

DAFTAR PUSTAKA

- Gok, N., & Khanna, N. (2013). *Building Hybrid Android Apps with Java and JavaScript*. USA: O'Reilly Media, Inc.
- Juanda, M. (2008). *Implementasi Progressive Web App sebagai Solusi untuk Meningkatkan Kinerja Aplikasi Berbasis Website*. Surabaya.
- Junaedi EP, F. (2005). *Panduan Lengkap Pemrograman HTML*. Yogyakarta: Divisi Percetakan dan Penerbitan PD. Anindaya.
- Lasa. (2007). *Manajemen Perpustakaan Sekolah*. Yogyakarta: Pinus.
- Peranginangin, K. (2006). *Aplikasi WEB dengan PHP dan MySQL*. Yogyakarta: Andi.
- Perry, W. E. (2006). *Effective Methods for Software Testing 3rd Edition*. Indianapolis: Wiley.
- Ridho, M. R., Pinandhito, A., & Dewi, R. K. (2018). *Perbandingan Performa Progressive Web Apps dan Mobile Web Terkait Waktu Respon, Penggunaan Memori dan Penggunaan Media Penyimpanan*. Makassar.
- Rosa, A., & Shalahuddin, M. (2013). *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*. Bandung: Informatika.
- Ruddamayanti. (2019). *Pemanfaatan Buku Digital dalam Peningkatan Minat baca*. Jakarta.
- Scott, E. A. (2016). *SPA Design and Architecture*. USA: Manning Publications Co.
- Sismanto. (2008, September 8). *Kupas Buku Manajemen Perpustakaan Digital*. Diambil kembali dari mkpd.wordpress:
<https://mkpd.wordpress.com/2008/09/08/kupas-buku-manajemen-perpustakaan-digital/>
- Soetejo, J. (2012). *Jurus Kilat Mahir Internet*. Jakarta: Dunia Komputer.

- Srikandi, A. S., & Paramitha, A. J. (2015). *Rancang bangun Digital Library Sistem dengan pendekatan konsep Object Oriented Hypermedia Design Method(OOHDM)*. Makassar.
- Sutarno, N. (2006). *Perpustakaan dan Masyarakat*. Jakarta: Segung Seto.
- Yakub. (2008). *Pengantar Sistem Informasi*. Yogyakarta: Graha Ilmu.
- Zaki, A. (2009). *Mari mengenal pemrograman komputer*. Yogyakarta: Yescom.

LAMPIRAN

1. Surat Izin Penelitian



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
UNIVERSITAS HASANUDDIN
FAKULTAS TEKNIK

Poros Malino Km.6Bontomarannu(92172) Gowa, Sulawesi Selatan 92172, Sulawesi Selatan
Telp. (0411) 586015, 586262 Fax (0411) 586015
<http://eng.unhas.ac.id>, Email : teknik@unhas.ac.id

17 Februari 2021

Nomor : 2570/UN4.7.1/PT.01.04/2021
Lamp : -
Hal : Permohonan Data Penelitian Mahasiswa

Yth.
Kepala Perpustakaan Fakultas Teknik Unhas
Jl. Poros Malino - Gowa

Dengan hormat, kami sampaikan bahwa dalam rangka penyelesaian skripsi / tugas akhir pada Program Studi Teknik Informatika Fakultas Teknik Universitas Hasanuddin, maka kami mohon kebijaksanaan Bapak/Ibu kiranya berkenan memberikan kesempatan melakukan pengambilan data penelitian bagi mahasiswa :

Nama (NIM) : Rosihan Ardiansyah D421 15 305
Judul TA : Perancangan Digital Library Berbasis Progresive Web APPS
Tujuan : Data Softcopy Buku Perpustakaan

Atas perhatian dan kerjasama yang baik kami sampaikan terima kasih.

a.n. Dekan,
Wakil Dekan Bidang Akademik

Prof. Baharuddin Hamzah, ST., M.Arch., Ph.D
NIP. 196903081995121001

Tembusan :
1. Dekan FT-UH
2. Ketua Departemen Teknik Informatika FT-UH



2. Surat Pernyataan Uji Coba



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
UNIVERSITAS HASANUDDIN
FAKULTAS TEKNIK

Poros Malino Km.6Bontomarannu(92172) Gowa, Sulawesi Selatan 92172, Sulawesi Selatan
Telp. (0411) 586015, 586262 Fax (0411) 586015
<http://eng.unhas.ac.id>, Email : teknik@unhas.ac.id

SURAT KETERANGAN

Yang bertanda tangan dibawah ini:

Nama : Andi Isran Ismail,S.E.
NIP : 19750416 200910 1 001
Jabatan : Pustakawan Ahli Muda

Dengan ini memberikan keterangan bahwa :

Nama : Rosihan Ardiansyah
NIM : D42115305

Telah melakukan uji coba pada aplikasi E-Library di Perpustakaan Fakultas Teknik Universitas Hasanuddin

Demikian surat ini kami sampaikan. Terima kasih atas perhatiannya

Gowa, 15 September 2021
Pustakawan Ahli Muda
Perpustakaan FTUH,



Andi Isran Ismail,S.E.
NIP. 19750416 200910 1 001

3. Script Pemrograman

1. App.js

```
import Vue from 'vue'

import VueMeta from 'vue-meta'

import PortalVue from 'portal-vue'

import { App, plugin } from '@inertiajs/inertia-vue'

import { InertiaProgress } from '@inertiajs/progress/src'

Vue.config.productionTip = false

Vue.mixin({ methods: { route: window.route } })

Vue.use(plugin)

Vue.use(PortalVue)

Vue.use(VueMeta)

InertiaProgress.init()

const el = document.getElementById('app')

// Initialize the service worker

if ('serviceWorker' in navigator) {
```

```

window.addEventListener('load', function() {
  navigator.serviceWorker.register('/offline.js').then(function(registration) {
    // Registration was successful
    console.log('ServiceWorker registration successful with scope: ', registration.scope)
  }, function(err) {
    // registration failed
    console.log('ServiceWorker registration failed: ', err)
  })
})
}

```

```

new Vue({
  metaInfo: {
    titleTemplate: title => (title ? `${title} – E-Library` : 'E-Library'),
  },
  render: h =>
    h(App, {
      props: {
        initialPage: JSON.parse(el.dataset.page),

```

```

    resolveComponent: name => import(`@/Pages/${name}`).then(module =>
module.default),

    },

  }),

}).$mount(el)

```

2. Login.vue

```

<template>

<div class="p-6 bg-red-800 min-h-screen flex justify-center items-center">

  <div class="w-full max-w-md">

    <form class="mt-8 bg-white rounded-lg shadow-xl overflow-
hidden" @submit.prevent="login">

      <div class="px-10 py-12">

        <h1 class="text-center font-bold text-3xl">Selamat Datang</h1>

        <div class="mx-auto mt-6 w-24 border-b-2" />

        <text-input v-model="form.email" :error="form.errors.email" class="mt-
10" label="Email" type="email" autofocus autocapitalize="off" />

        <text-input v-
model="form.password" :error="form.errors.password" class="mt-
6" label="Password" type="password" />

        <label class="mt-6 select-none flex items-center" for="remember">

```

```

        <input id="remember" v-model="form.remember" class="mr-
1" type="checkbox" />

        <span class="text-sm">Remember Me</span>

    </label>

</div>

<div class="px-10 py-4 bg-gray-100 border-t border-gray-100 flex justify-
between items-center">

    <a class="hover:underline" tabindex="-1" href="#reset-
password">Forget password?</a>

    <loading-button :loading="form.processing" class="btn-
indigo" type="submit">Login</loading-button>

</div>

</form>

</div>

</div>

</template>

<script>

import TextInput from '@/Shared/TextInput'

import LoadingButton from '@/Shared/LoadingButton'

```



```

export default {
  metaInfo: { title: 'Login' },
  components: {
    LoadingButton,
    TextInput,
  },
  data() {
    return {
      form: this.$inertia.form({
        email: null,
        password: null,
        remember: false,
      }),
    }
  },
  methods: {
    login() {
      this.form
        .transform(data => ({
          ...data,
          remember: data.remember ? 'on' : '',

```

```

    )))
    .post(this.route('login.attempt'))
  },
},
}
</script>

```

3. Dashboard.vue

```

<template>
  <div>
    <h1 class="mb-8 font-bold text-3xl">Dashboard</h1>
    <div class="bg-white rounded-md shadow overflow-x-auto">
      <table class="w-full table-auto whitespace-normal break-all lg:break-
words">
        <tr class="text-left font-bold">
          <th class="px-6 pt-6 pb-4 font-bold uppercase bg-gray-200 text-gray-
600">Department</th>
          <th class="px-6 pt-6 pb-4 font-bold uppercase bg-gray-200 text-gray-
600">Total</th>
        </tr>
        <tr v-for="doc in docs" :key="doc.department" class="hover:bg-gray-
100 focus-within:bg-gray-100">
          <td class="border-t">

```

```

        <inertia-link class="px-6 py-4 flex items-
center" :href="route('docs.search', doc.department)" tabindex="-1">

        {{ doc.department }}

    </inertia-link>

</td>

<td class="border-t">

    <span class="px-6 py-4 flex items-center focus:text-indigo-
500">{{ doc.sum }}</span>

</td>

</tr>

</table>

</div>

<div class="mt-8 bg-white rounded-md shadow overflow-x-auto">

    <table class="w-full table-auto whitespace-normal break-all lg:break-
words">

        <tr class="text-left font-bold">

            <th class="px-6 pt-6 pb-4 font-bold uppercase bg-gray-200 text-gray-
600">Bulan</th>

            <th class="px-6 pt-6 pb-4 font-bold uppercase bg-gray-200 text-gray-
600">Total</th>

        </tr>

```

```

    <tr v-for="report in reports" :key="report.months" class="hover:bg-gray-
100 focus-within:bg-gray-100">

      <td class="border-t">

        <span class="px-6 py-4 flex items-center" tabindex="-1">

          {{ report.month }}

        </span>

      </td>

      <td class="border-t">

        <span class="px-6 py-4 flex items-center focus:text-indigo-
500">{{ report.sum }}</span>

      </td>

    </tr>

  </table>

</div>

</div>

</template>

<script>

import Layout from '@Shared/Layout'

export default {

```

```
metaInfo: { title: 'Dashboard' },

layout: Layout,

props: {

  docs: Array,

  reports: Array,

},

created(){

  console.log(this.$page.props.auth.user)

  console.log(this.docs)

  console.log(this.reports)

},

}

</script>
```

4. Documents.vue

```
<template>

<div>

  <h1 class="mb-8 font-bold text-3xl">Daftar Buku</h1>

  <div class="mb-6 flex justify-between items-center">
```

```

<search-filter v-model="form.search" class="w-full max-w-md mr-
4" @reset="reset">

  <label class="block text-gray-700">Jenis Buku:</label>

  <select v-model="form.trashed" class="hidden mt-1 w-full form-select">

    <option :value="null" />

    <option value="with">With Trashed</option>

    <option value="only">Only Trashed</option>

  </select>

  <select v-model="form.types" class="mt-1 w-full form-select">

    <option :value="null" />

    <option v-
for="type in types" :key="type.id" :value="type.id">{{ type.name }}</option>

  </select>

</search-filter>

<inertia-link v-if="$page.props.auth.user.owner == 1" class="btn-
indigo" :href="route('docs.create')">

  <span>Tambah</span>

  <span class="hidden md:inline">Dokumen</span>

</inertia-link>

</div>

<div class="bg-white rounded-md shadow overflow-x">

  <table class="w-full table-fixed break-all md:break-words">

```

```

<tr class="text-left font-bold">

  <th class="w-1/2 px-6 pt-6 pb-4 font-bold uppercase bg-gray-200 text-
gray-600 hidden md:table-cell">Nama</th>

  <th class="w-1/5 px-6 pt-6 pb-4 font-bold uppercase bg-gray-200 text-
gray-600 hidden md:table-cell">Pengarang</th>

  <th class="w-1/5 px-6 pt-6 pb-4 font-bold uppercase bg-gray-200 text-
gray-600 hidden md:table-cell">Departemen</th>

  <th class="w-1/10 px-6 pt-6 pb-4 font-bold uppercase bg-gray-200 text-
gray-600 hidden md:table-cell" />

</tr>

<tr v-for="doc in docs.data" :key="doc.id" class="hover:bg-gray-
100 focus-within:bg-gray-100

flex md:table-row flex-col md:flex-row border-4 md:border flex-
wrap md:flex-no-wrap"

>

  <td class="pt-4 md:pt-0 md:border-t break-words block md:table-
cell relative md:static">

    <inertia-link class="px-6 py-4 flex items-center focus:text-indigo-
500" :href="route('docs.edit', doc.id)">

      {{ doc.docs_name }}

      <icon v-if="doc.deleted_at" name="trash" class="flex-shrink-0 w-3 h-
3 fill-gray-400 ml-2" />

    </inertia-link>

```

```

</td>

<td class="border-t block md:table-cell relative md:static">

  <inertia-link class="px-6 py-4 flex items-
center" :href="route('docs.edit', doc.id)" tabindex="-1">

    {{ doc.author }}

  </inertia-link>

</td>

<td class="border-t block md:table-cell relative md:static">

  <inertia-link class="px-6 py-4 flex items-
center" :href="route('docs.edit', doc.id)" tabindex="-1">

    {{ doc.department }}

  </inertia-link>

</td>

<td v-if="doc.pdf" class="px-6 md:px-0 py-4 pb-4 md:pb-0 border-
t block md:table-cell relative md:static items-center">

  <button class="items-center flex align-middle text-white-200 btn-
indigo" tabindex="-
1" @click="submit(doc.id,doc.docs_name)">Baca</button>

</td>

</tr>

<tr v-if="docs.data.length === 0">

  <td class="border-t px-6 py-4" colspan="4">No docs found.</td>

```



```
        </tr>
    </table>
</div>
<keep-alive>
    <pagination class="mt-6" :links="docs.links" />
</keep-alive>
</div>
</template>

<script>
import Icon from '@Shared/Icon'
import pickBy from 'lodash/pickBy'
import Layout from '@Shared/Layout'
import throttle from 'lodash/throttle'
import Pagination from '@Shared/Pagination'
import SearchFilter from '@Shared/SearchFilter'

export default {
  metaInfo: { title: 'Docs' },
  components: {
    Icon,
```

```
    SearchFilter,
    Pagination,
  },
  layout: Layout,
  props: {
    docs: Object,
    filters: Object,
    types: Array,
    page: Object,
    src: Object,
  },
  data() {
    return {
      form: {
        search: this.filters.search,
        trashed: this.filters.trashed,
        types: this.filters.types,
      },
      records: {
        docs_id: null,
        users_id: null,
      }
    }
  }
}
```

```

        user_name: null,
        department: null,
        doc_name: null,
    },
}
},
watch: {
    form: {
        handler: throttle(function() {
            let query = pickBy(this.form)
            this.$inertia.replace(this.route('docs', Object.keys(query).length ? query :
{ remember: 'forget' })))
        }, 150),
        deep: true,
    },
},
created(){
    if (this.src !== null){
        this.form.search = this.src.val
    }
    if(this.$page.props.auth.user.owner === 1){

```

```

    this.form.trashed = 'with'
  }
  // if(this.page.current_page == 1){
  //   this.form.types = null
  // }

  console.log('Page: ' + this.page.current_page)
  console.log(this.$page.props.auth.user)
},
methods: {
  reset() {
    this.form.search = null
    this.form.types = null
  },
  destroy() {
    if (confirm('Are you sure you want to delete this docs?')) {
      this.$inertia.delete(this.route('docs.destroy', this.doc.id))
    }
  },
  submit(id, doc_name){
    this.records.users_id = this.$page.props.auth.user.id
  }
}

```

```

    this.records.user_name = (this.$page.props.auth.user.first_name + ' ' + this.
$page.props.auth.user.last_name)

    this.records.doc_name = doc_name

    this.records.docs_id = id

    this.records.department = this.$page.props.auth.user.department

    this.$inertia.post(this.route('reports.store',this.records))

  },

},

}

</script>

```

5. Create.vue

```

<template>

  <div>

    <h1 class="mb-8 font-bold text-3xl">

      <inertia-link class="text-indigo-400 hover:text-indigo-
600" :href="route('docs')">Documents</inertia-link>

      <span class="text-indigo-400 font-medium"></span> Create

    </h1>

    <div class="bg-white rounded-md shadow overflow-hidden max-w-3xl">

```

```

<form @submit.prevent="store">

  <div class="p-8 -mr-6 -mb-8 flex flex-wrap">

    <textarea-input v-
model="form.docs_name" autocomplete="off" :error="form.errors.docs_name"
class="pr-6 pb-8 w-full" label="Judul Buku" type="text" />

    <text-input v-
model="form.author" autocomplete="off" :error="form.errors.author" class="pr
-6 pb-8 w-full" label="Pengarang" />

    <text-input v-
model="form.publisher" autocomplete="off" :error="form.errors.publisher" cla
ss="pr-6 pb-8 w-full" label="Penerbit" />

    <text-input v-
model="form.book_code" autocomplete="off" :error="form.errors.book_code"
class="pr-6 pb-8 w-full" label="Kode Buku" />

    <select-input v-
model="form.types_id" :error="form.errors.types_id" class="pr-6 pb-8 w-
full lg:w-1/2" label="Jenis Buku">

      <option :value="null" />

      <option v-
for="type in types" :key="type.id" :value="type.id">{{ type.name }}</option>

    </select-input>

    <text-input v-
model="form.NIM" autocomplete="off" :error="form.errors.NIM" class="pr-
6 pb-8 w-full" label="Stambuk" />

```

```

        <num-input v-
model="form.year" autocomplete="off" :error="form.errors.year" class="pr-
6 pb-8 w-full" label="Tahun" />

        <text-input v-
model="form.department" autocomplete="off" :error="form.errors.department"
class="pr-6 pb-8 w-full" label="Departemen" />

        <file-input v-model="form.pdf" :error="form.errors.pdf" class="pr-6 pb-
8 w-full" type="file" accept=".pdf" label="File PDF" />

    </div>

    <div class="px-8 py-4 bg-gray-50 border-t border-gray-100 flex justify-
end items-center">

        <loading-button :loading="form.processing" class="btn-
indigo" type="submit">Create Documents</loading-button>

    </div>

</form>

</div>

</div>

</template>

<script>

import Layout from '@/Shared/Layout'

import TextInput from '@/Shared/TextInput'

import SelectInput from '@/Shared/SelectInput'

```

```
import TextareaInput from '@/Shared/TextareaInput'

import FileInput from '@/Shared/FileInput'

import NumInput from '@/Shared/NumInput'

import LoadingButton from '@/Shared/LoadingButton'

export default {

  metaInfo: { title: 'Create Documents' },

  components: {

    LoadingButton,

    SelectInput,

    FileInput,

    TextInput,

    TextareaInput,

    NumInput,

  },

  layout: Layout,

  props: {

    types: Array,

  },

  remember: 'form',

  data() {
```



```
return {  
  form: this.$inertia.form({  
    docs_name: null,  
    types_id: null,  
    book_code:null,  
    author:null,  
    publisher:null,  
    NIM:null,  
    year:null,  
    department:null,  
    pdf:null,  
  }  
},  
  methods: {  
    store() {  
      this.form.post(this.route('docs.store'))  
    },  
  },  
}  
  
</script>
```

6. Edit.vue

```
<template>

<div>

  <div class="mb-8 flex justify-start max-w-3xl">

    <h1 class="mb-8 font-bold text-2xl">

      <inertia-link class="text-indigo-400 hover:text-indigo-
600" :href="route('docs')">{{ form.docs_name }}</inertia-link>

    </h1>

  </div>

  <trashed-message v-if="doc.deleted_at" class="mb-6" @restore="restore">

    Data telah dihapus.

  </trashed-message>

  <div class="bg-white rounded-md shadow overflow-hidden max-w-3xl">

    <form @submit.prevent="update">

      <div class="p-8 -mr-6 -mb-8 flex flex-wrap">

        <textarea-input v-if="$page.props.auth.user.owner == 1" v-
model="form.docs_name" :error="form.errors.docs_name" class="pr-6 pb-8 w-
full" label="Judul Buku" type="text" />

      </div>

    </form>

  </div>

</div>

</template>
```

```
<text-input v-if="$page.props.auth.user.owner == 1" v-  
model="form.author" :error="form.errors.author" class="pr-6 pb-8 w-  
full" label="Pengarang" />
```

```
<text-input v-if="$page.props.auth.user.owner == 1" v-  
model="form.publisher" :error="form.errors.publisher" class="pr-6 pb-8 w-  
full" label="Penerbit" />
```

```
<text-input v-if="$page.props.auth.user.owner == 1" v-  
model="form.book_code" :error="form.errors.book_code" class="pr-6 pb-8 w-  
full" label="Kode Buku" />
```

```
<select-input v-  
model="form.types_id" :error="form.errors.types_id" class="pr-6 pb-8 w-  
full lg:w-1/2" label="Jenis Buku">
```

```
<option :value="form.types_id" />
```

```
<option v-  
for="type in types" :key="type.id" :value="type.id">{{ type.name }}</option>
```

```
</select-input>
```

```
<text-input v-if="$page.props.auth.user.owner == 1" v-  
model="form.NIM" :error="form.errors.NIM" class="pr-6 pb-8 w-  
full" label="Stambuk" />
```

```
<num-input v-if="$page.props.auth.user.owner == 1" v-  
model="form.year" :error="form.errors.year" class="pr-6 pb-8 w-  
full" label="Tahun" />
```

```
<text-input v-if="$page.props.auth.user.owner == 1" v-  
model="form.department" :error="form.errors.department" class="pr-6 pb-8 w-  
full" label="Department" />
```

```

    <file-input v-if="$page.props.auth.user.owner == 1" v-
model="form.pdf" :error="form.errors.pdf" class="pr-6 pb-8 w-
full" type="file" accept=".pdf" label="PDF Files" />

    <div v-if="$page.props.auth.user.owner == 0">

        <span class="pr-6 pb-8 w-
full" label="Judul Buku"> {{ form.docs_name }} </span>

        <span class="pr-6 pb-8 w-
full" label="Pengarang"> {{ form.author }} </span>

        <span v-if="doc.publisher" class="pr-6 pb-8 w-
full" label="Penerbit"> {{ form.publisher }} </span>

        <span class="pr-6 pb-8 w-
full" label="Kode Buku"> {{ form.book_code }} </span>

        <!-- <span v-
for="type in types" :key="type.id" :value="type.id" class="pr-6 pb-8 w-
full" label="Jenis Buku"> {{ type.name }} </span> -->

        <span v-if="doc.NIM" class="pr-6 pb-8 w-
full" label="Stambuk"> {{ form.NIM }} </span>

        <span class="pr-6 pb-8 w-
full" label="Tahun"> {{ form.year }} </span>

        <span v-if="doc.department" class="pr-6 pb-8 w-
full" label="Department"> {{ form.department }} </span>

    </div>

</div>

```

```

    <div class="px-8 py-4 bg-gray-50 border-t border-gray-100 flex items-
center">

    <button v-
if="!doc.deleted_at && $page.props.auth.user.owner == 1" class="text-red-
600 hover:underline" tabindex="-
1" type="button" @click="destroy">Hapus Buku</button>

    <loading-button v-
if="$page.props.auth.user.owner == 1" :loading="form.processing" class="btn-
indigo ml-auto" type="submit">Ubah Buku</loading-button>

    </div>

</form>

<div v-if="doc.pdf" class="align-middle px-8 py-8 relative md:static items-
center">

    <button class="items-center flex align-middle text-white-200 btn-
indigo" tabindex="-
1" @click="submit(doc.id,doc.docs_name)">Baca</button>

    </div>

</div>

</div>

</template>

<script>

import Layout from '@/Shared/Layout'

```

```
import TextInput from '@/Shared/TextInput'

import SelectInput from '@/Shared/SelectInput'

import TextareaInput from '@/Shared/TextareaInput'

import NumInput from '@/Shared/NumInput'

import FileInput from '@/Shared/FileInput'

import LoadingButton from '@/Shared/LoadingButton'

import TrashedMessage from '@/Shared/TrashedMessage'

export default {

  metaInfo() {

    return { title: this.form.docs_name }

  },

  components: {

    FileInput,

    SelectInput,

    LoadingButton,

    TextInput,

    TextareaInput,

    NumInput,

    TrashedMessage,

  },
```

```
layout: Layout,

props: {
  doc: Object,
  types: Array,
},
remember: 'form',
data() {
  return {
    form: this.$inertia.form({
      _method: 'put',
      docs_name: this.doc.docs_name,
      book_code: this.doc.book_code,
      types_id: this.doc.types_id,
      author: this.doc.author,
      publisher: this.doc.publisher,
      NIM: this.doc.NIM,
      year: this.doc.year,
      department: this.doc.department,
      pdf: null,
    }),
    records: {
```

```

    docs_id: null,

    users_id: null,

    user_name: null,

    department: null,

    doc_name: null,

  },

}

},

created(){

  console.log(this.$page.props.auth.user)

},

methods: {

  update() {

    this.form.post(this.route('docs.update', this.doc.id), {

      onSuccess: () => this.form.reset('pdf'),

    })

  },

  destroy() {

    if (confirm('Apakah anda ingin menghapus dokumen ini?')) {

      this.$inertia.delete(this.route('docs.destroy', this.doc.id))

    }

  }

}

```



```
    },  
    restore() {  
      if (confirm('Apakah anda ingin mengembalikan dokumen ini?')) {  
        this.$inertia.put(this.route('docs.restore', this.doc.id))  
      }  
    },  
    submit(id,doc_name){  
      this.records.users_id = this.$page.props.auth.user.id  
      this.records.user_name = (this.$page.props.auth.user.first_name + ' ' + this.$  
page.props.auth.user.last_name)  
      this.records.doc_name = doc_name  
      this.records.docs_id = id  
      this.records.department = this.$page.props.auth.user.department  
      this.$inertia.post(this.route('reports.store',this.records))  
    },  
  },  
}  
</script>
```

7. Reports.vue

```
<template>

  <div>

    <h1 class="mb-8 font-bold text-3xl">Laporan</h1>

    <div class="mb-6 flex justify-between items-center">

      <search-filter v-model="form.search" class="w-full max-w-md mr-4" @reset="reset">

        <label class="mt-4 block text-gray-700">Trashed:</label>

        <select v-model="form.trashed" class="mt-1 w-full form-select">

          <option :value="null" />

          <option value="with">With Trashed</option>

          <option value="only">Only Trashed</option>

        </select>

      </search-filter>

    </div>

    <div class="bg-white rounded-md shadow overflow-x">

      <table class="w-full table-fixed break-all md:break-words whitespace-normal">

        <tr class="text-left font-bold">

          <th class="w-1/4 px-6 pt-6 pb-4 font-bold uppercase bg-gray-200 text-gray-600 hidden md:table-cell">Nama</th>
```

```
<th class="w-1/4 px-6 pt-6 pb-4 font-bold uppercase bg-gray-200 text-gray-600 hidden md:table-cell">Departemen</th>
```

```
<th class="w-1/4 px-6 pt-6 pb-4 font-bold uppercase bg-gray-200 text-gray-600 hidden md:table-cell">Judul Buku</th>
```

```
<th class="w-1/4 px-6 pt-6 pb-4 font-bold uppercase bg-gray-200 text-gray-600 hidden md:table-cell">Waktu</th>
```

```
</tr>
```

```
<tr v-for="report in reports.data" :key="report.id" class="hover:bg-gray-100 focus-within:bg-gray-100
```

```
flex md:table-row flex-col md:flex-row border-4 md:border flex-wrap md:flex-no-wrap"
```

```
>
```

```
<td class="border-t block md:table-cell relative md:static">
```

```
<span class="lg:hidden absolute top-0 left-0 bg-blue-200 px-2 py-1 text-xs font-bold uppercase">Nama</span>
```

```
<inertia-link class="px-6 py-4 flex items-center focus:text-indigo-500" :href="route('reports.show', report.id)">
```

```
  {{ report.user_name }}
```

```
<icon v-if="report.deleted_at" name="trash" class="flex-shrink-0 w-3 h-3 fill-gray-400 ml-2" />
```

```
</inertia-link>
```

```
</td>
```

```
<td class="border-t block md:table-cell relative md:static">
```

```
<span class="lg:hidden absolute top-0 left-0 bg-blue-200 px-2 py-1 text-  
xs font-bold uppercase">Departemen</span>
```

```
<inertia-link class="px-6 py-4 flex items-  
center" :href="route('reports.show', report.id)" tabindex="-1">
```

```
  {{ report.department }}
```

```
</inertia-link>
```

```
</td>
```

```
<td class="border-t block md:table-cell relative md:static">
```

```
<span class="lg:hidden absolute top-0 left-0 bg-blue-200 px-2 py-1 text-  
xs font-bold uppercase">Judul Buku</span>
```

```
<inertia-link class="px-6 py-4 flex items-  
center" :href="route('reports.show', report.id)" tabindex="-1">
```

```
  {{ report.doc_name }}
```

```
</inertia-link>
```

```
</td>
```

```
<td class="border-t block md:table-cell relative md:static">
```

```
<span class="lg:hidden absolute top-0 left-0 bg-blue-200 px-2 py-1 text-  
xs font-bold uppercase">Waktu</span>
```

```
<inertia-link class="px-6 py-4 flex items-  
center" :href="route('reports.show', report.id)" tabindex="-1">
```

```
  {{ report.created_at }}
```

```
</inertia-link>
```

```

        </td>

    </tr>

    <tr v-if="reports.data.length == 0">

        <td class="border-t px-6 py-4" colspan="3">No reports found.</td>

    </tr>

</table>

</div>

<pagination class="mt-6" :links="reports.links" />

</div>

</template>

<script>

import Icon from '@Shared/Icon'

import pickBy from 'lodash/pickBy'

import Layout from '@Shared/Layout'

import throttle from 'lodash/throttle'

import Pagination from '@Shared/Pagination'

// import mapValues from 'lodash/mapValues'

import SearchFilter from '@Shared/SearchFilter'

export default {

```

```
metaInfo: { title: 'Reports' },
components: {
  Icon,
  SearchFilter,
  Pagination,
},
layout: Layout,
props: {
  reports: Object,
  filters: Object,
},
data() {
  return {
    form: {
      search: this.filters.search,
      trashed: this.filters.trashed,
    },
  }
},
watch: {
  form: {
```

```

    handler: throttle(function() {

        let query = pickBy(this.form)

        this.$inertia.replace(this.route('reports', Object.keys(query).length ? query
: { remember: 'forget' }))

        }, 150),

        deep: true,

    },

},

mounted(){

    this.form.search = null

    this.form.trashed = null

    console.log(this.reports)

},

methods: {

    reset() {

        this.form.search = null

    },

},

}

</script>

```

8. `serviceWorker.js`

```
import {registerRoute} from 'workbox-routing'

import { precacheAndRoute,getCacheKeyForURL } from 'workbox-
precaching'

import {ExpirationPlugin} from 'workbox-expiration'

import * as strategies from 'workbox-strategies'

import {CacheableResponsePlugin} from 'workbox-cacheable-response'

precacheAndRoute(self.__WB_MANIFEST || [])

addEventListener('message', (event) => {

  if (event.data && event.data.type === 'SKIP_WAITING') {

    self.skipWaiting()

  }

})

registerRoute(

  ({request}) => request.destination === 'image',

  new strategies.CacheFirst({

    cacheName: 'images',
```



```
plugins: [  
  new ExpirationPlugin({  
    maxEntries: 60,  
    maxAgeSeconds: 30 * 24 * 60 * 60, // 30 Days  
  }),  
],  
}),  
)
```

```
registerRoute(  
  ({request}) => request.destination === 'script' ||  
    request.destination === 'style',  
  new strategies.StaleWhileRevalidate({  
    cacheName: 'static-resources',  
  }),  
)
```

```
const networkFirstHandler = new strategies.NetworkFirst({  
  cacheName: 'dynamic',  
  plugins: [  
    new ExpirationPlugin({
```

```

    maxEntries: 10,

  }),

  new CacheableResponsePlugin({

    statuses: [200],

  }),

],

})

const FALLBACK_URL = getCacheKeyForURL('/offline.html')

const matcher = ({ event }) => event.request.mode === 'navigate'

const handler = args =>

  networkFirstHandler

    .handle(args)

    .then(response => response || caches.match(FALLBACK_URL))

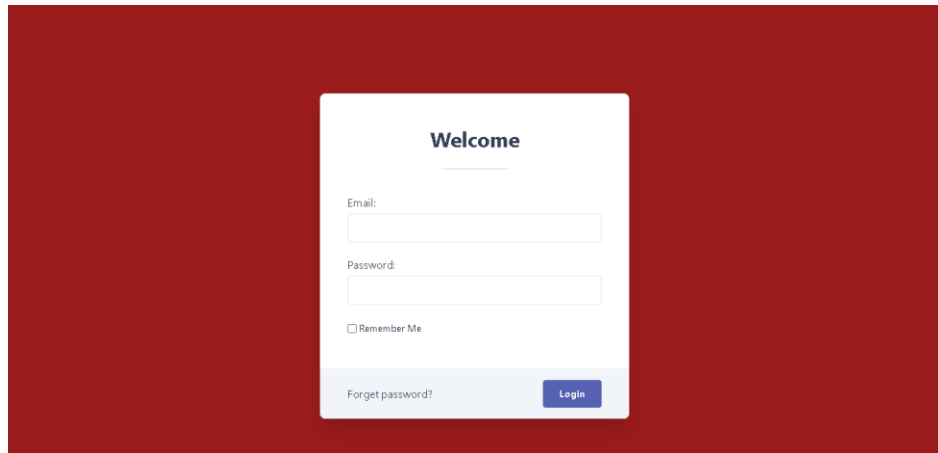
    .catch(() => caches.match(FALLBACK_URL))

registerRoute(matcher, handler)

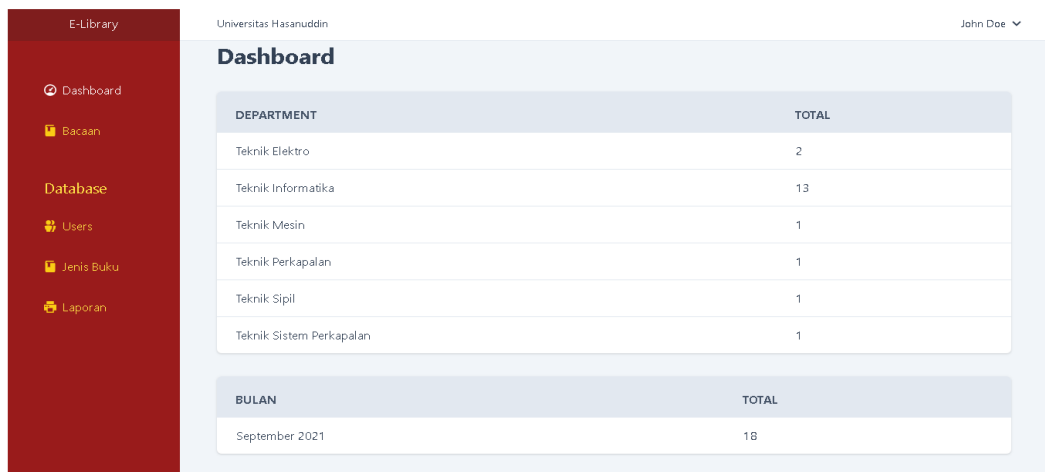
```

4. Petunjuk Penggunaan Aplikasi

1. Pada saat pertama kali masuk pada aplikasi, anda akan diarahkan ke tampilan login. Setelah login berhasil, anda akan diarahkan ke tampilan dashboard



2. Pada tampilan dashboard terdapat menu yang terletak di bagian samping yang terdiri dari menu dashboard, bacaan, dan database(hanya untuk admin) yang terdiri dari user, laporan, dan jenis buku dan menu pengguna pada bagian pojok kanan atas yang terdiri dari profil pengguna dan tombol untuk keluar dari aplikasi



The dashboard screenshot shows the following structure:

- Header:** "E-Library" on the left, "Universitas Hasanuddin" in the center, and "John Doe" with a dropdown arrow on the right.
- Sidebar (Left):** A dark red sidebar with the following menu items: "Dashboard" (selected), "Bacaan", "Database", "Users", "Jenis Buku", and "Laporan".
- Main Content Area:**
 - Dashboard Title:** "Dashboard"
 - DEPARTMENT Table:**

DEPARTMENT	TOTAL
Teknik Elektro	2
Teknik Informatika	13
Teknik Mesin	1
Teknik Perkapalan	1
Teknik Sipil	1
Teknik Sistem Perkapalan	1
 - BULAN Table:**

BULAN	TOTAL
September 2021	18

3. Pada bagian bacaan terdapat tabel daftar buku dan tombol tambah buku untuk menambahkan buku ke dalam basis data dan ketika pengguna menekan salah satu judul buku, maka akan diarahkan ke halaman detail buku

The screenshot shows the 'Daftar Buku' (Book List) page of the E-Library system. The page features a sidebar on the left with navigation options: Dashboard, Bacaan, Database, Users, Jenis Buku, and Laporan. The main content area displays a table of books with the following data:

NAMA	PENGARANG	DEPARTEMEN
A first course probability	Sheldon Ross	
A guide to fortran IV programming	DanielD. MCCracken	
A text book of electrical technology 2	B.L. Theraja	
Adaptime filter theory	Simon Haykin	
Adaptime Neural Network Control Of Robotic Manipulators	S.S. GE, T.H & Lee, C.J. Harris	

Below the table is a pagination control showing page 1 of 10, with 'Previous' and 'Next' buttons. A search bar and a 'Tambah Dokumen' button are located at the top of the main content area.

4. Pada bagian detail buku terdapat informasi lengkap mengenai buku dan tombol baca pada bagian paling bawah yang berfungsi untuk membaca buku tersebut dan dapat mengubah data buku (untuk admin)

The screenshot shows the 'Detail Buku' (Book Detail) page for the book 'A first course probability'. The page features a sidebar on the left with navigation options: Dashboard, Bacaan, Database, Users, Jenis Buku, and Laporan. The main content area displays the following information:

A first course probability

Judul Buku: A first course probability

Pengarang: Sheldon Ross

Penerbit: Person Education International

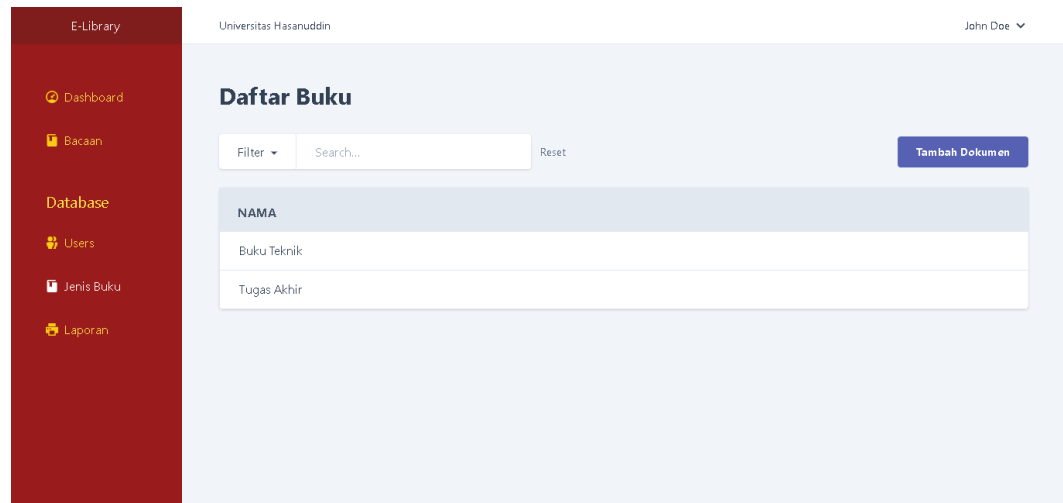
Kode Buku: 0-13-201817-9

5. Pada bagian database user terdapat tabel daftar user dan tombol tambah user untuk menambahkan user ke dalam basis user dan ketika pengguna menekan salah satu nama user, maka akan di arahkan ke halaman detail user

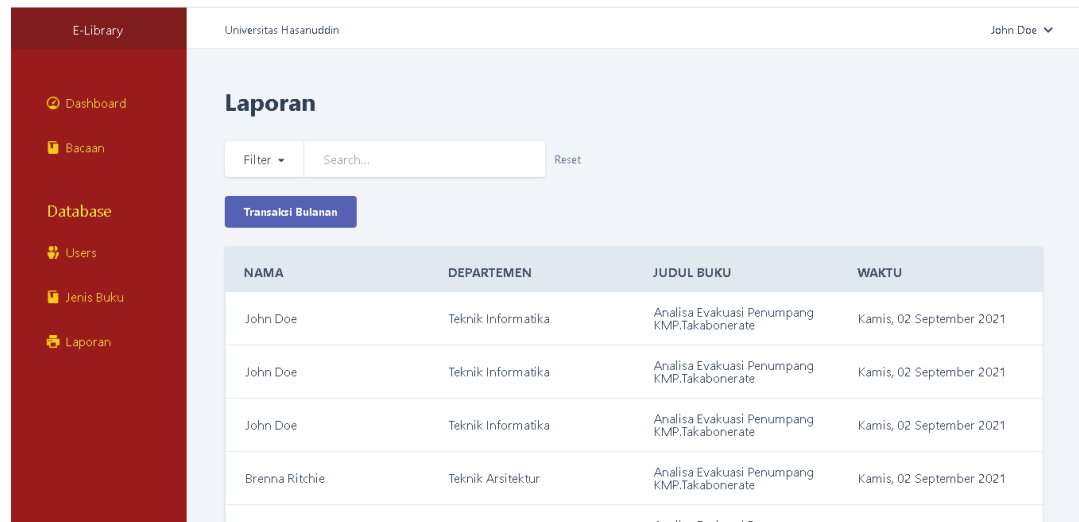
The screenshot displays the 'Daftar User' interface. On the left is a dark red sidebar with navigation options: Dashboard, Bacaan, Database, Users, Jenis Buku, and Laporan. The main content area has a light blue header with 'Universitas Hasanuddin' and a user profile 'John Doe'. Below the header is a search bar with a 'Filter' dropdown, a 'Search...' input field, and a 'Reset' button. A 'Tambah User' button is located on the right. The main content is a table with three columns: NAMA, EMAIL, and ROLE. The table contains eight rows of user data.

NAMA	EMAIL	ROLE
anu anu	anu@gmail.com	User
Gussie Dietrich	hgrimes@example.org	User
Laurine Gaylord	kane85@example.com	User
Theodora Hansen	richard88@example.net	User
Joyce Lockman	gglover@example.com	User
Brenna Ritchie	hill.winfred@example.net	User
tes tes	tes@mail.com	User

6. Pada bagian database jenis buku terdapat tabel daftar jenis buku dan tombol tambah jenis buku untuk menambahkan jenis buku ke dalam basis data dan ketika pengguna menekan salah satu jenis buku, maka akan diarahkan ke halaman detail



7. Pada bagian database laporan terdapat tabel daftar laporan yang menampilkan riwayat akses setiap pengguna ketika mengakses buku dan tombol transaksi bulanan untuk mengakses halaman laporan bulanan. Ketika pengguna menekan salah satu laporan, maka akan diarahkan ke halaman detail



8. Selamat menggunakan aplikasi DigitalLibrary

5. *Source Code* Aplikasi

Source Code dan cara *install* untuk aplikasi ini dapat diunduh melalui link berikut

<https://github.com/RosihanArdiansyah/Skripsi.git>