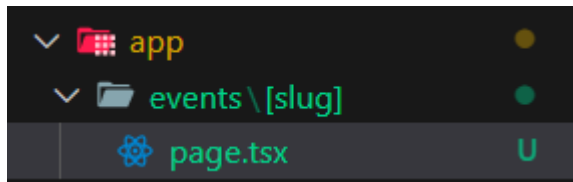


**NAMA : Ayuni Delmira**  
**NPM : 22312037**  
**KELAS : IF 22 B**

### **1. Membuat Halaman Detail Event**

Buat folder events lalu `/[slug]` dan buat file `page.ts`



```

1  'use client'
2
3  import { useEffect, useState } from 'react';
4  import axios from 'axios';
5  import { format } from 'date-fns';
6  import { id } from 'date-fns/locale';
7  import { useParams, useRouter } from 'next/navigation';
8  import NavHeader from 'components/NavHeader';
9  import Cookies from 'js-cookie';
10
11  export default function DetailEvent() {
12    const { slug } = useParams();
13    const router = useRouter();
14
15    const [event, setEvent] = useState<null>();
16    const [name, setName] = useState('');
17    const [nps, setNps] = useState('');
18    const [majorId, setMajor] = useState('');
19    const [whatsapp, setWhatsapp] = useState('');
20    const [majors, setMajors] = useState([]);
21
22    const fetchEvent = async () => {
23      try {
24        const res = await axios.get(`/front/events/${slug}`);
25        setEvent(res.data);
26      } catch (err: any) {
27        console.log('Failed to fetch event', err);
28      }
29    }
30
31    const fetchMajors = async () => {
32      try {
33        const res = await axios.get('/majors');
34        setMajors(res.data);
35      } catch (err) {
36        console.log('Failed to fetch majors', err);
37      }
38    }
39
40    useEffect(() => {
41      fetchEvent();
42      fetchMajors();
43    }, []);
44
45    const handleSubmit = async (e: React.FormEvent) => {
46      e.preventDefault();
47
48      try {
49        const res = await axios.post('/registrations', {
50          eventId: event.id, name, nps, majorId, whatsapp, {
51            headers: {
52              Authorization: `Bearer ${Cookies.get('token')}`
53            }
54          }, {
55            setName(''),
56            setNps(''),
57            setMajor(''),
58            setWhatsapp(''),
59            router.push(`/tickettrap-${res.data.id}`)
60          }
61        );
62      } catch (err: any) {
63        console.log('Failed to register', err);
64      }
65    }
66
67    if (!event) return <div>Loading...</div>;
68
69    return (
70      <div className="max-w-screen-sm mx-auto relative">
71        <navHeader title="Event" />
72
73        <img src={process.env.NEXT_PUBLIC_API_HOST + event.thumbnail} alt="" className="w-full" />
74
75        <div className="py-4 px-4 mx-auto max-w-screen-xl">
76          <h1 className="text-md font-semibold mb-3">{event.title}</h1>
77
78          <div className="grid grid-cols-3 mb-3">
79            <div className="col-span-1">
80              <h2 className="text-xs font-semibold mb-1">Date</h2>
81              <p className="text-xs font-light">{format(new Date(event.date), 'dd MMM yyyy', { locale: id })}</p>
82            </div>
83            <div className="col-span-1">
84              <h2 className="text-xs font-semibold mb-1">Time</h2>
85              <p className="text-xs font-light">{format(new Date(event.date), 'HH:mm', { locale: id })} WIB</p>
86            </div>
87            <div className="col-span-1">
88              <h2 className="text-xs font-semibold mb-1">Location</h2>
89              <p className="text-xs font-light">{event.location}</p>
90            </div>
91          </div>
92
93          <div className="mb-3">
94            <h2 className="text-xs font-semibold mb-1">Description</h2>
95            <p className="text-xs font-light text-justify">{event.description}</p>
96          </div>
97
98          <div className="mb-18">
99            <div className="flex justify-between">
100              <h2 className="text-xs font-semibold mb-1">Quota</h2>
101              <p className="text-xs font-light">{event.currentQuota} / {event.quota}</p>
102            </div>
103            <progress className="progress w-full" value={event.currentQuota} max={event.quota}></progress>
104          </div>
105
106          <div className="w-full px-4 py-4 fixed z-10 bottom-0 left-0 right-0 bg-base-100">
107            <button
108              className="btn btn-neutral w-full"
109              onClick={() => {
110                const modal = document.getElementById('modal_register') as HTMLDialogElement;
111                if (modal) modal.showModal();
112              }}
113              disabled={event.currentQuota <= 0}
114            >
115              Register
116            </button>
117          </div>
118
119          <dialog id="modal_register" className="modal">
120            <div className="modal-box">
121              <h3 className="font-semibold text-base mb-3">Modal</h3>
122              <form id="register-form" onSubmit={handleSubmit}>
123                <fieldset className="fieldset">
124                  <legend className="fieldset-legend">Full name</legend>
125                  <input type="text" className="input" placeholder="Ex: John Doe" value={name} onChange={(e) => setName(e.target.value)} required />
126                </fieldset>
127                <fieldset className="fieldset">
128                  <legend className="fieldset-legend">NPM</legend>
129                  <input type="text" className="input" placeholder="Ex: 2212089" value={nps} onChange={(e) => setNps(e.target.value)} required />
130                </fieldset>
131                <fieldset className="fieldset">
132                  <legend className="fieldset-legend">WhatsApp</legend>
133                  <input type="text" className="input" placeholder="Ex: 081234567890" value={whatsapp} onChange={(e) => setWhatsapp(e.target.value)} required />
134                </fieldset>
135                <fieldset className="fieldset">
136                  <legend className="fieldset-legend">Major</legend>
137                  <select defaultValues="Pick a major" className="select" onChange={(e) => setMajor(e.target.value)} required>
138                    <option disabled={true}>Pick a major</option>
139                    {majors && majors.map((major: any) => (
140                      <option key={major.id} value={major.id}>{major.name}</option>
141                    ))}
142                  </select>
143                </fieldset>
144              </form>
145              <div className="modal-action">
146                <form method="dialog">
147                  <button className="btn">Close</button>
148                </form>
149                <button type="submit" form="register-form" className="btn btn-neutral">Register</button>
150              </div>
151            </div>
152          </dialog>
153        </div>
154      </div>
155    );
156  }

```

## Tujuan Halaman DetailEvent

Halaman ini menampilkan informasi detail dari sebuah event berdasarkan **slug URL** dan menyediakan **form registrasi peserta** ke event tersebut. Fitur utamanya:

- Menampilkan data event (judul, tanggal, lokasi, deskripsi, kuota).
- Menampilkan progress kuota.
- Form pendaftaran peserta (dengan input nama, NPM, jurusan, WhatsApp).
- Redirect ke halaman tiket setelah berhasil mendaftar.

## Penjelasan kodingan :

'use client'

- Menandakan file ini adalah **Client Component**, memungkinkan pakai hook seperti useState, useEffect.

```
import { useEffect, useState } from "react";
```

```
import axios from "@lib/axios";
```

```
import { format } from 'date-fns'
```

```
import { id } from 'date-fns/locale'
```

```
import { useParams, useRouter } from "next/navigation";
```

```
import NavbarUser from "@lib/components/NavbarUser";
```

```
import Cookies from "js-cookie";
```

- useEffect, useState: hook React untuk state dan efek samping.
- axios: instance Axios custom dari folder lib, digunakan untuk panggil API.
- format, id: library date-fns dan locale Bahasa Indonesia untuk format tanggal.
- useParams, useRouter: dari Next.js, dipakai untuk ambil slug dan navigasi halaman.
- NavbarUser: komponen navigasi atas.
- Cookies: untuk ambil token dari cookie saat melakukan registrasi.

```
const [event, setEvent] = useState<any>(null)
```

```
const [name, setName] = useState("")
```

```
const [npm, setNpm] = useState("")
```

```
const [majorId, setMajor] = useState("")
```

```
const [whatsapp, setWhatsapp] = useState("")
```

```
const [majors, setMajors] = useState([])
```

- event: menyimpan detail event dari API.
- name, npm, majorId, whatsapp: data isian form registrasi.
- majors: list jurusan dari backend (/majors).

```
const fetchEvent = async () => {  
  const res = await axios.get(`/front/events/${slug}`)  
  setEvent(res.data)  
}
```

→ Ambil detail event berdasarkan slug.

```
const fetchMajors = async () => {  
  const res = await axios.get('/majors')  
  setMajors(res.data)  
}
```

→ Ambil list semua jurusan.

```
useEffect(() => {  
  fetchEvent()  
  fetchMajors()  
}, [])
```

→ Jalankan kedua fungsi fetch saat halaman pertama kali dimuat.

```

const handleSubmit = async (e: React.FormEvent) => {
  e.preventDefault()

  const res = await axios.post('/registrations', {
    eventId: event.id, name, npm, majorId, whatsapp
  }, {
    headers: { Authorization: `Bearer ${Cookies.get('token')}` }
  })

  router.push(`/ticket?reg=${res.data.id}`)
}

```

- Mencegah reload form default.
- Kirim data pendaftaran ke endpoint /registrations.
- Kirim token Bearer (didapat dari cookie).
- Jika berhasil, redirect ke /ticket?reg={id}.

```
if (!event) return <div>Loading...</div>
```

→ Sambil menunggu data event, tampilkan teks Loading....

```
<img src={process.env.NEXT_PUBLIC_API_HOST + event.thumbnail} ... />
```

→ Menampilkan gambar event.

```
<h1 className="text-md font-semibold">{event.title}</h1>
```

→ Judul event.

```
<p>{format(new Date(event.date), 'dd MMMM yyyy', { locale: id })}</p>
```

→ Format tanggal menggunakan date-fns dengan lokal Indonesia.

```
<progress value={event.currentQuota} max={event.quota}></progress>
```

→ Bar progress untuk kuota pendaftar saat ini.

```
<button onClick={() => modal.showModal()}>Register</button>
```

→ Jika diklik, tampilkan modal form registrasi (dengan id="modal\_register").

```
<dialog id="modal_register" className="modal">
```

```
  <form id="register-form" onSubmit={handleSubmit}>
```

```
    ...
```

```
    <select onChange={(e) => setMajor(e.target.value)}>
```

```
      {majors.map((major) => (
```

```
        <option key={major.id} value={major.id}>{major.name}</option>
```

```
      )))
```

```
    </select>
```

```
  </form>
```

```
</dialog>
```

- Form dengan input nama, npm, WhatsApp, dan jurusan.
- Data jurusan dimuat dinamis dari API.
- Ketika form disubmit, handleSubmit() dipanggil.
- Terdapat tombol Close (tutup modal) dan Register.

### Fungsi Lengkap & Alur

1. **Ambil slug dari URL** → digunakan untuk ambil data event.
2. **Ambil data event & jurusan** → disimpan di state.
3. **Render data event** → tampilkan di halaman.
4. **Klik Register** → tampilkan form modal.
5. **Isi form & submit** → data dikirim ke backend dengan token dari cookie.
6. **Redirect ke halaman tiket** jika berhasil.

## Ringkasan Kegunaan

### Bagian

### Fungsi

useParams

Ambil slug dari URL

fetchEvent()

Fetch data event dari API

fetchMajors()

Fetch data jurusan dari API

useEffect()

Menjalankan fetch saat pertama dimuat

handleSubmit()

Kirim data registrasi ke API

<dialog> & modal form

Menyediakan form pendaftaran event

router.push()

Pindah ke halaman tiket setelah pendaftaran sukses

Tampilan :

