app");

</script>

Listing 2.23

Fungsi Sebagai *Event Handler*

<div id="app">

<input v-model="angka1" placeholder="0" @change="hitung( angka1, angka2 )"> x

<input v-model="angka2" placeholder="0" @change="hitung( angka1, angka2 )">

<p>{{ angka1 }} x {{ angka2 }} = {{ hasil }}</p>

</div>

<script [src="http://unpkg.com/vue@next"></script>](http://unpkg.com/vue%40next)

<script>

const { createApp } = Vue;

const App = { data() { return {

angka1: 0,

angka2: 0,

hasil: 0

};

},

methods: {

hitung(a1, a2) { this.hasil = a1 \* a2;

}

}

};

createApp(App).mount("#app");

</script>

Listing 2.24 Fungsi dengan Argumen

<div id="app">

<p>Jumlah klik: {{ jumlahKlik }}</p>

<p>Faktorial dari {{ jumlahKlik }} adalah {{ faktorial }}</p>

<p>

<button @click="tambahKlik(), factorial(jumlahKlik)">Klik untuk menambah 1</ button>

</div>

<script [src="http://unpkg.com/vue@next"></script>](http://unpkg.com/vue%40next)

<script>

const { createApp } = Vue;

const App = { data() { return {

jumlahKlik: 0,

faktorial: 1

};

},

methods: { tambahKlik() { this.jumlahKlik += 1;

},

factorial(num) { var rval=1;

for (var i = 2; i <= num; i++)

rval = rval \* i; this.faktorial = rval;

}

}

};

createApp(App).mount("#app");

</script>

Listing 2.25

*Event Handler* Lebih Dari Satu

<div id="app">

<input v-model="isianTeks" placeholder="ketikkan sesuatu" />

<button @click="cekIsianTeks($event)">Kirim</button>

<p>

Anda mengisikan: {{ isianTeks }}

</p>

</div>

<script [src="http://unpkg.com/vue@next"></script>](http://unpkg.com/vue%40next)

<script>

const { createApp } = Vue;

const App = { data() { return {

isianTeks: ''

};

},

methods: { cekIsianTeks(event) {

if (this.isianTeks === '') {

event.preventDefault()

alert('Isikan teks terlebih dahulu ' + event)

} else {

alert('Terima kasih sudah mengisi teks ' + event)

}

}

}

};

createApp(App).mount("#app");

</script>

Listing 2.26 Mengakses DOM

<div id="app">

<button @click.once="tampilPesan">Klik sekali saja!</button

</div>

<script [src="http://unpkg.com/vue@next"></script>](http://unpkg.com/vue%40next)

<script>

const { createApp } = Vue;

const App = { data() { return {

jumlahKlik: 0

};

},

methods: { tampilPesan() {

alert("Hanya boleh klik sekali saja, setelah itu tidak bisa");

}

}

};

createApp(App).mount("#app");

</script>

Listing 2.27

*Modifier* dalam *Event Handler*

<template>

<div class="hello">

<h1>{{ msg }}</h1>

…

</div>

</template>

<script>

export default { name: 'HelloWorld',

…

…

}

</script>

<!-- Add "scoped" attribute to limit CSS to this component only -->

<style scoped> h3 {

…

…

}

</style>

Setelah didefinisikan, maka komponen tersebut bisa digunakan pada

template atau pada HTML setelah di-import dan didaftarkan:

<template>

<img alt="Vue logo" src="./assets/logo.png">

<HelloWorld msg="Welcome to Your Vue.js App"/>

</template>

<script>

import HelloWorld from './components/HelloWorld.vue'

export default { name: 'App', components: { HelloWorld

}

}

</script>

Listing 2.28

Contoh Komponen HelloWord.Vue

<template>

<p v-if="compShowHide">{{ strToHideShow }}</p>

<button @click="onOff">{{ aksi }}</button>

</template>

// <1>

<script>

export default { name: 'ButtonOnOff', data() {

return {

// <2>

aksi: 'Sembunyikan',

terlihat: true

}

},

props: { strToHideShow: String

},

computed: { compShowHide() {

if (this.terlihat === true) { return true;

} else { return false;

}

}

},

methods: { onOff() {

if (this.aksi === 'Sembunyikan') { this.aksi = 'Tampilkan'; this.terlihat = false;

} else {

this.aksi = 'Sembunyikan';

this.terlihat = true;

}

}

}

}

</script>

// <3>

// <4>

// <5>

<style scoped>

button {

color: #42b983;

}

</style>

// <6>

Listing 2.29 Komponen ButtonOnOff.Vue

<template>

<ButtonOnOff strToHideShow="Universitas Terbuka"/>

<ButtonOnOff strToHideShow="Indonesia"/>

</template>

// <1>

<script>

import ButtonOnOff from './components/ButtonOnOff.vue'

// <2>

export default { name: 'App', components: { ButtonOnOff

}

}

</script>

// <3>

<style> #app {

font-family: Avenir, Helvetica, Arial, sans-serif;

-webkit-font-smoothing: antialiased;

-moz-osx-font-smoothing: grayscale; text-align: center;

color: #2c3e50; margin-top: 60px;

}

</style>

Listing 2.30 App.Vue