var vm = New Vue({

el: “app”,

data: {},

template: ,

computed: , 🡪 sebagai objek, no param

methods , 🡪 sebagai fungsi, can input param. Get only get, set bisa mengambil dari console

**v-once**

<p v-once> {{ message }} </p>

Message tidak akan berubah meskpun pada console vm.message diubah

**v-html**

<h1 v-html=”message”></h1>

Message: ‘<span style=”color=green”>Hello Bandung</span>’

Menulis dalam kurung kurawal 🡪 hanya expression, no statement

<p> {{ ‘pesan: ‘ + message }} </p>

<p> {{ ‘number \* 20 = ${number\*20} }} </p> 🡪 template literal

<p> {{ var = angka }} </p> 🡪 tidak bisa!

directive (-v): menerapkan ‘efek samping’ kepada DOM saat terjadi perubahan data

**v-if** -> if directive

<p v-if="seen">Now you see me</p>

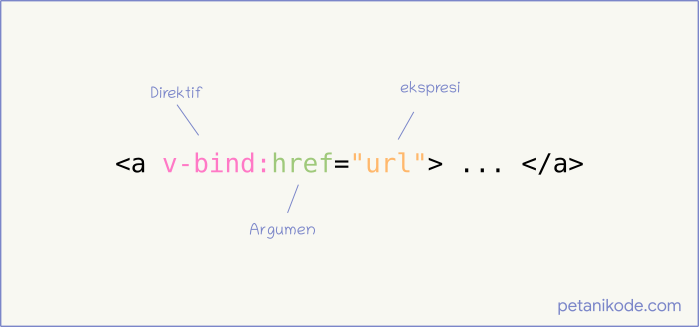
**v-bind** -> one way data binding, mengisi attribute html dengan variable data vue. hanya mengubah variabel pada line tersebut

<input type="text" v-bind:value="namaVariabel" />

<img v-bind:[attrName]="value" /> 🡪 pada dynamic attribut

**v-model** -> two way data binding, mengubah variable di seluruh tempat

<input type="text" v-model:value="namaVariabel" />



<p v-if="qty >= 10"> kamu hanya boleh beli {{ qty }} aja, soalnya kita gak punya banyak stok</p>

<p v-else-if="qty >= 5">Whoaa tukang borong nih!</p>

<p v-else-if="qty >= 2">Selamat! Kamu dapat bonus, karena membeli {{ qty }} buku</p>

<p v-else>Beli liebih banyak lagi untuk dapatkan bonus :)</p>

<li v-for="item in todolists"> 🡪 todolists adalah variable vue berisi list of object

<s v-if="item.done">{{ item.task }}</s>

<b v-else>{{ item.task }}</b>

</li>

<button @click="btnClick" @mouseover="hover = true" @mouseout="hover = false" >Klik Saya</button>

Modifier:

<input v-model.number="namaVariabel" />

<input v-model.lazy="namaVariabel" />

Quiz 2 : <https://www.youtube.com/watch?v=SZd7GsV9zzI&ab_channel=Sanbercode>

Tugas 11: <https://documenter.getpostman.com/view/15879306/UUy3AST5>

**Vue CLI**

Semacam framework untuk membuat project VueJS

1. Install Vue CLI

npm install -g @vue/cli

1. Create new project

vue create blog-project

“blog-project” = nama projek kita

pilih default (vue 2)

1. Jalankan project

cd blog-project

npm run serve

buka di http://localhost:8080/

1. Buka project di IDE

Buka cmd baru, arahkan ke folder projek, code .

1. Tiap mau run project setelah menambah sesuatu package, run:

npm run serve

**Vue router**

Semacam finite state machine untuk mengatur state2

1. Install Vue router

cd blog-project

vue add router

1. Create new project

vue create blog-project

vue history mode? Y

**Vuetify**

Framework ui library vue, semacam bootstrap di css

1. Install Vue router

cd blog-project

vue add vuetify

preset = default

1. Install vuetify dependency? Tidak usah, cukup npm run serve ulang saja

npm install --save vuetify/lib/framework

1. Akan ada styling yg baru dari vuetify, e.g. v-container, v-row, etc

Tambahkan component ke view seperti bootsrap

**Vue Axios**

Untuk request PI

1. Install Vue axios

cd blog-project

npm install --save axios vue-axios

1. Buat file plugins/axios.js, isi di dalamnya:

import \* as Vue from 'vue' // in Vue 3

import axios from 'axios'

import VueAxios from 'vue-axios'

Vue.use(VueAxios, axios)

1. Di main.js:

import axios from ' ./plugins/axios'

di new Vue: axios,

**Isi router, axios (req dan response API), view**

Atur komponen-komponen blog

1. Isi router, axios (req dan response API), view

bla

**Vuex**

Untuk state management: sentralisasi variabel data, sehingga semua component dalam aplikasi dapat mengakses dan memanipulasinya dengan aturan tertentu sehingga perubahannya dapat diprediksi.

1. Install Vuex

cd blog-project

npm install vuex --save

1. Buat folder src/store, buat file index.js, isi:

import Vue from 'vue'

import Vuex from 'vuex'

Vue.use(Vuex)

New Vuex dll

1. Di file main.js, isi:

import store from ‘./store’

1. Tambahkan berbagai fungsi menggunakan variable global dari store di views

Bisa gunakan mapGetters, mapMutations, dll

1. Modularisasi komponen2 store

Buat file baru, nama komponen store (e.g. counter.js) di folder store

Isikan state, getter, mutation, action dsb di file komponen store (counter.js)

Import komponen store ke file store

Pada file views, panggil nama komponen store (counter/komponen) di setiap pemangilan var global

**Vuex-persist**

Untuk menyimpan state secara persistent, missal jika direload maka data tetap sama

1. Install Vuex persist

npm install --save vuex-persist

1. di file store/index.js, isi:

import VuexPersist from 'vuex-persist'

const vuexPersist = new VuexPersist({

key: 'sanbercode',

storage: localStorage

})

plugins: [vuexPersist.plugin],

**Deploy ke Netlify**

Untuk menyimpan state secara persistent, missal jika direload maka data tetap sama

1. Signup Netlify
2. Pastikan semua http diubah ke https atau Netlify akan membloknya
3. Build project kita:

npm run build

aka nada folder dist

1. Drag and drop folder dist hasil build ke Netlify

https://unruffled-shannon-c48001.netlify.app/