

LAPORAN TUGAS BESAR



Disusun oleh

Raffel Firmasnyah Cahyono	1203210121
Nur Muhammad Nabil Krisandi	1203210146
Tedy Aditiya Andrianto	1203210144
Muhammad Ezra Firdaus	1203210151

INFORMATIKA

UNIVERSITAS TELKOM SURABAYA

Bab I Pendahuluan

1.1 Latar Belakang

Sebelum adanya revolusi industri tayangan film hanya ditampilkan melalui bioskop, pelanggan harus datang langsung ke bioskop, mengantri dan membeli tiket, lalu menikmati film di ruangan bersama puluhan penonton lain. Bagi beberapa kalangan cara ini dianggap kurang praktis, karena pelanggan yang ingin menonton film harus menyesuaikan waktu dengan jam tayang film yang bersangkutan.

Dengan adanya masalah seperti itu kami membuat web bernama 2KMovie. berbeda dengan website lain yang menyediakan film secara bajakan, 2KMovie menyediakan layanan film yang langsung bisa ditonton dari sumbernya sehingga pengunjung atau penonton mendapat pengalaman yang lebih baik lagi

1.2 Tujuan

- Menyediakan informasi tentang film: Sinopsis, Judul, Rating dan informasi lainnya terkait film-film terbaru dan terkini.
- Menyediakan film original tanpa bajakan dari sumbernya

1.3 Manfaat

- Pengguna dapat menonton trailer atau film dengan grafik yang HD
- pengguna dapat mengetahui informasi tentang film yang akan di tonton

Bab II Landasan Teori

1. VueJS

VueJS adalah sebuah kerangka kerja nan progresif untuk membangun antarmuka pengguna. Tidak seperti beberapa kerangka kerja monolitik yang lain, Vue dirancang dari dasar sekali agar dapat diadopsi secara bertahap. Pustaka intinya difokuskan pada layer tampilan saja, dan sangat mudah untuk diintegrasikan dengan pustaka yang lain atau dengan proyek yang sudah ada. Di sisi lain, Vue sangat mampu memberikan dan mendukung *single page application* yang canggih ketika dikombinasikan dengan perkakas modern dan dukungan pustaka.

Vue.JS sendiri merupakan sebuah framework JavaScript yang digunakan untuk membangun antarmuka pengguna pada aplikasi web. Framework ini diciptakan oleh Evan You dan pertama kali dirilis pada Februari 2014. Vue.js dirancang untuk memudahkan pengembangan aplikasi web yang interaktif dengan menyediakan struktur yang terorganisir untuk memanipulasi tampilan.

2. Evan You (Pencipta Vue.js)

Evan You menciptakan Vue.js dengan tujuan memberikan pengembang alat yang ringan dan fleksibel untuk membangun antarmuka pengguna yang dinamis. Dia menekankan pada kemudahan penggunaan dan keterbacaan kode.

3. Greg Pollack (CEO Vue Mastery)

Greg Pollack, sebagai CEO Vue Mastery, sering menyoroti kecepatan belajar Vue.js dan kepraktisannya. Dia menilai Vue.js sebagai framework yang dapat digunakan oleh pengembang dengan berbagai tingkat keahlian.

Bab III Implementasi

link repository github : <https://github.com/tesyadt/vue-movie-web>

1. App.vue

```
<template>
  <header></header>
  <router-view/>
</template>

<style lang="scss">
@font-face {
  font-family: Roboto;
  font-style: normal;
  font-display: swap;
  font-weight: 10;
  src: url('@/assets/fonts/Roboto/Roboto-Regular.ttf');
}

#app {
  font-family: 'Roboto';
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
  text-align: center;
  color: #000000;
  background-color: #9BBEC8;
}

header {
  width: 1600px;
  height: 65px;
  left: 0px;
  top: 0px;
  background: #164863;
  margin-bottom: 60px;
}

nav {
  padding: 30px;

  a {
    font-weight: bold;
  }
}
```

```

    color: #2c3e50;

    &.router-link-exact-active {
      color: #42b983;
    }
  }
}
@import '~bootstrap/dist/css/bootstrap.css'
</style>

```

2. HomeView.vue

untuk tampilan home pada web

```

<template>
  <div class="home container">

    <movies-header @searchMovieByDate="searchMovieByDate"/>

    <movies :allMovies="allMovies" :genres="genres"/>

    <pagination
      v-if="!filteredFlag"
      :itemsCount="movies.length"
      :pageSize="pageSize"
      :currentPage="currentPage"
      @onPageChange="onPageChange" />

    <p v-if="!filteredFlag" class="results">Showing results
      {{startIndex+1}} - {{pageSize*(currentPage)}}</p>
  </div>
</template>

<script>

import axios from 'axios'
import Movies from '../components/movies/MoviesWrapper.vue'
import _ from 'lodash';
import Pagination from '../components/shared/Pagination.vue'
import MoviesHeader from '../components/movies/MoviesHeader.vue';
export default {

  name: 'HomeView',

```

```

components: {
  Movies,
  Pagination,
  MoviesHeader,
},
data: () =>({
  movies:[],
  genres:[],
  currentPage:1,
  pageSize:10,
  allMovies:[],
  startIndex:1,
  filteredFlag:false,
}),
created(){
  this.getListOfMovies()
  this.getGenres()
},
watch:{
  currentPage(){
    this.allMovies = this.paginate(this.movies, this.currentPage,
this.pageSize)
  }
},
methods:{
  getListOfMovies(){
    axios.get(`https://api.themoviedb.org/3/discover/movie?api_key=f62f750b70a8ef11dad44670cfb6aa57&language=en-US&sort_by=popularity.desc&include_adult=false&include_video=false&page=1&with_watch_monetization_types=flatrate`)
      .then((response) =>{
        this.movies = response.data.results
        this.allMovies = this.paginate(this.movies, this.currentPage,
this.pageSize)
      })
  },
  getGenres(){
    axios.get('https://api.themoviedb.org/3/genre/movie/list?api_key=f62f750b70a8ef11dad44670cfb6aa57&language=en-US')
      .then((response) =>{
        this.genres = response.data.genres

```

```

    })
  },
  paginate(items, pageNumber, pageSize){
    this.startIndex = (pageNumber - 1) * pageSize;
    return _(items)
      .slice(this.startIndex)
      .take(pageSize)
      .value()
  },
  onPageChange(page){
    this.currentPage = page
  },
}
}
</script>
<style lang="scss" scoped>

.results{
  font-weight: 400;
  font-size: 16px;
  line-height: 19px;
  color: #000000;
  margin-top: 14px;
}
</style>

```

3. MovieVlew.vue untuk tampilan movie pada web

```

<template>
  <div class="container">
    <div class="header">
      <router-link to="/">
        <svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 448 512">
          <path
            d="M447.1 256C447.1 273.7 433.7 288 416 288H109.3l105.4
105.4c12.5 12.5 12.5 32.75 0 45.25C208.4 444.9 200.2 448 192
448s-16.38-3.125-22.62-9.375l-160-160c-12.5-12.5-12.5-32.75
0-45.25l160-160c12.5-12.5 32.75-12.5 45.25 0s12.5 32.75 0 45.25l109.3
224H416C433.7 224 447.1 238.3 447.1 256z"
          />

```

```

    </svg>
    Back
  </router-link>
  <div>
    <p class="title">{{ movieDetail.original_title }}</p>
    <p class="tag">{{ movieDetail.tagline }}</p>
  </div>
</div>
<div class="movie-detail">
  <div class="col-12 d-flex">
    <div class="col-4 movie-detail__img">
      <div
        v-bind:style="{
          backgroundImage: 'url(' + imgUrl +
movieDetail.poster_path + ')',
        }"
      ></div>
    </div>
    <div class="col-8 movie-detail__detail">
      <div class="item">
        <p class="item__title">Release date</p>
        <p class="item__desc">
          {{ movieDetail.release_date }}
        </p>
      </div>
      <div class="item">
        <p class="item__title">Runtime</p>
        <p class="item__desc">
          {{ formatTime(movieDetail.runtime) }}
        </p>
      </div>
      <div class="item">
        <p class="item__title">Score</p>
        <p class="item__desc">
          {{ movieDetail.vote_average }} ({{ movieDetail.vote_count
}}
          votes)
        </p>
      </div>
      <div class="item">
        <p class="item__title">Genres</p>
        <p class="item__desc">

```



```

        <span v-for="(genre, i) in movieDetail.genres"
:key="genre.id">
            {{ genre.name }}
            {{ i !== movieDetail.genres.length - 1 ? ", " : "" }}
        </span>
    </p>
</div>
<div class="item">
    <p class="item__title">Play from imdb</p>
    <p class="item__desc">
        <a :href="imdbLink + movieDetail.imdb_id">
            <font-awesome-icon :icon="['fas', 'play']" />
        </a>
    </p>
</div>
<div class="item">
    <p class="item__title">Home Page Movie</p>
    <p class="item__desc">
        <a :href="movieDetail.homepage">
            <font-awesome-icon :icon="['fas', 'home']" />
        </a>
    </p>
</div>
</div>
</div>
<div class="col-12 movie-detail__overview">
    {{ movieDetail.overview }}
</div>
</div>
</div>
</template>

```

```

<script>
import axios from "axios";
import { FontAwesomeIcon } from "@fortawesome/vue-fontawesome";
import { library } from "@fortawesome/fontawesome-svg-core";
import { faPlay, faHome } from "@fortawesome/free-solid-svg-icons";

library.add(faPlay, faHome);

export default {
  data() {
    return {

```

```

        id: this.$route.params.id,
        movieDetail: {},
        imgUrl: "https://image.tmdb.org/t/p/w500",
        imdbLink: "https://www.imdb.com/title/",
    };
},
created() {
    this.getMovieDetail();
},
components: {
    FontAwesomeIcon, // Tambahkan komponen FontAwesomeIcon di sini
},
methods: {
    getMovieDetail() {
        axios
            .get(
`https://api.themoviedb.org/3/movie/${this.id}?api_key=f62f750b70a8ef11dad44670cfb6aa57&language=en-US`
            )
            .then((response) => {
                this.movieDetail = response.data;
                console.log(this.movieDetail);
            });
    },
    formatPrice(value) {
        if (value === 0) return 0;
        else {
            let val = (value / 1).toFixed(3).replace(".", "");
            return val.toString().replace(/\B(?=(\d{3})+(?!\d))/g, "");
        }
    },
    formatTime(value) {
        let hour = Math.floor(value / 60);
        let min = Math.floor(value % 60);
        return `${hour}h ${min}m`;
    },
},
};
</script>
<style lang="scss" scoped>
.header {

```

```
background: #427d9d;
border-radius: 6px;
display: flex;
justify-content: flex-start;
padding: 18px 35px;
margin-bottom: 78px;
text-align: left;
a {
  background: #9bbec8;
  border-radius: 100px;
  color: white;
  align-self: center;
  padding: 7px 14px;
  text-decoration: none;
  svg {
    width: 13px;
    fill: white;
  }
}
div {
  margin-left: 64px;
  & > p {
    margin-bottom: 6px;
  }
  .title {
    font-weight: 700;
    font-size: 18px;
    line-height: 21px;
  }
  .tag {
    font-weight: 400;
    font-size: 18px;
    line-height: 21px;
  }
}
}
.movie-detail {
  padding-right: 16px;
  & __img {
    & > div {
      width: 100%;
      background-size: cover;
      background-position: center;
    }
  }
}
```

```

        height: 495px;
        width: 330px;
        border-radius: 12px;
    }
}
&__detail {
    margin-top: 20px;
    .item {
        display: flex;
        justify-content: space-between;
        &__title {
            font-weight: 700;
            font-size: 16px;
            line-height: 22px;
        }
        &__desc {
            font-weight: 400;
            font-size: 16px;
            line-height: 22px;
        }
    }
}
}
&__overview {
    margin: 53px 0px 80px 0px;
    text-align: left;
}
}
</style>

```

4. MovieItem.vue

untuk menampilkan list film yang ada

```

<template>

    <router-link :to="{ name: 'movie-detail', params: { id: movie.id
    }}">

        <div class="movie-item">
            <div class="pic col-4" v-bind:style="{ backgroundImage:
            'url(' + imageUrl + ')' }">
            </div>

            <div class="detail">

```

```

        <div class="title">{{movie.original_title}}</div>
        <div class="info">
            <div class="date">
                <svg width="16" height="16" viewBox="0 0 16 16"
fill="none" xmlns="http://www.w3.org/2000/svg"
xmlns:xlink="http://www.w3.org/1999/xlink">
                    <rect width="16" height="16"
fill="url(#pattern0)"/>
                    <defs>
                        <pattern id="pattern0"
patternContentUnits="objectBoundingBox" width="1" height="1">
                            <use xlink:href="#image0_8_18"
transform="translate(0 -0.015625) scale(0.03125)"/>
                        </pattern>
                        <image id="image0_8_18" width="32"
height="33"
xlink:href="data:image/png;base64,iVBORw0KGgoAAAANSUheUgAAACAAAAAhCAQAA
AASL2HaAAAABGdBTUEAALGPC/xhBQAAACBjSFJNAAB6JgAAgIQAAPoAAACA6AAAdTAAOpG
AAA6mAAAF3CculeE8AAAAAmJLR0QA/4ePzL8AAAAHdElNRQfiAwoOOCiX8bOWAAAAA8ELEQVR
Ix+2VTw7BUBCHv9afBVsShxHHEAcQcQaLLiyqV+AGIjiDOAzClkVbxkbrTUKpEpv+ZtHk99
58b+a9ZAr/lvXgtGkBa6YJv0MTWDFLBzrILYbKH8a+kw7YIH4CAfl7xFcPIRtOiBEKABnx
GjPQjgDBYRQJxRv3yp96oBtrI2eHmHjAVsmnEx7Efd4r0BHVEEUC809ZgYcdQs1A1YnfPK8
FrV4t5ERbfQV4h0F1Elc2kcyixpweTPLxk1aPqJgr44V/Ij1pXJADsgBPwIU4/lSopthHkC
greXDEH0Vcz0TK/RoZKh8x1j/F/6nK0D2d+Xdbd9cAAAAJXRFWHRkYXRlOmNyZWF0ZQAyMD
E4LTAzLTEwVDE0OjU2OjQwKzAwOjAwQn9tOgAAACV0RVh0ZGF0ZTptb2RpZnkAMjAxOC0wM
y0xMFQxND01Njo0MCswMDowMDMi1YAAAAAASUVORK5CYII="/>
                    </defs>
                </svg>
                {{movie.release_date}}
            </div>
            <div class="genre">
                <span v-for="genre in genresOfMovie"
:key="genre.id">
                    {{ genre.name}}
                </span>
            </div>
        </div>
    </div>
</router-link>

```

```

</template>
<script>
export default {
  props:['movie', 'genres'],
  computed:{
    genresOfMovie(){
      let genresList=[]
      if(this.movie && this.genres)
      {
        let genresTitle = this.movie.genre_ids.map(m =>
this.genres.filter(g => g.id === m))
        genresTitle.map(genre => genre.map(g =>
genresList.push(g)))
        return genresList
      }
    },
    imageUrl(){
      return
`https://image.tmdb.org/t/p/w500${this.movie.poster_path}`
    }
  }
}
</script>
<style lang="scss" scoped>
  a{
    text-decoration: none;
  }
  .movie-item{
    width:295px;
    background-color: #DDF2FD;
    // border: 1px solid #DDF2FD;
    border-radius: 6px;
    padding: 3px;
    margin:45px 67px;
    display: flex;
    color:black;
    .pic{
      width: 128px;
      height:189px;
      overflow: hidden;
      background-image:
url(https://image.tmdb.org/t/p/w500/fOy2Jurz9k6RnJnMUMRDagBwru2.jpg);

```

```
        background-size: cover;
        background-position: center;
        border-radius: 6px 0px 0px 6px;
    }
    .detail{
        margin:14px;
        text-align: left;
        display: flex;
        flex-direction: column;
        justify-content: space-between;
        .title{
            font-weight: 700;
            font-size: 16px;
            line-height: 19px;
        }
        .info{
            font-weight: 400;
            font-size: 12px;
            line-height: 14px;
            .genre{
                display: flex;
                flex-wrap: wrap;
                color: #505050;
                span{
                    margin-right:4px;
                }
                span:not(:last-child):after{
                    display: inline-block;
                    content: "";
                    border-radius: 0.375rem;
                    height: 0.25rem;
                    width: 0.25rem;
                    margin-right: 0.5rem;
                    background-color: black;
                    margin-left: 4px;
                }
            }
        }
        .date{
            color: #4E4E4E;
            margin-bottom: 14px;
        }
    }
}
```

```
    }  
</style>
```

5. MovieHeader.vue sebagai header pada web film

```
<template>  
  <div class="header">  
      
  </div>  
</template>  
<style>  
  
.logo {  
  
  max-width: 200px;  
  max-height: 200px;  
}  
</style>
```

6. MovieWrapper.vue sebagai wrapper yang digunakan pada web film

```
<template>  
  
  <div class="movies" >  
    <movie-item :movie="movie" :genres="genres" v-for="movie in  
allMovies" :key="movie.id"/>  
  </div>  
</template>  
<script>  
import MovieItem from './MovieItem.vue'  
export default {  
  components:{  
    MovieItem,  
  },  
  props:["allMovies", "genres"]  
}  
</script>  
<style lang="scss" scoped>  
  .movies{
```



```

display: flex;
flex-direction: row;
flex-wrap: wrap;
justify-content: flex-start
}
</style>

```

7. Pagination.vue

sebagai perpindahan Page untuk next dan previous pada web film

```

<template>
  <nav aria-label="Page navigation example">
    <ul class="paginations">
      <li class="page-item">
        <a @click="onPageChange('previous')" :class="{ 'active':
currentPage > 1 }">Previous page</a>

      </li>
      <li class="page-item">
        <a @click="onPageChange('next')" :class="{ 'active':
currentPage < this.pages.length }">Next page</a>
      </li>
    </ul>

  </nav>
</template>
<script>
import _ from 'lodash'
export default {
  data: () => ({
    page: 0
  }),
  props: {
    itemCount: {
      type: Number,
      required: true
    },
    pageSize: {
      type: Number,
      required: true,
      default: 10
    },
    currentPage: {

```

```

        type: Number,
        required: true
    },
    created() {
        this.page = this.currentPage
    },
    computed: {
        pageCount() {
            return Math.ceil(this.itemsCount / this.pageSize) === 1 ?
null : Math.ceil(this.itemsCount / this.pageSize)
        },
        pages() {
            return _.range(1, this.pageCount + 1)
        }
    },
    methods: {
        onPageChange(value) {
            switch (value) {
                case 'previous':
                    if (this.page > 1) {
                        this.page--
                    }
                    break;
                case 'next':
                    if (this.page < this.pages.length) {
                        this.page++
                    }
                    break;
            }
            this.$emit('onPageChange', this.page)
        },
    }
}
</script>
<style lang="scss" scoped>
nav {
    padding-bottom: 0px;
    .paginations {
        text-align: center;
        li {
            list-style: none;
            justify-content: center;

```

```

        align-items: center;
        display: inline-flex;
        height: 40px;
        padding-right: 30px;
        padding-left: 30px;
        box-sizing: unset;
        font-weight: 700;
        font-size: 16px;
        line-height: 19px;
        color: rgba(0, 0, 0, 0.48);
        &:first-child{
            border-right: 1px solid #6A6A6A;
        }
        a{
            text-decoration: none;
        }
        .active{
            color: #318FE7;
        }
    }
}
}
</style>

```

8. index.js untuk router pada aplikasinya

```

import { createRouter, createWebHistory } from 'vue-router'
import HomeView from '../views/HomeView.vue'
import MovieDetail from '../views/MovieView.vue'

const routes = [
  {
    path: '/',
    name: 'home',
    component: HomeView
  },
  {
    path: '/movie-detail/:id',
    name: 'movie-detail',
    component: MovieDetail
  }
]

```

```

]

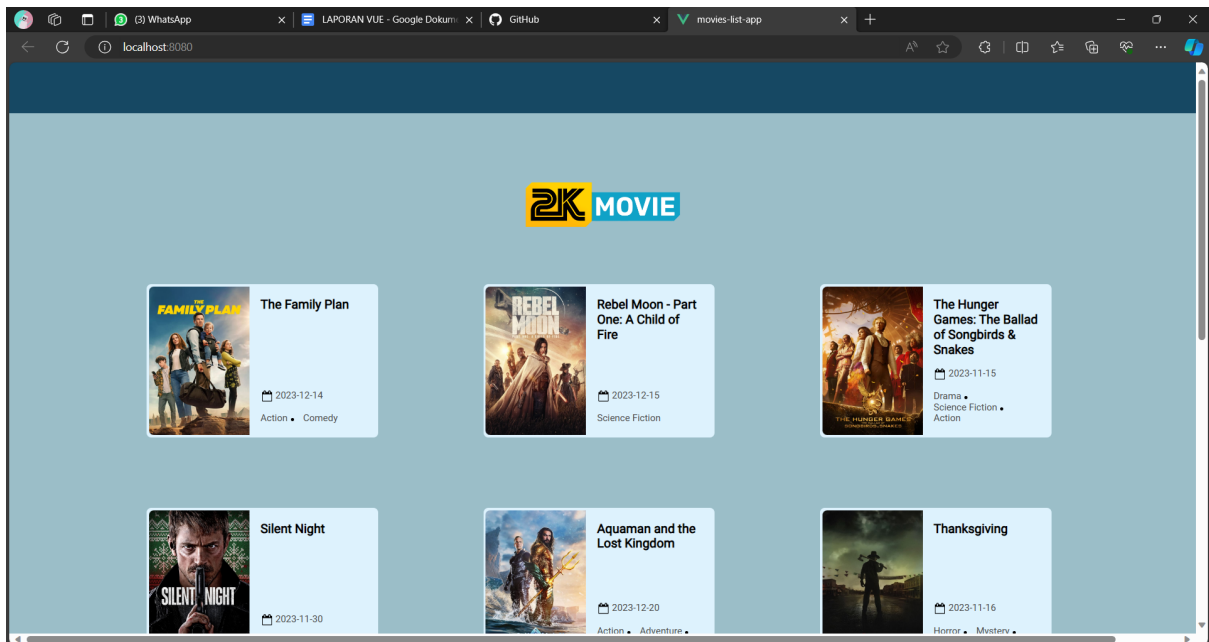
const router = createRouter({
  history: createWebHistory(process.env.BASE_URL),
  routes
})

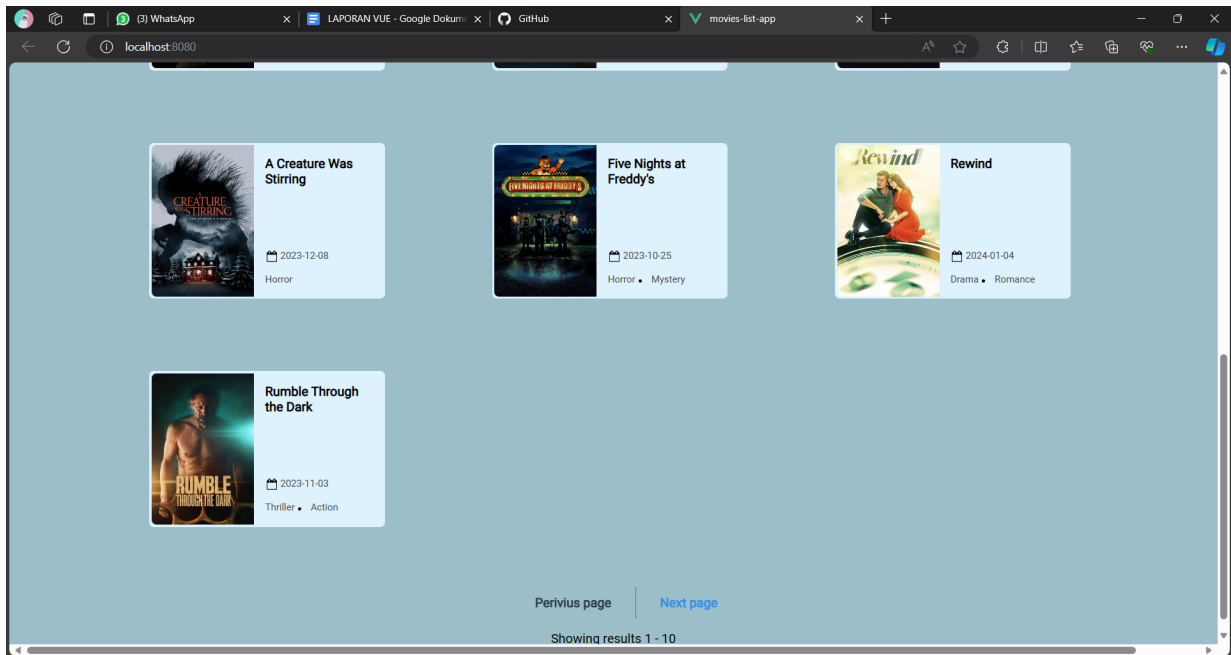
export default router

```

Bab III Tampilan

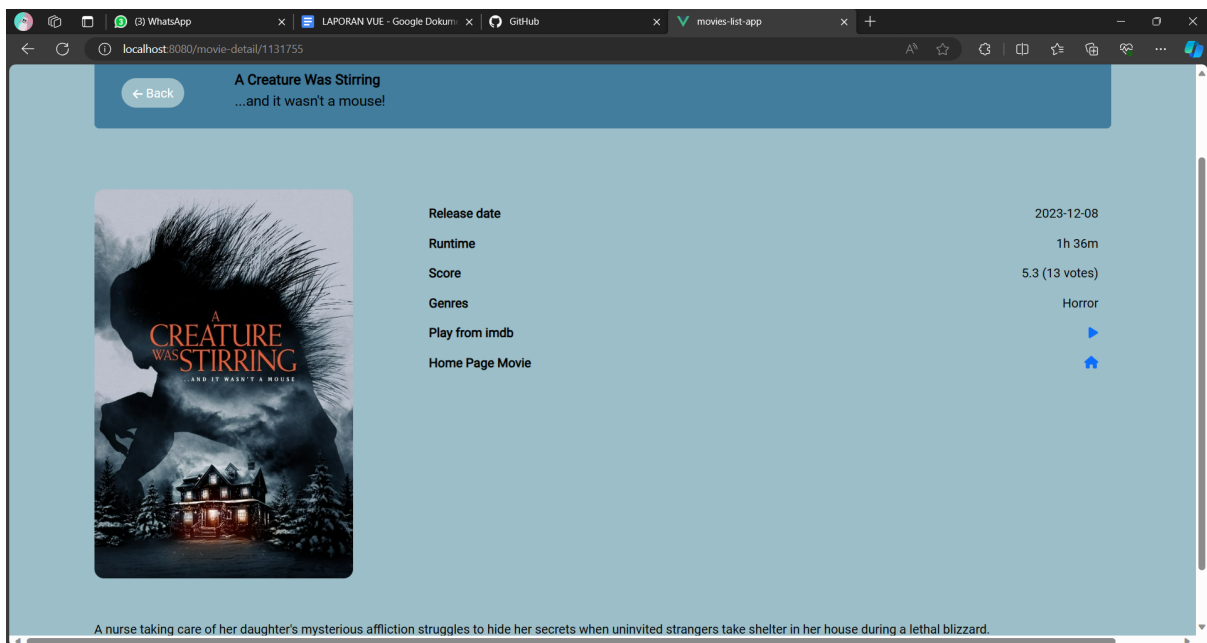
Tampilan Home





- menggunakan library axios untuk melakukan request pengampilan API yang akan digunakan untuk menampilkan list film
- untuk 1 page berisi 10 list film yang ditampilkan

Tampilan Detail film



- untuk detail film API yang digunakan di ambil API website IMDb.com
- pada detail film ini berisi gambar film, Release date, Runtime, Score, Genres dan deskripsi singkat dari film tersebut
- ketika tombol play pada **"Play from imdb"** maka akan di arahkan ke web IMDb.com untuk melihat trailer atau cuplikan dari film tersebut
- ketika tombol home pada **"Home Page Movie"** makan akan di arahkan ke web resmi film tersebut