



Mengimplementasikan User Interface

J.620100.005.01



**ARIE RAHMADI,
S.T.**

INSTRUKTUR AHLI PERTAMA
KEJ. TIK

BALAI PELATIHAN VOKASI DAN
PRODUKTIVITAS SORONG

SISTEM KOMPUTER
UNIVERSITAS DIPONEGORO

TUJUAN

Peserta pelatihan mampu mengimplementasikan user interface menggunakan HTML, CSS, dan React sesuai standar desain UI yang baik

OUTLINE

Pengenalan
User Interface
(UI)

Tools yang
digunakan
(HTML, CSS,
React)

Mengidentifikasi
rancangan user
interface

Melakukan
implementasi
rancangan user
interface

APA ITU USER INTERFACE (UI)?



Tampilan visual dari aplikasi yang digunakan oleh pengguna



Tombol, form, layout, warna, teks, ikon



UI \neq UX (UI adalah tampilan, UX adalah pengalaman)

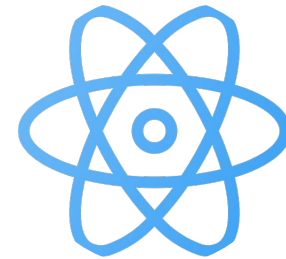
TOOLS



HTML



CSS



**REAC
T**



TOOLS HTML (PENGENALAN)

- **HTML (HyperText Markup Language)** adalah bahasa markup standar untuk membuat dan menyusun halaman web.
 - **Tujuan utama:** Menyusun konten (teks, gambar, link, form) agar dapat ditampilkan oleh browser.
 - **HTML bukan bahasa pemrograman**, melainkan bahasa markup (penanda).
-

TOOLS HTML (STRUKTUR DASAR)

BAGIAN	FUNGSI
<!DOCTYPE html>	Mendeklarasikan dokumen HTML5
<html>	Elemen induk semua isi halaman
<head>	Informasi meta (judul, charset, link CSS, dsb.)
<body>	Konten utama halaman yang ditampilkan ke pengguna

```
<!DOCTYPE html>
<html lang="id">
<head>
  <meta charset="UTF-8">
  <title>Judul Halaman</title>
</head>
<body>
  <h1>Halo Dunia!</h1>
  <p>Ini adalah paragraf pertama.</p>
</body>
</html>
```


TOOLS HTML (STRUKTUR DASAR)

Elemen	Fungsi
<h1> – <h6>	Judul (heading) dari besar ke kecil
<p>	Paragraf
	Tautan (link)
	Gambar
 / 	Daftar (bullet / angka)
<input> / <form>	Elemen formulir
<div> / <section>	Kontainer / bagian dari halaman

Setiap bagian dari HTML adalah **elemen**, biasanya memiliki struktur seperti ini:

<tagname>Isi konten</tagname>

TOOLS HTML (HIRARKI)

HTML memiliki struktur **bertingkat (nested)**
— elemen bisa berada di dalam elemen lain.

```
<body>
  <header>
    <h1>Judul Halaman</h1>
  </header>
  <main>
    <section>
      <h2>Konten Utama</h2>
      <p>Deskripsi isi...</p>
    </section>
  </main>
  <footer>
    <p>© 2025</p>
  </footer>
</body>
```



TOOLS CSS

- **CSS (Cascading Style Sheets)** adalah bahasa desain untuk mempercantik tampilan halaman web.
 - **Fungsi CSS:**
 - Mengatur warna, ukuran, font, dan posisi elemen HTML
 - Membuat halaman **responsif** (mobile friendly)
 - Mengelola layout dan tampilan antar halaman secara konsisten
-

TOOLS CSS (CARA PENULISAN)

Cara	Penjelasan	Contoh
Inline	Di dalam atribut HTML	<code><p style="color: red;">Teks merah</p></code>
Internal	Di dalam tag <code><style></code> pada <code><head></code>	<code><style>p { color: blue; }</style></code>
Eksternal	Di file <code>.css</code> terpisah lalu di-link	<code><link rel="stylesheet" href="style.css"></code>

TOOLS CSS (STRUKTUR DASAR)

```
h1 {  
  color: green;  
  font-size: 24px;  
}
```

- h1 → selector (elemen yang ditarget)
 - color & font-size → properti CSS
 - green & 24px → nilai properti
-



TOOLS REACT

- **React** adalah *JavaScript library* yang digunakan untuk membangun **User Interface (UI)** secara **komponen**.
 - Dikembangkan oleh **Facebook**
 - Fokus pada **tampilan (view)** dalam arsitektur *MVC*
 - Cocok untuk membuat aplikasi web modern dan interaktif
-

TOOLS REACT (KONSEP INTI)

Konsep	Penjelasan
Component	Blok UI yang dapat digunakan ulang
JSX	Sintaks penulisan mirip HTML di JavaScript
State	Data lokal dalam komponen
Props	Data yang dikirim dari komponen induk ke anak
Hook	Fitur React seperti useState , useEffect

TOOLS REACT (STRUKTUR & SIFAT)

STRUKTUR

```
function Greeting() {  
  return <h1>Halo, Dunia!</h1>;  
}
```

Fungsi Greeting() adalah **komponen**
return menghasilkan **elemen UI** dalam JSX
JSX = HTML + JavaScript

BERSIFAT DINAMIS

```
const [count, setCount] = useState(0);
```

```
<button onClick={() => setCount(count + 1)}>  
  Klik: {count}  
</button>
```

useState → React Hook untuk menyimpan dan memperbarui state

UI akan **otomatis merender ulang** saat state berubah

TOOLS REACT (INSTALASI)

- **npm create vite@latest my-app --template react**
- **cd my-app**
- **npm install**

Bagian Perintah	Artinya
npm create vite@latest	Menjalankan perintah pembuatan project Vite versi terbaru
my-app	Nama folder project (bisa diganti dengan nama lain)
--template react	Template project yang digunakan adalah React (bukan vanilla HTML/CSS)
cd my-app	Masuk ke folder project my-app
npm install	Mengunduh dan menginstal semua dependency diperlukan



TOOLS REACT (AXIOS)

- Digunakan untuk **mengambil/mengirim data** dari dan ke server (REST API).
- **Memudahkan komunikasi** antara frontend (React) dan backend (Node.js, Laravel, dsb).
- Contoh penggunaan: menampilkan daftar peserta dari database.
- Install Axios

npm install axios



TOOLS REACT (ROUTER)

- Digunakan untuk membuat **navigasi antar halaman** dalam aplikasi React.
 - Memungkinkan aplikasi menjadi **SPA (Single Page Application)**, di mana navigasi tidak memuat ulang seluruh halaman.
 - Contoh penggunaan: pindah dari halaman Beranda ke halaman Profil atau Laporan.
 - Install React Router
`npm install react-router-dom`
-

TOOLS REACT (TAILWIND CSS)

- Framework CSS berbasis **utility class** seperti `bg-blue-500, text-center, p-4`.
- Mempercepat pembuatan tampilan tanpa menulis CSS panjang.
- Cocok untuk developer karena langsung styling di elemen React/JSX.
- Install Tailwind CSS
`npm install tailwindcss @tailwindcss/vite`
- Konfigurasi vite plugin
`import tailwindcss from '@tailwindcss/vite'` – tambakan pada bagian import
`tailwindcss()`, – tambakan pada bagian plugins
- Import Tailwind CSS
`@import "tailwindcss";` – tambakan pada bagian atas index.css

Mengidentifikasi Rancangan User Interface

Siapa
penggunanya?
(admin, user umum,
teknisi, dsb.)

Apa kebutuhannya?
(login, input data,
melihat laporan)

Apa perangkatnya?
(desktop, mobile)

Mengidentifikasi Komponen User Interface Dialog Sesuai Konteks

01

Komponen UI **Dialog**:
komponen interaktif yang
memicu aksi pengguna.

02

Komponen Umum:
Button
Input field
Modal/Dialog Box
Dropdown, Checkbox

03

Identifikasi Berdasarkan Proses:
Jika prosesnya **login**, maka
butuh:
Username input
Password input
Tombol "Login"

Menjelaskan Urutan dari Akses Komponen User Interface

Urutan interaksi UI harus **logis dan ramah pengguna**

Gunakan tab-index di HTML

Gunakan urutan alami (top-to-bottom, left-to-right)

Hindari elemen yang membingungkan atau tidak perlu

Membuat Simulasi (Mock-up)

01

Mock-up:
Representasi visual dari
UI yang akan
dikembangkan

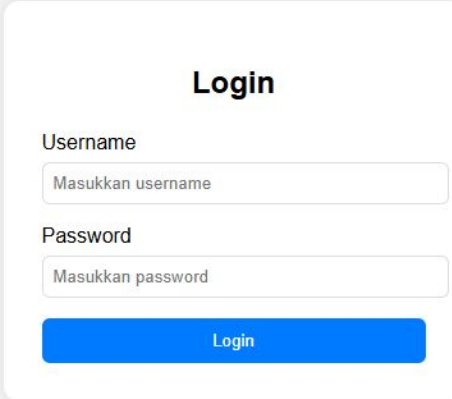
02

Bisa berupa:
Sketsa tangan
figma / balsamiq
langsung dengan HTML
+ CSS (high-fidelity
prototype)

03

Tujuan mockup:
Menyepakati desain
sebelum implementasi
memastikan alur dan
fungsi UI tepat

Contoh Mockup Sederhana: Form Login



Login

Username

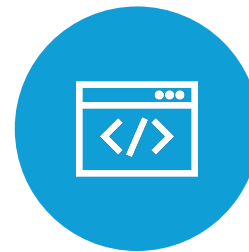
Password

Login

Contoh Mockup Sederhana: Form Login

Folder dan File

- Buat folder Mockup
- Buat file index.html dan style.css dalam folder Mockup
- Tulis kode program



Contoh
Kode Program

Job Sheet Mockup

Form Registrasi

Nama Lengkap

Email

Password

Daftar

Melakukan Implementasi User Interface


Implementasi = Mewujudkan Mock-up menjadi halaman kerja nyata

- **HTML** → Struktur konten
 - **CSS/Tailwind** → Gaya dan tata letak
 - **React** → Komponen interaktif dan modularisasi
-



Menerapkan Menu Program Sesuai Rancangan

- Menu program adalah **navigasi utama** yang digunakan pengguna untuk berpindah antar halaman atau fitur dalam aplikasi.
-



Menerapkan Menu Program Sesuai Rancangan (React + Tailwind)

Menu Program

Beranda Data Peserta

Menerapkan Menu Program Sesuai Rancangan (React + Tailwind)

Instalasi

- [Inisialisasi Project React dengan Vite](#)
- [Instalasi Tailwind CSS](#)
- [Instalasi React Router DOM](#)
- [Instalasi Luide-React](#)

Folder dan File

- Buat folder components dan pages dalam folder src
 - Buat file Navbar.jsx, Layout.jsx, dan Footer.jsx dalam folder components
 - Buat file DataPeserta.jsx, dan Home.jsx, dalam folder pages
-

Menerapkan Menu Program Sesuai Rancangan (React + Tailwind)



Contoh
Kode Program

Tulis Kode Program

- Navbar.jsx
- Layout.jsx
- Footer.jsx
- DataPeserta.jsx,
- Home.jsx

Setup Routing

- App.jsx

Mengatur Penempatan UI Dialog Secara Sekuensial



Dialog: Form, modal, popup input



Sekuensial = Sesuai urutan proses (step-by-step)

Mengatur Penempatan UI Dialog Secara Sekuensial (React + Tailwind)

Penempatan UI Dialog

[Beranda](#) [Data Peserta](#) [Register](#)

Form Pendaftaran

Kirim

Mengatur Penempatan UI Dialog Secara Sekuensial (React + Tailwind)

Instalasi

- [Inisialisasi Project React dengan Vite](#)
- [Instalasi Tailwind CSS](#)
- [Instalasi React Router DOM](#)
- [Instalasi Luide-React](#)
- [Instalasi Axios](#)

Folder dan File

- Ulangi langkah kerja Menerapkan Menu Program
- Buat file .env dalam root folder
- Buat file FormPendaftaran.jsx dalam folder pages
- Buat folder config
- Buat file Constants.jsx dalam folder config

Mengatur Penempatan UI Dialog Secara Sekuensial (React + Tailwind)



Contoh
Kode Program

Tulis Kode Program

- .env
- FormPendaftaran.jsx
- Constants.jsx

Update Kode Program

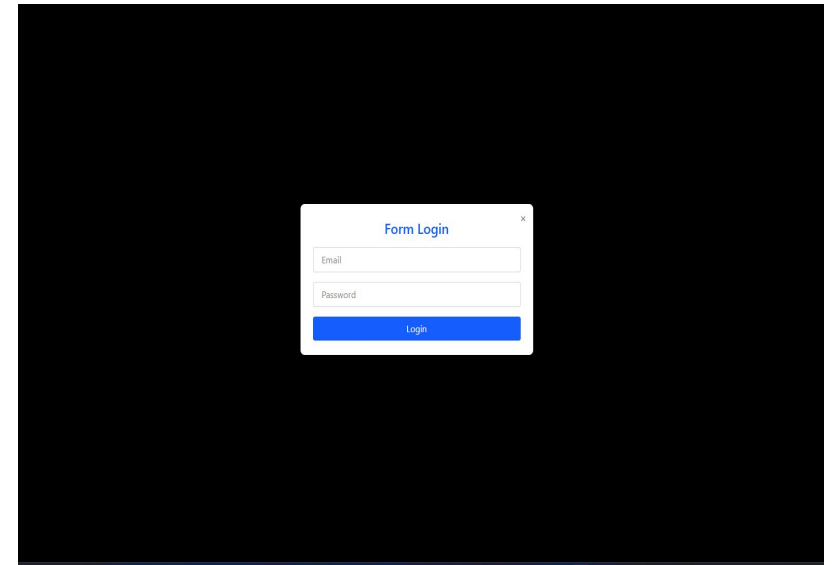
- Navbar.jsx
- App.jsx
- Footer.jsx

Mengatur Aktif-Pasif Komponen Dialog

Komponen **aktif**:
bisa diisi/diklik

Komponen
pasif: disable,
tidak bisa
diakses
(sementara)

Mengatur Aktif-Pasif Komponen Dialog (React + Tailwind)



Mengatur Aktif-Pasif Komponen Dialog (React + Tailwind)

Instalasi

- [Inisialisasi Project React dengan Vite](#)
- [Instalasi Tailwind CSS](#)
- [Instalasi React Router DOM](#)
- [Instalasi Luide-React](#)
- [Instalasi Axios](#)

Folder dan File

- Ulangi langkah kerja Mengatur Penempatan UI Dialog
- Buat folder context dan Routes
- Buat file AuthContext.jsx dalam folder context
- Buat file Index.jsx dalam folder routes
- Buat file Login.jsx dalam folder pages

Mengatur Aktif-Pasif Komponen Dialog (React + Tailwind)



Contoh
Kode Program

Tulis Kode Program

- AuthContext
- Index.jsx
- Login.jsx

Update Kode Program

- Navbar.jsx
- App.jsx
- Footer.jsx



Menentukan Bentuk Style Komponen UI

Gunakan prinsip desain UI yang baik:

- **Konsistensi:** semua tombol bentuk & warna seragam
 - **Aksesibilitas:** kontras warna cukup
 - **Responsivitas:** tampil baik di berbagai ukuran layar
-

Menentukan Bentuk Style Komponen UI (React + Tailwind)

Data Peserta

+ Tambah Peserta			
No.	Nama	Status	Aksi
1	ani	aktif	<button>View</button> <button>Edit</button> <button>Hapus</button>
2	ana	aktif	<button>View</button> <button>Edit</button> <button>Hapus</button>
3	anu	aktif	<button>View</button> <button>Edit</button> <button>Hapus</button>
4	ano	aktif	<button>View</button> <button>Edit</button> <button>Hapus</button>
5	ane	nonaktif	<button>View</button> <button>Edit</button> <button>Hapus</button>



Menentukan Bentuk Style Komponen UI (React + Tailwind)

Instalasi

- [Inisialisasi Project React dengan Vite](#)
- [Instalasi Tailwind CSS](#)
- [Instalasi React Router DOM](#)
- [Instalasi Luide-React](#)
- [Instalasi Axios](#)
- [Instalasi Sweetalert2](#)

Folder dan File

- Ulangi langkah kerja Mengatur Aktif-Pasif Komponen Dialog
 - Buat file Button.jsx, Form.jsx Input.jsx, dan Table.jsx dalam folder components
 - Buat file FormPeserta.jsx dalam folder pages
-

Menentukan Bentuk Style Komponen UI (React + Tailwind)



Contoh
Kode Program

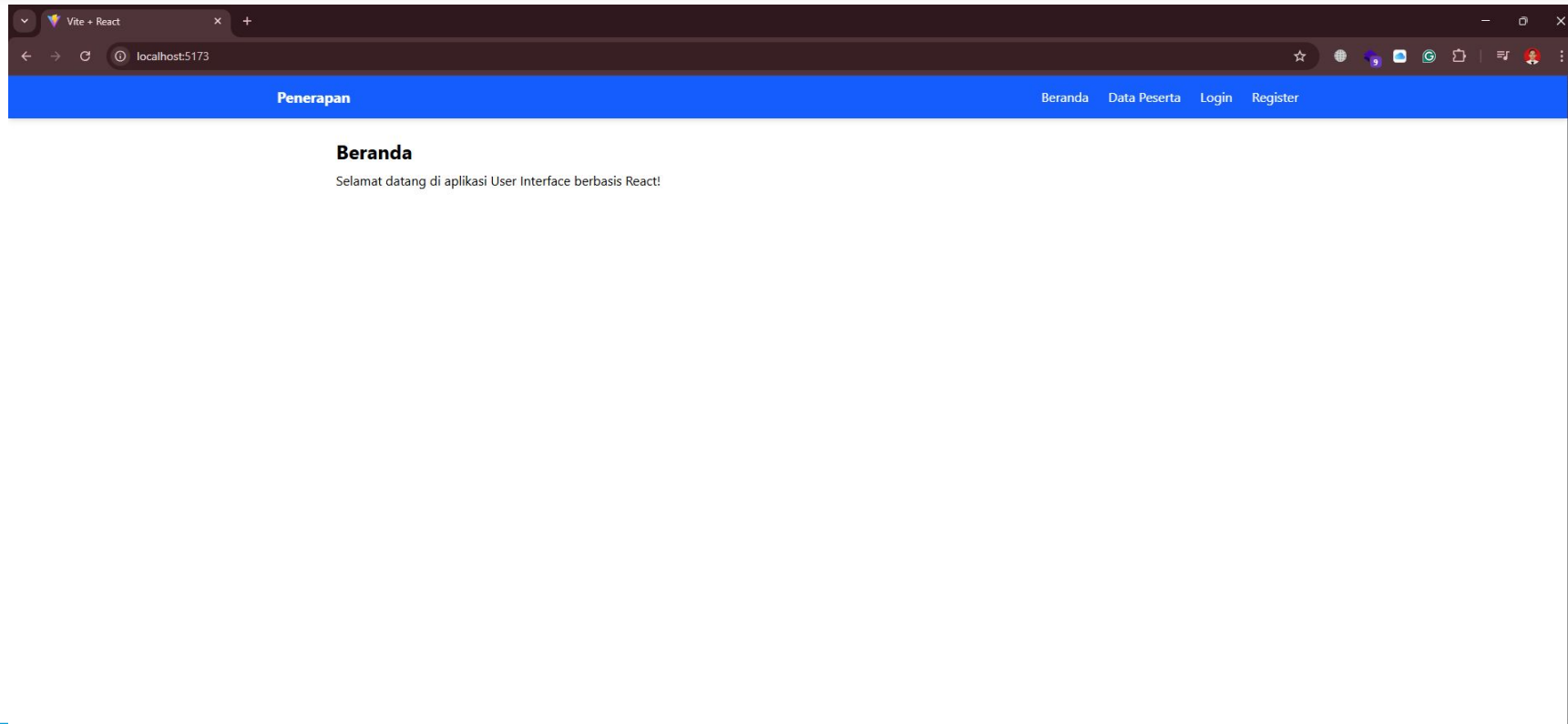
Tulis Kode Program

- Button.jsx
- Form.jsx
- Input.jsx
- Table.jsx
- FormPeserta.jsx

Update Kode Program

- DataPeserta.jsx
- Navbar.jsx
- Index.jsx
- Footer.jsx

Penerapan UI (React + Tailwind)



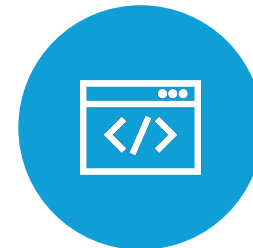
Penerapan UI (React + Tailwind)

Instalasi

- [Inisialisasi Project React dengan Vite](#) – ubah **my-app** menjadi **penerapan-ui**
- [Instalasi Tailwind CSS](#)
- [Instalasi React Router DOM](#)
- [Instalasi Luide-React](#)
- [Instalasi Axios](#)
- [Instalasi Sweetalert2](#)

Folder dan File

- Ulangi seluruh langkah kerja yang sebelumnya telah dilakukan



Contoh
Kode Program

Job Sheet Penerapan UI

