

MODUL 1 – DASAR Vue.js

Ikuti seluruh instruksi yang diberikan lengkap dengan penjelasannya untuk memahami materi dasar-dasar Vue.js dengan baik

Environment Setup (CDN)

1. Buat file baru dengan nama lat1.html
2. Di dalam lat1.html, buat struktur dasar HTML
3. Di dalam tag <head>, tambahkan tag script untuk menambahkan library Vue.js, seperti berikut.
<script src="https://cdn.jsdelivr.net/npm/vue"></script>
4. Simpan.

Introduction

5. Di dalam lokasi yang sama dengan lat1.html, buat file baru dengan nama app.js
6. Tambahkan tag script untuk file app.js

```
<!DOCTYPE html>
<html lang="en">
<head>
  <script src="https://cdn.jsdelivr.net/npm/vue"></script>
  <title>Hello World</title>
</head>
<body>

  <script src="app.js"></script>
</body>
</html>
```

7. Di dalam body, tambahkan tag div dengan id="app".

```
<!DOCTYPE html>
<html lang="en">
<head>
  <script src="https://cdn.jsdelivr.net/npm/vue"></script>
  <title>Hello World</title>
</head>
<body>

  <div id="app">
    <h1></h1>
  </div>

  <script src="app.js"></script>
</body>
</html>
```

Elemen div#app akan dikontrol melalui instance dari Vue yang dideklarasikan di dalam app.js

8. Buka file app.js dan ketikkan baris kode berikut untuk membuat inisialisasi Vue object.

```
new Vue({ });
```

Vue object ini terdiri atas berbagai macam property yang sudah ditentukan oleh Vue. Inisialisasi diperlukan untuk mengontrol elemen `div#app`, misalnya untuk menampilkan teks tertentu.

9. Property pertama yang akan ditambahkan adalah `el`, dimana value atau nilainya merupakan id dari elemen yang menjadi target untuk dikontrol. Untuk menentukan `div#app` sebagai target, ketikkan baris kode berikut.

```
var app = new Vue({  
  |   el : '#app'  
});
```

10. Di dalam `div#app` terdapat elemen `<h1>`. Teks "Hello World" akan ditampilkan sebagai `<h1>` melalui property `data`, seperti berikut.

```
var app = new Vue({  
  el : '#app',  
  data : {  
    'message' : 'Hello World'  
  }  
});
```

Property `data` merupakan object sehingga juga memiliki property dan value, dimana 'message' merupakan property dengan 'Hello World' sebagai value

n.b: Perhatikan penambahan koma di antara property `el` dan `data`

11. Teks "Hello World" telah ditambahkan melalui `data`. Berikutnya kembali ke `index.html` untuk menempatkan teks tersebut di dalam `<h1>` menggunakan ekspresi `{{ }}`, sebagai berikut.

```
<div id="app">  
  <h1>{{ message }}</h1>  
</div>
```

12. Simpan dan jalankan di browser.

Methods

1. Buat file baru dengan nama `lat2.html`. Untuk latihan ini, kode Javascript ditambahkan di dalam `lat2.html` menggunakan tag `script`.
2. Di dalam `lat2.html` tambahkan kode berikut.

```

<!DOCTYPE html>
<html lang="en">
<head>
  <script src="https://cdn.jsdelivr.net/npm/vue"></script>
  <title>Hello World</title>
</head>
<body>

  <div id="app">

  </div>

  <script>

  </script>

</body>
</html>

```

3. Di dalam tag script, tambahkan baris kode untuk inisialisasi Vue object seperti yang dilakukan di latihan sebelumnya.
4. Setelah property el, tambahkan property methods untuk mendeklarasi method baru.

```

<script>
  var app = new Vue({
    el: "#app",
    methods: {
    }
  })
</script>

```

Lengkapi dengan menambahkan nama method serta barisan kode yang dieksekusi ketika method dipanggil

```

methods: {
  greet: function () {
    return 'Good morning';
  }
}

```

Ketika method 'greet' dipanggil di dalam div#app, akan dikembalikan string 'Good morning'.

5. Kembali ke lat2.html, kemudian tambahkan elemen <p> di dalam div#app untuk tempat memanggil method greet.
6. Baris kode yang ditambahkan untuk memanggil method greet di dalam <p> adalah _____

7. Simpan dan jalankan
8. Untuk melewati nilai ke dalam method greet, tambahkan parameter sebagai berikut.
9. Baris kode yang ditambahkan untuk memanggil method greet yang sudah dimodifikasi adalah
10. Simpan dan jalankan di browser.

Good morning Bias Damiasa

Data Binding

Data binding adalah cara atau teknik untuk mengikat (*binding*) data pada view. Terdapat beberapa directive yang digunakan, salah satunya adalah v-bind. Directive ini dapat digunakan untuk binding value pada atribut HTML.

1. Buat file baru dengan nama lat2.html
2. Di dalamnya tambahkan CDN Vue.js, elemen div#app, dan tag script untuk menambahkan barisan kode Javascript (Vue.js)
3. Inisialisasi object Vue.js baru, kemudian target elemen div#app dengan property el (silahkan mengacu pada latihan sebelumnya)
4. Setelah property el, tambahkan property data dengan value berupa object.
5. Object tersebut memiliki property 'url' dengan value sebagai berikut.

```
<script>
  var app = new Vue({
    el: "#app",
    data: {
      "url" : "https://www.fullstackpython.com/img/logos/vuejs-wide.png"
    }
  })
</script>
```

url: <https://www.fullstackpython.com/img/logos/vuejs-wide.png>

6. URL tersebut adalah url menuju source dari gambar logo Vue JS yang akan ditampilkan di dalam div#app. Untuk menampilkannya, tambahkan tag seperti berikut.

```
<div id="app">
  <img >
</div>
```

7. Untuk menambahkan url gambar pada atribut src, maka digunakan v-bind sebagai berikut.

```

```

8. Simpan dan jalankan pada browser.
9. Tugas: Buat link untuk menuju ke halaman website SMK Telkom Malang. Simpan nilai address website di dalam data, kemudian manfaatkan v-bind agar link tersebut bekerja.

Two Way Data Binding

Teknik data binding berikutnya disebut dengan two way data binding.

1. Buat file baru dengan nama lat3.html
2. Tambahkan CDN Vue.js, div#app, dan tag script
3. Inisialisasi Vue.js object dan target elemen div#app menggunakan property el
4. Tambahkan satu kolom isian dan satu label di dalam div#app

```
<form>
  <input type="text" placeholder="Masukkan nama anda">
  <br> <br>
  <label>Nama saya adalah: </label>
</form>
```

Dengan teknik two way data binding, nilai yang diketikkan di kolom isian dapat ditampilkan secara langsung pada user. Directive yang dibutuhkan adalah v-model

5. Tambahkan property namaSaya dalam data. Nilainya dibuat kosong terlebih dulu.

```
<script>
  var app = new Vue({
    el: "#app",
    data: {
      "namaSaya" : ""
    }
  })
</script>
```

6. Berikutnya, pada tag input, tambahkan directive v-model , seperti berikut.

```
<input type="text" placeholder="Masukkan nama anda" v-model="namaSaya">
```

7. Untuk menampilkan value pada label, panggil property namaSaya seperti berikut.

```
<label>Nama saya adalah: {{namaSaya}} </label>
```

Simpan dan jalankan pada browser. Isi kolom isian dan lihat apakah nilainya langsung ditampilkan pada label.

Condition

Pada bagian ini, akan ditampilkan item dengan kondisi-kondisi tertentu menggunakan directive v-if, v-else-if, dan v-else.

1. Buat file baru dengan nama lat4.html
2. Tambahkan CDN Vue.js, div#app, dan tag script untuk barisan kode Vue.js
3. Inisialisasi object Vue.js, kemudian target div#app dengan property el
4. Di dalam div#app, tambahkan tiga komponen paragraf yang akan ditampilkan dalam tiga kondisi berbeda.

```
<p v-if="itemInStock > 10">{{itemInStock}} in stock</p>
<p v-else-if="itemInStock > 0">Hurry up, only {{itemInStock}} items left</p>
<p v-else>Sorry, we are all out!</p>
```

- Paragraf 1 ditampilkan jika nilai dari itemInStock > 10

- Paragraf2 ditampilkan jika kondisi sebelumnya tidak terpenuhi dan nilai dari itemInStock > 0
 - Paragraf3 ditampilkan jika dua kondisi sebelumnya tidak terpenuhi
5. Setelah property el, tambahkan property data dengan value berupa object seperti berikut.

```
<script>
  var app = new Vue({
    el: "#app",
    data: {
    }
  })
</script>
```

6. Tambahkan property itemInStock dengan value 10

```
data: {
  "itemInStock" : 10
}
```

7. Simpan dan jalankan hasilnya pada browser.
8. Ubah nilai itemInStock untuk menampilkan paragraf lainnya.

Loop

Pada bagian ini, akan diterapkan perulangan array menggunakan directive v-for

1. Buat file baru dengan nama lat5.html
2. Tambahkan CDN Vue.js, div#app, dan tag script
3. Inisialisasi Vue.js object dan target elemen div#app menggunakan property el
4. Tambahkan property data dengan value berupa object, dimana property-nya adalah 'kelas' dengan value berupa array, seperti berikut.

```
<script>
  var app = new Vue({
    el: "#app",
    data: {
      "kelas" : ["XRPL1", "XRPL2", "XIRPL1", "XIRPL2", "XIIRPL1", "XIIRPL2"]
    }
  })
</script>
```

5. Di dalam div#app, tambahkan tag dan untuk menampilkan semua elemen dari array kelas.

```
<ol>
  <li>

  </li>
</ol>
```

6. Di dalam tag tambahkan directive v-for untuk melakukan loop terhadap array kelas dan menampilkan semua isinya.

```

<ol>
  <li v-for="k in kelas">{{k}}</li>
</ol>

```

k adalah variabel yang menampung item-item kelas satu per satu. Oleh karena itu, untuk menampilkan setiap item, dipanggil variabel k menggunakan {{}}

7. Simpan dan jalankan hasilnya di browser

Tambahan: untuk menampilkan index dari perulangan, dapat digunakan kode v-for seperti berikut.

```

<li v-for="(k, index) in kelas" :key="index">

</li>

```

Dimana index adalah counter dari struktur looping, dan menyimpan nilai 0

8. Untuk menampilkan hasil seperti berikut, silahkan memodifikasi struktur loop yang sudah ada.

```

1. Item ke-0 adalah XRPL1
2. Item ke-1 adalah XRPL2
3. Item ke-2 adalah XIRPL1
4. Item ke-3 adalah XIRPL2
5. Item ke-4 adalah XIIRPL1
6. Item ke-5 adalah XIIRPL2

```

Petunjuk: manfaatkan index dari setiap item di dalam array kelas untuk mencapai hasil seperti yang diminta.

Event Handling

Menangani event dalam Vue.js dapat menggunakan directive v-on.

1. Buat file baru dengan nama lat6.html
2. Tambahkan CDN Vue.js, elemen div#app, dan tag script untuk baris kode Javascript (Vue.js)
3. Inisialisasi object Vue.js, kemudian target div#app dengan property el
4. Setelah property el, tambahkan property methods. Di dalamnya, tambahkan method baru untuk memanggil method pesan(). Method ini akan menampilkan pesan "Selamat belajar Vue.js" ketika dipanggil.

```

<script>
  var app = new Vue({
    el: "#app",
    methods: {
      pesan: function() {
        alert("Selamat belajar Vue.js")
      }
    }
  })
</script>

```

5. Di dalam div#app, tambahkan button untuk memanggil method pesan()
`<button v-on:click="pesan">Tampil Pesan</button>`
v-on:click digunakan untuk menangani event pemanggilan fungsi melalui klik tombol.
6. Simpan dan jalankan aplikasi di browser.
7. Vue.js juga menyediakan shorthand untuk penggunaan v-on, sehingga ketika ditulis lengkap adalah `v-on:click="pesan"`
dan ketika dipersingkat menjadi `@click="pesan"`

Vue.js menyediakan key modifiers untuk penanganan event keyboard menggunakan keyup. Contohnya penanganan event ketika menekan tombol Enter. Untuk contoh penggunaan key modifier, akan dibuat form dimana data disimpan dengan menekan tombol Enter atau menekan tombol.

8. Tambahkan property data dengan value object seperti berikut.

```
<script>
  var app = new Vue({
    el: "#app",
    methods: {
      pesan: function() {
        alert("Selamat belajar Vue.js")
      }
    },
    data: {
  },
  })
</script>
```

9. Di dalamnya, tambahkan property 'dataGuru' dan 'guru'


```

<script>
  var app = new Vue({
    el: "#app",
    methods: {
      pesan: function() {
        alert("Selamat belajar Vue.js")
      }
    },
    data: {
      'dataGuru' : [
        {'nama' : 'Arya', 'kelas' : 'XRPL1'},
        {'nama' : 'Bella', 'kelas' : 'XRPL2'},
        {'nama' : 'Cintya', 'kelas' : 'XIRPL1'},
        {'nama' : 'Doni', 'kelas' : 'XIRPL2'},
        {'nama' : 'Elma', 'kelas' : 'XIIRPL1'},
        {'nama' : 'Farhan', 'kelas' : 'XIIRPL2'},
      ],
      'guru' : ''
    }
  })
</script>

```

Perhatikan bahwa value dari 'dataGuru' adalah array dengan item berupa enam object sedangkan property 'guru' adalah string kosong.

10. Tambahkan method baru untuk menyimpan data dari form

```

  methods: {
    pesan: function() {
      alert("Selamat belajar Vue.js")
    },
    simpan: function () {
      this.dataGuru.push(this.guru);
      this.guru = "";
    }
  },

```

guru nantinya akan di-bind dengan kolom isian form menggunakan two-way data binding. Input user disimpan di dalam guru kemudian ditambahkan sebagai elemen dari array dataGuru.

11. Di div#app, tambahkan form yang berisi kolom isian dan tombol submit (hapus elemen tombol sebelumnya).

```

<h3>Masukkan Nama Guru</h3>
<form v-on:submit.prevent>
  <input type="text" name="namaSiswa" id="namaSiswa" @keyup.enter="simpan" v-model="guru">
  <button type="button" @click="simpan">Simpan</button>
</form>
<br><br>

```

- v-on:submit.prevent ditambahkan agar halaman tidak dimuat ulang ketika data disubmit.
- @keyup.enter (shorthand dari v-on:keyup.enter) ditambahkan untuk memanggil method simpan ketika tombol Enter ditekan setelah input data di kolom isian
- v-model ditambahkan untuk two-way data binding dengan property guru yang akan ditambahkan di object data nanti
- @click akan memanggil method simpan

12. Setelah tag
 tambahkan baris kode berikut untuk menambahkan <table>. Seluruh isian dari form akan ditampilkan dalam tabel ini.

```

<table border="1" style="border-collapse: collapse" v-if="dataGuru.length > 0"> ①
  <tr>
    <thead>
      <th style="width: 30px">#</th>
      <th style="width: 200px">Nama Guru</th>
    </thead>
  </tr>
  <tr>
    <tbody v-for="(guru, index) in dataGuru" :key="index"> ②
      <td>{{index+1}}</td>
      <td>{{guru}}</td>
    </tbody>
  </tr>
</table>
<h4 v-else>Tidak ada data</h4> ③

```

- 1) v-if menguji kondisi jumlah elemen yang disimpan di dalam array dataGuru. Jika jumlah lebih dari 0 maka tabel ditampilkan
- 2) Menampilkan semua isi dari array dataGuru. Index adalah counter dari v-for dan nilainya ditampilkan sebagai nomor tabel.
- 3) Ditampilkan jika kondisi nomor 1 tidak terpenuhi

13. Simpan dan jalankan di browser.

Membuat Component

Component adalah instance Vue yang reusable atau dapat digunakan berulang-ulang. Melalui Vue component, dapat dibuat custom element.

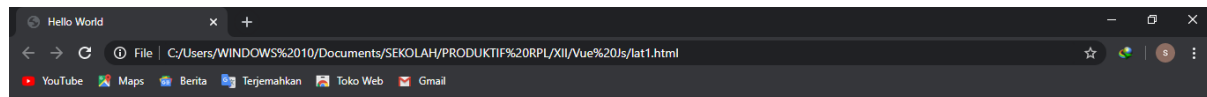
1. Buat file baru dengan nama lat7.html
2. Tambahkan CDN Vue.js, elemen div#app, dan tag script untuk baris kode Javascript (Vue.js)
3. Di dalam tag script, tambahkan baris kode berikut untuk mendeklarasi component baru.

```
Vue.component('counter', ①
{
  data: function () {
    return {
      count: 0 ②
    }
  },
  template: '<button @click="count++">{{count}}</button>' ③
}
);
```

- 1) 'counter' ditambahkan sebagai nama tag dari custom element yang dibuat
 - 2) Data memiliki value berupa function yang mengembalikan object. Di dalam object yang dikembalikan tersebut, terdapat satu property counter dengan nilai 0
n.b: Perhatikan bahwa object tidak secara langsung dideklarasikan sebagai value dari property data seperti pada latihan sebelumnya.
 - 3) Pada template, dideklarasikan kode HTML yang akan di-render oleh browser ketika custom element dipanggil
4. Di dalam div#app, tambahkan baris kode berikut untuk menggunakan custom element yang baru saja dibuat
- ```
<div id="app">
 <counter></counter>
 <counter></counter>
 <counter></counter>
</div>
```
5. Simpan dan jalankan di browser. Hasilnya adalah tiga tombol counter. Perhatikan nilai count dari masing-masing tombol. Walaupun berasal dari instance yang sama, nilai count pada ketiga tombol berbeda tergantung tombol mana yang ditekan.

Link Github : <https://github.com/saddamrr/Vue-Js>

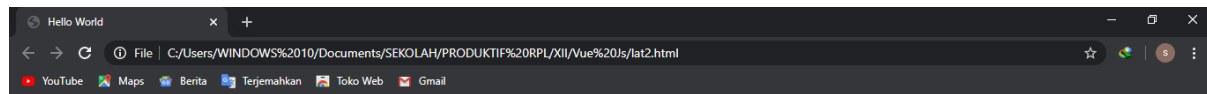
## Latihan 1



**Hello World**



## Latihan 2

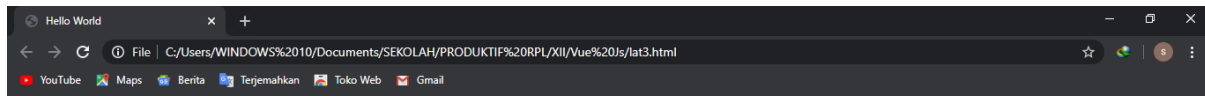


Good Morning Saddam

[My Github](#)



## Latihan 3

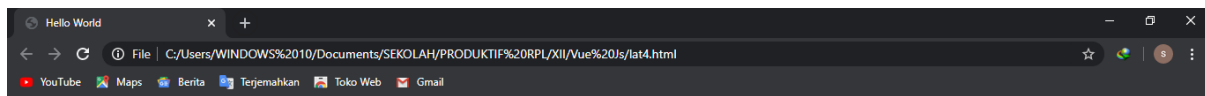


Saddam Raihan Ramadhan

Nama saya adalah : Saddam Raihan Ramadhan



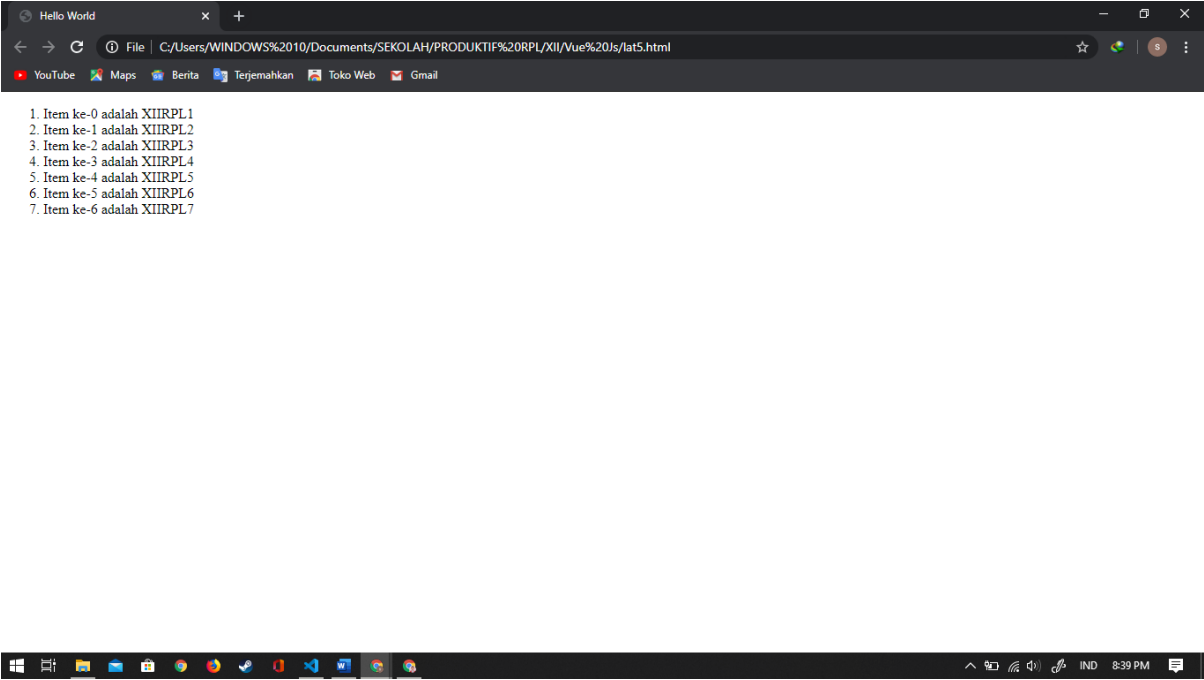
## Latihan 4



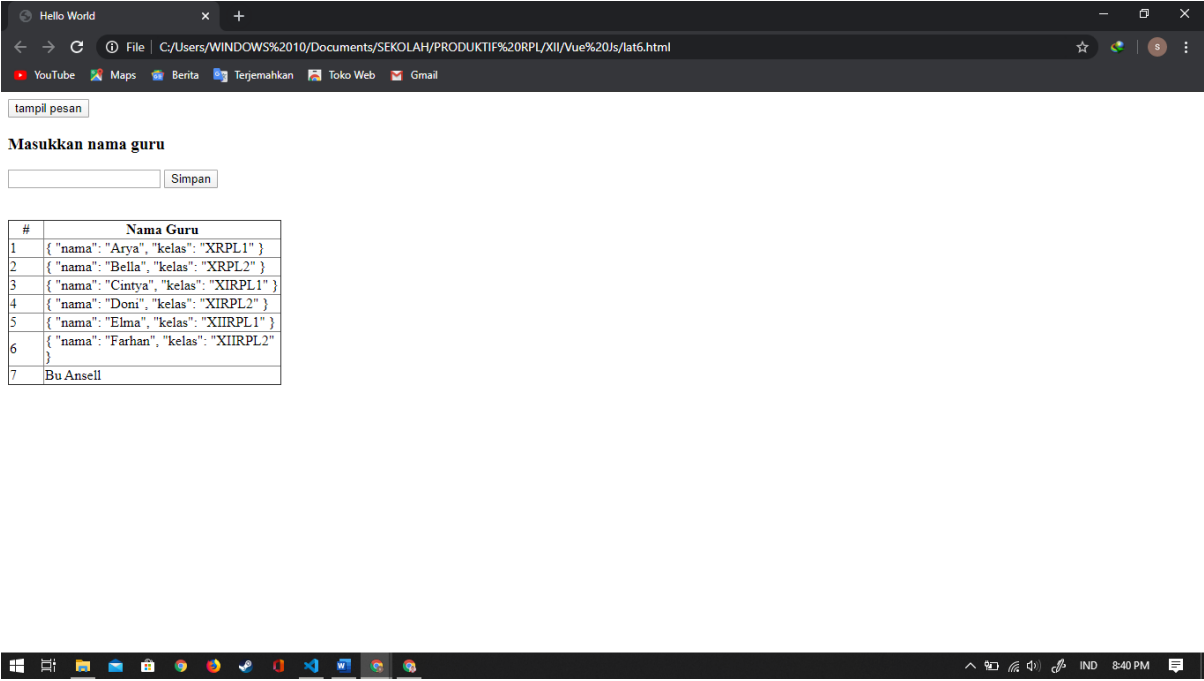
Hurry up, only 10 items left.



## Latihan 5



Latihan 6



Latihan 7

