**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>

1. Simpan.

**Introduction**

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



1. Di dalam body, tambahkan tag div dengan id=”app”.



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

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



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.

1. 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.



1. Di dalam div#app terdapat elemen <h1>. Teks “Hello World” akan ditampilkan sebagai <h1> melalui property data, seperti berikut.



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*

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



1. Simpan dan jalankan di browser.

**Methods**

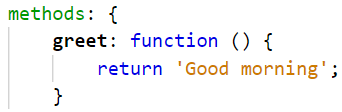
1. Buat file baru dengan nama lat2.html. Untuk latihan ini, kode Javascript ditambahkan di dalam lat2.html menggunakan tag script.
2. Di dalam lat2.html tambahkan kode berikut.



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

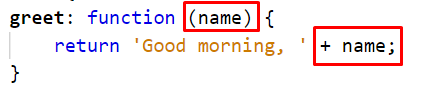


Lengkapi dengan menambahkan nama method serta barisan kode yang dieksekusi ketika method dipanggil



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

1. Kembali ke lat2.html, kemudian tambahkan elemen <p> di dalam div#app untuk tempat memanggil method greet.
2. Baris kode yang ditambahkan untuk memanggil method greet di dalam <p> adalah \_\_\_\_\_\_\_\_
3. Simpan dan jalankan
4. Untuk melewatkan nilai ke dalam method greet, tambahkan parameter sebagai berikut.



1. Baris kode yang ditambahkan untuk memanggil method greet yang sudah dimodifikasi adalah \_\_\_\_\_\_\_\_\_\_
2. Simpan dan jalankan di browser.



**Data Binding**

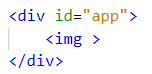
Data binding adalah cara atau teknik untuk mengikat (*binding*) data pada view. Terdapat beberapa directive yang digunakan, salah satunya adalah v-bind. Directive ini dapat digunakan untuk binding value pada atribut HTML.

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.



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

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



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

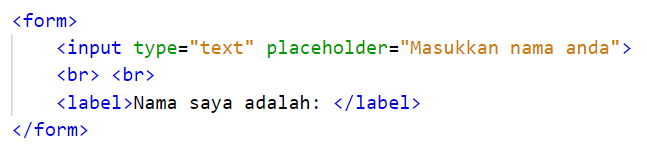


1. Simpan dan jalankan pada browser.
2. Tugas: Buat link untuk menuju ke halaman website SMK Telkom Malang. Simpan nilai address website di dalam data, kemudian manfaatkan v-bind agar link tersebut bekerja.

**Two Way Data Binding**

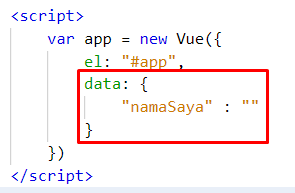
Teknik data binding berikutnya disebut dengan two way data binding.

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



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

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



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



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

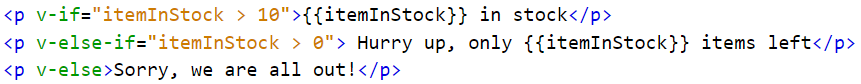


Simpan dan jalankan pada browser. Isi kolom isian dan lihat apakah nilainya langsung ditampilkan pada label.

**Condition**

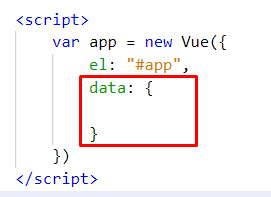
Pada bagian ini, akan ditampilkan item dengan kondisi-kondisi tertentu menggunakan directive v-if, v-else-if, dan v-else.

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.

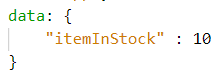


* Paragraf 1 ditampilkan jika nilai dari itemInStock > 10
* Paragraf 2 ditampilkan jika kondisi sebelumnya tidak terpenuhi dan nilai dari itemInStock > 0
* Paragraf 3 ditampilkan jika dua kondisi sebelumnya tidak terpenuhi

1. Setelah property el, tambahkan property data dengan value berupa object seperti berikut.



1. Tambahkan property itemInStock dengan value 10

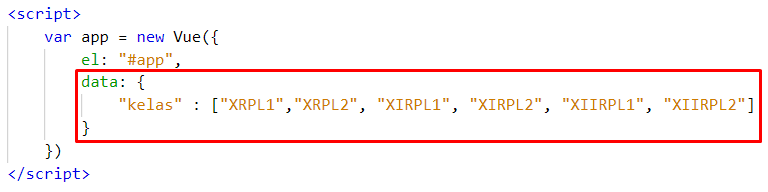


1. Simpan dan jalankan hasilnya pada browser.
2. Ubah nilai itemInStock untuk menampilkan paragraf lainnya.

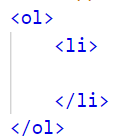
**Loop**

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

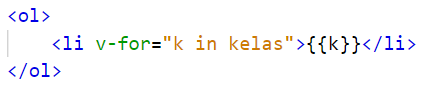
1. Buat file baru dengan nama lat5.html
2. Tambahkan CDN Vue.js, div#app, dan tag script
3. Inisialisasi Vue.js object dan target elemen div#app menggunakan property el
4. Tambahkan property data dengan value berupa object, dimana property-nya adalah ‘kelas’ dengan value berupa array, seperti berikut.



1. Di dalam div#app, tambahkan tag <ol> dan <li> untuk menampilkan semua elemen dari array kelas.



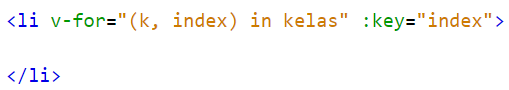
1. Di dalam tag <li> tambahkan directive v-for untuk melakukan loop terhadap array kelas dan menampilkan semua isinya.



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

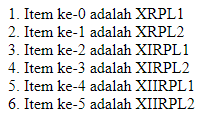
1. Simpan dan jalankan hasilnya di browser

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



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

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

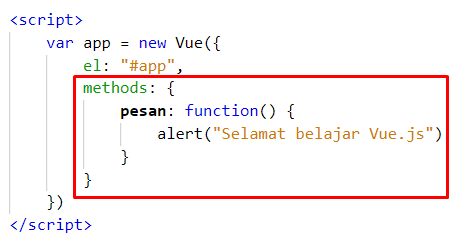


Petunjuk: manfaatkan index dari setiap item di dalam array kelas untuk mencapai hasil seperti yang diminta.

**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.



1. Di dalam div#app, tambahkan button untuk memanggil method pesan()



v-on:click digunakan untuk menangani event pemanggilan fungsi melalui klik tombol.

1. Simpan dan jalankan aplikasi di browser.
2. Vue.js juga menyediakan shorthand untuk penggunaan v-on, sehingga ketika ditulis lengkap adalah

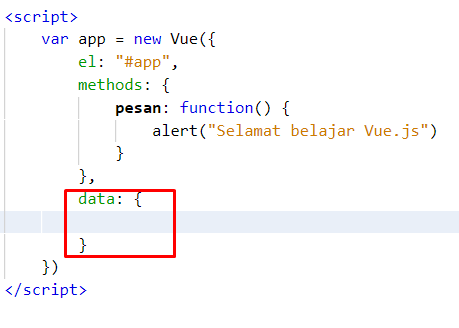
v-on:click=”pesan”

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.

1. Tambahkan property data dengan value object seperti berikut.

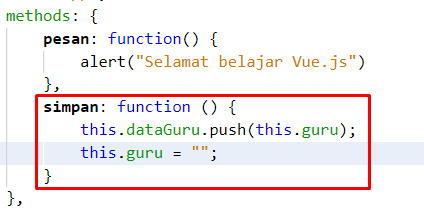


1. Di dalamnya, tambahkan property ‘dataGuru’ dan ‘guru’



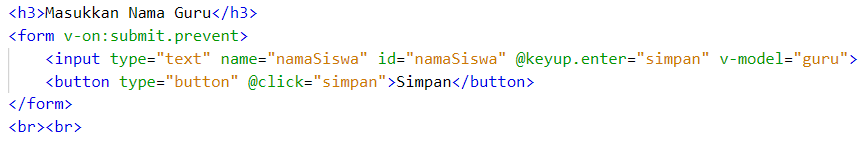
Perhatikan bahwa value dari ‘dataGuru’ adalah array dengan item berupa enam object sedangkan property ‘guru’ adalah string kosong.

1. Tambahkan method baru untuk menyimpan data dari form



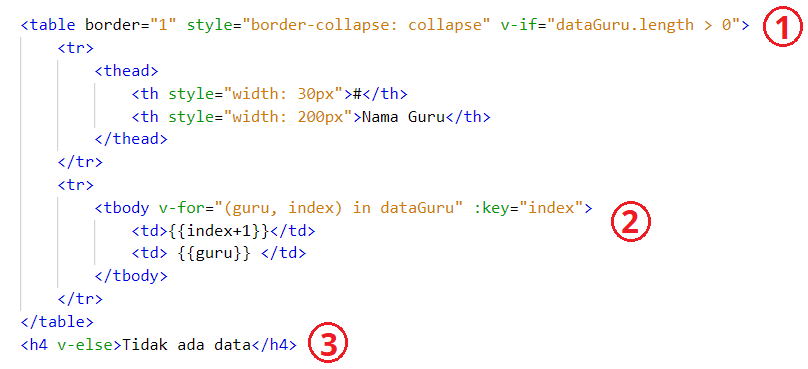
guru nantinya akan di-bind dengan kolom isian form menggunakan two-way data binding. Input user disimpan di dalam guru kemudian ditambahkan sebagai elemen dari array dataGuru.

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



* 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

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

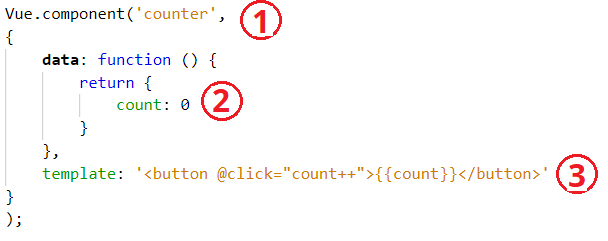


1. v-if menguji kondisi jumlah elemen yang disimpan di dalam array dataGuru. Jika jumlah lebih dari 0 maka tabel ditampilkan
2. Menampilkan semua isi dari array dataGuru. Index adalah counter dari v-for dan nilainya ditampilkan sebagai nomor tabel.
3. Ditampilkan jika kondisi nomor 1 tidak terpenuhi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

**Membuat Component**

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

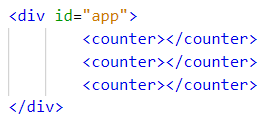
1. Buat file baru dengan nama lat7.html
2. Tambahkan CDN Vue.js, elemen div#app, dan tag script untuk baris kode Javascript (Vue.js)
3. Di dalam tag script, tambahkan baris kode berikut untuk mendeklarasi component baru.



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.*

1. Pada template, dideklarasikan kode HTML yang akan di-render oleh browser ketika custom element dipanggil
2. Di dalam div#app, tambahkan baris kode berikut untuk menggunakan custom element yang baru saja dibuat



1. 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.