

Dokumentasi Projek Pemrograman Web Berorientasi Layanan

# **Implementasi RoomSync sebagai Sistem Reservasi Hotel Berbasis Arsitektur API**



Disusun Oleh:

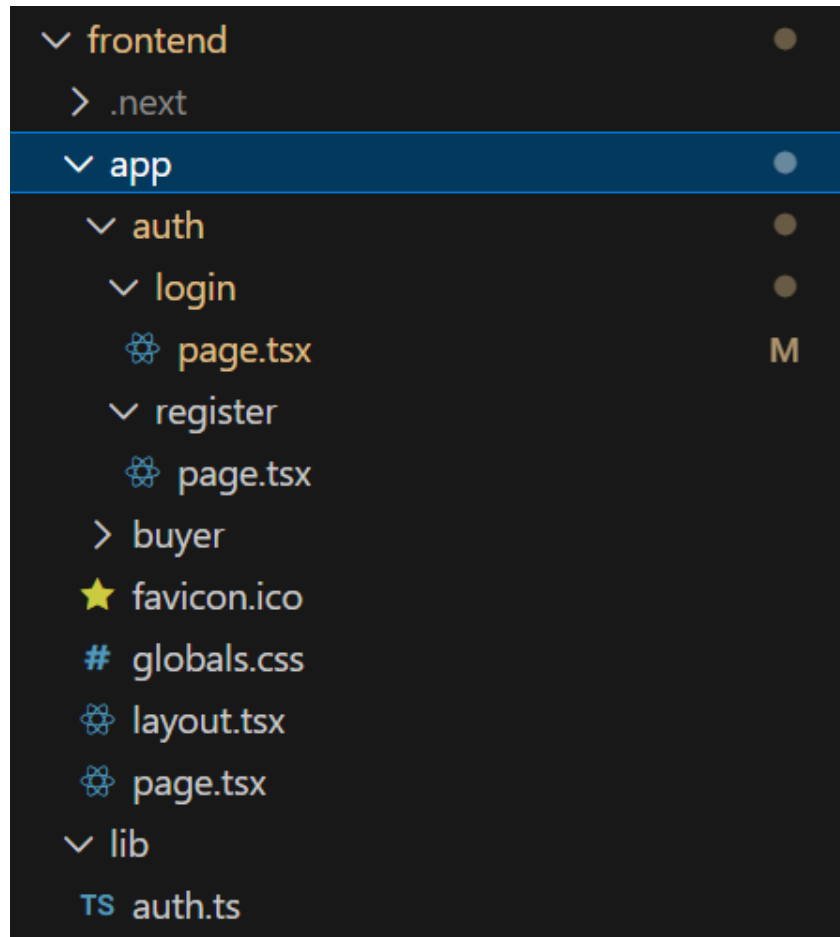
Nama : Gery Prananta

NPM : 23313002

Program Studi S1 Teknologi Informasi  
Fakultas Teknik dan Ilmu Komputer  
Universitas Teknokrat Indonesia  
2026

## 1. Struktur File

Dalam proyek PWBL ini, saya bergabung dalam salah satu kelompok yang membahas **Implementasi RoomSync sebagai Sistem Reservasi Hotel Berbasis Arsitektur API**. Penelitian ini mengembangkan RoomSync, yaitu sistem reservasi hotel berbasis web yang menggunakan arsitektur API untuk mempermudah proses pemesanan kamar serta memastikan data tersedia secara cepat, terstruktur, dan mudah diintegrasikan



Pada proyek ini, saya bertugas mengerjakan bagian **frontend**. Saya telah membuat beberapa folder dan file yang sudah dibagi sebelumnya, di antaranya folder **auth** yang berisi subfolder **login** dan **register**, di mana masing-masing subfolder memiliki file `page.tsx` sebagai halaman utama. Untuk menangani logika autentikasi, saya membuat folder **lib** yang berisi file `auth.ts`. Selain itu, saya juga membuat program utama sebagai tampilan awal aplikasi, yaitu file `page.tsx`, `layout.tsx`, serta `globals.css` yang digunakan untuk mengatur struktur dan tampilan keseluruhan web

## 2. page.tsx

Program page.tsx ini digunakan sebagai halaman utama (landing page) pada aplikasi **VGA International Hotel Booking**. Halaman ini digunakan untuk tampilan pertama yang dilihat oleh pengguna ketika membuka web. Di dalamnya ditampilkan judul aplikasi, deskripsi singkat tentang layanan hotel, serta beberapa tombol navigasi seperti **Home, Login, Register, dan Katalog Hotel** yang mengarahkan pengguna ke halaman autentikasi. Ini menunjukkan bahwa pengguna perlu melakukan login terlebih dahulu sebelum mengakses fitur utama aplikasi



Tampilan utama:

Tulisan VGA International Hotel

```
<h1
  style={{
    fontSize: 46, // Ukuran teks besar
    color: "gold", // Warna emas
    marginBottom: 14, // Jarak bawah judul
    letterSpacing: 1.5, // Jarak antar huruf
    textTransform: "uppercase", // Huruf kapital semua
  }}
>
  VGA International Hotel
</h1>
```

Deskripsi

```
<p
  style={{
    fontSize: 18, // Ukuran teks deskripsi
    color: "#eee", // Warna abu terang
    maxWidth: 650, // Lebar maksimal teks
    margin: "0 auto", // Posisi teks di tengah
    lineHeight: 1.7, // Jarak antar baris
  }}
>
  Temukan pengalaman menginap terbaik dengan kemewahan kelas dunia.
  Pilih hotel favorit Anda dan lakukan booking hanya dalam hitungan detik.
</p>
```



Bagian ke 2 saya menambahkan tombol-tolbol tersebut, dengan arah tombol home, dan katalog hotel ke form login karena belum login:

Tombol home, login, registrasi, dan katalog hotel

```
<div
  style={{
    marginTop: 40, // Jarak dari hero section
    display: "flex", // Mengatur tombol secara horizontal
    justifyContent: "center", // Posisi tombol di tengah
    gap: 20, // Jarak antar tombol
    flexWrap: "wrap", // Agar responsif di layar kecil
  }}
>
  <Link href="/auth/login" className="btn-gold">
    Home
  </Link>
  <Link href="/auth/login" className="btn-white">
    Login
  </Link>
  <Link href="/auth/register" className="btn-white">
    Register
  </Link>
  <Link
    href="/auth/login"
    className="btn-gold"
    style={{ background: "#111" }}
  >
    Katalog Hotel
  </Link>
</div>
```

Selain itu, halaman ini juga menampilkan beberapa fitur yang menjelaskan keunggulan layanan, seperti kemewahan hotel, kualitas pelayanan, dan kemudahan proses booking. Secara keseluruhan, program ini bertujuan untuk memberikan gambaran awal tentang aplikasi, menarik minat pengguna, serta mengarahkan alur penggunaan agar sesuai dengan sistem autentikasi yang telah dibuat sebelumnya



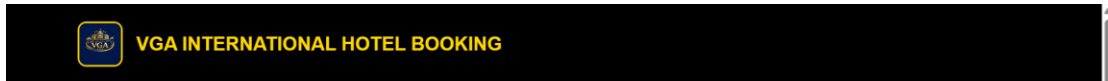
Kemudian bagian terakhir dari beranda utamanya:

Deskripsi keunggulan hotel

```
{
  title: "Kemewahan",
  text: "Hotel premium dengan kenyamanan kelas dunia.",
},
{
  title: "Pelayanan",
  text: "Tim profesional siap melayani Anda 24/7.",
},
{
  title: "Booking Cepat",
  text: "Proses pemesanan mudah dan aman.",
},
}
```

### 3. layout.tsx

Program layout.tsx ini berfungsi sebagai kerangka utama dari aplikasi VGA International Hotel Booking. File ini membungkus seluruh halaman yang ada di dalam folder app, sehingga elemen seperti header, footer, dan css global akan selalu muncul di setiap halaman. Dengan adanya layout ini, tampilan aplikasi menjadi konsisten tanpa perlu menulis ulang struktur yang sama di setiap halaman



Kemudian layout header dan footer, ini bagian header:

Logo hotel

```

```

Teks Vga International Hotel Booking

```
<h1
  style={{
    margin: 0, // Hilangkan margin default
    color: "gold", // Warna teks emas
    fontSize: 22, // Ukuran font
    whiteSpace: "nowrap", // Teks tidak turun baris
  }}
>
  VGA INTERNATIONAL HOTEL BOOKING
</h1>
```



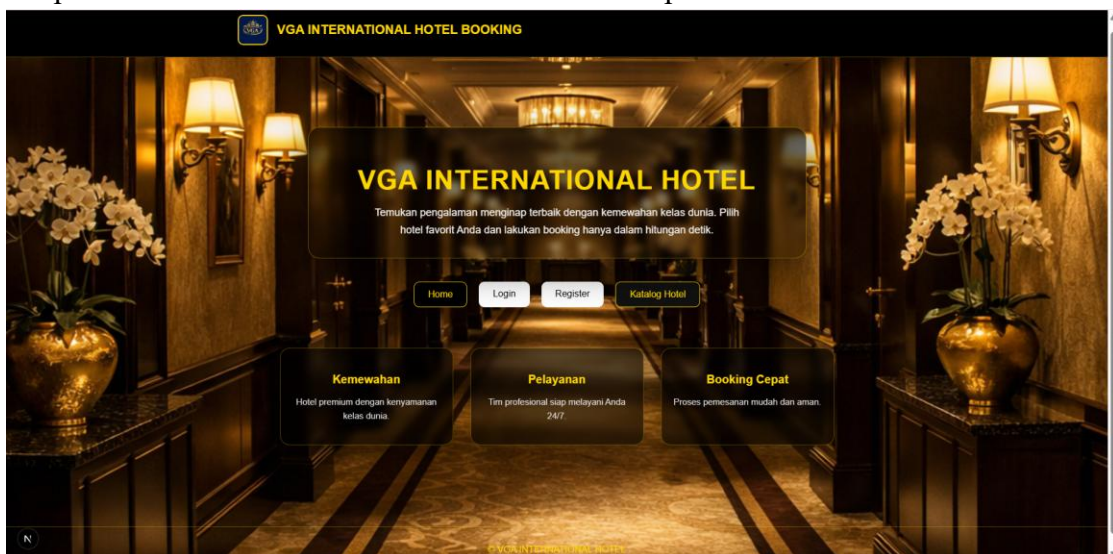
Dan ini bagian footer:

```
Teks © VGA INTERNATIONAL HOTEL
<footer
  style={{
    textAlign: "center", // Teks di tengah
    color: "gold", // Warna teks emas
    marginTop: 60, // Jarak dari konten utama
    padding: 30, // Ruang dalam footer
    borderTop: "1px solid rgba(255,215,0,0.3)", // Garis atas emas
  }}
>
  © VGA INTERNATIONAL HOTEL
</footer>
```

Pada file ini juga dijelaskan metadata seperti judul dan deskripsi web yang digunakan untuk judul tab browser dan kebutuhan SEO. Bagian header menampilkan logo serta nama aplikasi sebagai identitas web, bagian main digunakan untuk menampilkan konten halaman yang berubah-ubah melalui children, sedangkan footer berisi informasi hak cipta. Secara keseluruhan, layout.tsx berperan penting dalam mengatur struktur dasar dan tata letak aplikasi agar lebih rapi dan tersusun rapi

#### 4. global.css

CSS ini saya gunakan untuk mengatur tampilan keseluruhan web supaya terlihat rapi dan konsisten di setiap halaman. Pada bagian body, saya menghilangkan margin bawaan browser, mengatur font utama, dan menambahkan background gambar lorong hotel agar tampilan web terlihat lebih elegan dan sesuai dengan tema hotel. Style pada tag a dibuat supaya link tidak memiliki garis bawah dan warnanya mengikuti teks di sekitarnya. Class .card-shape saya gunakan untuk desain kartu konten, dengan background transparan, border emas, dan efek hover agar tampil lebih menarik. Untuk tombol, saya membuat dua jenis yaitu .btn-gold sebagai tombol utama dan .btn-white sebagai tombol sekunder, lengkap dengan efek hover supaya terlihat interaktif. Selain itu, class .nav-link digunakan pada menu navigasi untuk menambahkan efek garis bawah saat di-hover. Secara keseluruhan, CSS ini berfungsi untuk keren/mempercantik tampilan web dan memberikan kesan modern serta profesional



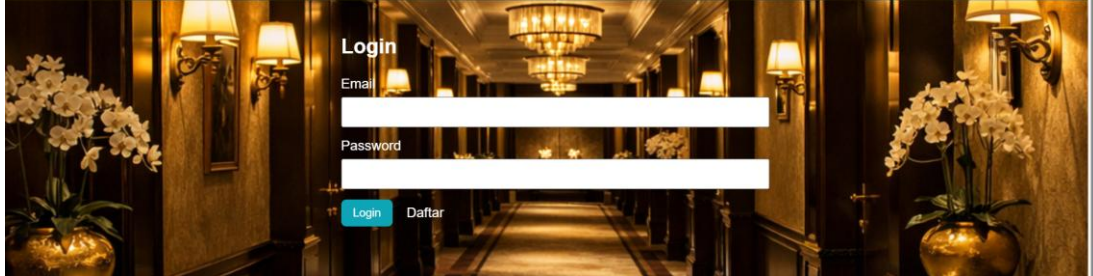
Di css bagian di bawah ini Adalah bagian unntuk background utama pada web ini

```
body {  
    margin: 0; /* Tetap menghilangkan margin */  
    font-family: "Poppins", Arial, sans-serif; /* Font tetap sama */  
    background: url("../public/lorong.png") center/cover no-repeat fixed;  
    /*  
        - center      : posisi gambar di tengah  
        - cover       : gambar menutupi layar  
        - no-repeat    : gambar tidak diulang  
        - fixed       : background tidak ikut scroll  
    */  
    color: white; /* Warna teks tetap putih */  
}
```



## 5. auth/login.tsx

Program auth/login/page.tsx ini digunakan sebagai halaman login user pada aplikasi VGA International Hotel Booking. Di file ini, user diminta memasukkan email dan password. Data yang di input kemudian dikirim ke API login menggunakan metode POST, lalu hasilnya diproses di sisi frontend



Ini adalah form loginnya:

Pengisian email

```
<form onSubmit={handleSubmit} style={{ display: "grid", gap: 12 }}>
  { /* Input Email */ }
  <label>
    Email
    <input
      type="email"
      value={email}
      onChange={(e) => setEmail(e.target.value)} // Update state email
      required
      style={{ width: "100%", padding: 8, marginTop: 6 }}
    />
  </label>
```

Pengisian password

```
  { /* Input Password */ }
  <label>
    Password
    <input
      type="password"
      value={password}
      onChange={(e) => setPassword(e.target.value)} // Update state password
      required
      style={{ width: "100%", padding: 8, marginTop: 6 }}
    />
  </label>
```

Tombol login dan daftar(register) nya

```
<div style={{ display: "flex", gap: 8, alignItems: "center" }}>
  {/* Tombol Login */}
  <button
    type="submit"
    disabled={loading} // Nonaktif saat loading
    style={{
      padding: "8px 14px",
      background: "#0ea5b7",
      color: "white",
      border: "none",
      borderRadius: 6,
    }}
  >
    {loading ? "Memproses..." : "Login"}
  </button>

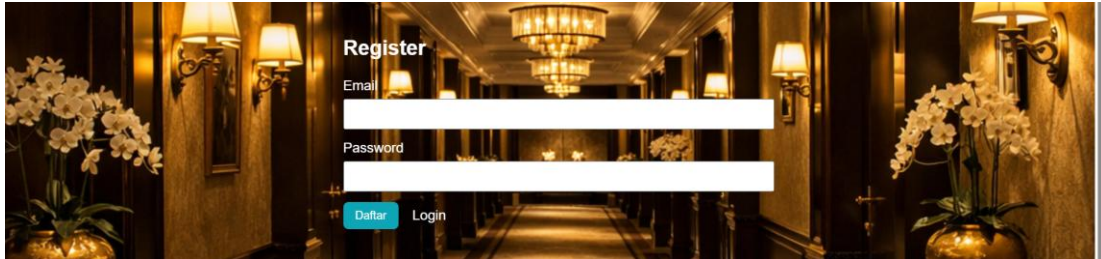
  {/* Link ke halaman register */}
  <a href="/auth/register" style={{ marginLeft: 8 }}>
    Daftar
  </a>
</div>
```

Jika login berhasil, data user akan disimpan ke localStorage sebagai penanda bahwa user sudah login. Setelah itu, sistem melakukan redirect: apabila user sebelumnya mengakses halaman booking (ditandai dengan parameter room), maka user akan diarahkan kembali ke halaman form booking, sedangkan jika login dilakukan secara normal, user akan diarahkan ke halaman beranda user. Secara keseluruhan, file ini berperan untuk menangani proses autentikasi user dari sisi tampilan dan mengatur alur navigasi setelah login

Setelah selesai mengisi form login dan menekan tombol login, user akan diarahkan ke tampilan buyer yang sudah di buat rekan saya

## 6. auth/register.tsx

Program auth/register/page.tsx ini digunakan sebagai halaman registrasi user pada aplikasi VGA International Hotel Booking. Halaman ini berfungsi untuk menampung data user baru berupa email dan password yang kemudian dikirim ke backend melalui API menggunakan metode POST



Form daftar:

Pengisian email

```
<form onSubmit={handleSubmit} style={{ display: "grid", gap: 12 }}>
  { /* Input Email */ }
  <label>
    Email
    <input
      type="email"
      value={email}
      onChange={(e) => setEmail(e.target.value)} // Update state email
      required
      style={{ width: "100%", padding: 8, marginTop: 6 }}
    />
  </label>
```

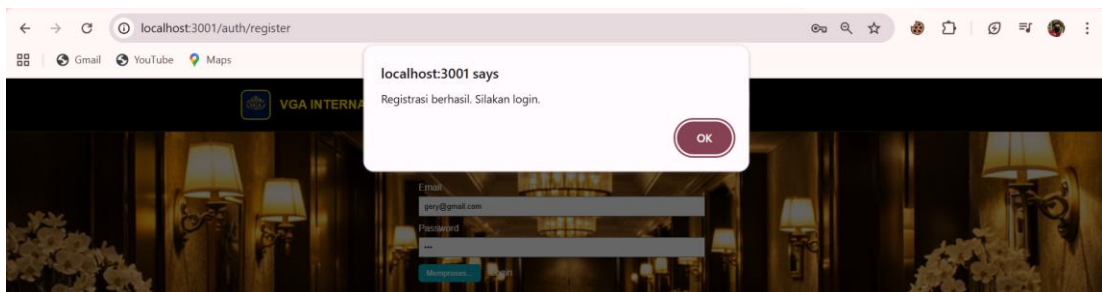
Pengisian password

```
{ /* Input Password */ }
<label>
  Password
  <input
    type="password"
    value={password}
    onChange={(e) => setPassword(e.target.value)} // Update state password
    required
    style={{ width: "100%", padding: 8, marginTop: 6 }}
  />
</label>
```

Tombol daftar(register) dan login nya

```
<div style={{ display: "flex", gap: 8, alignItems: "center" }}>
  {/* Tombol Daftar */}
  <button
    type="submit"
    disabled={loading} // Nonaktif saat loading
    style={{
      padding: "8px 14px",
      background: "#0ea5b7",
      color: "white",
      border: "none",
      borderRadius: 6,
    }}
  >
    {loading ? "Memproses..." : "Daftar"}
  </button>

  {/* Link menuju halaman login */}
  <a href="/auth/login" style={{ marginLeft: 8 }}>
    Login
  </a>
</div>
```



Jika proses registrasi berhasil, sistem akan menampilkan notifikasi bahwa registrasi sukses dan secara otomatis mengarahkan user ke halaman login agar bisa masuk ke aplikasi. Sebaliknya, jika terjadi kesalahan (misalnya email sudah terdaftar atau server bermasalah), pesan error akan ditampilkan. Secara keseluruhan, file ini berperan untuk menangani proses pendaftaran akun user dari sisi frontend serta mengatur alur perpindahan halaman setelah registrasi berhasil.

## 7. lib/auth.tsx

Program lib/auth.tsx

```
// lib/auth.ts
export function getUser() {
  if (typeof window === "undefined") return null;
  const user = localStorage.getItem("user");
  return user ? JSON.parse(user) : null;
}
export function isLoggedIn() {
  return !!getUser();
}
export function logout() {
  localStorage.removeItem("user");
}
```

File auth.ts berfungsi sebagai helper atau fungsi bantu untuk mengelola autentikasi user di sisi client. Modul ini digunakan untuk mengecek apakah user sudah login, mengambil data user yang sedang login, serta melakukan proses logout.

Fungsi getUser() digunakan untuk mengambil data user dari localStorage. Pada fungsi ini terdapat pengecekan `typeof window === "undefined"` untuk memastikan kode hanya dijalankan di sisi client, sehingga tidak menimbulkan error saat proses server-side rendering. Jika data user tersedia, maka data tersebut akan dikonversi dari format JSON menjadi objek JavaScript, sedangkan jika tidak ada data, fungsi akan mengembalikan nilai null.

Fungsi isLoggedIn() digunakan untuk mengecek status login user. Fungsi ini bekerja dengan memanfaatkan getUser(), di mana jika data user tersedia maka user dianggap sudah login.

Selain itu, terdapat fungsi logout() yang digunakan untuk menghapus data user dari localStorage. Fungsi ini dijalankan saat user melakukan logout agar sesi login benar-benar berakhir.