

**Dokumentasi Projek Pemrograman Web Berorientasi Layanan
(FRONTEND)**



Annisa Zulfa Ulya

TI 23 A

23313014

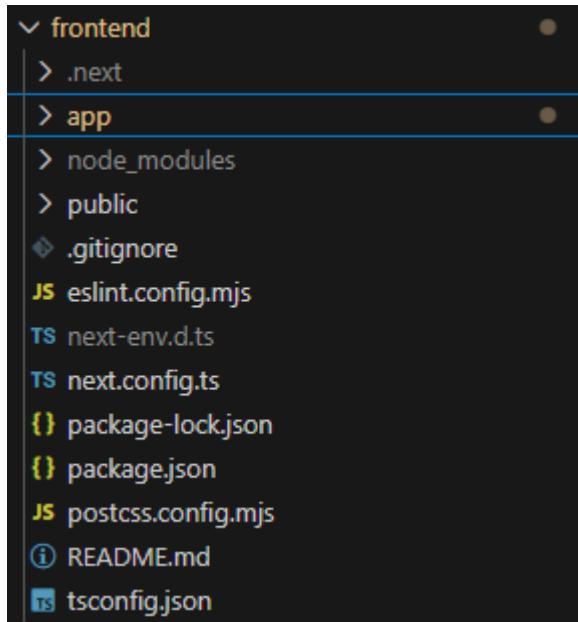
**S1 TEKNOLOGI INFORMASI
FAKULTAS TEKNIK DAN ILMU KOMPUTER
UNIVERSITAS TEKNOKRAT INDONESIA**

LOGIN

1. Menginstall Frontend (Untuk tampilan & interaksi pengguna)

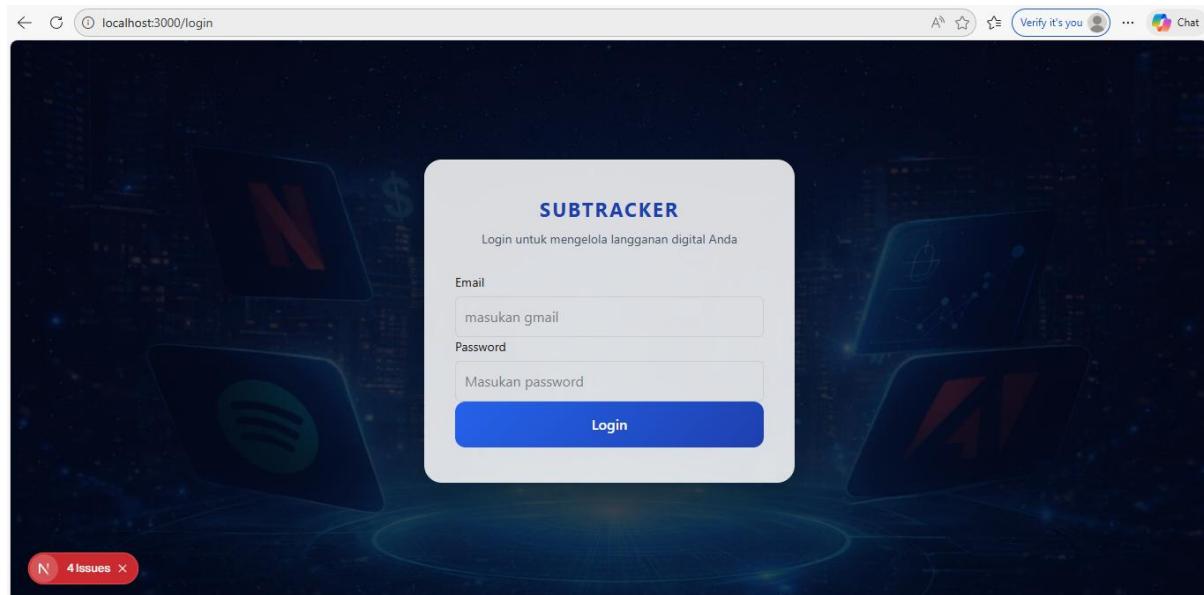
annisa : menginstall frontend

 annisazulfaulya committed on Dec 12, 2025



Ketika sudah didownload akan tampil folder frontnd Dimana berisi folder/file yang bisa diedit untuk tampilan web.

TAMPILAN AWAL DARI FORM LOGIN



Tampilan awal ini merupakan halaman login SubTracker yang digunakan pengguna untuk masuk ke sistem dengan memasukkan email dan password guna mengelola langganan digital mereka.

1.



```
return ()<>
  /* CARD LOGIN LANGSUNG */
  <div className="login-card">
    <h2>SUBTRACKER</h2>
    <p className="subtitle">
      Login untuk mengelola langganan digital Anda
    </p>
```

Kode tersebut berfungsi untuk **menampilkan kartu login** berisi **judul aplikasi dan deskripsi singkat** sebagai pengantar sebelum pengguna memasukkan data login.

2.



```
<form onSubmit={handleLogin}>
  <div className="input-group">
    <label>Email</label>
    <input
      type="email"
      placeholder="masukan gmail"
      value={email}
      onChange={(e) => setEmail(e.target.value)}>
    />
  </div>
```

Kode ini digunakan untuk **menampilkan form login** dengan **input email** serta **menangani perubahan nilai email** sebelum proses login dijalankan.

3. Password

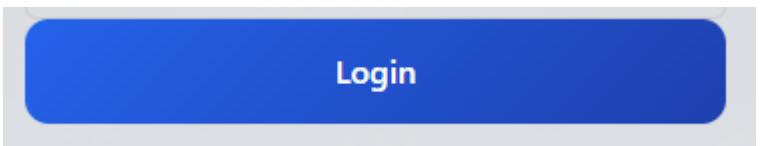


Masukan password

```
<div className="input-group">
  <label>Password</label>
  <input
    type="password"
    placeholder="Masukan password"
    value={password}
    onChange={(e) => setPassword(e.target.value)}>
  />
</div>
```

Kode ini berfungsi untuk **menyediakan input password pada form login dan menyimpan nilai password yang dimasukkan pengguna.**

4.



Login

```
</div>

<button type="submit">Login</button>
</form>
</div>
```

Kode ini digunakan untuk **menampilkan tombol Login yang akan mengirim form dan menjalankan proses autentikasi saat diklik.**

5. **localhost:3000 says**

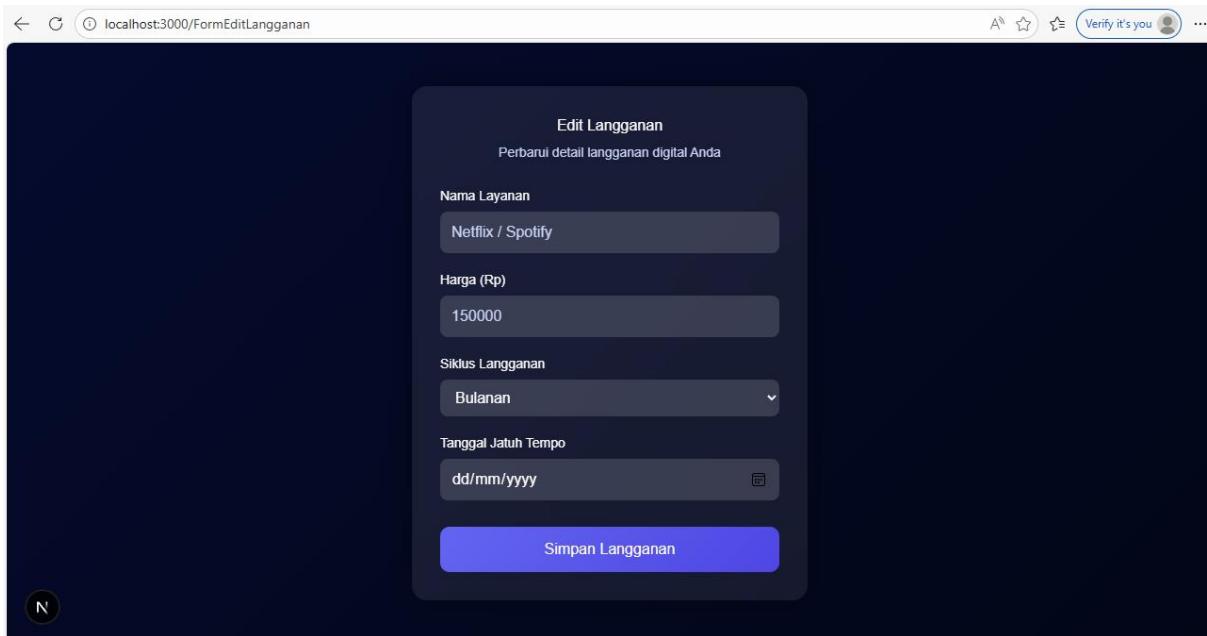
Email dan password wajib diisi

OK

```
if (!email || !password) {  
  alert("Email dan password wajib diisi");  
  return;  
}
```

Kode ini berfungsi untuk **memvalidasi input login dengan memastikan email dan password telah diisi sebelum proses login dilanjutkan.(SEBAGAI PENGINGAT)**

FORM EDIT LANGGANAN



Tampilan ini merupakan **halaman edit langganan** yang digunakan pengguna untuk memperbarui nama layanan, harga, siklus langganan, dan tanggal jatuh tempo langganan digital.

1. localhost:3000 says

Langganan berhasil diperbarui!

OK

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

  // Contoh kirim ke REST API
  console.log(formData);

  alert("Langganan berhasil diperbarui!");
};
```

Kode ini digunakan untuk **mensimulasikan pengiriman data hasil edit ke REST API** dan **menampilkan notifikasi bahwa langganan berhasil diperbarui**.

2.

Edit Langganan

Perbarui detail langganan digital Anda

```
return (
  <div className="container">
    <div className="card">
      <h1>Edit Langganan</h1>
      <p className="subtitle">
        Perbarui detail langganan digital Anda
      </p>
```

Kode ini berfungsi untuk **menampilkan struktur utama halaman edit langganan berupa container dan card dengan judul serta deskripsi singkat.**

3.

Harga (Rp)

17000

```
<div className="form-group">
  <label>Harga (Rp)</label>
  <input
    type="number"
    name="harga"
    placeholder="150000"
    value={formData.harga}
    onChange={handleChange}
    required
  />
</div>
```

Kode ini digunakan untuk **menyediakan input harga langganan dalam bentuk angka dan menyimpan perubahannya ke state form.**

4.

Siklus Langganan

Bulanan

```
</div>

<div className="form-group">
  <label>Siklus Langganan</label>
  <select
    name="siklus"
    value={formData.siklus}
    onChange={handleChange}>
    <option value="Bulanan">Bulanan</option>
    <option value="Tahunan">Tahunan</option>
  </select>
</div>
```

Kode ini berfungsi untuk **menyediakan pilihan siklus langganan (bulanan atau tahunan)** dan menyimpan pilihan pengguna ke dalam state form.

5.

Tanggal Jatuh Tempo

21/12/2026



```
<div className="form-group">
  <label>Tanggal Jatuh Tempo</label>
  <input
    type="date"
    name="jatuhTempo"
    value={formData.jatuhTempo}
    onChange={handleChange}
    required
  />
</div>
```

Kode ini digunakan untuk **menyediakan input tanggal jatuh tempo langganan** dan menyimpan perubahannya ke dalam state form.

6.

Nama Layanan

Netflix / Spotify

```
<form onSubmit={handleSubmit}>
  <div className="form-group">
    <label>Nama Layanan</label>
    <input
      type="text"
      name="nama"
      placeholder="Netflix / Spotify"
      value={formData.nama}
      onChange={handleChange}
      required
    />
  </div>
```

Kode ini berfungsi untuk menyediakan input nama layanan langganan serta menangani pengiriman form saat proses edit dilakukan.

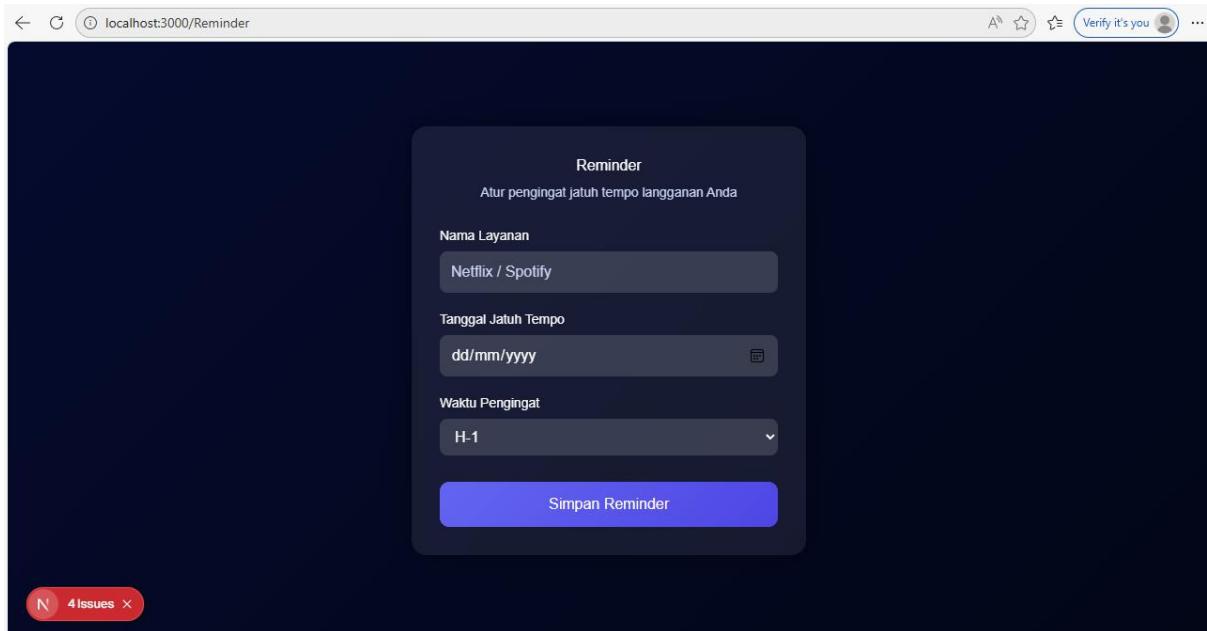
7.

Simpan Langganan

```
</div>
<button type="submit" className="btn-submit">
  Simpan Langganan
</button>
</form>
</div>
```

Kode ini digunakan untuk menampilkan tombol “Simpan Langganan” yang akan mengirim form untuk menyimpan perubahan data langganan.

REMINDER



Tampilan ini merupakan **halaman reminder** yang digunakan untuk mengatur pengingat jatuh tempo langganan digital berdasarkan nama layanan, tanggal jatuh tempo, dan waktu pengingat.

1. **Reminder**

Atur pengingat jatuh tempo langganan Anda

```
return (
  <div className="container">
    <div className="card">
      <h1>Reminder</h1>
      <p className="subtitle">
        Atur pengingat jatuh tempo langganan Anda
      </p>
```

Kode ini berfungsi untuk **menampilkan struktur utama halaman reminder berupa judul dan deskripsi singkat sebagai panduan pengguna dalam mengatur pengingat langganan**.

2. localhost:3000 says

Reminder berhasil disimpan!

OK

```
Windsurf: Refactor | Explain | Generate JSDoc | X
const handleSubmit = (e: React.FormEvent) => {
  e.preventDefault();
  console.log(reminder);
  alert("Reminder berhasil disimpan!");
};
```

Kode ini digunakan untuk **menangani pengiriman form reminder, menampilkan data reminder di console, dan memberi notifikasi bahwa reminder berhasil disimpan.**

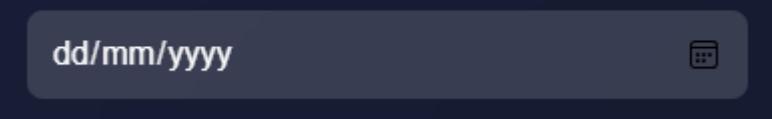
3. Nama Layanan

Netflix / Spotify

```
<form onSubmit={handleSubmit}>
  <div className="form-group">
    <label>Nama Layanan</label>
    <input
      type="text"
      name="layanan"
      placeholder="Netflix / Spotify"
      value={reminder.layanan}
      onChange={handleChange}
      required
    />
  </div>
```

Kode ini berfungsi untuk **menyediakan input nama layanan pada form reminder dan menyimpan perubahan nilai ke state reminder saat pengguna mengetik.**

4. Tanggal Jatuh Tempo

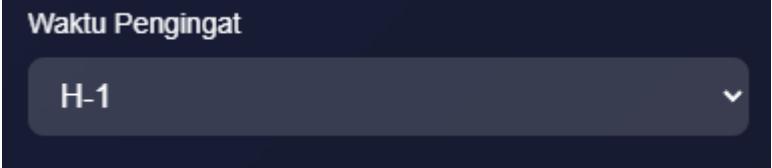


dd/mm/yyyy

```
<div className="form-group">
  <label>Tanggal Jatuh Tempo</label>
  <input
    type="date"
    name="tanggal"
    value={reminder.tanggal}
    onChange={handleChange}
    required
  />
</div>
```

Kode ini digunakan untuk **menyediakan input tanggal jatuh tempo pada form reminder dan menyimpan nilainya ke state reminder saat pengguna memilih tanggal.**

5. Waktu Pengingat

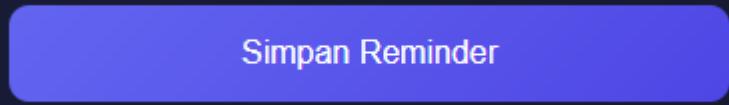


H-1

```
<div className="form-group">
  <label>Waktu Pengingat</label>
  <select
    name="waktu"
    value={reminder.waktu}
    onChange={handleChange}
  >
    <option value="1">H-1</option>
    <option value="3">H-3</option>
    <option value="7">H-7</option>
  </select>
</div>
```

Kode ini berfungsi untuk **menyediakan pilihan waktu pengingat (H-1, H-3, H-7) pada form reminder dan menyimpan pilihan pengguna ke state reminder.**

6.

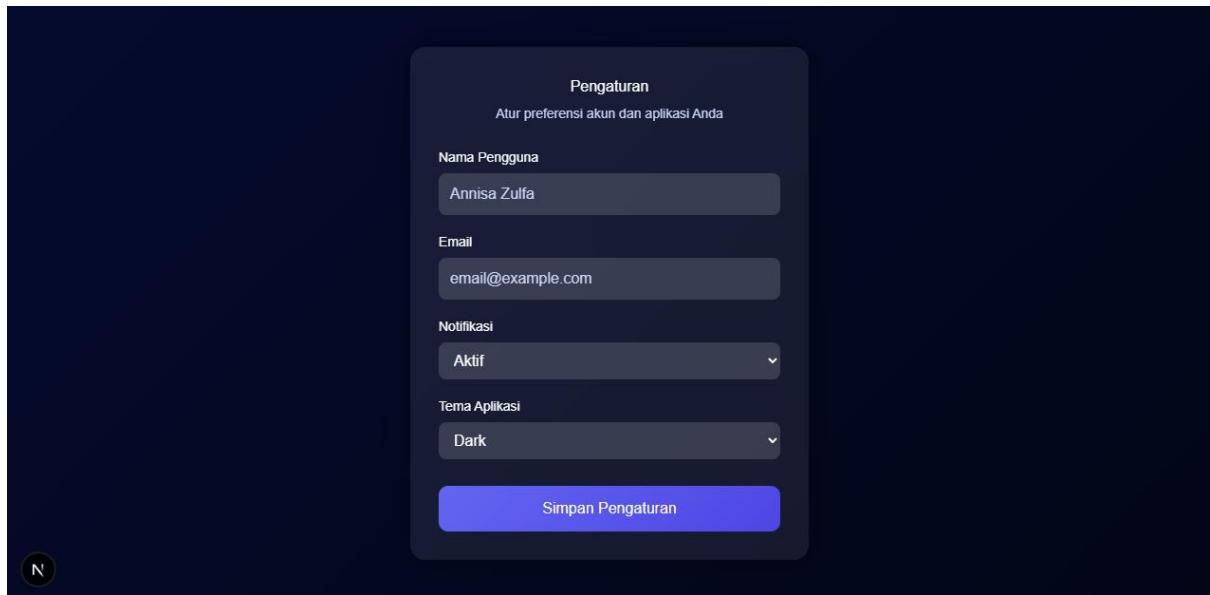


Simpan Reminder

```
</div>
<button type="submit" className="btn-submit">
  Simpan Reminder
</button>
</form>
</div>
```

Kode ini digunakan untuk menampilkan tombol “Simpan Reminder” yang akan mengirim form dan menyimpan pengaturan pengingat langganan yang telah diisi pengguna.

SETTING



Sama dengan program sebelumnya untuk setting yang berbeda hanya pada tema aplikasi nya saja dengan code sebagai berikut :

```
<div className="form-group">
  <label>Tema Aplikasi</label>
  <select
    name="tema"
    value={settings.tema}
    onChange={handleChange}>
    <option value="Dark">Dark</option>
    <option value="Light">Light</option>
  </select>
</div>
```