

Pemrograman Web Modern

Insan Taufik, S.Kom., M.Kom.



Pengantar Pemrograman Web Modern & React JS

Sub-Materi

Konsep SPA (Single Page Application). Apa itu React? Sejarah, keunggulan, dan ekosistem React. Persiapan Lingkungan (Node.js, npm/yarn, VS Code). Membuat proyek React pertama (Vite/Create React App).

Tujuan Pembelajaran (Capaian)

Memahami tren web modern, mengenal React sebagai library utama, dan berhasil menjalankan proyek React sederhana.

Web Klasik vs Web Modern

Web Klasik (Multi-Page Application - MPA)

Setiap klik pindah halaman, full reload server. (Contoh: Website statis/PHP tradisional).

VS

Web Modern (Single-Page Application - SPA)

Hanya reload sebagian konten, navigasi cepat. (Contoh: Gmail, Twitter, Aplikasi yang akan kita buat).

Keunggulan Single-Page Application - SPA

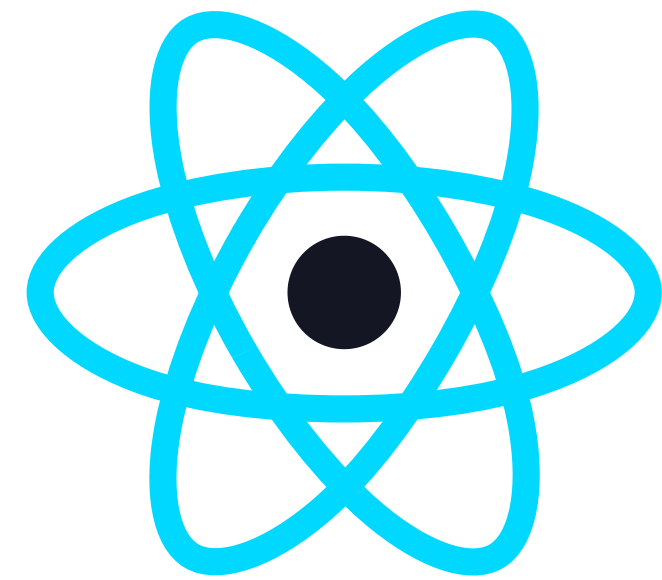
- ⚡ Respons Cepat, Hanya bagian tertentu dari halaman yang diperbarui tanpa reload penuh, sehingga interaksi terasa cepat seperti aplikasi desktop.
- 📦 Penghematan Bandwidth, Resource utama (HTML, CSS, JS) diunduh sekali di awal — setelah itu hanya data (JSON, API) yang ditukar dengan server.
- 💻 Pengalaman Pengguna (UX) Lebih Halus, Navigasi terasa mulus tanpa transisi halaman yang patah, cocok untuk aplikasi modern seperti Gmail atau Trello.
- 🔄 Pemanfaatan API Lebih Efisien, SPA berkomunikasi dengan backend lewat REST API atau GraphQL, memisahkan frontend dan backend dengan jelas (decoupled architecture).
- 🚀 Mudah Dikembangkan dan Diperluas, Framework seperti React, Vue, atau Angular memudahkan modularisasi, reusability, dan integrasi dengan teknologi lain.

Tantangan Single-Page Application - SPA

- 🕵️🔗 SEO Sulit, Karena konten dimuat secara dinamis lewat JavaScript, mesin pencari bisa kesulitan mengindeks halaman SPA (kecuali menggunakan server-side rendering).
- 🛡️ Manajemen Keamanan Lebih Kompleks, SPA banyak berinteraksi dengan API, sehingga perlu perhatian ekstra pada autentikasi (misalnya token JWT) dan proteksi data.
- 🧭 Navigasi & History Rumit, Harus mengelola URL dan riwayat browser secara manual (misalnya dengan React Router) agar fitur “Back/Forward” tetap berfungsi dengan benar.
- 🐢 Waktu Muat Awal Lebih Lama, Semua script (React, Vue, library, dll.) diunduh di awal, sehingga initial load bisa terasa lambat pada koneksi lemah.
- 🧰 Debugging & Monitoring Lebih Sulit, Karena logika banyak di sisi client, pelacakan error atau performa butuh alat tambahan seperti Sentry, Redux DevTools, atau Chrome DevTools.

Ekosistem Frontend Modern

- 🧩 1. HTML, CSS, JavaScript (ES6+) – Core
 - HTML (HyperText Markup Language) → Struktur dasar halaman web (membentuk elemen seperti teks, gambar, form).
 - CSS (Cascading Style Sheets) → Mengatur tampilan dan layout (warna, font, posisi, animasi).
 - JavaScript (ES6+) → Menambahkan interaktivitas dan logika dinamis; versi ES6+ membawa fitur modern seperti let/const, arrow function, module, dan class.
- ⚙️ 2. Library/Framework (React, Angular, Vue)
 - React → Library dari Meta untuk membangun UI berbasis komponen; fokus pada tampilan (View).
 - Angular → Framework lengkap dari Google; punya sistem modular, routing, dan dependency injection bawaan.
 - Vue → Framework progresif; mudah dipelajari dan fleksibel, bisa diintegrasikan secara bertahap ke proyek lama.
- 🏗️ 3. Build Tools (Webpack, Vite)
 - Webpack → Bundler yang menggabungkan file JS, CSS, dan aset lain menjadi satu paket siap produksi.
 - Vite → Build tool modern yang sangat cepat; menggunakan ES modules dan dev server instan untuk pengembangan.



React

Apa itu react

1. Definisi

React adalah library JavaScript yang digunakan untuk membangun Antarmuka Pengguna (User Interface) secara komponen-based. Artinya, tampilan web dipecah menjadi bagian-bagian kecil (komponen) yang dapat digunakan kembali dan dikelola dengan mudah.

2. Bukan Framework! (Perbedaan Library dan Framework)

- Library → Kamu memanggil fungsinya sesuai kebutuhan (kamu yang mengontrol alur aplikasi).
- Framework → Framework yang memanggil kode kamu (alur ditentukan oleh framework).
- React hanya fokus pada lapisan tampilan (View), bukan seluruh arsitektur aplikasi — itulah sebabnya ia disebut library, bukan framework.

3. Dibuat oleh Facebook (Jordan Walke)

React dikembangkan oleh Jordan Walke, seorang insinyur di Facebook, dan pertama kali dirilis pada tahun 2013. Awalnya digunakan untuk mempercepat dan mempermudah pembuatan UI dinamis di Facebook dan Instagram.

Konsep Utama React (Pondasi)

1. Komponen (Component)

Komponen adalah blok bangunan utama UI di React.

Setiap bagian tampilan (misalnya tombol, form, navbar) dibuat sebagai komponen terpisah yang bisa digunakan ulang dan digabung seperti potongan LEGO.

2. JSX (JavaScript XML)

JSX adalah sintaks khusus yang memungkinkan kamu menulis HTML di dalam JavaScript.

Contoh:

```
function Hello() {  
  return <h1>Hello, World!</h1>;  
}
```

JSX membuat kode lebih mudah dibaca dan mendekati struktur tampilan yang sebenarnya.

3. Props dan State

- Props (Properties) → Data dikirim dari luar komponen (bersifat tetap/read-only).
- State → Data internal komponen yang bisa berubah seiring interaksi pengguna.

Props dan State bekerja sama untuk membuat UI React dinamis dan interaktif.

Persiapan

Alat yang Dibutuhkan:

- Node.js & npm/yarn: Runtime & Package Manager (tunjukkan cara cek versi).
- Editor Teks: VS Code (sertakan rekomendasi extension seperti ES7 React/Redux/GraphQL Snippets).
- Web Browser: Chrome/Firefox (dengan React Developer Tools Extension).

Membuat Proyek React

⚙️ 1. Pilih Build Tool – Rekomendasi: Vite

Vite adalah build tool modern pengganti Create React App (CRA).

Keunggulannya:

- Startup proyek sangat cepat (hot reload instan).
- Mendukung ES Modules dan React 18 secara default.
- Ukuran bundle kecil dan waktu build cepat.

Membuat Proyek React

2. Perintah Membuat Proyek

Jalankan perintah berikut di terminal:

```
npm create vite@latest my-app -- --template react
```

Penjelasan:

- `npm create vite@latest` → memanggil tool pembuat proyek Vite versi terbaru.
- `my-app` → nama folder proyek yang akan dibuat.
- `--template react` → memilih template React sebagai basis proyek.

Membuat Proyek React

3. Struktur Folder Dasar

Berikut struktur utama proyek React dengan Vite:

```
my-app/  
|  
├─ public/           → File statis (ikon, gambar, dsb.)  
├─ src/              → Kode utama aplikasi  
|   ├─ App.jsx        → Komponen utama aplikasi  
|   ├─ main.jsx        → Entry point yang merender App  
|   └─ assets/         → Aset gambar/CSS khusus  
|  
├─ package.json      → Berisi daftar dependensi & script npm  
├─ index.html         → Template utama aplikasi  
└─ vite.config.js     → Konfigurasi build tool Vite
```

Penjelasan singkat:

- src/ → tempat menulis komponen React.
- public/ → file statis yang tidak di-bundle.
- package.json → daftar library & perintah seperti npm run dev.

Membuat Proyek React

👉 1. Ubah App.jsx

Buka file src/App.jsx, lalu ubah isinya menjadi:

```
function App() {  
  return <h1>Hello React!</h1>;  
}  
  
export default App;
```

File App.jsx adalah komponen utama (root component) yang menampilkan isi halaman — di sini hanya menampilkan teks sederhana “Hello React!”.

Membuat Proyek React

🧱 2. Peran index.html dan Elemen

#root

File index.html berada di folder utama proyek.

Di dalamnya terdapat elemen:

```
<div id="root"></div>
```

👉 Elemen #root inilah tempat (container) di mana seluruh aplikasi React akan “ditanam” (mounted).

React tidak mengganti seluruh halaman, tetapi hanya mengontrol isi dari elemen div#root ini.

Membuat Proyek React

⚙️ 3. Peran main.jsx — Tempat Mounting Komponen Utama
File src/main.jsx bertugas memasang (mount) komponen React ke dalam elemen #root.
Contoh isi default-nya:

Penjelasan:

- ReactDOM.createRoot() → memilih elemen #root di index.html.
- .render(<App />) → menampilkan komponen App ke dalam elemen tersebut.
- <React.StrictMode> → membantu mendeteksi error dan praktik buruk selama pengembangan.

```
import React from 'react'
import ReactDOM from 'react-dom/client'
import App from './App.jsx'

ReactDOM.createRoot(document.getElementById('root')).render(
  <React.StrictMode>
    <App />
  </React.StrictMode>,
)
```

Menjalankan Project

⚙️ 1. Instalasi Dependensi

Setelah membuat proyek (misalnya my-app), buka folder proyek di terminal:

```
cd my-app
```

Lalu jalankan perintah: `npm install`

Perintah ini akan mengunduh semua dependensi yang tercantum di file package.json (misalnya React, ReactDOM, Vite, dll). Tanpa langkah ini, proyek belum bisa dijalankan.

🚀 2. Menjalankan Server Development

Setelah instalasi selesai, jalankan: `npm run dev`

Perintah ini akan:

Menjalankan development server Vite,

Melakukan hot reload otomatis (setiap kali file disimpan, browser langsung diperbarui),

Menampilkan URL lokal, biasanya: `→ Local: http://localhost:5173/`

Tugas Mandiri/Penutup:

Mahasiswa diwajibkan untuk menginstal semua alat dan berhasil menjalankan proyek "Hello React" mereka sendiri sebelum pertemuan berikutnya.