***VueJs***

## Apa itu dan kenapa VueJs ?

*Vue* (cara pengucapannya /vjuː/, seperti view) adalah sebuah kerangka kerja nan progresif untuk membangun antarmuka pengguna.

*Vue* merupakan framework javascript yang biasa digunakan oleh developer untuk membangun single-page application (SPA). Vue juga sangat ringan dan kecil dibandingkan react dan angular sehingga memiliki run-time performa yang tinggi dan sangat cepat.

*Vue* menganut system MVVM yang akan tersambung ke element HTML (DOM) ketika ada perubahan yang terjadi makan akan secara langsung (reactive).

## Installasi dan Penggunaan Vue JS

### *Installasi Vue dengan CDN*

Untuk installasi vue dengan menggunakan CDN dapat diakses melalui link berikut, pastikan vue disimpan sebelum tutup body, dan sebelum script code vue kalian dipanggil.

[*https://docs.vuejs.id/v2/guide/installation*](https://docs.vuejs.id/v2/guide/installation)

### *Installasi Project Vue*

…

## Object Vue

Object ini akan mengatur data data yang berkaitan dengan aplikasi dan bisa membuat sebuah property atau method untuk menjalankannya. Penulisan object Vue dapat dilakuan dengan menuliskan *new Vue();* .

Keyword *new Vue();* wajib dilakukan saat kita ingin insialisasi object Vue.

Object Vue dapat kita anggap seperti class, Dimana didalamnya terdapat property dan juga method.

A computer screen with green text

Description automatically generated

Contoh code vue sederhana

A computer screen shot of a computer program

Description automatically generated

*el*  : itu merepresentasikan sebuah element html yang kita tangkap dengan DOM.

*data* : itu bisa kita anggap sebagai property atau variable di dalam Object.

Pemanggilan data di vue dapat kita gunakan *{{ … }}* atau dapat disebut *interpolasi.*

A screen shot of a computer

Description automatically generated

### Ternary Operator

Anggap kita memiliki Data dan data tersebut bernilai true, kita bisa jalankan sebuah ternary operator di dalam interpolasi vue.

A blue background with white text

Description automatically generated

### Method

Berikut adalah contoh penerapan method di dalam Object Vue, dan pemanggilan Method di dalam interpolasi. Agar dapat mengakses property di dalam data kita gunakan keyword *this* untuk merujuk data di dalam object Vue tersebut.

A computer screen shot of a program code

Description automatically generated

### Method Parameter

Kita juga bisa mengirimkan parameter kedalam method yang kita miliki ini berfungsi untuk menangkap data yang dikirimkan ke method.

Berikut adalah contoh codenya :

Untuk mengakses parameternya kita gunakan keyword *this,* sama seperti memanggil data. Lalu bagaimana jika data dan parameter Bernama sama? Bagaimana dengan konsep *this*?

Ada yang Namanya lingkup scope yang pertama dipanggil itu adalah method dari parameternya terlebih dahulu.

*v-model*  : digunakan untuk menghubungkan nilai.

*@input* : merupakan shortcut dari *v-on:input* artinya ketika ada inputan jalankan method-nya.

A computer screen shot of a program code

Description automatically generated

### Computed Property

Ini seperti kita mendifinisikan sebuah property / variable di vue.js namun dalam bentuk function.

Cara ini tidak disimpan di dalam methods tapi disimpan di dalam property / keyword yang Namanya *computed*. Dan untuk penulisan code di dalamnya mirip mirip seperti membuat sebuah method.

*Perbedaan antara Computed dan Methods…*

* *Method* hanya akan dijalankan ketika dipanggil saja dan hasilnya tidak di cache, meskipun dapat melihat rectivity dari *data* yang ada di dalam object vue.
* *Computed* digunakan untuk melakuan kalkulasi atau perhitungan informasi yang lebih kompleks atau rumit dan hasilnya di cache maka jika dependencynya berubah maka akan dihitung ulang.
* Perbedaan diantara keduanya yaitu jika *Computed* itu akan di cached dan ketergantungan **dependency**, *dependency* yang dimaksud seperti property yang ada di dalam data. *Computed* cocok digunakan untuk data yang bergantung pada property data yang selalu tetap atau constant.

A screen shot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

Dengan menggunakan method maka datanya akan ditampilkan ketika dipanggil, sedangkan jika computed datanya akan disimpan ke dalam Cache.

Sebagai contoh saya melakukan perubahan pada data, maka cache dari computed akan dibuild ulang, sedangkan untuk Method itu dipanggil kembali sehingga jika menggunakan method maka code akan dipanggil berulang ulang kali apabila ada perubahan data.

Contoh penerapan computed

Computed cocok digunakan untuk data yang bergantung pada state lain di dalam Object Vue. Sedangkan method cocoknya digunakan untuk data di dalam Object vue itu dan parameter yang ada di dalam method itu sendiri.

A computer screen shot of a program code

Description automatically generated

## Directive

Directive merupakan sebuah fungsi yang ada di dalam vue js dan disimpan di html. Berikut adalah beberapa contoh directive

*v-once* = digunakan agar datanya tidak bisa diubah / tetap.

*v-text ­*= digunakan agar datanya dapat ditampilkan kedalam element html, ini merupakan cara lain dalam memanggil data selain menggunakan interpolasi

*v-html* = digunakan untuk menuliskan code Html di dalam sebuah syntax html.

### Directive Binding

Di dalam directive kita juga bisa melakuan sebuah expression, berikut contoh nya.

<h1 v-text=” ‘Halo ‘ + name”></h1>

<h1 v-text="name.toUpperCase()”></h1>

### Attribute Binding

Bagaimana dengan attribute html seperti *href, src* dan lain sebagainya. Kita bisa menggunakan keyword

*v-bind:namaAttr*

terdapat juga shortcut dalam penulisannya cukup dengan *:namaAttr*

*A computer screen with green text

Description automatically generated*

### Event Binding

Mungkin sebelumnya kita sudah melakukan binding menggunakan event pada saat terjadi inputan.

*v-on:event* digunakan ketika akan ada perubahan yang terjadi / terjadi sesuatu.

Berikut contoh code sederhananya, code ini sama seperti counter.

A black background with yellow and blue text

Description automatically generated

### Class Binding

Cara binding class ini sama seperti melakukan binding pada attribute html. Berikut ini kita lakukan pengecekkan akan dibere class active jika menu itu sama dengan ‘*namaMenu*’ dan ketika button di klik maka jalankan juga sebuah directive event untuk mengubah datanya.

A computer screen shot of a program code

Description automatically generated

### Two Ways Data Binding

Singkatnya seperti ketika ada perubahan yang kita lakukan pada javascript kita maka akan berpengaruh ke tampilan, dan juga sebaliknya perubahan pada tampilan juga akan berpengaruh juga terhadap value yang ada di dalam javascriptnya.

Keyword yang digunakan untuk melakukan Two Ways Data Binding itu adalah

*v-model*

*perintah untuk binding data dengan menggunakan v-bind: sedangkan perintah untuk two ways binding menggunakan v-model.*

Perhatikan code tersebut, v-model itu akan menangkap setiap inpuan dari user.

Apabila kita membuat *v-model* lalu menghubungkan ke property yang ada di dalam data maka data property itu akan langsung berubah.

Sedangkan dari contoh berikut kita tidak melakukan hal tersebut melainkan kita bungkus ke dalam sebuah even ­*v-on:click* jalankan sebuah method sekalian mengisi ulang property yang dipanggilnya.

A computer screen shot of a program code

Description automatically generated

Apabila *v-model* sama dengan nama property di dalam data maka field input akan sudah terisi dengan secara otomatis mengambil di property data.

Berikut juga contoh penerapan dari *v-model* dengan menggunakan checkbox, kita dapat menangkap data dari inputan dengan merubah property yang ada di dalam data.

A screen shot of a computer program

Description automatically generated

### Looping Directive

*v-for* digunakan dalam melakukan perulangan pada template directivenya.

Berikut contohnya

*v-for* ini loopingnya mirip seperti foreach dimana kita memberikan nama alias dari data yang mau diloopoingnya, kemudian di dalam interpolasinya kita cukup panggil nama aliasnya / kita bisa gunakan directive binding.



Jika cara ini adalah ketika kita ingin melakukan looping dari angka 1 sampai 5.

Bagaimana dengan Index?

A computer code on a black background

Description automatically generated

Sama seperti looping dengan menggunakan foreach berikut contoh menampilkan index dari data arraynya cukup diawali denan (*aliasItem, index*).

Latihan dengan menggunakan *v-for*

Code berikut contoh melakukan filter pada numbers dan mengembalikan nilai genap.

A computer screen shot of a program code

Description automatically generated

### Conditional Directive

Biasanya digunakan untuk menampilkan atau menyembunyikan element HTML, misal yang sering dilakukan pada dialog Modal. Atau dapat kita gunakan untuk menambahkan / bahkan menghapus element HTML.

*v-if* biasanya yang digunakan untuk kasus seperti ini.

Berikut adalah contoh kondisi pengecekan pada directive dengan memeriksa data di dalam object vue.

A computer screen shot

Description automatically generated

*v-else* terkadang ini digunakan untuk kondisi ketika v-if tidak terpenuhi.

Jika misalkan datanya tidak ada di dalam v-if maka element HTML akan dihapus dari struktur element HTML itu sendiri.

Berikut adalah contoh kode ketika datanya bernilai kosong, maka akan mengembalikan tampilan dari v-else.

A computer code with text

Description automatically generated with medium confidence

*v-show* directive ini sama seperti v-if bertujuan untuk menambahkan atau menghapus element HTML.

v-show tidak bisa disandingkan dengan v-else.

Berikut adalah contoh dari v-show, kondisi ini lebih berfokus pada pengecekan kondisi Boolean, atau jadi

Jika hasilnya bernilai true akan ditampilkan.

A computer screen with text and symbols

Description automatically generated

### Conditional Group Directive

Conditional *v-if* biasanya disimpan di dalam parent dari element HTML, menurut dokumentasi vue <https://docs.vuejs.id/v2/guide/conditional> kita dapat bukus pengkondisian tersebut dengan keyword *template*.

A computer screen shot of a program code

Description automatically generatedBerikut adalah contohnya

A screenshot of a computer program

Description automatically generated

*Element html template itu tidak akan dirender saat tampilan dibangun*.

Namun kita tidak selalu harus membungkus itu ke dalam template kita tetap bisa menggunakannya di dalam div / element html lainnya.

A screenshot of a computer program

Description automatically generated

*v-show tidak mendungkung element html template.*

### List Rendering Object

Terkadang kita ingin melakukan looping pada array yang didalam nya sebuah object, kita dapat melakukannya dengan melakukan looping seperti biasa.

Perhatikan contoh berikut agar kita dapat mengambil data di dalam objectnya kita cukup gunakan tanda (titik) agar dapat mengakses isi objectnya.

*Contoh Object di dalam Array.*

A screenshot of a computer

Description automatically generated

*Contoh Array di dalam Object.*

A computer screen shot of text

Description automatically generated

### Event Handler

A computer code with text

Description automatically generated with medium confidence

Contoh code diatas itu menggunakan *v-on:event* Bagaimana pun logika untuk setiap event handler akan terlihat sangat rumit jika menempatkan code javascript di dalam event handler itu sangat tidak tepat, oleh karena itu v-one dapat menerima nama method yang ingin dijalankan.

A computer screen shot of text

Description automatically generated

Contoh berikut adalah kondisi ketika button di klik maka akan menampilkan sebuah alert.

Terkadang kita selalu membuat method dengan () atau bahkan tanpa (), sebenarnya dalam keduanya sama saja fungsinya untuk menjalankan method hanya saja jika tana () maka tidak akan ada argument yang dikirimkan, jika dengan () akan ada argument yang dikirimkan dan method langsung dijalankan.

#### Global Variabel

Terkadang ketika kita melakukan redirect ke halaman dengan menggunakan tag a biasanya halaman terjadi refresh dan halaman mereload ulang, lantas bagaimana agar link tersebut di block?

Kita bisa gunakan preventDefault() code javascript agar mematikan aksi yang dilakukan element html nya, misal dalam kasus ini yaitu redirect tag a.

*preventDefault biasanya digunakan saat mengirim form, meload element agar tidak bisa di klik kanan, dilakukan pada tag a, pada element input.*

Perhatikan contoh berikut untuk memuat global variable kita harus mendefinisikannya dengan menggunakan $ sebagai parameter tambahan yang mendefinisikan bahwa method ini akan menjalankan event dimana event ini bersifat preventDefault.

A screen shot of a computer

Description automatically generated

#### Event Modifiers

Mungkin sebelumnya kita mendefinisikan ke dalam method event nya, kitab isa membuat method tersebut menjadi lebih clean dengan memisahkan proses event nya.

berikut adalah contoh dari model event modifiers tujuannya mempermudah / mempersingkat argument dari method yang dijalankan.

<https://docs.vuejs.id/v2/guide/events#Memantau-Event>

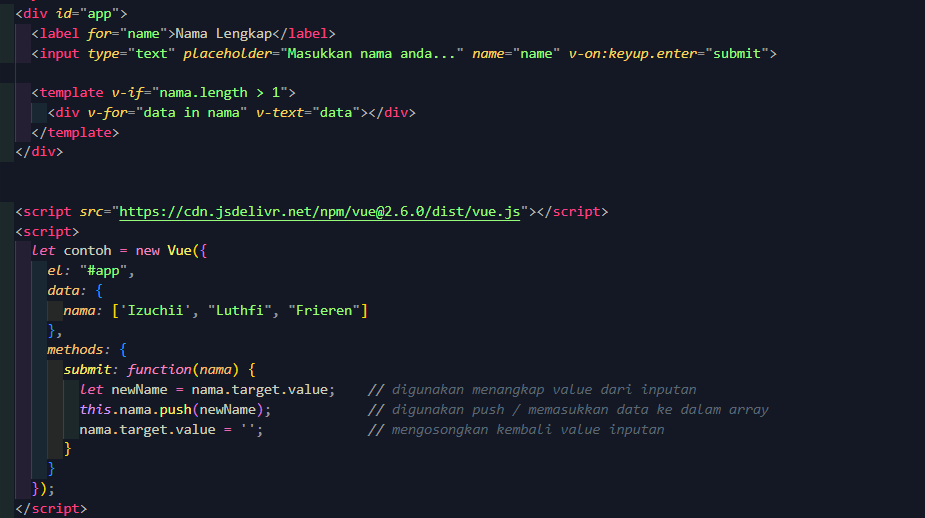
A screen shot of a computer program

Description automatically generated

#### Key Modifiers

Event ini dilakukan ketika ada perubahan pada keyword atau page, misal ketika di klik tab, enter atau ketika di scroll dll. Untuk dokumentasinya dapat dilihat disini. <https://docs.vuejs.id/v2/guide/events#Pengubah-Modifier-Key>

Berikut adalah contoh ketika kita mengisikan inputan dan kita klik enter maka data akan di masukkan ke dalam array



### Form Input Binding

Seperti contoh seblumnya kita bisa menggunakan *v-model* untuk membuat data binding dua arah / *two ways binding*. *V-model*  ini agak Ajaib karena pada dasarnya ini adalah syntax sugar dimana ketika ada aksi yang dilakukan oleh user maka akan memperbarui datanya.

*v-model* ini merupakan two ways binding, jadi akan mengabaikan attribute *value, checked* dan *selected*. Maksudya bagaimana? Dengan memanggil v-model maka attribute itu sudah termasuk di dalam v-model misal kita tambahkan v-model=”nama” maka akan menghubungkan nama yang ada di object vue sehingga kita tidak perlu mengisikan value dengan nama.

Berikut adalah contoh penerapan v-model

#### v-model input text

A screenshot of a computer program

Description automatically generated

#### v-model input textarea

Contoh ini adalah penerapan dengan multiline dengan menggabungkan style css dan vue.

A computer code with text

Description automatically generated with medium confidence

#### v-model input checkbox

Contoh v-model ini yang ada di dalam tipe checkbox

A screen shot of a computer program

Description automatically generated

#### v-model input radio button

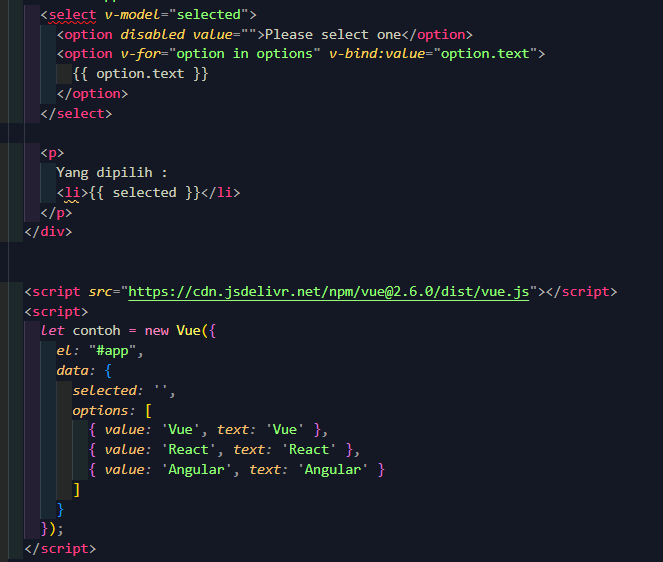
Berikut adalah contoh pada radio button.

A screen shot of a computer program

Description automatically generated

Terkadang kita ingin membuat opsi pilihan di dalam radio button secara dinamis, kita dapat menyimpan data sebelumnya dan melakukan looping terlebih dahulu.

Berikut adalah contoh penulisan codenya



## Component Basics

Component pada vue js merupakan sebuah object dari vue instance yang dapat digunakan sebagai fungsi untuk menyimpan sebuah template, dan template tersebut bisa kita panggil berkali kali.

Component ini merupakan customisasi dari element apa yang dibuat, oleh karena itu nanti nama element html akan sama seperti nama component. Component juga tidak akan merubah data component yang lain, karena ketika dicetak di dalam DOM instance baru akan dibuat secara terpisah.

*Untuk component pastikan data berbentuk function dan mengembalikan return karena jika tidak data tersebut akan mempengaruhi nilai yang lain.*

<https://docs.vuejs.id/v2/guide/components#Pengorganisasian-Komponen> biasanya aplikasi akan dibentuk dalam component component bersarang. Dan ada dua tipe component Global dan local.

Vue global component cukup lakukan ­*Vue.component(‘nama-component’,{});*

*A screen shot of a computer program

Description automatically generated*

*A screenshot of a computer

Description automatically generated*

Ketika kita membuat compnonent pastikan kita panggil atau buat sebuah object instance vue baru dengan keyword *new* dan pastikan element yang ditujunya.

Jika tanpa menggunakan keyword *new* maka itu merujuk penggunaan class, atau object vue itu sendiri.

### Analogi Object Vue

A black screen with white text

Description automatically generated

Pastikan ketika membuat object Vue component sebelum instance Vue dipanggil menggunakan keyword new karena jika sesudahnya component tidak akan tampil.

### Data Component

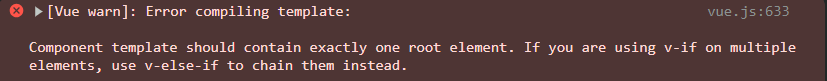
Dari contoh sebelumnya kita sudah membuat data di dalam component pastikan bahwa data berbentuk function dan mengembalikan return karena jika tidak nilainya akan mempengaruhi yang lain.

*A screen shot of a computer program

Description automatically generated*

### Component Template

Terkadang kita menemui error seperti ini, artinya dimana kita membuat component isinya berisi element element.



A screen shot of a computer code

Description automatically generatedseperti ini, dan cara ini salah. Sebaiknya semuanya kita bungkus terlebih dahulu misal ke dalam div.

### Props

Bagaimana cara kita mengoper / mengirimkan datanya ke dalam child component? Disinilah kita gunakan *props.*

*Props* adalah custom attribute yang dapat anda daftarkan di dalam component, pada saat data dikirim ke dalam attribute props maka nilai tersebut akan menjadi property data dalam component.

Untuk mengaksesnya sama seperti property yang ada di dalam data. Ini bisa dianalogikan sebagai parameter yang dikirimkan ke dalam component.

A screenshot of a computer program

Description automatically generated

Dengan begini title yang dikirimkan sebagai attribute di element htmlnya akan dikirimkan dan ditangkap oleh props.

Data di dalam aplikasi seharusnya memiliki penyimpanan khusus sehingga datanya bisa dibagikan antar component. Jadi misal saja id #app ini sebagai component parent, dan component lainnya adalah child component-nya.

### Props Menangkap Data

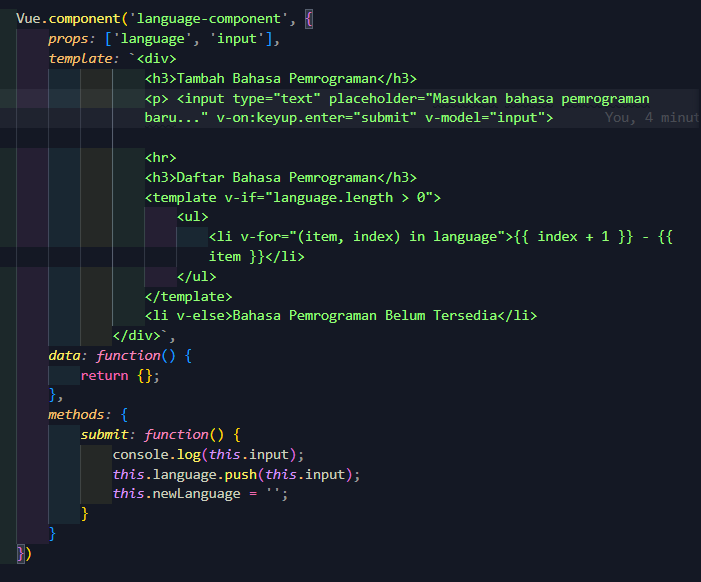
Mungkin sebelumnya kita sudah bisa dalam menyimpan data dan menampilkannya, pada kali ini data tersbut akan dikirimkan ke dalam component, dan disimpan ke dalam props.

A computer screen shot of a code

Description automatically generated with medium confidence

Ini adalah contoh component nya

*Props* akan menangkap datanya, dan menampilkan datanya ke dalam template component-nya.



Kita bisa mengirimkan data ke dalam props dengan melakukan binding data nya ke dalam custom attribute, dan untuk inputan pastikan binding input sama dengan binding v-model

*Input sudah bind dengan Object Vue Parent nya maka saat di component tidak perlu newLanguage.*

A screen shot of a computer

Description automatically generated

### Single Root Element

Pada saat membuat sebuah component, template yang ditulis harus berisi 1 root element, namun template kita tidak mungkin 1 element saja dan kemungkinan akan banyak.

A computer code with text

Description automatically generated with medium confidenceApabila kita menuliskan seperti ini, vue.js nya akan menampilkan kesalahan di console.

*Errornya menjelaskan setiap element yang dirender harus memiliki maksimal 1 element root*.

Untuk mengatasinya biasanya element” itu disimpan di dalam 1 element pembungkus biasanya menggunakan element div.

#### Penamaan Component

Kita bisa menggunakan Kebab Case.

Contoh: my-component-name

Atau menggunakan Pascal Case.

Contoh: MyComponentName

#### Lingkup Component

Untuk Global Registration Component, kita gunakan Vue.component

Untuk Local Registration Component dapat seperti berikut

A computer screen shot of green text

Description automatically generated

Untuk menampilkannya sama seperti biasa kita memanggil component ke dalam element HTML.

### Listening to Child Components Events

<https://docs.vuejs.id/v2/guide/components#Elemen-Root-Tunggal> Mungkin sebelumnya kita berhasil mengirimkan data parent ke dalam child, untuk kasus kali ini kita akan mengirimkan atau menjalankan sebuah event ke dalam parent nya.

Mungkin disini kita berikan contoh code sebelum penjelasan dijabarkan

A screen shot of a computer

Description automatically generated

*$emit* digunakan untuk pada component child untuk berkomunikasi dengan induk-nya.

A screen shot of a computer program

Description automatically generated

Perhatikan pada contoh Code diatas

Sebelumnya kita buat sebuah tag a, dengan menjalankan sebuah fungsi *prevent* dengan cara akan mengakses *parent* dari component nya.

Kita bisa gunakan keyword method *$emit*, dengan menggunakan *$emit* kitab isa menjalankan sebuah method yang ada di dalam parent component nya

*$emit(‘nama\_method’, parameter)*

Kemudian di dalam penulisan component di dalam tag html kita bisa berikan *v-on:click=”nama\_method”*

Maka nanti nama method ini akan merujuk ke parent dari methodnya.

Untuk parameter di dalam tag html kitab isa memberikan seperti ini *v-on:click=”nama\_method($params)”* namun kita juga bisa secara eksplisit tidak memasukkannya

Ketika event hapus dipancarkan oleh komponen anak, Vue.js akan secara otomatis memanggil metode hapus di komponen induk dan meneruskan parameter yang diperlukan tanpa perlu menyebutkan parameter tersebut secara eksplisit di dalam template

*Kesimpulan*

*Jika Parent to Child kita gunakan props dan mengirimkan datanya melalui custom binding.*

*Sedangkan jika Child to Parent kitab isa gunakan $emit sebagai perizinan ke parent method.*

#### Emit Submit Data

Kita bisa melakukan hal yang sama pada event:keyup submit yang dikirimkan dari child ke parent dengan menerapkan emit.

A screen shot of a computer program

Description automatically generated

*$emit(‘submit’ ini merupakan menandakan menjalankan sebuah fungsi submit*

*Lalu $event merupakan data yang dikirimkan*

Hapus juga v-model yang sebelumnya ada di dalam input karena v-model sudah tidak terhubung langsung dengan data apapun.

Kemudian daftarkan emit di custom component, lalu buat sebuah methodnya di Parent Object Vue nya.

A screen shot of a computer code

Description automatically generated

Parameter *data* merupakan argument / parameter yang ditangkap, karena data masih berupa object kita harus mengambil valuenya dari target.value

### Slots

Sama seperti element html pada umumnya kitab isa menyimpan sebuah content pada custom component. Seperti contoh berikut

A screenshot of a chat

Description automatically generated

A screen shot of a computer program

Description automatically generatedBerikut adalah contoh code pengginaan slot.

## Vue Router

Untuk membuat *Router* pada Vue.js kita perlu memasang sebuah package *vue-router* supaya web applikasi kita mempunyai link navigasi dan dapat melakukan perpindahan halaman secara sederhana. Ini sangat cocok ketika kita ingin membuat *SPA (Single Page Application).*

*A screenshot of a computer

Description automatically generated*

[*https://router.vuejs.org/introduction.html*](https://router.vuejs.org/introduction.html)

### Latihan Membuat Single Page Application (SPA)

Untuk dokumentasi resmi vue dapat dilihat disini <https://router.vuejs.org/installation.html>.

Mungkin pada sebelumnya kita telah menerapkan contoh Listening Component pada kasus penambahan Bahasa Pemrograman, Kita disini akan gunakan Kembali file tersebut semjadi sebuah Single Page Application.

A screenshot of a computer program

Description automatically generated

Mungkin sebelumnya kita pisahkan code nya ke dalam sebuah variable yang berisi component, lalu kita akan mengenal dengan yang Namanya Router vue.

A computer screen shot of a program code

Description automatically generatedBerikut adalah contoh code untuk route nya

Untuk path pastikan component mana yang akan dipanggil

Dan jangan lupa untuk memanggil Object VueRouter untuk membuat route baru

Dan panggil ketika Object Vue dijalankan.

Lalu untuk tampilannya kita buat seperti ini

A screen shot of a computer

Description automatically generated

Router link itu merupakan router yang dibuat oleh vue js, sebenarnya itu sama saja melakukan tag a, namun jika kita menggunakan vue kita harus menggunakan router link dan *to* merupakan url yang mau ditujunya

Router view digunakan untuk menampilkan tampilan, karena kasus ini kitab isa melakukan penambahan data maka yang sebelumnya kita menyimpan binding di dalam component syntaxnya kita bisa pindahkan ke dalam router view.

Lalu bagaimana jika ada kasus yang sama lebih dari 1?

*Kita bahas di materi selanjutnya.*

### History Mode

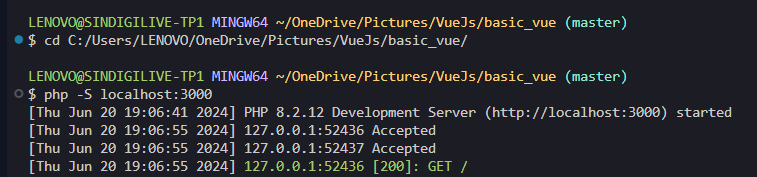
Untuk membuat sebuah route yang lebih clean tidak seperti berikut <file:///C:/Users/LENOVO/OneDrive/Pictures/VueJs/basic_vue/latihan_spa.html#/lang> kita bisa gunakan history mode di dalam pembuatan object vue, secara default vue menggunakan mode hash.

Dokumentasi

<https://v2.vuejs.org/v2/guide/migration-vue-router.html?redirect=true#history-true-replaced>

Ketika kita menggunakan fitur History maka Vue Router akan memanfaatkan history.pushState Navigasi Url tanpa memuat ulang / reload. Kita harus menjalankan sebuah server agar methode history ini dapat berjalan dengan baik.

Kita bisa gunakan sebuah server yang ada di php dengan menjalankan code berikut, pastikan kita masuk terlebih dahulu ke dalam folder applikasinya.

**

*A screenshot of a computer

Description automatically generated*Nanti kita bisa mengakses halaman seperti ini.

Perubahan code yang dilakukan hanya memberikan mode pada Vue Router

A screen shot of a computer

Description automatically generated

### Merubah Array menjadi Array of Object

Kita tidak lagi menggunakan array untuk menampilkannya kita coba dengan membuat sebuah array of object karena API juga strukturnya sama seperti ini.

A computer code with colorful text

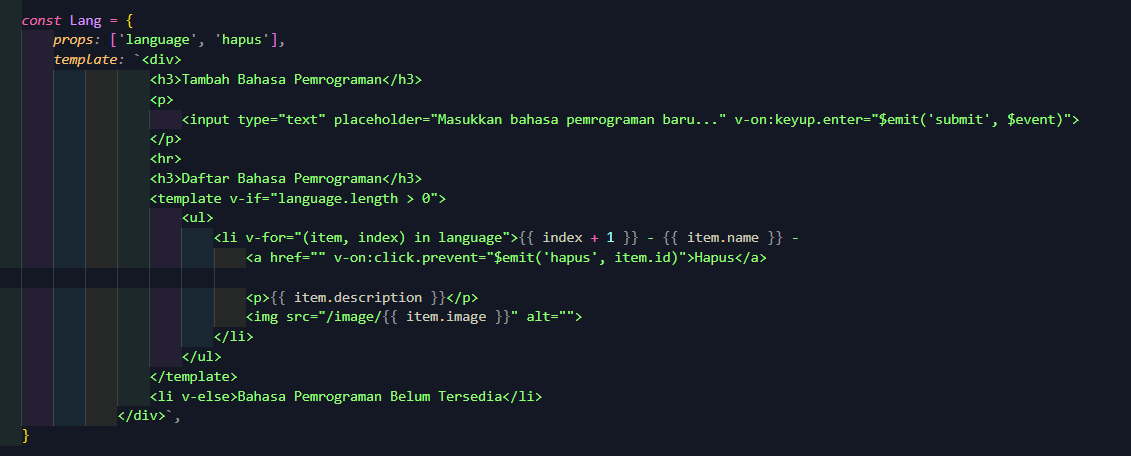
Description automatically generated with medium confidence

Nanti jika kita ubah hasilnya akan seperti ini

A close-up of a text

Description automatically generated

Kita cocokkan pemanggilan data agar lebih spesifik, seperti misalkan kita butuh nama maka cukup *item.name*. Pada contoh ini juga kita mengubah id berdasarkan item nya.



Pada method hapus, kita melakukan pengecekkan apakah id yang dihapus itu ada di dalam item.id? jika ada maka tampilkan jika tidak ada maka jangan tampilkan cara ini juga mirip seperti menghapus data.

A screen shot of a computer code

Description automatically generated

### Router

Mulai pada kali ini kita akan bahas mengenai Vue Router dengan Studi Kasus yang berbeda namun mirip dengan kasus sebelumnya. Untuk materi mengenai vue router dapat kita akses melalui dokumentasi vue aslinya [disini](https://v3.router.vuejs.org/guide/#javascript).

Berikut adalah contoh code dan penjelasannya :

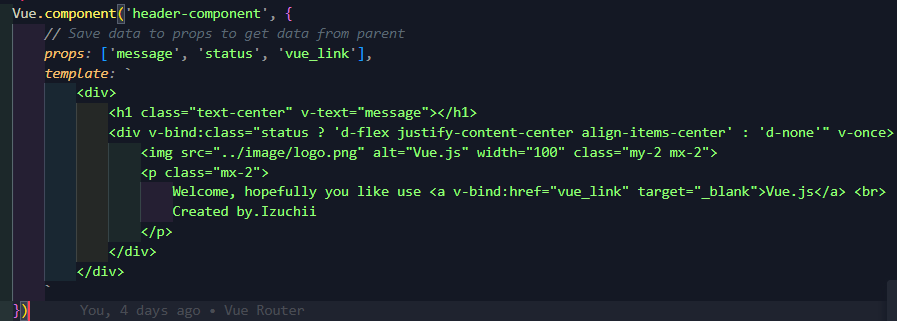
A computer screen shot of a computer program

Description automatically generatedPada kasus ini kita akan gunakan cdn Bootstrap 5 dan juga sedikit styling pada css.

*Component*

Terdapat 2 cara dalam menggunakan Component di Vue.js

Menggunakan *Vue.component*, dengan menggunakan ini kita membuat sebuah component Global, sehingga dapat digunakan di halaman mana saja tanpa perlu mengimpor atau mendaftarkan ulang.





Menggunakan *Variabel*, dengan menggunakan cara ini kita membuat sebuah component local, dengan cara ini kita mendaftarkannya secara local di instance Vue atau component lainnya, sehingga dapat membuat component lebih modular dan menghindari pencemaran namespace global.

Mudahnya perbedaan diantara keduanya dapat kita analogikan pada kasus seperti ini, misal ada sebuah aplikasi SPA (*single page application*) lalu untuk header & footer kita gunakan global Component karena component ini akan digunakan dan dipanggil Kembali, namun untuk content pasti bisa berubah sesuai dengan route yang dituju maka disini kita gunakan Local Component.

Baik kita lanjut pada code Kasus Vue Router

Pastikan kita panggil cdn vue, sebelum script vue dijalankan.

A computer screen shot of a computer screen

Description automatically generated

Berikut adalah contoh component global dan local.

A computer code with green text

Description automatically generated

*Props*, merupakan data yang ditangkap oleh component setelah datanya dikirimkan oleh parent nya.

Sama seperti Object Instance Vue Local component juga memiliki sebuah methods, pada code berikut ada method yang langsung menjalankan script code, dan ada yang melakukan *$emit*.

*\*sedikit diulas apa itu $emit, ini merupakan cara child component berkomunikasi ke parent component (Object Vue) nanti data yang emit lakukan akan dikirimkan ke parent dan diolah oleh parent component.*

A screen shot of a computer

Description automatically generated

Perhatikan pada bagian code ini kita memiliki sebuah data, di vue js penulisan data pada child component ada 2 cara:

Cara 1 : data: function() { … } = Mengembalikan object data baru setiap kali instance component dibuat.

Cara 2 : data() { … } = Cara ini lebih singkat dan ringkas untuk mendefinisikan fungsi dalam ES6.

Kenapa data langsung mengembalikan return ?

Di sini, properti di dalam objek data yang melekat pada instance Vue utama dapat diakses dari dalam komponen yang merupakan bagian dari hierarki Vue tersebut (Child).

Sedangkan data di dalam child component lebih dominan sebagai template / variabel kosong untuk menampung data, yang kemudian data tersebut dikirimkan ke parent nya. Meskipun sebenarnya data di child component juga bisa digunakan sebagai menyimpan data beserta valuenya yang dapat kita panggil.

A screenshot of a computer program

Description automatically generated

A screen shot of a computer code

Description automatically generatedperhatikan pada data disini kita anggap membuat sebuah variabel object yang berisi nilai & value di dalamnya.

Value tersebut akan terisi berdasarkan inputannya, dengan bantuan *v-model* maka data akan langsung dihubungkan dengan data tersebut.

A screen shot of a computer program

Description automatically generatedDisini kita buat sebuah validasi sederhana apabila content newBlog masih kosong maka akan diisi data Error dan informasinya.

Ketika berhasil maka akan menjalankan emit dengan mengirimkan data newBlog tersebut. Ingat newBlog yang dikirimkan isinya yaitu { title, description, image}

Setelah emit dilakukan kita bisa melakukan reset pada newBlog agar datanya Kembali kosong

Untuk image kita hanya menangkap nama image yang kita upload, & pastikan imagenya ada di local folder.



Bagaimana dengan Route ?

A computer screen shot of a program code

Description automatically generatedUntuk membuat Route kita cukup buat sebuah data array of object yang berisi path dan component yang mengarah ke child component yang akan di tampilkan.

*Terkadang kita juga menghadapi pada route not found / fallback / wildcard, di vue kita mengatasi itu dengan menggunakan tanda \* dan arahkan ke component yang menampilkan notFound.*

Selanjutnya setelah kita definisikan router kita panggil atau buat Instance dari VueRouter, mode *history* digunakan agar membuat code kita lebih clean (agar tidak Adaya # pada url) kita dapat menggunakan server untuk menjalankannya, contoh kali ini kita gunakan ***php -S localhost:3000***

A screenshot of a computer program

Description automatically generated

Berikut adalah vue parent Object / Instance Vue data mengembalikan data yang dapat ditampilkan / dikirimkan ke child component, router menjabarkan instance VueRouter.

Method tersebut akan menambah dan menghapus data blog yang didapoatkan dari child.

A computer screen shot of text

Description automatically generated

Router link itu sebuah navigasi pengganti tag *a* yang dimana kita bisa berpindah page (SPA) lalu untuk menampilkan tampilannya pastikan memanggil router-view.

v-bind:author & blogs itu artinya mendefinisikan data author dan blog ke dalam tampilan namun ini akan diambil datanya apabila ada props yang menampung.

Sedangkan v-on:add-blog itu adalah emit yang dilakukan dimana jika emit tersebut dijalankan maka akan menjalankan method parent nya.

### Local Storage

A screenshot of a computer program

Description automatically generated

Mungkin pada code sebelumnya mungkin code sepreti ini

Created adalah fungsi yang otomatis dijalankan saat vue dibuat, created pada code tersebut akan mengambil data pada local storage dengan nama ‘blog’.

JSON.parse memugkinkan kita merubah string json menjadi object javascript

{"nama": "John", "usia": 30} menjadi { nama: 'John', usia: 30 }

Dan untuk stringify melakukan kebalikannya.

Dengan melakukan local Storage memungkinkan kita menyimpan data ke dalam sebuah local di computer kita sehingga ketika computer di refresh data masih kita miliki dan tidak terhapus sepenuhnya / dideklarasikan Kembali.

Local Storage adalah fitur HTML 5 yang sudah ada di dalam web browser, fitur ini menyimpan data sementara sebelum dikirimkan ke server database. Biasanya penggunaan local storage untuk menampung isi form yang gagal disubmit pada saat internet putus misalkan sehingga user melakukan submit ulang datanya dapat kita submit Kembali melalui local Storage.

Caranya dengan menggunakan Object javascript yaitu *Local Storage* (window.localStorage).

*Created()* ini adalah metode lifecycle dari vue js fungsi ini secara otomatis dijalankan dan merupakan konsep aliran data secara keseluruhan.

### Redirect Route Vue Js

Terkadang setelah ada kasus dimana aplikasi kita melakukan redirecting pada halaman lain, kita bisa gunakan *$route*.

*$ digunakan terhadap property* ***vue instance****, beda dengan cara mengakses datanya jika di child component kita tidak perlu melakukan seperti itu. Misalkan $emit(‘nama-method’) emit itu mencoba mengakses method yang ada di vue instance namun tidak semua menggunakan $.*

A computer code with colorful text

Description automatically generated with medium confidence

A screen shot of a computer program

Description automatically generated

Dengan cara begitu $route akan meng akses Vue Instance dari Vue Router yang ada di inctance utama, apabila jika kita tidak menggunakan $router maka redirect tidak akan berjalan.

Beberapa contoh penggunaan $ dalam Vue.js:

$emit: Digunakan untuk memancarkan event dari komponen anak ke komponen induk.

$router: Mengakses instance router dalam aplikasi Vue yang menggunakan Vue Router.

$route: Mengakses informasi rute saat ini (seperti path, params, dan query).

$nextTick: Menjalankan fungsi callback setelah DOM diperbarui.

$refs: Mengakses referensi ke elemen atau komponen anak.

### Sweet Alert & Fix Delete Data

Pastikan kita sisipkan cdn dari sweet alart jika sudah maka kita akan bisa jalankan sweet alert Swal seperti berikut.

A computer screen shot of a computer code

Description automatically generated

### Lifecycle Hooks

Lifecycle hooks di Vue.js adalah metode yang dipanggil pada titik-titik tertentu dalam siklus hidup sebuah instance Vue. Mereka memberikan kesempatan untuk mengeksekusi kode pada saat instance tersebut dibuat, ditambahkan ke DOM, diperbarui, atau dihancurkan. Lifecycle hooks ini sangat berguna untuk menginisialisasi data, mengatur event listener, melakukan operasi cleanup, dan banyak lagi.

A screen shot of a computer program

Description automatically generatedA screen shot of a computer program

Description automatically generated

Berikut adalah contoh lifecycle hooks <https://vuejs.org/guide/essentials/lifecycle.html>

<https://v2.vuejs.org/v2/guide/instance#Instance-Lifecycle-Hooks>

Berikut adalah lifecycle hooks Vue.js

Ada pengkondisian dimana jika ada element elm aka akan dirender jika tidak ad akita bisa gunakan vm.$mount(el)

A black background with white text

Description automatically generated

Berikut adalah contohnya, biasanya Teknik ini digunakan ketika membuat aplikasi menggunakan vue CLI.

Terdapat before update dan updated ini akan dipanggil ketika berpidah page / merender ulang tampilan, dan tidak hanya disitu saja terkadang setelah mengisikan form itu juga akan dipanggil.

Dan bagusnya fungsi created, before create tidak akan dipanggil lagi?

### Route Params

Terkadang di dalam aplikasi kita mengarahkan ke halaman yang memerlukan parameter, misalkan pada kasus ini adalah detail blog.

Untuk membuat param kita gunakan :

*Artinya ketika mau mengarahkan ke page lain pastikan mengirimkan data melalui path url dengan nama parameter-nya.*

A blue and white text

Description automatically generated

Lalu pada component blogs kita cukup buat sebuah router-linkA computer code with green and blue text

Description automatically generated

Pastikan gunakan :to karena *:* dapat diartikan sebagai parameter yang dikirimkan ke route tersebut.

A screen shot of a computer program

Description automatically generatedSelanjutnya buat component untuk detail blog

$route bisa kita ulas Kembali pada materi redirect route vue js.

Dengan mengambil $route.params.title kita mengambil parameter title yang disimpan di url.

Kemudian ambil blogs di localstorage

Kemudian filter dengan mencari data yang judulnya sama. Jika ada maka data blog akan diisi dengan data tersebut.

## Firebase

Sebelumnya kita akan mencoba menggunakan database realtime yang disediakan oleh firebase, pastikan daftar firebase dan buat sebuah akun kemudian buat aplikasi dan pastikan aplikasi test mode terlebih dahulu, setelah selesai create di halaman dashboard pergi ke project setting dan copy kan SDK setup dan konfigurasinya (pada kasus ini kita gunakan cdn terlebih dahulu)

Kemudian buat sebuah Realtime Database dengan testmode dan buat sebuah data, contohnya sebagai berikut.

A blue screen with white text

Description automatically generated

Jika sudah maka pastikan script kita sudah ditambahkan setup konfigurasinya

A screenshot of a computer program

Description automatically generated

Tambahkan sebuah import ke firebase-database dan setelah inisialisasiApp firebase ambil database dan referensikan ke database yang barusaja dibuat.

A screen shot of a computer code

Description automatically generatedJika sudah maka ketika vue dibuat ke dalam object DOM maka jalankan fungsi yang disediakan oleh SDK firebase

*onValue(‘referensi data’, ‘result callback’, ‘error callback’)*

*val()* digunakan untuk menangkap data snapshotnya dan dikembalikan sebagai object murni.