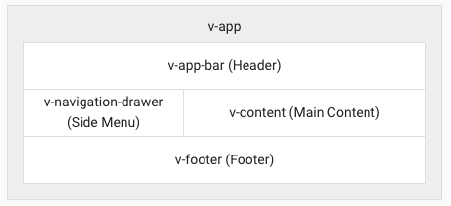
Dokumentasi Vue.js

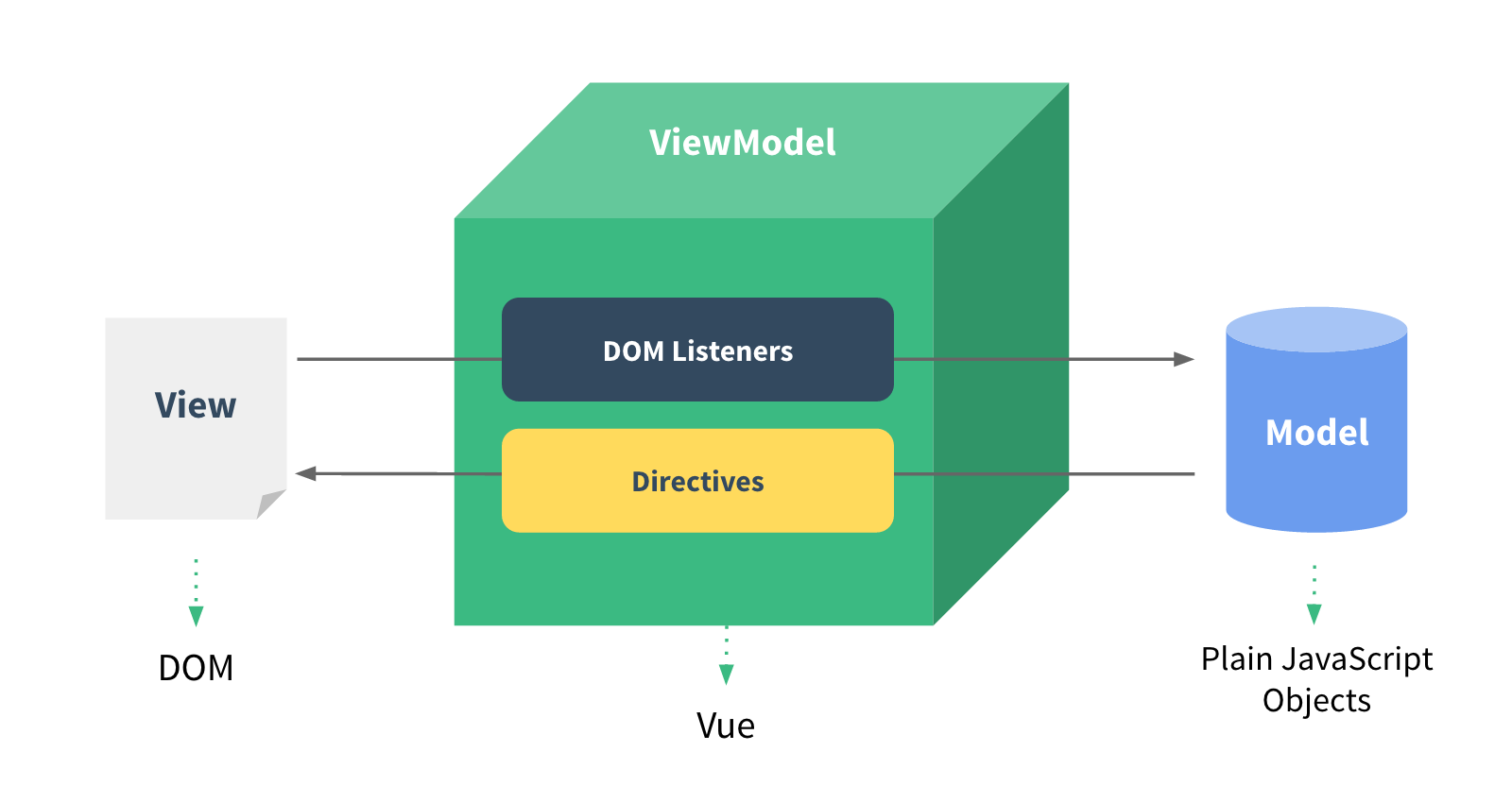
Pada web ini saya menggunakan axios(<https://github.com/axios/axios>) untuk ajaxnya sebenarnya kita dapat menggunakan fetch untuk ajaxnya akan tetapi secara umum pada vue js menggunakan plugin axios, vuetify untuk material designnya(hampir sama dengan bootstrap), vue-router(<https://router.vuejs.org/>) untuk routingnya dan vuex(<https://vuex.vuejs.org/>) untuk state-management.

Gambaran layut vuetify :



Selengkapnya tentang component vuetify bisa dilihat pada <https://vuetifyjs.com/en/getting-started/quick-start/>

* Direktori dist berisi hasil dari mode production pada vue dengan perintah npm run build.
* Direktori node\_modules berisi node-node module yang digunakan seperti halnya menggunakan webpack. konsep vue.js mirip dengan konsep webpack(module bundler) webpack, apabila kita menggunakan webpack maka code yang kita gunakan akan lebih complex karena webpack harus menginstall dan menkonfigurasi satu persatu loader, plugins, dll. Sedangkan pada vue kita dapat menggunakan vue-cli yang bisa digunakan dalam mode cli dan mode gui untuk menginstall dan langsung terkonfigurasi. Untuk pertama kali saya merekomendasikan untuk belajar tentang web-component(shadow-Dom dan component) dan modul bundler seperti webpack sebelum beranjak ke Vue.js karena dengan begitu akan lebih mudah dalam belajara framework seperti vue js atau react yang berbasis module bundler dan web-component. Pada module bundler apabila kita memakai webpack kita dapat menkonfigurasi web kita dalam model development atau dalam mode production, dengan menggunakan module web-common. Pada vue js kita tinggal membuat sebuah file .env.development dan .env.production seperti pada projek saya. Selebihnya dapat dilihat pada <https://webpack.js.org/> dan <https://vuejs.org/>
* direktori src yang berisi projek kita disaat kita dalam mode development
* didalam direktori src ini terdapat beberapa direktori seperti components, plugins, dll.
* Direktori Components berisi component-component yang digunakan pada web(tentang component bisa dibaca dibawah atau mirip dengan web-component pada javascript tanpa framework).
* Direktori plugin berisi plugin seperti vuetify dan axios.
* Direktori router berisi vue-router yang digunakan untuk routing pada vue
* direktori store yang berisi vuex yang digunakan untuk state-management pada vue file utama adalah index.js.
* Direktori View berisi file vue yang berisi halaman-halaman pada web.
* File app.vue adalah inti dari web(disini saya menggunakan SPA atau Single Page Application dan tampilan yang digunakan secara umum ini adalah pada file app.vue ini).
* File main.js adalah untuk mengimport module-module yang digunakan pada web agar bisa digunakan(sistem module).
* Service worker yang dijalankan oleh browser di background, yang terpisah dengan kode lain di halaman web browser. Dengan menggunakan service worker, kita dapat memanfaatkan resource yang telah disimpan di dalam cache untuk ditampilkan kembali bahkan dalam mode jaringan offline.
* disini saya juga menggunakan fitur Progressive Web Apps (PWA) adalah gabungan dari web apps dan mobile apps. Sebelumnya, Mobile Apps adalah aplikasi JavaScript yang bekerja di browser dan bertujuan untuk membawa fitur-fitur native app ke website.
* Contoh dari penerapan PWA adalah pada file registerServiceworker yang memungkinkan web berjalan secara offline, jadi komponent-komponent tertentu yang terdaftar pada service worker meskipun online akan tetap berjalan.
* Disini saya juga menerapkan penggunakan babel preset seperti pada penggunaan module bundler webpack yang memungkinkan javascript dan web kita tetap berjalan pada browser-browser lama. Karena browser lama belum mendukung javascrip ES6



Vue fokus pada layer ViewModel dari pattern MVVM. ViewModel merupakan penghubung antara View(DOM/tampilan) dan Model(objek data), manipulasi pada View dipantau oleh DOM listeners kemudian diterima oleh ViewModel untuk digunakan mengupdate Model, demikian juga perubahan data pada model akan diteruskan oleh ViewModel ke View melalui Directives. Pada Vue, obejak Vue inilah yang berperan menjadi ViewModel pada konsep MVVM. pustaka vue terbagi menjadi 2 yaitu mode development dan mode production sangat disarankan menggunakan mode development saat mengembangan aplikasi menggunakan VUE sebab semua informasi umum(warning) jika terjadi kesalahan kode akan dimunculkan. <https://vuejs.org/js/vue.js> versi development <https://vuejs.org/js/vue.min.js> versi production.

LifeCycle Objek Vue

terdapat 8 hooks pada siklus dibawah dan 8 siklus ini diklompokkan menjadi 4 bagian:

Create:

1. beforeCreate yaitu hook sesaat setelah objek Vue dibuat dan komponent diinisialiasi, properti data belum dapat diakses atau digunakan pada hook ini.

2. created yaitu hook ketika objek Vue telah selesai diciptakan. Hook ini bersifat reactivity pada properti data juga sudah didefinisikan sehingga kita sudah diizinkan untuk mengakses dan memanipulasi data.

Mount:

1. beforeMount yaitu hook ketika template dicompile.

2. mounted yaitu hook ketika elemen telah diinisialisasi, data telah dimuat dan view telah dirender.

Update:

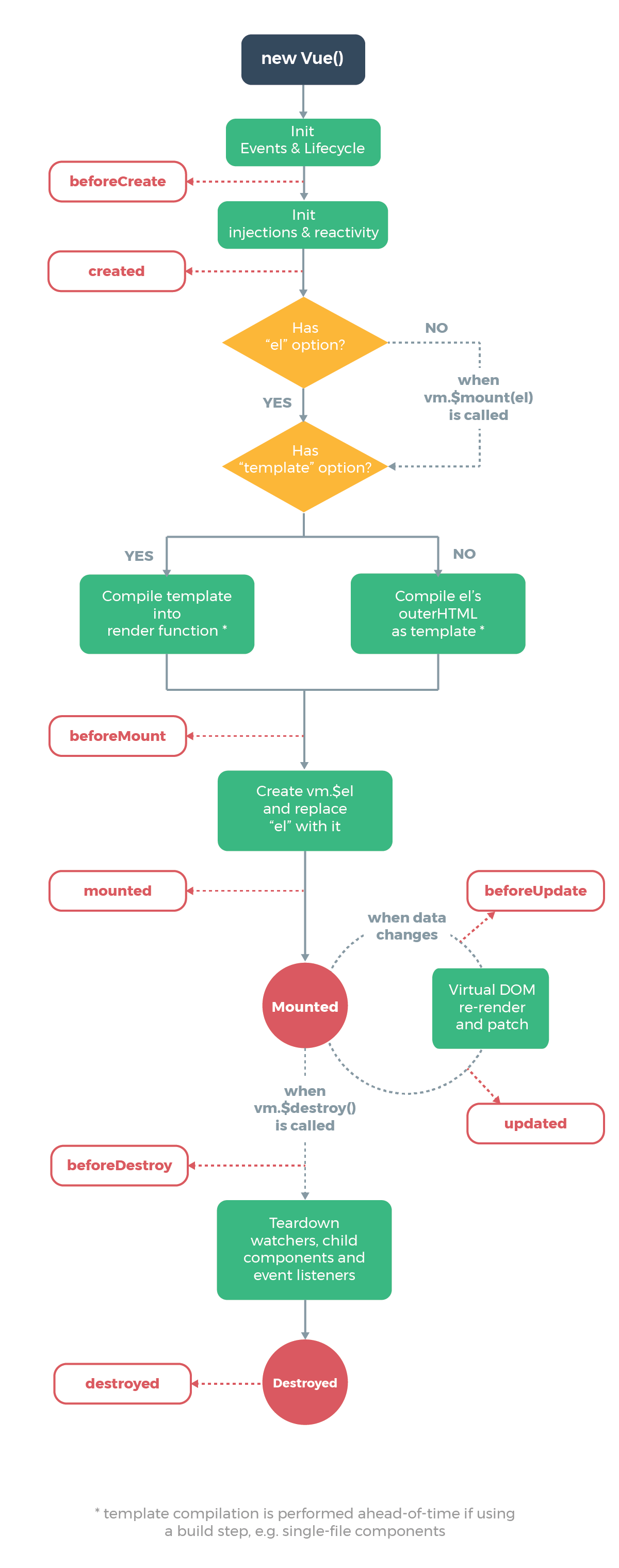
1.beforeUpdate yaitu hook yang terjadi setelah mounted dan hanya terjeadi jika ada perubahan data yang mengakibatkan render ulang. Tepatnya terjadi saat sebelum view dirender ulang.

2. updated yaitu hook yang terjadi setelah beforeUpdate atau view sudah dirender ulang.

Destroy:

1.beforeDestroy yaitu hook yang terjadi sebelum component dihapus.

2.destroyed yaitu hook yang terjadi setelah objek vue dihapus.



Penulisan Template

Pada Vue, template merupakan kode yang menjadi dasar dari suatu tampilan, umunya kode template ditulis dengan menguunakan bahasa HTML, variable pada template ditulis dengan menggunakan tanda kurung kurawal. Template dari variable(data) dicompile oleh Vue menjadi Virtual DOM sebelum akhirnya dirender atau ditampilkan dalam bentuk HTML DOM. Ketika terjadi perubahan data maka akan memicu render ulang dari DOM. Bentuk-bentuk data terkait hubungan dengan template ada beberapa macam, dan hal itu menuntu perlakuan yang berbeda.

Properi Method

Properti methods dapat berisi fungsi-fungs js yang dapat dipangil disemua tempat pada aplikasi berbasis Vue. Jika ada action atau event yang memanggil suatu funsi, maka fungsi tersebut cocok dikategorikan sebagai methods.

Properti Computed

Properti computed berisi fungsi-fungsi yang nilainya akan senantiasa dievaluasi ketika terjadi perubahan variable data yang menjadi dependensinya. Fungsi pada computed umumnya mengembalikan nilai / return value.

Properti Filters

Properti filters yang dapat berisi fungsi yang digunakan memanipulasi tampilan atau format teks pada template. Filters ditulis dengan menggunakan simbol | atau pipe.

Directive

directive merupakan atribut khusus yang disematkan pada element atau markup HTML sebagai penanda bahwa elemen DOM tersebut akan dikenai perlakuan tertentu oleh Vue. Directive berbentuk ekspresi javascript yang secara reaktif menerapkan efek tertentu ke elemen DOM ketika nilai ekspresinya berubah. Directive diawali dengan prefix v-. contoh v-for, v-if,dll.

V-if

v-if merupakan directive yang digunakan untuk merender atau tidak merender suatu element DOM (conditional rendering).

V-on

merupakan directive yang berperan sebagai sebuah event listener pada element HTML/komponent Vue. Directive ini bertugas memantau aktifitas(aksi) yang dilakukan terhadap suatu elemen HTML/komponen Vue, directive v-on dapat disingkat menjadi @

v-bind

directive ini berfungsi untuk membinding atribut html atau komponen agar nilainya terupdate secara reactive sesuai dengan datanya. Directive v-bind dapat disingkat dengan titik dua :.

v-for

directive v-for yang berfungsi untuk melakukan perulangan sebanyak elemen data yang ada pada variable array ataupun objek dan kita dapat menambahkan argumen seperti index pada array yang digunakan untuk menampilkan key dari array suatu data. Dan apabila pada objek ada value dan key.

V-model

digunakan untuk menangani two way data binding.

Perubahan atau Mutation data pada array dan objek

mutation data pada array pada vue js mengikuti kaidah javascript yaitu dengan menggunakan fungsi-fungsi yang tersedia di javascript, sehingga kita tidak bisa langsung mengeset data array secara langsung menggunakan indexnya. Jika kita mengubah data tidak sesuai kaidah javascript maka akan terjadi error. Javascript memanipulasi data bentuk array menggunakan fungsi-fungsi bulit-in seperti push(), pop(), dll. Dan kita dapat mutation data array ataupun objek dengan menggunakan fungsi built-in set.

Validation data

proses validasi adalah proses memastikan setiap isian yang diinput oleh user melalui form tersebut sesuai dengan persyaratan minimal yang kita tentukan. Kita bisa juga menambahkan method setFokus pada input yang belum memenuhi syarat. Sehingga akan memudahkan dari sisi user. dengan menambahkan directive ref sebagai penanda unik pada field input. Kita bisa menggunakan objek FormData(https://developer.mozilla.org/en-US/docs/Web/API/FormData/FormData) untuk mempacking data hasil isian form menjadi sebuah objek.

Component pada Vue js

component merupakan sub class/objek dari Vue yang bisa kita buat untuk berbagai tujuan misalnya memca kompleksitas kode, reusabilitas kode, dan modularitas. Setiap component memiliki lifecycle yang sama dengan objek Vuenya. Untuk membuat sebuah component baru, cukup dengan kode menjalankan method Vue.component(‘nama-component’, {option}).

Mixin

mixin merupakan cara pada Vue untuk mendefinisikan suatu kumpulan fungsi atau option yang akan digunakan pada aplikasi atau component tertentu. Ketika objek Vue atau component menggunakan mixins maka semua option dari mixin tersebut akan digabungkan kedalam component yang menggunakannya tersebut.

Plugins

Plugins digunakan sebagai wrapper untuk menambahkan atau mendaftarkan suatu fitur global pada Vue, misalnya plugin untuk menambahkan

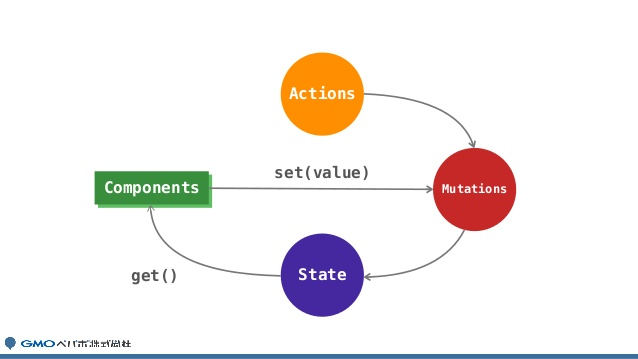
Routing

Vue secara official membuat pustaka bernama Vue Router yang bisa diintegrasikan secara baik tentunya dengan Vue untuk membuat Single Page Applications. [https://router.vuejs.org](https://router.vuejs.org/).

Untuk bekerja dengan Vue Router, ada dua component penting yang akan kita gunakan yaitu router-linkdan router-view. router-link berfungsi menggenerate menu link, sedangkan router-view berfungsi sebagai tempat penampung component yang ditampilkan. Disamping kita kita perlu mendeklarasikan class VueRouter dan mendefiniskan routing aplikasi kita.

State-Managemen

state management merupakan sentralisasi variable data sehingga semua component dalam aplikasi dapat mengakses dan memanipulasinya dengan aturan-aturan tertentu sehingga perubahannya dapat diprediksi. Pustaka state-management yang dibuat oleh vue adalah Vuex, ide dasar dari Vuex terinspirasi oleh arsitektur flux, Redux dan elm. Pustaka Vuex juga terintegrasi dengan Vue devtools extension sehinga memudahkan kita melakukan debugging state.



Pustaka vuex menangani dan terdiri dari 3 hal utama yaitu state, mutation, dan action. Cara kerjanya adalah : awalnya component melakukan request kepada suatu fungsi pada action, kemudian action yang berisi kumpulan fungsi tersebut bertugas memanggil fungsi pada mutation, fungsi-fungsi pada mutation bertugas mengupdate state. Perubahan pada state yang bersifat reactive akan memicu rendering component. State atau data pada vuex disimpan dalam sebuah objek yang disebut dengan store.

Getters

vuex store memiliki properti getters sehingga pemanggilan state tidak langsung ke statenya, melainkan melalui perantara fungsi getters distore tersebut.

Mutation

mutation merupakan kumpulan fungsi untuk memanipulasi state atau bisa juga disebut sebagai setters. Tiap fungsi mutations memiliki minimal 2 hal yaitu type dan handler, dimana type merupakan nama fungsinya dan handler merupakan state.

Action

Vuex juga memiliki properties action. Action sebenernya mirip dengan mutation, namun perbedaannya adalah actions bertugas mengcommit mutation dan action mendukung operasi asynchronous.