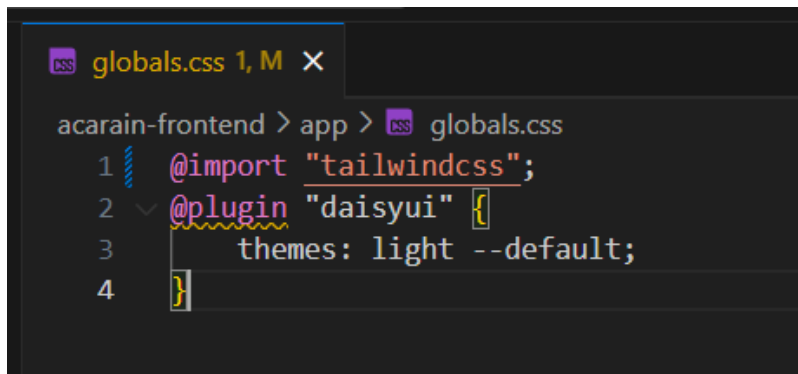


NAMA : Ayuni Delmira
NPM : 22312037
KELAS : IF 22 B

DOKUMENTASI KEDUA

1. Globals.css



```
globals.css 1, M X
acarain-frontend > app > globals.css
1 @import "tailwindcss";
2 @plugin "daisyui" {
3   themes: light --default;
4 }
```

`import "tailwindcss";` → mengimpor semua utility Tailwind CSS agar bisa digunakan di seluruh project.

`plugin "daisyui"` → menambahkan plugin DaisyUI untuk komponen UI siap pakai.

`themes: light --default;` → menetapkan tema default menjadi light.

Fungsi:

- Mengatur gaya global aplikasi (reset bawaan browser + styling dasar).
- Mengimpor utilitas CSS dari Tailwind CSS dan plugin DaisyUI.

Manfaat:

- `tailwindcss`: framework CSS utility-first, membuat styling lebih cepat.
- `daisyui`: komponen UI berbasis Tailwind, mempercepat pembuatan UI yang konsisten dan responsif.

2. Install Js-cookie

```
PS D:\KULIAH SEMESTER 6\Pemrograman Berorientasi Service\ProjectPBS\2\TOOS-Acarain\acarain-frontend> npm install js-cookie
```

`npm install js-cookie` digunakan untuk menginstal library `js-cookie`, yaitu sebuah library JavaScript ringan yang digunakan untuk mengelola cookie di sisi klien (browser). Dalam konteks aplikasi web, library ini sangat berguna untuk menyimpan, mengambil, dan menghapus data pengguna yang tersimpan di cookie, seperti token autentikasi setelah pengguna login. Misalnya, pada halaman Detail Event, `js-cookie` digunakan untuk mengambil token dari cookie melalui `Cookies.get('token')` agar bisa mengakses endpoint yang memerlukan otorisasi (seperti mendaftar event). Dengan bantuan `js-cookie`, proses pengelolaan token menjadi lebih mudah dan efisien tanpa perlu menulis kode manipulasi cookie secara manual.

3. Install axios

```
PS D:\KULIAH SEMESTER 6\Pemrograman Berorientasi Service\ProjectPBS\2\TOOS-Acarain\acarain-frontend> npm install axios
```

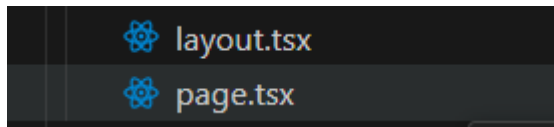
Axios merupakan library JavaScript yang digunakan untuk melakukan HTTP request ke server, seperti mengambil data (GET) atau mengirim data (POST) ke backend. Pada halaman `DetailEvent`, Axios berperan penting dalam proses pengambilan data event dan jurusan dari API, serta untuk mengirimkan data pendaftaran pengguna ke server. Instalasi Axios dilakukan dengan perintah `npm install axios` agar dapat digunakan dalam proyek. Penggunaan Axios memberikan kemudahan dalam komunikasi antara frontend dan backend serta menangani response dan error dengan lebih efisien dibandingkan menggunakan `fetch` bawaan.

4. Install date-fns@latest

```
PS D:\KULIAH SEMESTER 6\Pemrograman Berorientasi Service\ProjectPBS\2\TOOS-Acarain\acarain-frontend> npm install date-fns@latest
```

Perintah `npm install date-fns@latest` digunakan untuk menginstal `date-fns`, yaitu sebuah library JavaScript yang menyediakan fungsi-fungsi lengkap dan efisien untuk memanipulasi, memformat, dan mengelola tanggal dan waktu. Dalam proyek halaman Detail Event, `date-fns` digunakan untuk menampilkan tanggal dan waktu acara secara lebih rapi dan sesuai dengan format lokal. Contohnya, fungsi format dari `date-fns` dipakai untuk mengubah data tanggal dari format mentah (misalnya ISO string) menjadi format yang mudah dibaca seperti "11 Juni 2025". Dengan menambahkan opsi `locale: id`, tampilan tanggal juga disesuaikan dengan bahasa Indonesia. `date-fns` sangat berguna karena memiliki banyak fungsi kecil yang modular, cepat, dan tidak membebani performa aplikasi.

5. Membuat page.tsx atau landing page



app/page.tsx

Fungsi:

- Komponen halaman utama (Home) yang ditampilkan di route /.
- Menampilkan:
 - 📌 Judul aplikasi dan deskripsi.
 - 📌 Input pencarian event.
 - 📌 Daftar event dari API.
 - 📌 Navigasi bawah (Dock).

Fitur:

- Menggunakan axios untuk fetch data event dari API.
- Menyaring event berdasarkan input pencarian.
- Menampilkan tanggal dan kuota event menggunakan date-fns dengan lokalisasi Bahasa Indonesia.

Manfaat:

- UI dinamis dan interaktif dengan React Hooks (useState, useEffect).
- Menyediakan tampilan user-friendly dan informatif.

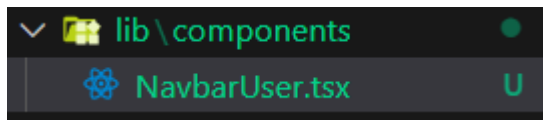
[illegible]

- use client :
= Menandakan bahwa komponen ini adalah Client Component (Next.js 13+), bisa pakai useState, useEffect :
- const [events, setEvents] = useState([]) dan const [search, setSearch] = useState("")
= State events untuk menampung data event dari API, search untuk input pencarian.
- const fetchEvents = async () => {
const res = await axios.get('/front/events')
setEvents(res.data)
}
= Fungsi ambil data event dari backend (/front/events).
- useEffect(() => { fetchEvents() }, [])
= Menjalankan fetch sekali ketika komponen pertama kali dimount.
- events.filter(...).map(...)
= Menampilkan daftar event yang cocok dengan pencarian (filter berdasarkan event.title).

6. Membuat komponen Navigasi dan QR

Membuat folder lib dan components lalu membuat file NavbarUser, dock, dan QRView.

a. Membuat NavbarUser.tsx



Fungsi:

- Komponen navigasi atas.
- Menampilkan judul halaman dan tombol Back.

Manfaat:

- Navigasi antar halaman terasa natural.
- Tombol back menggunakan router.back() agar pengguna bisa kembali ke halaman sebelumnya.

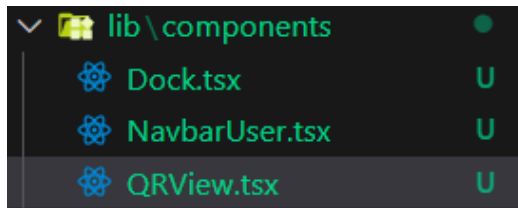
```
<button onClick={() => router.back()}>...</button>
```

– Tombol kembali (seperti tombol back di browser).

```
<a className="btn btn-ghost text-xl">{title}</a>
```

– Menampilkan judul halaman (dikirim dari props).

c. Membuat QRView



Fungsi:

- Menampilkan QR Code dari string tertentu.
- Digunakan untuk **halaman tiket** (misalnya untuk validasi saat scan di venue).

Manfaat:

- QR Code mudah di-scan.
- Bisa dipakai untuk bukti kepemilikan tiket.

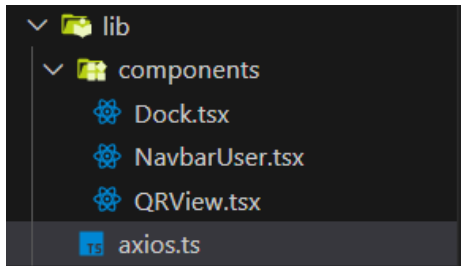
`<QRCodeSVG value={value} />`

– Menggunakan library `qrcode.react` untuk menampilkan QR code dari string yang dikirim via props `value`.

```
1 import { QRCodeSVG } from "qrcode.react"
2
3 export default function QrView({ value }: { value: string }) {
4   return (
5     <div className="flex justify-center">
6       <QRCodeSVG value={value} />
7     </div>
8   )
9 }
10
```

7. Membuat konfigurasi API Axios.ts untuk API backend

Dalam folder lib buat file axios.ts

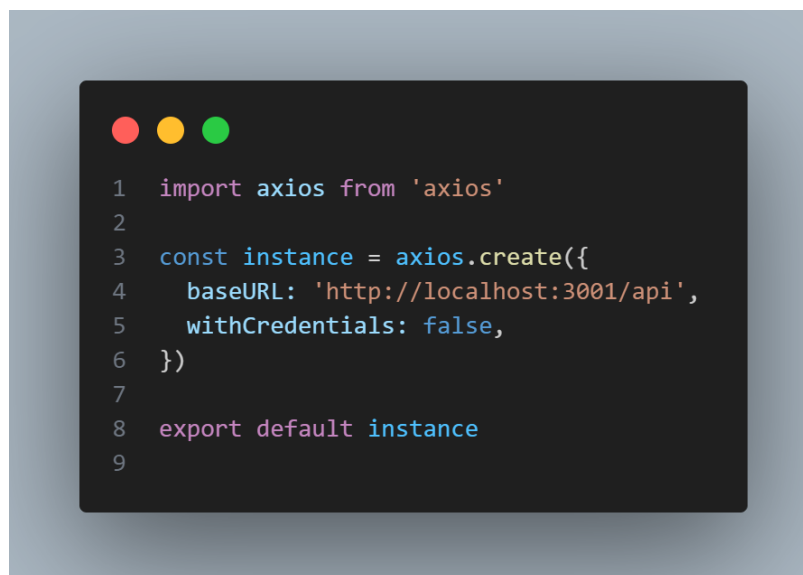


Fungsi:

- Setup instance **Axios** untuk komunikasi dengan API backend (<http://localhost:3001/api>).
- Bisa dipakai di seluruh aplikasi tanpa mengulang konfigurasi.

Manfaat:

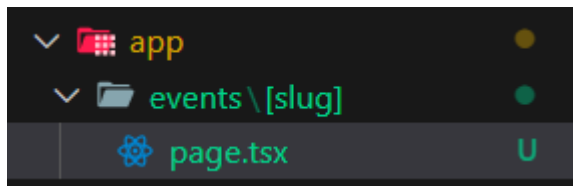
- Mempermudah dan menstandarkan komunikasi API.
- Bisa disesuaikan jika pindah ke environment production



- baseURL: prefix URL untuk semua endpoint.
- withCredentials: false: tidak mengirim cookie/session dari browser (kecuali kamu set true untuk auth yang pakai cookie).

8. 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 '9/lib/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            <div>
104              <progress className="progress w-full" value={event.currentQuota} max={event.quota}></progress>
105            </div>
106          </div>
107
108          <div className="w-full px-4 py-4 fixed z-10 bottom-0 left-0 right-0 bg-base-100">
109            <button
110              className="btn btn-neutral w-full"
111              onClick={() => {
112                const modal = document.getElementById('modal_register') as HTMLDialogElement;
113                if (modal) modal.showModal();
114              }}
115              disabled={event.currentQuota <= 0}
116            >
117              Register
118            </button>
119          </div>
120
121          <dialog id="modal_register" className="modal">
122            <div className="modal-box">
123              <h3 className="font-semibold text-base mb-3">Modal</h3>
124              <form id="register-form" onSubmit={handleSubmit}>
125                <fieldset className="fieldset">
126                  <legend className="fieldset-legend">Full name</legend>
127                  <input type="text" className="input" placeholder="Ex: John Doe" value={name} onChange={(e) => setName(e.target.value)} required />
128                </fieldset>
129                <fieldset className="fieldset">
130                  <legend className="fieldset-legend">NPM</legend>
131                  <input type="text" className="input" placeholder="Ex: 2212089" value={nps} onChange={(e) => setNps(e.target.value)} required />
132                </fieldset>
133                <fieldset className="fieldset">
134                  <legend className="fieldset-legend">WhatsApp</legend>
135                  <input type="text" className="input" placeholder="Ex: 081234567890" value={whatsapp} onChange={(e) => setWhatsapp(e.target.value)} required />
136                </fieldset>
137                <fieldset className="fieldset">
138                  <legend className="fieldset-legend">Major</legend>
139                  <select
140                    defaultValues="Pick a major"
141                    className="select"
142                    onChange={(e) => setMajor(e.target.value)}
143                    required
144                  >
145                    <option disabled={true}>Pick a major</option>
146                    {majors && majors.map((major: any) => (
147                      <option key={major.id} value={major.id}>{major.name}</option>
148                    ))}
149                  </select>
150                </fieldset>
151              </form>
152              <div className="modal-action">
153                <form method="dialog">
154                  <button className="btn">Close</button>
155                </form>
156                <button type="submit" form="register-form" className="btn btn-neutral">Register</button>
157              </div>
158            </div>
159          </dialog>
160        </div>
161      </div>
162    );
163  }
164

```

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

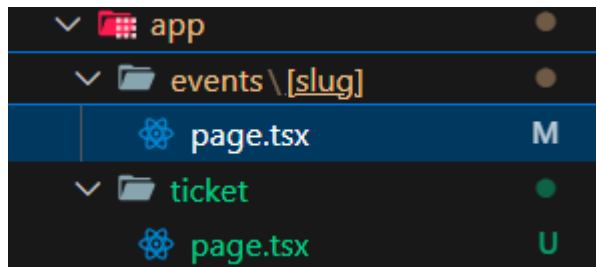
1. **Ambil slug dari URL** → digunakan untuk ambil data event.
2. **Ambil data event & jurusan** → disimpan di state.
3. **Render data event** → tampilkan di halaman.
4. **Klik Register** → tampilkan form modal.
5. **Isi form & submit** → data dikirim ke backend dengan token dari cookie.
6. **Redirect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

7. Membuat Halaman Ticket

Buat halaman tiket pada app\ticket dan buat file page.tsx



Pada halaman app/ticket/page.tsx, terdapat fitur untuk menampilkan data tiket pendaftaran user berdasarkan ID registrasi. Data tersebut diambil dari endpoint backend menggunakan axios, sebuah library HTTP client berbasis promise yang diinstal melalui perintah npm install axios. Axios memungkinkan pengambilan data dari server secara efisien, seperti dalam fungsi fetchReg, yang bertugas mengambil detail registrasi pengguna berdasarkan ID dari query parameter URL. Untuk menampilkan waktu dan tanggal dengan format yang mudah dibaca pengguna Indonesia, digunakan library date-fns yang dipasang menggunakan npm install date-fns@latest. Library ini memiliki banyak fungsi manipulasi waktu, dan dalam kode digunakan fungsi format untuk mengubah data tanggal (createdAt, event.date) menjadi format lokal Indonesia seperti “11 Juni 2025, 13:00 WIB” dengan bantuan locale: id.

Selain itu, terdapat komponen QrView, yang digunakan untuk menampilkan QR Code milik peserta yang telah berhasil mendaftar dan disetujui (APPROVED). QR Code ini ditampilkan dalam sebuah modal dialog dan hanya bisa dibuka ketika status registrasi memenuhi syarat.

Tampilan ini membantu peserta menunjukkan bukti pendaftaran mereka secara cepat dalam bentuk digital saat hari acara.

Secara keseluruhan, halaman ini memanfaatkan integrasi dari beberapa library penting (axios, date-fns, QRView) untuk menghadirkan fitur pencarian tiket registrasi, menampilkan data lengkap pengguna dan event, serta menyediakan QR Code untuk keperluan verifikasi kehadiran secara efisien.

Penjelasan kodingan :

'use client'

- Menandakan file ini dijalankan di sisi **client** (bukan server-side rendering)

```
import Dock from "@/lib/components/Dock";  
  
import { useState, useEffect } from "react";  
  
import axios from "@/lib/axios";  
  
import { useSearchParams } from "next/navigation";  
  
import { format } from 'date-fns'  
  
import { id } from 'date-fns/locale'  
  
import Link from "next/link";  
  
import QrView from "@/lib/components/QRView";
```

- **Dock**: Komponen navigasi di bagian bawah halaman.
- **useState, useEffect**: Hook React untuk state dan lifecycle.
- **axios**: Untuk HTTP request.
- **useSearchParams**: Mengambil parameter query dari URL (?reg=...).
- **date-fns + locale id**: Format tanggal ke Bahasa Indonesia.
- **Link**: Navigasi Next.js.
- **QrView**: Komponen yang menampilkan QR Code.

```
const [search, setSearch] = useState("")
const [reg, setReg] = useState<any>(null)
const searchParams = useSearchParams()
```

- search: Menyimpan ID registrasi.
- reg: Menyimpan data registrasi dari API.
- searchParams: Untuk membaca parameter reg dari URL.

```
const fetchReg = async (id: string) => {
  try {
    const res = await axios.get(`/registrations/${id}`)
    setReg(res.data)
  } catch (err) {
    console.log('Failed to fetch registration', err)
    setReg(null)
  }
}
```

- Fungsi ini mengambil data pendaftaran dari endpoint /registrations/:id.
- Jika berhasil, disimpan ke state reg.

```
useEffect(() => {
  const regParam = searchParams.get('reg')
  if (regParam) {
    setSearch(regParam)
    fetchReg(regParam)
  }
}, [])
```

- Ketika halaman pertama kali dibuka, jika ada parameter reg, maka ambil datanya.


```
useEffect(() => {
  if (search) fetchReg(search)
}, [search])
```

- Ketika search berubah (user mengetik ID), data akan di-fetch ulang.

```
const handleCancel = async () => {
  const ok = confirm('Are you sure you want to cancel this registration?')
  if (!ok) return
  try {
    await axios.post(`/registrations/cancel?token=${reg.cancelToken}`)
    fetchReg(search)
  } catch (err) {
    console.log('Failed to cancel registration', err)
  }
}
```

- Fungsi ini menjalankan POST request untuk membatalkan pendaftaran menggunakan cancelToken.

```
<input type="search" value={search} onChange={(e) => setSearch(e.target.value)} required
placeholder="Registration ID" />
```

- Input field untuk memasukkan ID registrasi.

```
<table>
  <tr><td>Name</td><td>{reg.name}</td></tr>
  <tr><td>NPM</td><td>{reg.npm}</td></tr>
  ...
</table>
```

- Tabel menampilkan detail registrasi: nama, NPM, WhatsApp, jurusan, waktu daftar.

```
{reg.status === 'PENDING' && <span className="badge badge-warning">Pending</span>}
```

...

```
<button onClick={() => document.getElementById('modal_qrcode')?.showModal()}  
disabled={!reg.status === 'APPROVED' && reg.attendance === 'ABSENT'}}>
```

- Menampilkan badge status registrasi.
- Tombol untuk membuka modal QR Code hanya aktif jika status = APPROVED dan belum hadir.

```
{reg.status !== 'REJECTED' && reg.status !== 'CANCELLED' && reg.attendance ===  
'ABSENT' ?
```

```
<button onClick={handleCancel}>Cancel</button> : null
```

```
}
```

- Tombol Cancel hanya muncul jika status belum ditolak/dibatalkan dan belum hadir.

```
<table>
```

```
<tr><td>Title</td><td>{reg.event.title}</td></tr>
```

...

```
</table>
```

```
<Link href={` /events/${reg.event.slug}`}>Show Event</Link>
```

- Menampilkan info event yang didaftarkan: judul, lokasi, tanggal, waktu.
- Link ke detail event.

```
<dialog id="modal_qrcode">  
  {reg.qrCode && <QrView value={reg.qrCode} />}  
</dialog>
```

- QR Code ditampilkan dalam modal dialog jika tersedia (QrView).

```
<Dock />
```

- Navigasi tetap (biasanya untuk ke Home, Ticket, Profile, dsb.).

8. Modifikasi package-lock.json

```

1  {
2    "name": "quartz4j-frontend",
3    "version": "3.0.0-RC",
4    "description": "3.0",
5    "license": "MIT",
6    "package": {
7      "name": "quartz4j-frontend",
8      "version": "3.0.0-RC",
9      "dependencies": {
10        "react": "16.8.6",
11        "react-dom": "16.8.6",
12        "react-router": "4.3.1",
13        "react-router-dom": "4.3.1",
14        "react-scripts": "1.0.10",
15        "react": "16.8.6",
16        "react-dom": "16.8.6",
17        "react-router": "4.3.1",
18        "react-router-dom": "4.3.1",
19        "react-scripts": "1.0.10"
20      },
21      "scripts": {
22        "start": "react-scripts start",
23        "build": "react-scripts build",
24        "test": "react-scripts test",
25        "eject": "react-scripts eject"
26      },
27      "devDependencies": {
28        "react-scripts": "1.0.10"
29      }
30    },
31    "scripts": {
32      "start": "react-scripts start",
33      "build": "react-scripts build",
34      "test": "react-scripts test",
35      "eject": "react-scripts eject"
36    },
37    "devDependencies": {
38      "react-scripts": "1.0.10"
39    }
40  },
41  {
42    "name": "quartz4j-backend",
43    "version": "3.0.0-RC",
44    "description": "3.0",
45    "license": "MIT",
46    "package": {
47      "name": "quartz4j-backend",
48      "version": "3.0.0-RC",
49      "dependencies": {
50        "react": "16.8.6",
51        "react-dom": "16.8.6",
52        "react-router": "4.3.1",
53        "react-router-dom": "4.3.1",
54        "react-scripts": "1.0.10",
55        "react": "16.8.6",
56        "react-dom": "16.8.6",
57        "react-router": "4.3.1",
58        "react-router-dom": "4.3.1",
59        "react-scripts": "1.0.10"
60      },
61      "scripts": {
62        "start": "react-scripts start",
63        "build": "react-scripts build",
64        "test": "react-scripts test",
65        "eject": "react-scripts eject"
66      },
67      "devDependencies": {
68        "react-scripts": "1.0.10"
69      }
70    },
71    "scripts": {
72      "start": "react-scripts start",
73      "build": "react-scripts build",
74      "test": "react-scripts test",
75      "eject": "react-scripts eject"
76    },
77    "devDependencies": {
78      "react-scripts": "1.0.10"
79    }
80  },
81  {
82    "name": "quartz4j-backend",
83    "version": "3.0.0-RC",
84    "description": "3.0",
85    "license": "MIT",
86    "package": {
87      "name": "quartz4j-backend",
88      "version": "3.0.0-RC",
89      "dependencies": {
90        "react": "16.8.6",
91        "react-dom": "16.8.6",
92        "react-router": "4.3.1",
93        "react-router-dom": "4.3.1",
94        "react-scripts": "1.0.10",
95        "react": "16.8.6",
96        "react-dom": "16.8.6",
97        "react-router": "4.3.1",
98        "react-router-dom": "4.3.1",
99        "react-scripts": "1.0.10"
100      },
101      "scripts": {
102        "start": "react-scripts start",
103        "build": "react-scripts build",
104        "test": "react-scripts test",
105        "eject": "react-scripts eject"
106      },
107      "devDependencies": {
108        "react-scripts": "1.0.10"
109      }
110    },
111    "scripts": {
112      "start": "react-scripts start",
113      "build": "react-scripts build",
114      "test": "react-scripts test",
115      "eject": "react-scripts eject"
116    },
117    "devDependencies": {
118      "react-scripts": "1.0.10"
119    }
120  },
121  {
122    "name": "quartz4j-backend",
123    "version": "3.0.0-RC",
124    "description": "3.0",
125    "license": "MIT",
126    "package": {
127      "name": "quartz4j-backend",
128      "version": "3.0.0-RC",
129      "dependencies": {
130        "react": "16.8.6",
131        "react-dom": "16.8.6",
132        "react-router": "4.3.1",
133        "react-router-dom": "4.3.1",
134        "react-scripts": "1.0.10",
135        "react": "16.8.6",
136        "react-dom": "16.8.6",
137        "react-router": "4.3.1",
138        "react-router-dom": "4.3.1",
139        "react-scripts": "1.0.10"
140      },
141      "scripts": {
142        "start": "react-scripts start",
143        "build": "react-scripts build",
144        "test": "react-scripts test",
145        "eject": "react-scripts eject"
146      },
147      "devDependencies": {
148        "react-scripts": "1.0.10"
149      }
150    },
151    "scripts": {
152      "start": "react-scripts start",
153      "build": "react-scripts build",
154      "test": "react-scripts test",
155      "eject": "react-scripts eject"
156    },
157    "devDependencies": {
158      "react-scripts": "1.0.10"
159    }
160  },
161  {
162    "name": "quartz4j-backend",
163    "version": "3.0.0-RC",
164    "description": "3.0",
165    "license": "MIT",
166    "package": {
167      "name": "quartz4j-backend",
168      "version": "3.0.0-RC",
169      "dependencies": {
170        "react": "16.8.6",
171        "react-dom": "16.8.6",
172        "react-router": "4.3.1",
173        "react-router-dom": "4.3.1",
174        "react-scripts": "1.0.10",
175        "react": "16.8.6",
176        "react-dom": "16.8.6",
177        "react-router": "4.3.1",
178        "react-router-dom": "4.3.1",
179        "react-scripts": "1.0.10"
180      },
181      "scripts": {
182        "start": "react-scripts start",
183        "build": "react-scripts build",
184        "test": "react-scripts test",
185        "eject": "react-scripts eject"
186      },
187      "devDependencies": {
188        "react-scripts": "1.0.10"
189      }
190    },
191    "scripts": {
192      "start": "react-scripts start",
193      "build": "react-scripts build",
194      "test": "react-scripts test",
195      "eject": "react-scripts eject"
196    },
197    "devDependencies": {
198      "react-scripts": "1.0.10"
199    }
200  },
201  {
202    "name": "quartz4j-backend",
203    "version": "3.0.0-RC",
204    "description": "3.0",
205    "license": "MIT",
206    "package": {
207      "name": "quartz4j-backend",
208      "version": "3.0.0-RC",
209      "dependencies": {
210        "react": "16.8.6",
211        "react-dom": "16.8.6",
212        "react-router": "4.3.1",
213        "react-router-dom": "4.3.1",
214        "react-scripts": "1.0.10",
215        "react": "16.8.6",
216        "react-dom": "16.8.6",
217        "react-router": "4.3.1",
218        "react-router-dom": "4.3.1",
219        "react-scripts": "1.0.10"
220      },
221      "scripts": {
222        "start": "react-scripts start",
223        "build": "react-scripts build",
224        "test": "react-scripts test",
225        "eject": "react-scripts eject"
226      },
227      "devDependencies": {
228        "react-scripts": "1.0.10"
229      }
230    },
231    "scripts": {
232      "start": "react-scripts start",
233      "build": "react-scripts build",
234      "test": "react-scripts test",
235      "eject": "react-scripts eject"
236    },
237    "devDependencies": {
238      "react-scripts": "1.0.10"
239    }
240  },
241  {
242    "name": "quartz4j-backend",
243    "version": "3.0.0-RC",
244    "description": "3.0",
245    "license": "MIT",
246    "package": {
247      "name": "quartz4j-backend",
248      "version": "3.0.0-RC",
249      "dependencies": {
250        "react": "16.8.6",
251        "react-dom": "16.8.6",
252        "react-router": "4.3.1",
253        "react-router-dom": "4.3.1",
254        "react-scripts": "1.0.10",
255        "react": "16.8.6",
256        "react-dom": "16.8.6",
257        "react-router": "4.3.1",
258        "react-router-dom": "4.3.1",
259        "react-scripts": "1.0.10"
260      },
261      "scripts": {
262        "start": "react-scripts start",
263        "build": "react-scripts build",
264        "test": "react-scripts test",
265        "eject": "react-scripts eject"
266      },
267      "devDependencies": {
268        "react-scripts": "1.0.10"
269      }
270    },
271    "scripts": {
272      "start": "react-scripts start",
273      "build": "react-scripts build",
274      "test": "react-scripts test",
275      "eject": "react-scripts eject"
276    },
277    "devDependencies": {
278      "react-scripts": "1.0.10"
279    }
280  },
281  {
282    "name": "quartz4j-backend",
283    "version": "3.0.0-RC",
284    "description": "3.0",
285    "license": "MIT",
286    "package": {
287      "name": "quartz4j-backend",
288      "version": "3.0.0-RC",
289      "dependencies": {
290        "react": "16.8.6",
291        "react-dom": "16.8.6",
292        "react-router": "4.3.1",
293        "react-router-dom": "4.3.1",
294        "react-scripts": "1.0.10",
295        "react": "16.8.6",
296        "react-dom": "16.8.6",
297        "react-router": "4.3.1",
298        "react-router-dom": "4.3.1",
299        "react-scripts": "1.0.10"
300      },
301      "scripts": {
302        "start": "react-scripts start",
303        "build": "react-scripts build",
304        "test": "react-scripts test",
305        "eject": "react-scripts eject"
306      },
307      "devDependencies": {
308        "react-scripts": "1.0.10"
309      }
310    },
311    "scripts": {
312      "start": "react-scripts start",
313      "build": "react-scripts build",
314      "test": "react-scripts test",
315      "eject": "react-scripts eject"
316    },
317    "devDependencies": {
318      "react-scripts": "1.0.10"
319    }
320  },
321  {
322    "name": "quartz4j-backend",
323    "version": "3.0.0-RC",
324    "description": "3.0",
325    "license": "MIT",
326    "package": {
327      "name": "quartz4j-backend",
328      "version": "3.0.0-RC",
329      "dependencies": {
330        "react": "16.8.6",
331        "react-dom": "16.8.6",
332        "react-router": "4.3.1",
333        "react-router-dom": "4.3.1",
334        "react-scripts": "1.0.10",
335        "react": "16.8.6",
336       
```

dan seterusnya

File `package-lock.json` adalah file yang secara otomatis dihasilkan oleh Node Package Manager (npm) ketika menjalankan perintah `npm install`. File ini memiliki peran penting dalam proyek berbasis Node.js atau framework JavaScript modern seperti Next.js. Fungsi utama dari `package-lock.json` adalah untuk mengunci versi dari semua package dan sub-package yang diinstal, sehingga memastikan bahwa setiap orang yang mengerjakan proyek tersebut mendapatkan versi dependency yang sama, baik di komputer lokal maupun di lingkungan produksi. Dengan kata lain, file ini menjaga konsistensi dan stabilitas proyek agar tidak terjadi perbedaan versi package yang bisa menyebabkan bug atau error yang sulit dilacak.

Selain itu, `package-lock.json` juga menyimpan informasi lengkap mengenai dependency, seperti versi yang diunduh, sumber URL, serta hash untuk memverifikasi integritas file. Hal ini sangat berguna untuk meningkatkan keamanan dan mencegah modifikasi berbahaya pada paket-paket yang digunakan. Berbeda dengan `package.json` yang hanya mencantumkan dependency utama secara umum, `package-lock.json` mencatat seluruh struktur dependency hingga level terdalam, termasuk sub-dependency yang tidak ditulis langsung oleh developer.

File ini sangat penting untuk dikomit ke dalam version control seperti Git, karena dapat menjamin bahwa semua developer dalam tim bekerja menggunakan lingkungan dependency yang identik. Oleh karena itu, file ini tidak boleh diedit secara manual karena berisiko merusak struktur dependency yang sudah stabil. Jika terjadi kerusakan atau konflik pada file ini, solusinya adalah dengan menghapus file `package-lock.json` dan folder `node_modules`, lalu menjalankan kembali `npm install` untuk menghasilkan file yang baru dan bersih.

Secara keseluruhan, `package-lock.json` adalah bagian penting dalam manajemen dependency sebuah proyek JavaScript modern. Keberadaannya mendukung praktik pengembangan yang konsisten, aman, dan dapat diandalkan, baik dalam skala individu maupun tim.

9. Modifikasi package.json



```
1  {
2    "name": "acarain-frontend",
3    "version": "0.1.0",
4    "private": true,
5    "scripts": {
6      "dev": "next dev --turbo",
7      "build": "next build",
8      "start": "next start",
9      "lint": "next lint"
10   },
11   "dependencies": {
12     "axios": "^1.9.0",
13     "date-fns": "^4.1.0",
14     "js-cookie": "^3.0.5",
15     "next": "15.3.3",
16     "qrcode.react": "^4.2.0",
17     "react": "^19.0.0",
18     "react-dom": "^19.0.0"
19   },
20   "devDependencies": {
21     "@tailwindcss/postcss": "^4.1.10",
22     "@types/js-cookie": "^3.0.6",
23     "@types/node": "^20",
24     "@types/react": "^19",
25     "@types/react-dom": "^19",
26     "daisyui": "^5.0.43",
27     "tailwindcss": "^4",
28     "typescript": "^5"
29   }
30 }
```

File `package.json` adalah file inti dalam proyek berbasis Node.js dan digunakan untuk mendefinisikan metadata proyek, daftar dependensi, serta perintah-perintah (scripts) yang dapat dijalankan melalui NPM. File ini bertindak seperti "blueprint" dari proyek, yang menjelaskan bagaimana proyek dijalankan, dependensi apa saja yang dibutuhkan, dan versi paket yang digunakan. Dalam proyek ini, `package.json` milik aplikasi **"acarain-frontend"** menyimpan berbagai informasi penting yang dibutuhkan agar aplikasi dapat berjalan dengan baik.

Properti "name": "acarain-frontend" dan "version": "0.1.0" menunjukkan nama dan versi dari aplikasi. private: true mencegah proyek ini dipublikasikan ke registry publik seperti npmjs.com, yang berguna untuk menjaga kerahasiaan aplikasi internal.

Pada bagian "scripts", terdapat beberapa perintah yang memudahkan developer menjalankan tugas tertentu:

- "dev": "next dev --turbo" digunakan untuk menjalankan server development dengan Next.js menggunakan teknologi Turbo agar lebih cepat.
- "build": "next build" akan membangun aplikasi untuk produksi (production-ready).
- "start": "next start" menjalankan aplikasi yang sudah dibangun dalam mode produksi.
- "lint": "next lint" digunakan untuk mengecek kualitas dan kesalahan dalam kode dengan alat linter.

Selanjutnya, bagian "dependencies" berisi library yang dibutuhkan saat aplikasi dijalankan:

- axios: library untuk melakukan HTTP request.
- date-fns: untuk memanipulasi dan memformat tanggal.
- js-cookie: untuk mengelola cookie di sisi klien (browser).
- next, react, dan react-dom: merupakan inti dari framework Next.js dan library React yang digunakan untuk membangun UI.
- qrcode.react: library untuk menampilkan QR code berbasis React.

Kemudian, bagian "devDependencies" berisi paket yang hanya digunakan saat pengembangan (bukan untuk runtime di produksi):

- tailwindcss dan daisyui: digunakan untuk styling tampilan antarmuka dengan utilitas CSS dan komponen UI siap pakai.
- @tailwindcss/postcss: plugin pendukung untuk Tailwind dengan PostCSS.
- typescript: bahasa pemrograman berbasis JavaScript dengan dukungan tipe.
- @types/*: adalah paket definisi tipe (type definitions) untuk mendukung TypeScript pada berbagai library seperti React, Node, dan js-cookie.

Kesimpulannya, package.json merupakan pusat konfigurasi dan pengelolaan dependensi aplikasi. File ini memungkinkan proyek untuk dijalankan dengan konsisten di berbagai lingkungan dan sangat penting untuk kolaborasi tim pengembang serta deployment ke server produksi.

10. Modifikasi postcss.config.mjs



File `postcss.config.mjs` adalah file konfigurasi untuk PostCSS, sebuah alat (tool) yang digunakan dalam proses build CSS modern. PostCSS bekerja dengan mengubah CSS melalui berbagai plugin, seperti untuk auto-prefixing, nested rule, hingga integrasi framework CSS seperti Tailwind CSS. Dalam proyek ini, file konfigurasi ini berperan untuk mengatur agar Tailwind CSS dapat diproses dengan benar saat aplikasi dijalankan atau dibangun.

Isi dari file ini adalah konfigurasi berbentuk JavaScript module modern (menggunakan `.mjs` dan `export default`), yang artinya ditulis dengan standar ECMAScript Module. Pada baris:

```
const config = {  
  plugins: ["@tailwindcss/postcss"],  
};
```

didefinisikan bahwa PostCSS akan menggunakan plugin `@tailwindcss/postcss`. Plugin ini adalah antarmuka agar Tailwind CSS dapat berfungsi melalui PostCSS dan memproses utility class yang digunakan dalam file HTML, JSX, atau TSX menjadi CSS akhir yang bisa digunakan oleh browser.

Kemudian, pada baris:

```
export default config;
```

konfigurasi tersebut diekspor sebagai default export agar dapat digunakan oleh tool build (seperti Next.js atau Vite) saat melakukan kompilasi CSS.

Secara keseluruhan, `postcss.config.mjs` berfungsi untuk mengaktifkan plugin Tailwind di dalam proses build CSS, memastikan bahwa semua utility class Tailwind yang ditulis di dalam proyek bisa dikompilasi menjadi CSS siap pakai di browser.