

LISTING PROGRAM

1. QrScanner.vue

```
<template>
  <v-layout row wrap class="mt-4">
    <v-flex v-if="qrcode" xs12 class="text-xs-center">
      <h3>Scan Kode QR disini</h3>
      <span style="color: crimson" class="ma-2">{{ error }}</span>
      <v-layout justify-center align-center>
        <!-- menampilkan komponen QR Scanner di HTML -->
        <qrcode-drop-zone @decode="initStream" @init="promise =>
promise.catch(console.error)">
          <qrcode-stream v-if="qrcode" style="width: 275px"
@decode="decodeStream" @init="initStream">
            <v-progress-circular v-
              if="loading"
              :size="50"
              color="purple"
              indeterminate
            />
          </qrcode-stream>
        </qrcode-drop-zone>
        <qrcode-capture v-if="noStreamApiSupport" @decode="onDecode" />
        <!-- end menampilkan komponen QR Scanner di HTML -->
      </v-layout>
    </v-flex>

    <v-flex xs10 offset-xs1>
      <v-btn small block :loading="loading" :color="qrcode ? 'warning' : 'info'"
@click="handleStream()">
        <v-icon small> fa
          fa-qrcode
        </v-icon>
        &nbsp;{{ qrcode ? 'Nonaktifkan' : 'Aktifkan' }} Kode QR Scanner
      </v-btn>
    </v-flex>
  </v-layout>
</template>
```

```

<script>
// import komponen QR Scanner dari library QR
import { QrcodeStream, QrcodeDropZone, QrcodeCapture } from 'vue-qrcode-reader'

export default {
  components: {
    QrcodeStream, QrcodeDropZone, QrcodeCapture
  },
  data: () => ({
    qrcode: false,
    loading: false,
    noStreamApiSupport: false,
    error: ''
  }),
  methods: {
    // fungsi untuk switch toggle scanner on/off
    handleStream() {
      this.qrcode = !this.qrcode
      this.$emit('active')
    },
    // fungsi untuk mengembalikan nilai yang sudah di decode dari QR CODE
    decodeStream(result) {
      this.$emit('decode', result)
    },
    // init komponen scan barcode ketika di aktifkan async
    initStream(promise) {
      this.loading = true try
      {
        await promise
      } catch (err) {
        if (err.name === 'NotAllowedError') {
          this.error = 'ERROR: Anda harus mengizinkan aplikasi ini untuk mengakses
kamera'
        } else if (err.name === 'NotFoundError') {
          this.error = 'ERROR: Tidak ada kamera di perangkat ini'
        } else if (err.name === 'StreamApiNotSupportedError') { this.error = 'ERROR:
Stream API tidak support di browser ini' this.noStreamApiSupport = true
        }
      } finally {

```

```

        this.loading = false
    }
}
}
}
</script>

```

2. SearchCode.vue

```

<template>
  <div>
    <v-card>
      <v-toolbar flat height="50px" color="primary">
        <v-icon color="white">
          fas fa-{{ barcode ? 'barcode' : 'qrcode' }}
        </v-icon>
        <v-toolbar-title class="white--text">
          Cari Data dengan {{ barcode ? 'Barcode' : 'Kode QR' }}
        </v-toolbar-title>
      </v-toolbar>
      <v-card-text>
        <v-layout justify-center align-center>
          <v-flex sm9 xs12>
            <v-text-field
              v-show="showInput" v-
              model="val"
              type="text"
              :prepend-icon="`fas fa-${barcode ? 'barcode' : 'qrcode'}`"
              :label="`Masukkan ${barcode ? 'Barcode' : 'Kode QR'}`"
              :mask="'#'.repeat(13)"
              autofocus
            />
            <hr v-if="showInput">
            <div class="text-xs-center">
              <slot name="extend-top" />
            </div>
          </v-flex>
        </v-layout>
      </v-card-text>
    </v-card>
  </div>
</template>
<script>
  export default {
    data() {
      return {
        val: '',
        showInput: false
      }
    },
    methods: {
      search() {
        if (this.val.trim().length < 13 && Object.keys(data).length
        === 0)

```

```

        class="text-xs-center"
    >
        <v-icon size="50" color="grey">
            search
        </v-icon>
        <h4>
            Silahkan isi {{ barcode ? 'Barcode' : 'Kode QR' }}
secara lengkap untuk mencari data
        </h4>
    </v-container>
    <v-container
        v-if="val.trim().length === 13 &&
Object.keys(data).length === 0 && error.trim().length > 0"
        class="text-xs-center"
    >
        <v-icon size="50" color="grey">
            cancel
        </v-icon>
        <h4>
            Data dengan {{ barcode ? 'Barcode' : 'Kode QR' }}
{{ val }} tidak ditemukan!
        </h4>
    </v-container>
    <Progress text="Mencari.. mohon tunggu" :loading="loading"
/>

</div>
<div class="text-xs-center">
    <slot name="extend-bottom" />
</div>
<div v-if="Object.keys(data).length > 0" class="text-xs-
center">
    <slot :data="data" />
</div>
</v-flex>
</v-layout>
</v-card-text>
</v-card>
</div>
</template>

<script>

```

```

import Progress from '~/components/Base/Progress'

export default { components: [
  Progress ], props: {
    barcode: { type:
      Boolean,
      default: true
    },
    value: {
      type: String,
      default: ''
    },
    apiEndpoint: {
      type: String,
      required: true
    },
    history: { type:
      Boolean,
      default: false
    }
  },
  data: () => ({
    loading: false,
    error: '',
    data: {},
    val: '',
    showInput: true
  }),
  watch: {
    // fungsi untuk mendeteksi apakah nilai barcode sudah 13 karakter
    // jika sudah 13 karakter maka program akan melakukan pencarian
    // dengan barcode tsb
    val(val) {
      this.$emit('change', val)
      if (val.trim().length === 13) {
        this.search(val)
      }
    }
  },
  mounted() {

```

```

const codeIn = this.value.trim().length if
(codeIn > 0)
  this.val = this.value
if (codeIn < 13 && codeIn > 0)
  this.$store.dispatch('setSnackbar', {
    visible: true,
    color: 'warning',
    msg: 'Kode belum lengkap, silahkan lengkapi form kode!'
  })
},
methods: {
  // fungsi untuk mencari data buku dengan barcode yang di input
  // dari web service
  search(code) {
    this.error = ''
    this.loading = true
    this.data = {}
    setTimeout(() => {
      this.$axios.get(`${this.apiEndpoint}/${code}`)
        .then(({ data: res }) => { if
          (!this.history)
            this.val = ''
            this.data = res.data
            this.$emit('done', this.data)
        })
        .catch(({ response: err }) => {
          this.$store.dispatch('setSnackbar', {
            visible: true,
            color: 'error',
            msg: err.data.message
          })
          this.error = err.data.message
        })
        .finally(() => {
          this.loading = false
        })
      }, 800)
    }
  }
}
</script>

```

3. Login. vue

```
<template lang="html">
  <v-content class="py-1">
    <v-container fluid fill-height>
      <v-layout align-center justify-center>
        <v-flex xs12 sm8 md5 class="text-xs-center">
          <v-card class="elevation-8">
            <v-toolbar dark color="deep-purple">
              <v-toolbar-title>Form Login</v-toolbar-title>
              <v-spacer />
            </v-toolbar>
            <div class="text-xs-center text--grey pa-3">
              <h3>Selamat Datang di Aplikasi Perpustakaan Teknik Komputer
POLSRI</h3>
            </div>
            <v-img
              src="/assets/img/polsri/polsri.svg" lazy-
              src="/assets/img/polsri/polsri.svg" max-
              width="150px"
              max-height="150px"
              style="margin: auto"
              class="my-3"
            />
            <v-form ref="login" v-model="valid" lazy-validation>
              <v-card-text>
                <v-select
                  v-model="selectedRole"
                  :items="roles" item-
                  text="name" item-
                  value="value"
                  label="Login Sebagai" prepend-
                  icon="fas fa-user"
                  append-icon="fas fa-caret-down"
                  @change="changeLoginAs"
                />
                <v-text-field
                  v-model="id.value"
                  prepend-icon="fas fa-id-card-alt"
                  :label="id.name.toUpperCase()"
                  :rules="id.rules"
                />
              </v-card-text>
            </v-form>
          </v-card>
        </v-flex>
      </v-layout>
    </v-content>
  </template>
```

```

        :mask="id.mask"
        type="text"
    />
    <Password
        toggle-show
        :value.sync="password.value"
        :rules="password.rules"
        :on-enter="login"
    />
    <br>
    <v-alert
        :value="error.length > 0"
        color="error"
        icon="fas fa-exclamation-triangle"
        outline
    >
        {{ error }}
    </v-alert>
</v-card-text>
<v-card-actions>
    <v-spacer />
    <v-btn      dark      :loading="loading"      color="deep-purple"
@click="login()">
        Login
    </v-btn>
</v-card-actions>
</v-form>
</v-card>
</v-flex>
</v-layout>
</v-container>
</v-content>
</template>

<script>
import Password from '~/components/Form/Password'

export default {
  auth: false,
  head: {
    title: 'Halaman Login'
  }
}

```

```

    },
    middleware: ({ app, redirect }) => { if
      (app.$auth.loggedIn) redirect('/')
    },
    components: { Password },
    data: () => ({
      valid: true,
      loading: false,
      error: '',
      roles: ['Mahasiswa', 'Dosen', 'Admin'],
      selectedRole: 'Mahasiswa',
      password: {
        value: '',
        rules: [
          val => !!val || 'Password tidak boleh kosong!',
          val => (val && val.length >= 6) || 'Password minimal 6 karakter!'
        ]
      }
    })),
    computed: {
      id() {
        return { name:
          'nim',
          value: '',
          mask: '#####',
          rules: [
            val => !!val || `${this.id.name.toUpperCase()} tidak boleh
kosong!`,
            val => (val && val.length === this.id.mask.length) ||
`${this.id.name.toUpperCase()} harus ${this.id.mask.length} karakter!`
          ]
        }
      },
      loginCreds() {
        return {
          role: this.selectedRole,
          [this.id.name]: this.id.value,
          password: this.password.value
        }
      }
    },
  },

```

```

methods: {
  changeLoginAs(value) {
    this.selectedRole = value
    this.$refs.login.resetValidation()
    if (value.toLowerCase() === 'mahasiswa') {
      this.id.name = 'nim'
      this.id.mask = '#####'
    } else {
      this.id.name = 'nip'
      this.id.mask = '#####'
    }
  },
  // fungsi login akun dengan merequest token dari web service login() {
  if (this.$refs.login.validate()) {
    this.loading = true
    this.$auth.loginWith('local', { data: this.loginCreds })
      .then(async () => {
        // menyimpan data id user kedalam local storage untuk
        // menyimpan sesi login this.$storage.local.set('userid',
        this.$auth.user._id)

        if (this.$auth.user._role === 'Admin') await
          this.$router.push('/admin')
        else
          await this.$router.push('/')

        this.$store.dispatch('setSnackbar', {
          visible: true,
          msg: 'Berhasil Login',
          timeout: 3000
        })

        if (this.$vuetify.breakpoint.lgAndUp) {
          setTimeout(() => this.$store.commit('setDrawer', true), 100)
        }
      })
  }
}

```

```

        // mengecek apakah user yang login bukan 'Admin' dan
        // apakah permissi notifikasi di browser sudah di aktifkan ?
        // jika iya maka program akan mendaftarkan perangkat untuk
        // selalu siap menerima broadcast notifikasi dari server
        if (this.$auth.user._role !== 'Admin' &&
this.$push.Permission.has()) {
            this.$OneSignal.push() => {
                this.$OneSignal.isPushNotificationsEnabled(isEnabled => {
                    if (isEnabled) {
                        console.log('Push notifications are enabled!')
                        this.$OneSignal.getUserId()
                            .then(pushid => {
                                console.log('Registering Notification ID..') return
this.$axios.post(`/api/user/${this.$auth.user._id}/add/${pushid}`)
                                    })
                                .then(() => {
                                    console.info('Notification ID has been
registered')

                                    this.$push.create('Notifikasi diaktifkan', { body:
                                        'Notifikasi aplikasi sudah diaktifkan', icon:
                                        '/icon.png',
                                        vibrate: [200, 100, 200, 100, 200, 100, 200]
                                    })
                                })
                            } else {
                                console.log('Push notifications are not enabled yet.')
                            }
                        })
                    })
                })
            }
        })
        .catch(err => {
            this.error = err.response.data.message
            this.password.value = ''
            this.$refs.login.resetValidation()
        })
        .finally(() => this.loading = false)
    }
}
}

```

```

}
</script>

```

4. _qrcode.vue

```

<template>
  <v-container fluid>
    <SearchCode
      ref="qrcode"
      api-endpoint="/api/loan"
      :barcode="false"
      :value="code"
      @done="onDone"
    >
    <template v-slot:extend-top>
      <QrScanner
        @active="$refs.qrcode.showInput = !$refs.qrcode.showInput"
        @decode="result => $refs.qrcode.val = result"
      />
    </template>
    <template slot-scope="{ data }">
      <v-container>
        <h2 class="mb-4">
          Informasi Peminjaman
        </h2>
        <h3>Dipinjam Oleh: {{ data.borrower._role }}</h3>
        <p>{{
          `${data.borrower.nip} || ${data.borrower.nim} -
          ${data.borrower.name}`
        }}</p>
        <h3>Status</h3>
        <p>
          <v-chip
            text-color="white"
            :color="
              data.returned
                ? 'success'
                : 'warning'
            >
          </v-chip>
          <v-avatar :class="data.returned ? 'primary' : 'error'">
            <v-icon small>
              {{ data.returned ? 'fa fa-check' : 'fa fa-times' }}
            </v-icon>
          </v-avatar>

```

```

        {{ data.returned ? 'Sudah dikembalikan' : 'Belum
dikembalikan' }}
    </v-chip>
</p>
<h3>Waktu Peminjaman</h3>
<p>{{ $moment(data.date.in).format('LLLL') }}</p>
<h3>Batas Waktu Peminjaman</h3>
<p>{{ $moment(data.date.due).format('LL') }}</p>
<div v-if="data.date.out">
    <h3>Waktu Pengembalian</h3>
    <p>{{ $moment(data.date.out).format('LLLL') }}</p>
</div>
<div v-else>
    <h3>Denda yang harus dibayar</h3>
    <p>
        <v-chip text-color="white" :color="fine(data.date.due) > 0 ? 'warning' :
'grey'">
            <v-avatar class="primary"> Rp.
            </v-avatar>
            {{ data.borrower._role === 'Mahasiswa' ?
fine(data.date.due) : 0 }}
        </v-chip>
    </p>
</div>
<h3>Total Buku yang dipinjam</h3>
<p>{{ data.books.length }}</p>
<v-list three-line>
    <v-subheader>
        Buku yang dipinjam
    </v-subheader>
    <v-divider />

    <div v-for="(book, i) in data.books" :key="i">
        <v-list-tile avatar>
            <v-list-tile-avatar>
                <v-avatar tile>
                    
    </v-avatar>
  </v-list-tile-avatar>

  <v-list-tile-content>
    <v-list-tile-title>
      {{ book.title }}
    </v-list-tile-title>
    <v-list-tile-sub-title>
      {{ `${book.publisher} - ${book.year}` }}
    </v-list-tile-sub-title>
  </v-list-tile-content>

  <v-list-tile-action>
    <v-btn icon
      small :to="`/books/details/${book.barcode_id}`">
      <v-icon small color="primary lighten-1"> fa fa-eye
    </v-icon>
    </v-btn>
  </v-list-tile-action>
</v-list-tile>
</div>
</v-list>
<v-layout row wrap>
  <v-flex xs12>
    <v-btn
      block
      color="primary"
      :disabled="data.returned"
      :loading="loading"
      @click="returnBook(data)"
    >
      {{ data.returned ? 'Peminjaman sudah dikembalikan' :
        'Konfirmasi Pengembalian' }}
    </v-btn>
  </v-flex>
</v-layout>
</v-container>
</template>

```

```
    </SearchCode>
  </v-container>
</template>
```

```
<script>
// import komponen QR Scanner yang sudah dibuat import
QrScanner from '~/components/Base/QrScanner' import SearchCode
from '~/components/Form/SearchCode'

export default {
  components: {
    // definisi komponen
    SearchCode, QrScanner
  },
  middleware: 'admin',
  head: {
    title: 'Konfirmasi Pengembalian'
  },
  data: () => ({
    loading: false
  }),
  asyncData: ({ params }) => ({ code:
    params.qrcode
  }),
  methods: {
    fine(duedate) {
      const diff = this.$moment(duedate).diff(this.$moment().format(), 'days')
      return diff < 0 ? process.env.APP_FINE * -diff : 0
    },
    returnBook(data) { this.loading
      = true setTimeout(async () =>
      {
        try {
          await this.$axios.post(`/api/loan/${data._id}`, { returned:
            true,
            date: {
              in: data.date.in, due:
                data.date.due,
              out: this.$moment().format()
            }
          }
        }
      }
    )
  }
}
```

```

    })
    this.$store.dispatch('setSnackbar', {
      visible: true,
      color: 'success',
      msg: 'Proses pengembalian berhasil!'
    })
    this.$refs.qrcode.search(data.qr_id)
  } catch ({ response }) {
    this.$store.dispatch('setSnackbar', {
      visible: true,
      color: 'error',
      msg: `Proses pengembalian gagal! ${response.data.message}`
    })
  } finally { this.loading =
    false
  }
}, 500)
},
onDone() { this.$store.dispatch('setSnackbar',
  {
    visible: true,
    color: 'primary',
    msg: 'Data Peminjaman ditemukan'
  })
}
}
}
</script>

```

5. _barcode.vue

```

<template>
  <v-container fluid>
    <SearchCode
      api-endpoint="/api/book"
      history
      :value="code"
      @change="onChange"
    >
    <template slot-scope="{ data: book }">
      <v-container>
        <v-img

```

```

        style="margin: auto"
        class="mb-4"
        height="350" width="280"
        :src="book.cover_url" ?
`/uploads/coverbooks/${book._id}/${book.cover_url}` :
'/assets/img/placeholders/cover.png' "
        :lazy-src="book.cover_url" ?
`/uploads/coverbooks/${book._id}/${book.cover_url}` :
'/assets/img/placeholders/cover.png' "
    >
    <template v-slot:placeholder>
        <v-layout fill-height align-center justify-center ma-0>
            <v-progress-circular indeterminate color="grey lighten-5">
</v-layout>
</template>
</v-img>
<hr style="width: 75%; margin: auto" class="mb-3">
<h3>Judul Buku</h3>
<p>{{ book.title }}</p>
<h3>Kategori</h3>
<p>{{ book.category }}</p>
<h3>Penulis</h3>
<p>
    <span v-for="(author, i) in book.author" :key="i">
        {{ author + (i === Object.keys(book.author).length - 1 ? '' :
',') }}
    </span>
</p>
<h3>Penerbit</h3>
<p>{{ book.publisher }} - {{ new
Date(book.year).getFullYear() }}</p>
<h3>Jumlah Halaman</h3>
<p>{{ book.pages || 'Tidak Diketahui' }}</p>
<div v-if="book.tags.length > 0">
    <h3>Tags</h3>
    <p>
        <v-chip
            v-for="(tag, i) in book.tags"
            :key="i"

```

```

        outline
        color="primary"
    >
        {{ '#' +tag }}
    </v-chip>
</p>
</div>
<h3>Jumlah Stok Buku</h3>
<p>{{ book.stock }}</p>
<h3>Total Dipinjam</h3>
<p>{{ book.total_borrowed }}</p>
<h3>Deskripsi</h3>
<p>{{ book.description || 'Tidak ada deskripsi' }}</p>
<v-layout row wrap>
    <v-flex xs12 px-2>
        <v-btn
            block
            :color="isFavorited(book._id) ? 'warning' : 'success'"
            :loading="loading"
            @click="handleFavorite(book._id)"
        >
            {{ isFavorited(book._id) ? 'Hapus dari Favorit' :
'Tambahkan ke Favorit' }}
        </v-btn>
    </v-flex>
    <v-flex xs12 md6 px-3>
        <v-btn
            v-if="userRole !== 'mahasiswa'" block
            target="_blank"
            color="info"
            :disabled="!book.soft_url"
            :href="book.soft_url"
            ?
`/api/book/soft/${book._id}_${book.soft_url}?token=${$auth.getToken('local')}.split(
')[1]` : '#'
        >
            {{ book.soft_url ? 'Lihat PDF' : 'PDF tidak tersedia' }}
        </v-btn>
    </v-flex>
    <v-flex xs12 md6 px-3>
        <v-btn

```

```

        v-if="userRole === 'admin'"
        block
        color="primary"
        :to="`/admin/books/${book._id}`"
      >
        Edit Informasi Buku
      </v-btn>
    </v-flex>
  </v-layout>
</v-container>
</template>
</SearchCode>
</v-container>
</template>

<script>
import SearchCode from '~/components/Form/SearchCode'

export default {
  components: {
    SearchCode
  },
  data: () => ({
    data: {}, loading:
    false,
    loanBarcode: ''
  }),
  head: {
    title: 'Rincian Buku'
  },
  computed: {
    userRole() {
      return this.$auth.user._role.toLowerCase() || ''
    }
  },
  asyncData: ({ params }) => ({ code:
    params.barcode
  }),
  methods: {
    handleFavorite(id) {
      this.loading = true

```

```

const method = this.isFavorited(id) ? 'remove' : 'add'
setTimeout(() => {

this.$axios.post(`/api/user/${this.$auth.user._id}/favorite/${method}/${id}`)
  .then(({ data: res }) => {
    this.$store.dispatch('setSnackbar', {
      visible: true,
      color: method === 'add' ? 'success' : 'warning', msg:
      res.message
    })
    this.$auth.setToken('local', `Bearer ${res.data.token}`)
    this.$auth.setUser({ ...res.data.user })
  })
  .catch(({ response }) => {
    this.$store.dispatch('setSnackbar', {
      visible: true,
      color: 'error',
      msg: response.data.message
    })
  })
  .finally(() => {
    this.loading = false
  })
}, 500)
},
isFavorited(id) {
  try {
    return this.$auth
      .user
      .books
      .favorite
      .find(book => book === id)
  } catch (err) {
    return false
  }
},
onChange(val) {
  window.history
    .replaceState(
      {},

```

```

        null,
        `/books/details/${val}`
    )
  }
}
}
</script>

```

6. BellButton.vue

```

<template>
  <div>
    <v-fab-transition>
      <v-btn
        v-model="fab"
        fab
        fixed
        bottom
        left
        dark
        small
        color="warning"
        @click="modal = true"
      >
        <v-icon>fa {{ $auth.user._role === 'Admin' ? 'fa-broadcast-
tower' : 'fa-bell' }}</v-icon>
      </v-btn>
    </v-fab-transition>
    <Modal :visible.sync="modal" :title="$auth.user._role === 'Admin' ? 'Broadcast
Notifikasi' : 'Mengaktifkan Notifikasi'">
      <template slot="body">
        {{
          $auth.user._role === 'Admin'
            ? 'Anda yakin ingin membroadcast notifikasi ke semua user sebagai pengingat
pengembalian buku ?'
            : 'Anda yakin ingin mengaktifkan notifikasi di perangkat ini ? (Izinkan
Notifikasi)'
        }}
      </template>
      <template slot="actions">
        <v-spacer />

```

```

        <v-btn color="primary" darken-1" flat="flat" @click="modal =
false">
            BATAL
        </v-btn>
        <v-btn color="red" darken-1" flat="flat" :loading="loading"
@click="$auth.user._role === 'Admin' ? notifyAllUser() : subscribe()">
            {{ $auth.user._role === 'Admin' ? 'YAKIN' : 'IZINKAN' }}
        </v-btn>
    </template>
</Modal>
</div>
</template>

<script>
// mengimport komponen dialog untuk broadcast atau
// mengaktifkan notifikasi pada perangkat import
Modal from '~/components/Base/Modal'

export default { components:
    { Modal }, data: () => ({
        loading: false,
        modal: false, fab:
        false
    }),
    methods: {
        // fungsi untuk trigger event broadcast notifikasi ke server
        // agar server mendengar event tersebut dan langsung dibroadcast
        // ke semua pengguna
        notifyAllUser() {
            this.loading = true
            this.$socket.emit('broadcast' loans notification',
this.$auth.user.name)
            this.$store.dispatch('setSnackbar', {
                visible: true,
                color: 'info',
                msg: 'Notifikasi telah dikirim'
            })
            this.loading = false
            this.modal = false
        },

```

```

// fungsi untuk mengaktifkan atau menerima permisi notifikasi
// pada perangkat dan browser
// jika sudah di aktifkan maka langsung trigger notifikasi aktif
// dari browser
subscribe() {
  this.loading = true
  this.$OneSignal.push(() => {
    if (this.$OneSignal.isPushNotificationsSupported()) {
      this.$OneSignal.on('subscriptionChange', isSubscribed => {
        console.info('OneSignal Subscription status:', isSubscribed) if
        (isSubscribed) {
          this.$OneSignal.getUserId()
            .then(pushid => {
              console.log('Registering Notification ID..') return
this.$axios.post(`/api/user/${this.$auth.user._id}/add/${pushid}`)
            })
            .then(() => {
              console.info('Notification ID has been registered')
              this.$push.create('Notifikasi diaktifkan', {
                body: 'Notifikasi aplikasi sudah diaktifkan', icon:
                '/icon.png',
                vibrate: [200, 100, 200, 100, 200, 100, 200]
              })
            })
            .catch(({ response }) => {
              console.warn('Failed to register Notification ID')
              this.$store.dispatch('setSnackBar', {
                visible: true,
                color: 'warning',
                msg: response.data.message
              })
            })
            .finally(() => {
              this.loading = false
              this.modal = false
            })
          }
        })
      this.$push.Permission.request(this.pushAllowed, this.pushDenied)
    } else {

```

```

        this.loading = false
        this.modal = false
        this.$store.dispatch('setSnackbar', {
            visible: true,
            color: 'warning',
            msg: 'Browser yang digunakan tidak support push notifikasi (anda
disarankan untuk menggunakan chrome atau mozilla)',
            timeout: 3000
        })
    }
})
},
pushAllowed() { this.$OneSignal.isPushNotificationsEnabled(isEnabled =>
{
    if (isEnabled) {
        this.loading = false
        this.modal = false
        this.$store.dispatch('setSnackbar', {
            visible: true,
            color: 'warning',
            msg: 'Perangkat sudah terdaftar'
        })
    } else { this.$OneSignal.registerForPushNotifications()
    }
})
},
// fungsi untuk menghandle ketika pengguna tidak mengizinkan
// permission push notifikasi
pushDenied() {
    console.log('Push notifications are not enabled yet.')
    this.loading = false
    this.modal = false
    this.$store.dispatch('setSnackbar', {
        visible: true,
        color: 'warning',
        msg: 'Izin Notifikasi di tolak'
    })
}
}
}
}

```

</script>

