Implementasi Vue.Js Pada Sistem Direktorat Kerja Sama UK Maranatha

Kasyfi Aghitya[#1], Maresha Caroline Wijanto[#2]

[#]*Program Studi S1 Teknik Informatika, Universitas Kristen Maranatha*
Jl. Prof. drg. Surya Sumantri No. 65 Bandung
[1]kasyfi997@gmail.com
[2]maresha.cw@it.maranatha.edu

Abstract — **In this research, we made a dashboard website for the Directorate of Cooperation Maranatha Christian University. This website is using the VueJs for programming language of front-end part. This research will focus more on the interface, all the data will be obtained from the API of the back-end part. HTTP is used to make a request call the data from backend using Axios because it is easier to use. After the data is called it will be displayed on a certain page in the table and the data can be managed by the admin. Main features for this website is Dashboard. Dashboard shows only important information needed in a form of bar chart, pie chart, or others. This information can help the management to see the update of their business in a glance. Dashboard will show the number of active cooperation per country in bar chart, the number of active cooperation per faculty in bar chart, and list of cooperation that will expired in the next 30 days.**

Keywords— **API, Chart, Dashboard, Front-End, VueJS.**

I. PENDAHULUAN

Teknologi sudah berkembang pesat dan sangat dibutuhkan oleh orang banyak. Banyak teknologi modern yang sudah digunakan saat ini salah satunya adalah *website*. *Website* berperan penting dalam teknologi sekarang. Selain dapat menyimpan informasi-informasi penting, *website* juga dapat menjadi sarana komunikasi dalam suatu komunikasi atau Kerja sama. *Website* sudah menjadi hal yang lumrah untuk organisasi besar atau yang memiliki data banyak. Salah satu contoh *website* yang digunakan oleh organisasi adalah *website* Direktorat Kerja Sama (DKS) Universitas Kristen Maranatha.

DKS Maranatha adalah salah satu organisasi resmi di UK Maranatha yang bekerja dalam mengatur semua Kerja sama dalam bentuk apapun. Dikarenakan jumlah kerja sama yang banyak maka dibutuhkan suatu sistem untuk membantu mengelola hal tersebut. Dengan adanya *website*, pengolahan data kerja sama yang ada di DKS menjadi lebih rapi dan terstruktur. Hanya saja, fitur dan *User Interface* (UI) dari *website* tersebut masih terbilang sederhana.

Maka oleh sebab itu, diperlukan adanya peningkatan pada *website* DKS. Peningkatan ini khususnya agar lebih mudah dalam mencari data dan perbaikan dari segi tampilan. Penelitian ini merupakan sub proyek dari pengembangan *website* DKS UK Maranatha. Fokus penelitian ini adalah pengembangan bagian *front-end*, terkait UI (User Interface) dan UX (User Experience) dan tidak membahas pengaturan dalam bagian *back-end*. Untuk bagian *backend* dikerjakan oleh Joses Vito Chandra dari Fakultas IT Universitas Kristen Maranatha. Selain perbaikan tersebut, diperlukan juga fitur tambahan untuk melihat secara grafik kerja sama yang sudah terjadi dan sebuah *reminder* untuk kerja sama yang sedang terlaksana. Hal ini diperlukan karena pihak DKS terkadang telat melakukan perpanjangan perpanjian. Selain itu adanya kesulitan untuk melihat keseluruhan data kerja sama yang ada.

II. KAJIAN TEORI

A. Node JS dan Vue JS

Node.js adalah perangkat lunak yang didesain untuk mengembangkan aplikasi berbasis web dan ditulis dalam sintaks bahasa pemrograman JavaScript [2]. Node.js melengkapi peran JavaScript sehingga bisa juga berlaku sebagai bahasa pemrograman yang berjalan di sisi server, seperti halnya PHP, Ruby, Perl, dan sebagainya.

Vue.js adalah sebuah *library* Javascript yang dapat digunakan untuk membangun UI atau desain antar muka dalam sebuah *website* interaktif [5]. Vue.js dapat diimplementasikan dan diintegrasikan dengan *framework/library* lain yang sudah ada dengan contoh Laravel. Vue.js juga mengandung unsur-unsur dasar HTML,CSS, dan Javascript. Tersedia juga *library* tambahan Vue-ChartJS untuk mengembangkan *chart* pada VueJS dengan lebih banyak pilihan *chart* [4].

B. Dashboard

Dashboard adalah jenis antarmuka pengguna yang membantu dalam mengolah data dalam jumlah yang banyak. Dashboard juga dapat disesuaikan untuk memenuhi kebutuhan spesifik departemen dan perusahaan [9]. Hanya informasi penting saja yang ditampilkan pada Dashboard sehingga pihak manajemen perusahaan dapat langsung memahami dalam sekali lihat [10]. Pada bagian *back-end*, Dashboard dapat terhubung ke *file*, layanan, atau API lainnya. Di bagian *front-end*, Dashboard menampilkan semua data dalam bentuk tabel, diagram garis, diagram batang, atau model *chart* lainnya.

C. Web API

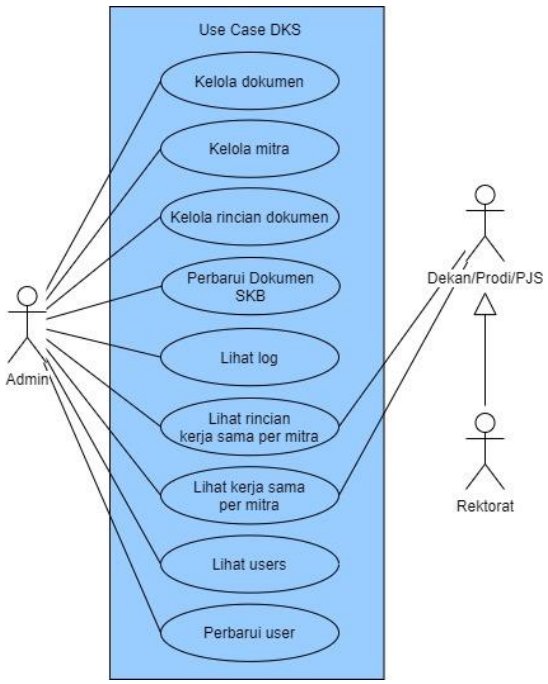
API adalah singkatan dari *Application Programming Interface*. Dengan adanya API, *developer* lebih mudah untuk mengintregrasikan dua bagian dari aplikasi [11]. API terdiri dari berbagai elemen seperti *function*, *protocols*, dan *tools* lainnya yang bisa digunakan *developer* untuk membantu dalam pengembangan sistem. Tujuan penggunaan API adalah untuk mempercepat *development* dengan menyediakan *function* secara terpisah sehingga *developer* tidak perlu membuat fitur serupa secara berulang.

III. ANALISIS DAN RANCANGAN SISTEM

A. Use Case Diagram

Usulan sistem baru akan mempunyai fitur utama yaitu untuk mengelola data Perjanjian dan Dokumen Kerja Sama. Terdapat juga fitur untuk mengelola data Mitra yang akan melakukan Perjanjian Kerja Sama dengan Universitas Kristen Maranatha. Use case diagram dapat dilihat pada Gambar 1.

Pada sistem tersebut hanya terdapat 1 aktor utama yaitu admin dari DKS UKM. Fitur-fitur yang dapat dilakukan oleh admin DKS UKM antara lain login, *logout*, kelola data Kerja sama yang memiliki *chart* dan *reminder*, kelola data rincian dokumen, dan kelola data mitra. Semua proses kelola memiliki fitur tambah, perbarui, hapus, dan lihat data. Terdapat juga aktor lain dari Rektorat atau PJS yang hanya dapat melihat daftar kerja sama per mitra dan rinciannya.



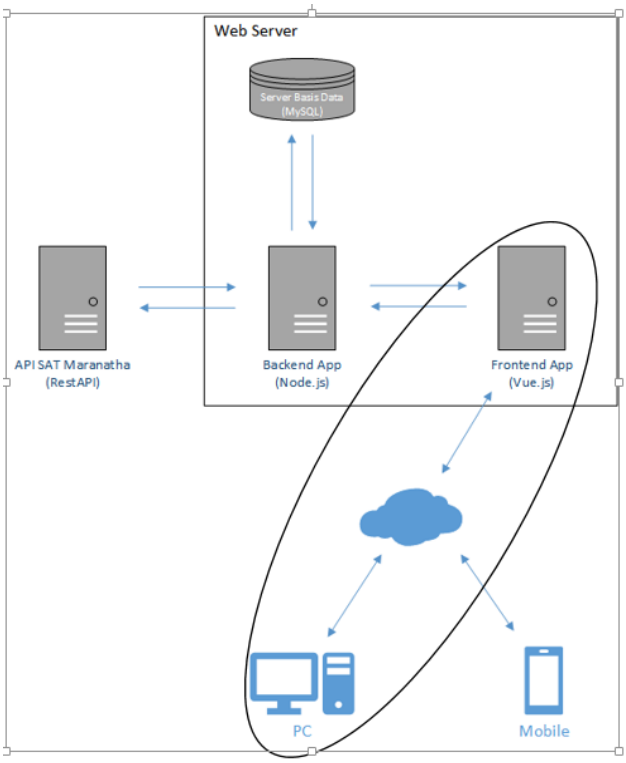
Gambar 1 Use Case Diagram

B. Arsitektur dan Lingkup Proyek

Gambar 2 menunjukkan bahwa ruang lingkup pengerjaan proyek ini adalah hanya pada bagian *front-end* saja. Pada bagian *front-end* mengerjakan dahulu bagian *website* utama saja, sedangkan untuk *mobile* akan menjadi saran pengembangan ke depannya. Untuk bagian *back-end* dikerjakan oleh mahasiswa lain dan API yang dihasilkan akan digunakan di bagian *front-end* ini.

C. Activity Diagram

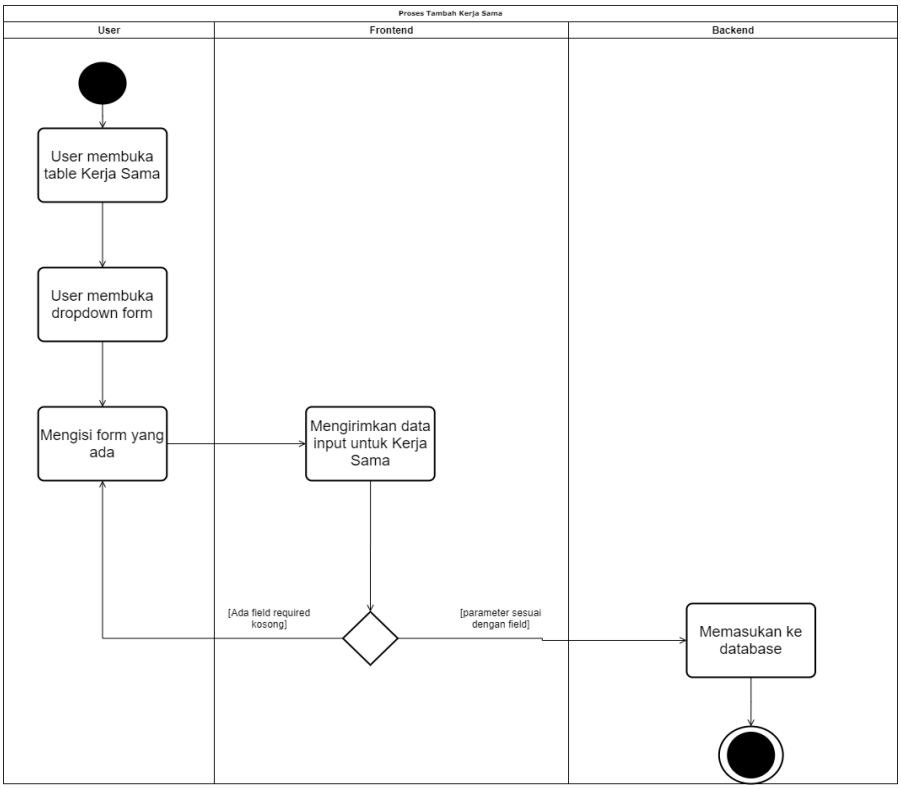
Secara umum, proses kelola itu sama. Sebagai contoh akan ditampilkan activity diagram tambah dan ubah Kerja sama.



Gambar 2 Arsitektur dan Ruang Lingkung Proyek

1) Activity Tambah Kerja sama

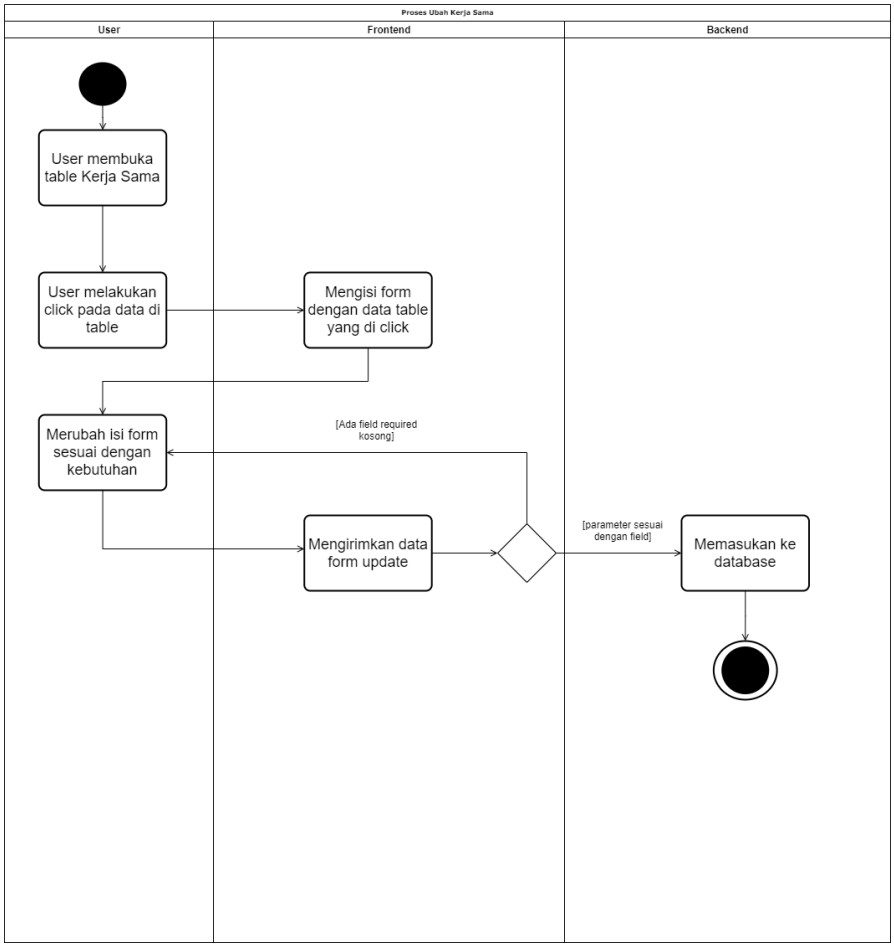
Proses Tambah Kerja sama dapat dilihat pada Gambar 3. Pada bagian atas *Table* akan ada tombol tambah yang berisikan *form* kebutuhan data. Apabila persyaratan tidak terpenuhi maka akan muncul *alert* tambah Kerja sama gagal.



Gambar 3 Activity Diagram Tambah Data Kerja Sama

2) Activity Ubah Kerja sama

Proses ubah kerja sama dapat dilihat pada Gambar 4. Prosesnya mirip dengan proses tambah, hanya saja pengguna perlu memilih data yang akan diubah, dan data awal akan diisikan secara otomatis pada *form* yang tersedia. Pengguna ubah dengan data yang baru lalu simpan lagi.



Gambar 4 Activity Diagram Ubah Data Kerja Sama

IV. IMPLEMENTASI

Sesudah pengguna berhasil melakukan *login, website* akan menampilkan halaman Dashboard seperti pada Gambar 5.

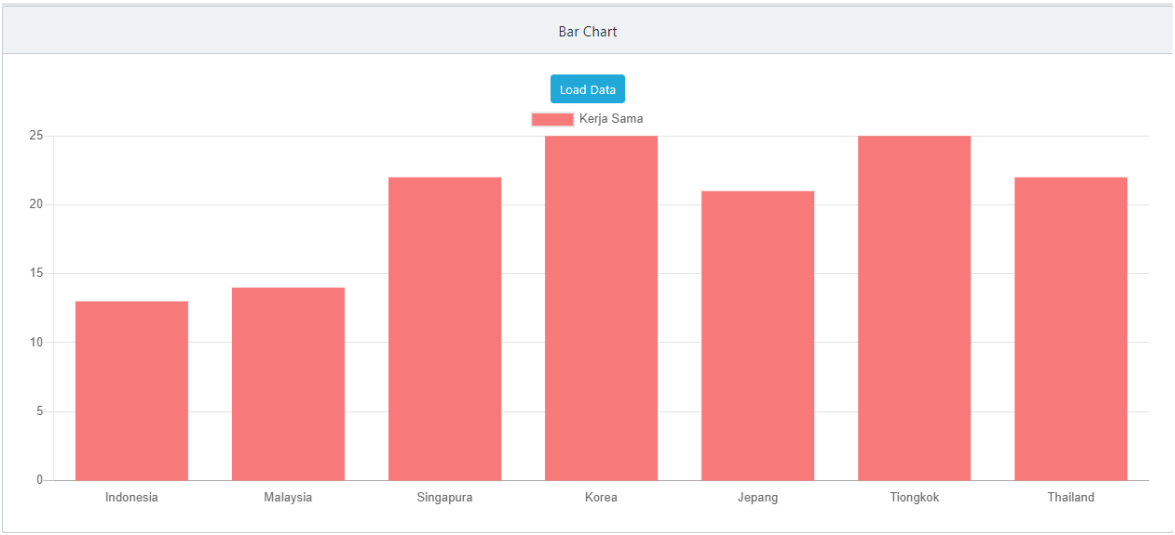
Gambar 5 Contoh Halaman Dashboard

Dashboard menampilkan beberapa data, antara lain: jumlah kerja sama aktif per negara dalam bentuk diagram batang, jumlah kerja sama aktif per fakultas dalam bentuk diagram batang, dan daftar kerja sama yang akan habis masa berlaku dalam waktu 30 hari ke depan dalam bentuk tabel. Untuk memanggil data isian chart menggunakan kode *async function* seperti terlihat di Kode Program 1.

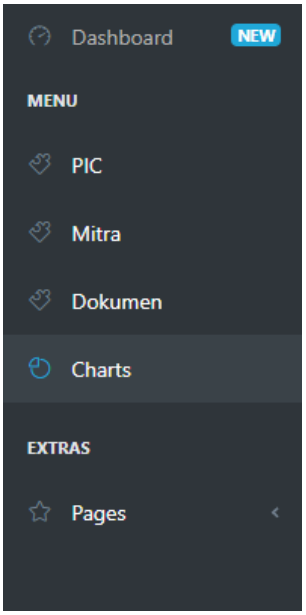
```
async mounted () {
    try {
      token = localStorage.getItem('tokena')
      var Getmitra  = await axios.post('http://gbi.sytes.net:3000/mitra',{token: token, _method: "GET"}).catch(error => console.log('Ada Error') )
      var temp = []
      Object.values(Getmitra.data.values).forEach((entry) => {
        temp.push(entry)
      })

      var obj = JSON.stringify(temp)
      var stringify = JSON.parse(obj);
      for (var i = 0; i < stringify.length; i++)
      {
        console.log(stringify[i]['nama_negara']);
        temp2.push(stringify[i]['nama_negara'])
      }
      jumlah = temp2.length
    } catch (e) {
      console.error(e)
    }
 }
}
```

Kode Program 1 Pemanggilan Data Chart

Contoh tampilan diagram batang untuk jumlah kerja sama aktif per negara dapat dilihat pada Gambar 6.



Gambar 6 Contoh Chart pada Dashboard



Gambar 7 Navigasi Menu pada Website

Untuk navigasi atau membuka menu yang lain ada di bagian kiri dari *website* (Gambar 7) dapat memanfaatkan *slide bar* sehingga terkesan dinamis. Untuk pada tampilan awal hanya menggunakan <template> seperti terlihat pada Kode Program 2 dan memanggil sisanya dari *component* yang sudah dibuat.

```
<template>
  <div class="app">
    <DefaultHeader/>
    <div class="app-body">
      <AppSidebar fixed>
        <SidebarHeader/>
        <SidebarForm/>
        <SidebarNav :navItems="nav"></SidebarNav>
        <SidebarFooter/>
        <SidebarMinimizer/>
      </AppSidebar>
      <main class="main">
        <Breadcrumb :list="list"/>
        <div class="container-fluid">
          <router-view></router-view>
        </div>
      </main>
      <AppAside fixed>
        <!--aside-->
        <DefaultAside/>
      </AppAside>
    </div>
    <DefaultFooter/>
  </div>
</template>
```

Kode Program 2 Template Navigasi Utama

Kode Program 3 menampilkan kode *export* data di VueJS yang akan dimasukkan ke dalam tag <template> tampilan.

```
<script>
import nav from '@/_nav'
```

```
3.  import { Sidebar as AppSidebar, SidebarFooter, SidebarForm, SidebarHeader, SidebarMinimizer, Side
    barNav, Aside as AppAside, Breadcrumb } from '@coreui/vue'
4.  import DefaultAside from './DefaultAside'
5.  import DefaultHeaderDropdownAccnt from './DefaultHeaderDropdownAccnt'
6.  import DefaultHeader from './DefaultHeader'
7.  import DefaultFooter from './DefaultFooter'
8.  export default {
9.    name: 'DefaultContainer',
10.   components: {
11.     AppSidebar,
12.     AppAside,
13.     Breadcrumb,
14.     DefaultAside,
15.     DefaultHeaderDropdownAccnt,
16.     SidebarForm,
17.     SidebarFooter,
18.     SidebarHeader,
19.     SidebarNav,
20.     SidebarMinimizer,
21.     DefaultFooter,
22.     DefaultHeader
23.   },
24.   data () {
25.     return {
26.       nav: nav.items
27.     }
28.   },
29.   computed: {
30.     name () {
31.       return this.$route.name
32.     },
33.     list () {
34.       return this.$route.matched.filter((route) => route.name || route.meta.label )
35.     }
36.   }
37. }
38. </script>
```

Kode Program 3 Export Vue.JS untuk Navigasi Utama

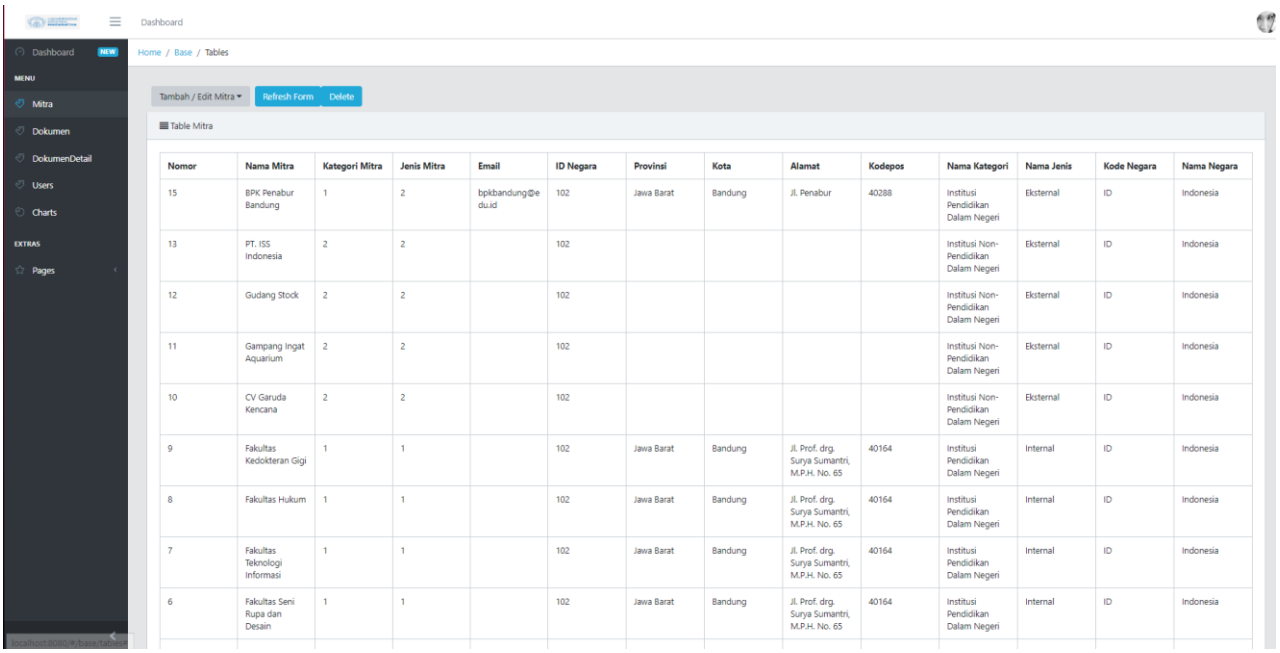
Component adalah file VueJs yang berisikan kode program berupa halaman *website* sehingga dapat dipanggil dari template UI utama. Untuk menu sidebar seperti pada Gambar 7 dipanggil oleh nav: nav.items dimana nav bar tersebut memanggil isi dari file _nav.js yang berisikan seperti pada Kode Program 4.

```
1.      export default {
2.        items: [
3.          {
4.            name: 'Dashboard',
5.            url: '/dashboard',
6.            icon: 'icon-speedometer',
7.            badge: {
8.              variant: 'primary',
9.              text: 'NEW'
10.           },attributes: { disabled: true },
11.         },
12.         {
13.           title: true,
14.           name: 'Menu',
15.           class: '',
16.           wrapper: {
```

```
            element: '',
            attributes: {}
          }
        },
        {
          name: 'PIC',
          url: '/base/tablepic',
          icon: 'icon-puzzle'
        },
        {
          name: 'Mitra',
          url: '/base/tables',
          icon: 'icon-puzzle'
        },
        {
          name: 'Dokumen',
          url: '/base/tabledokumen',
          icon: 'icon-puzzle'
        },
        {
          name: 'Charts',
          url: '/charts',
          icon: 'icon-pie-chart'
        },
        {
          divider: true
        },
        {
          title: true,
          name: 'Extras'
        },
        {
          name: 'Pages',
          url: '/pages',
          icon: 'icon-star',
          children: [
            {
              name: 'Login',
              url: '/pages/login',
              icon: 'icon-star'
            },
          ]
        },
      ]
    }
```

Kode Program 4 Kode Component nav.js

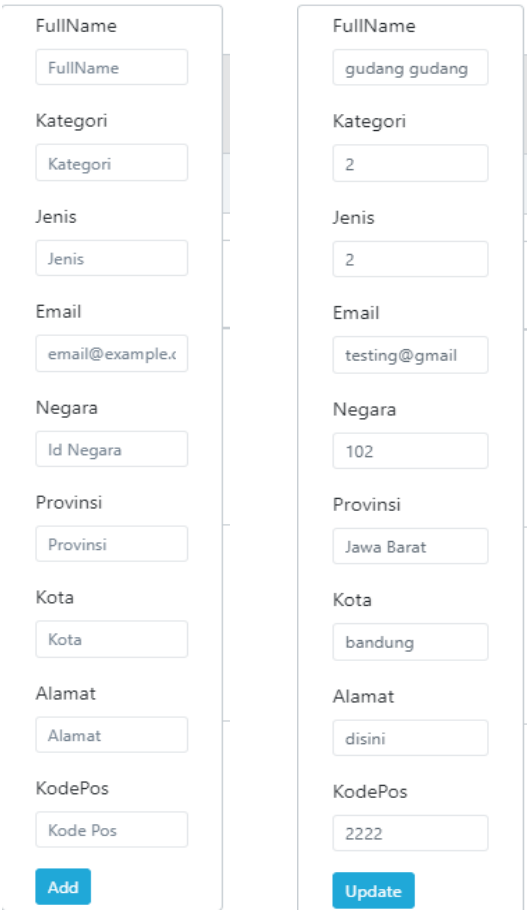
Pengguna dapat melakukan kelola data dengan memanfaatkan *table* yaitu untuk data Mitra, Rincian Dokumen, dan Kerja sama seperti yang dijelaskan pada Use Case Diagram. Sebagai contoh, tampilan fitur lihat data mitra dapat dilihat pada Gambar 8. Menu tambah dan ubah data dapat diakses melalui tombol Tambah/Edit di bagian kiri atas *table*.



Nomor	Nama Mitra	Kategori Mitra	Jenis Mitra	Email	ID Negara	Provinsi	Kota	Alamat	Kodepos	Nama Kategori	Nama Jenis	Kode Negara	Nama Negara
15	BPK Penabur Bandung	1	2	bpkbandung@edu.id	102	Jawa Barat	Bandung	Jl. Penabur	40288	Institusi Pendidikan Dalam Negeri	Eksternal	ID	Indonesia
13	PT. ISS Indonesia	2	2		102					Institusi Non-Pendidikan Dalam Negeri	Eksternal	ID	Indonesia
12	Gudang Stock	2	2		102					Institusi Non-Pendidikan Dalam Negeri	Eksternal	ID	Indonesia
11	Gampang Ingat Aquarium	2	2		102					Institusi Non-Pendidikan Dalam Negeri	Eksternal	ID	Indonesia
10	CV Garuda Kencana	2	2		102					Institusi Non-Pendidikan Dalam Negeri	Eksternal	ID	Indonesia
9	Fakultas Kedokteran Gigi	1	1		102	Jawa Barat	Bandung	Jl. Prof. drg. Surya Sumantri, M.P.H. No. 65	40164	Institusi Pendidikan Dalam Negeri	Internal	ID	Indonesia
8	Fakultas Hukum	1	1		102	Jawa Barat	Bandung	Jl. Prof. drg. Surya Sumantri, M.P.H. No. 65	40164	Institusi Pendidikan Dalam Negeri	Internal	ID	Indonesia
7	Fakultas Teknologi Informasi	1	1		102	Jawa Barat	Bandung	Jl. Prof. drg. Surya Sumantri, M.P.H. No. 65	40164	Institusi Pendidikan Dalam Negeri	Internal	ID	Indonesia
6	Fakultas Seni Rupa dan Desain	1	1		102	Jawa Barat	Bandung	Jl. Prof. drg. Surya Sumantri, M.P.H. No. 65	40164	Institusi Pendidikan Dalam Negeri	Internal	ID	Indonesia

Gambar 8 Table Lihat Data Mitra

Secara *default*, sesudah menekan tombol Tambah/Edit Data, akan menampilkan *dropdown* berisi *form* untuk *add* data seperti terlihat pada Gambar 9-a.

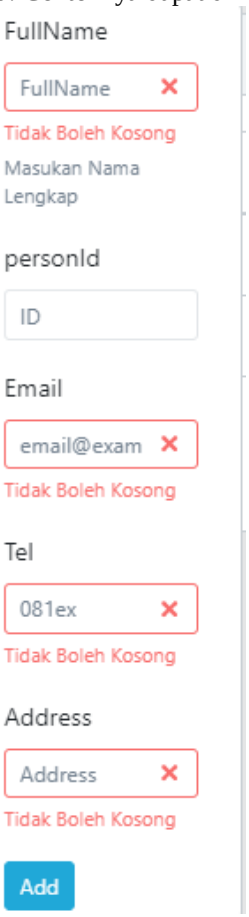


Gambar 9 Form Tambah Data (a) dan Form Ubah Data (b)

Pengguna juga dapat melakukan proses ubah di halaman yang sama. Caranya dengan menekan baris data yang akan diubah pada *table* seperti Gambar 10. Lalu data asal akan terisi langsung di *form* yang sama tetapi *button* sudah berubah menjadi Update seperti pada Gambar 9-b.

Gambar 10 Data Row Selected

Baik fitur tambah maupun ubah data, memiliki fungsi validasi *form*. Apabila ada data yang kosong atau isi tidak sesuai maka akan muncul pesan *error* langsung di *form*-nya. Contohnya dapat dilihat pada Gambar 11.



Gambar 11 Form dengan Validasi

V. PENGUJIAN

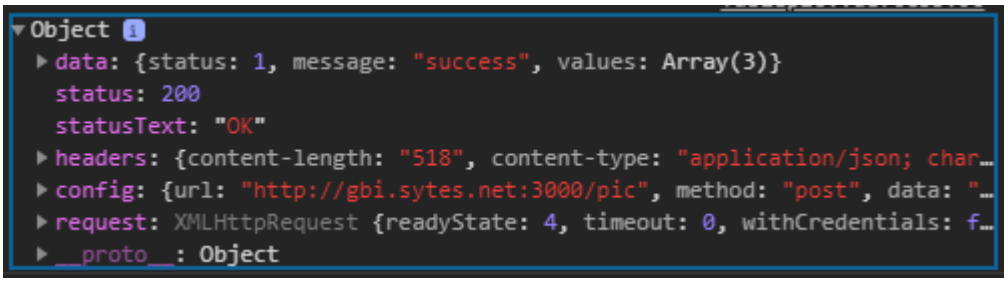
Pengujian dilakukan secara *black box* dan pengujian dari segi API yang digunakan. Pada TABEL I menampilkan hasil pengujian dengan *black box*.

TABEL I
HASIL PENGUJIAN BLACK BOX

No.	*Test Case*	Hasil Harapan	Hasil Keluaran	Hasil Uji
1	Melakukan proses *Login*	Berhasil *Login*	Berhasil *Login*	Valid
2	Melakukan proses *add Dokumen Detail* oleh admin	Berhasil *add Dokumen Detail*	Berhasil *add Dokumen Detail*	Valid
3	Melakukan proses *update*	Berhasil *update*	Berhasil *update Dokumen*	Valid

No.	*Test Case*	Hasil Harapan	Hasil Keluaran	Hasil Uji
	Dokumen Detail oleh admin	*Dokumen Detail*	*Detail*	
4	Melakukan proses delete *Dokumen Detail* oleh admin	Berhasil *delete Dokumen Detail*	Berhasil *delete Dokumen Detail*	Valid
5	Melakukan proses *Download Dokumen* oleh admin	Berhasil *Download Dokumen*	Berhasil *download* Dokumen	Valid
6	Menampilkan chart	*Chart* berhasil ditampilkan	*Chart* berhasil di tampilkan	Valid
7	Menampilkan data API	Data API Tampil di dalam *table*	Data Berhasil ditampilkan	Valid
8	Melakukan proses view data oleh admin	Berhasil lihat view data	Berhasil lihat view data	Valid
9	Melakukan *Logout*	Berhasil *logout*	Berhasil *logout*	Valid

Pada pengujian API, sistem akan mengambil data kepada *back-end* lalu menampilkannya di *table* dan dilihat apakah kode akan berjalan dengan baik atau tidak. Dengan Axios post dan menggunakan method GET maka sistem mengirimkan data token sebagai parameter untuk mendapatkan data yang dibutuhkan seperti pada Kode Program 5.



Kode Program 5 Data API

Data tersebut dimasukan ke dalam *array* lalu ditampilkan pada *table* di setiap halamannya masing-masing.

VI. KESIMPULAN DAN SARAN

Pembuatan *website* dengan memanfaatkan VueJS dapat menampilkan *user interface* yang lebih sederhana untuk dibuat dan lebih menarik untuk digunakan. Dengan VueJS juga memudahkan dalam melakukan berbagai macam kelola data untuk dokumen yaitu dalam pembuatan kerja sama ataupun dalam pembuatan data lainnya. Dikarenakan VueJS berasal dari JavaScript maka dalam proses pembuatan *Chart* dapat menggunakan *library* dari JavaScript. Sehingga memudahkan dalam pembuatan fitur *chart. Website* ini dilengkapi dengan fitur dashboard untuk menampilkan informasi penting terkait data kerja sama yang ada. Informasi tersebut adalah jumlah kerja sama aktif per negara dalam bentuk diagram batang dan jumlah kerja sama aktif per fakultas dalam bentuk diagram batang. Serta adanya fitur *reminder* akan membantu pengguna dalam mengingat waktu kerja sama. Fitur *reminder* dibuat dengan menampilkan daftar kerja sama yang akan habis masa berlakunya dalam waktu 30 hari ke depan.

Untuk pengembangan selanjutnya, dapat dibuat sistem berbasis *mobile* dengan memanfaatkan API yang sama. Selain itu juga dapat menambahkan aktor yang berperan dalam *website* ini. Yaitu adanya admin dari fakultas/prodi yang dapat lebih melengkapi data dengan lebih cepat.

DAFTAR PUSTAKA

[1] N. Ibrahim, Definition of Consistency Rules between UML Use Case and Activity Diagram, Berlin, Heidelberg: Springer, 2011.
[2] P. Teixeira, Professional Node. js: Building Javascript based scalable software, John Wiley & Sons, 2012.
[3] F. Nelli, Beginning JavaScript Charts: With JqPlot, D3, and Highcharts, Apress, 2014.
[4] S. Jake, Bootstrap: responsive web development, O'Reilly Media, Inc, 2013 *ECOC'00*, 2000, paper 11.3.4, p. 109.
[5] F. Olga, Learning Vue.js 2, Packt Publishing Ltd, 2016.
[6] B. Frain, Responsive web design with HTML5 and CSS3, Packt Publishing Ltd, 2012.
[7] F. Strazzullo, Frameworkless Front-End Development, Berkeley, CA: Apress, 2019.
[8] A. a. K. M. Kyriakidis, The Majesty of Vue. js., Packt Publishing Ltd, 2016.
[9] K. e. a. Verbert, Learning analytics dashboard applications, American Behavioral Scientist , 2013.

[10] R. L. Rothfeld, Advancing Web-based Dashboards:Providing ContextualisedComparisons in an Air TrafficDiscovery Dashboard, University of Glasgow, 2015.
[11] J. e. a. Li, How does web service API evolution affect clients?, IEEE, 2013.
[12] P. So, Vue. js, Berkeley, CA: Apress, 2018.