

# Evolusi Web: HTML, REST API, SPA, PWA, hingga WebAssembly

Nazril Kanahaya Akbar

Politeknik Negeri Lhokseuawe Jalan Medan-Banda Aceh Km 280, Buketrata, Kota  
Lhokseumawe, Aceh

nzril389@gmail.com

**Abstrak:** Artikel ini membahas perkembangan teknologi web dari masa ke masa, dimulai dari era HTML tradisional hingga munculnya berbagai teknologi modern seperti REST API, SPA (Single Page Application), PWA (Progressive Web App), dan WebAssembly. Pembahasan mencakup sejarah singkat HTML, peran REST API dalam menghubungkan front-end dan back-end, konsep SPA yang membuat aplikasi web lebih dinamis, keunggulan PWA dalam menghadirkan pengalaman seperti aplikasi mobile, hingga potensi WebAssembly dalam meningkatkan performa aplikasi di browser. Artikel ini bertujuan memberikan pemahaman menyeluruh tentang evolusi dan inovasi dalam dunia pemrograman web.

**Kata Kunci:** *Teknologi Web, HTML, REST API, SPA, PWA, WebAssembly*

## PENDAHULUAN

Perkembangan teknologi web telah melalui berbagai tahap yang menarik. Dimulai pada awal 1990-an dengan terciptanya World Wide Web oleh Tim Berners-Lee, internet awalnya digunakan untuk bertukar informasi melalui halaman statis. HTML (HyperText Markup Language) digunakan untuk menyusun konten dasar, sementara HTTP (HyperText Transfer Protocol) memungkinkan komunikasi antar server dan klien.ilmiah.

Pada masa itu, website masih sangat sederhana, dengan desain statis yang hampir tidak memiliki interaktivitas. Pengguna hanya bisa membaca konten tanpa banyak interaksi langsung dengan website. Seiring berjalannya waktu, permintaan akan website yang lebih dinamis dan interaktif semakin meningkat. Pada pertengahan 2000-an, muncul teknologi baru seperti AJAX (Asynchronous JavaScript and XML), yang memungkinkan website untuk memperbarui konten tanpa harus memuat ulang seluruh halaman. Hal ini membuka jalan bagi munculnya konsep Single Page Application (SPA), yang memungkinkan aplikasi web berfungsi hampir seperti aplikasi desktop, dengan pengalaman pengguna yang lebih mulus.

Pada saat yang sama, API (App

lication Programming Interface) mulai berkembang. Salah satu yang paling signifikan adalah REST (Representational State Transfer), yang memungkinkan aplikasi berbicara satu sama lain secara lebih efisien dan sederhana, dengan menggunakan protokol HTTP yang sudah ada.

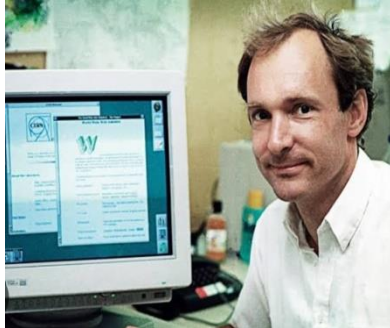
Lebih baru lagi, teknologi seperti WebAssembly (Wasm) dan Progressive Web Apps (PWA) semakin memperkaya dunia web. WebAssembly memungkinkan pengembang untuk menjalankan kode dalam berbagai bahasa pemrograman, termasuk C, C++, dan Rust, langsung di dalam browser dengan kecepatan hampir setara dengan aplikasi native. Sementara itu, PWA memungkinkan website untuk memiliki fungsionalitas seperti aplikasi mobile, seperti bekerja offline dan mengirim notifikasi push.

## PEMBAHASAN

### 1. Perkembangan Teknologi Web Tradisional

Pada tahun 1989, seorang ilmuwan komputer asal Inggris, **Tim Berners-Lee**, mengajukan sebuah ide revolusioner di CERN (Organisasi Eropa untuk Riset Nuklir) yang nantinya akan mengubah dunia: **World Wide Web**. Pada saat itu, internet sudah ada, tetapi masih sangat

terbatas dalam hal akses dan kegunaannya. Internet hanya digunakan untuk komunikasi antar ilmuwan dan peneliti di seluruh dunia melalui protokol yang disebut **ARPANET**.



*Gambar 1. Tim Barnes Lee Saat di CREN*

Tim Berners-Lee melihat potensi untuk menghubungkan informasi yang tersebar di seluruh dunia dengan cara yang lebih terstruktur dan mudah diakses. Tujuannya adalah untuk **mempermudah akses informasi ilmiah** dengan menggunakan sebuah sistem yang menghubungkan dokumen dokumen melalui **hypertext**, sebuah format yang memungkinkan teks untuk disertai dengan link yang bisa mengarahkan pengguna ke informasi terkait lainnya.

Pada tahun 1990, dia mulai mengembangkan proyek ini dengan tujuan agar **para ilmuwan di seluruh dunia** dapat dengan mudah berbagi informasi secara real-time. Proyek ini menjadi lebih konkret dengan munculnya tiga komponen utama yang dikenal sebagai **World Wide Web**:

- **HTML (HyperText Markup Language)**: Bahasa markup untuk menulis halaman web.
- **URL (Uniform Resource Locator)**: Sistem alamat yang digunakan untuk mengidentifikasi lokasi halaman web.
- **HTTP (HyperText Transfer Protocol)**: Protokol yang memungkinkan transfer data antar server dan klien (browser).

Di tahun 1991, Tim Berners-Lee akhirnya meluncurkan web pertama di dunia, yang berfungsi di server CERN, memungkinkan pengguna untuk mengakses berbagai informasi melalui web. Yang dapat anda akses di <http://info.cern.ch/hypertext/WWW/TheProject.html>

### **Proposal yang Diajukan Tim Berners-Lee**

Tim Berners Lee mengajukan proposal "**Information Management: A Proposal**" pada bulan Maret 1989 dan revisi proposalnya pada Mei 1990. Proposal ini adalah dasar dari seluruh proyek World Wide Web. Di dalam proposal ini, Tim menjelaskan secara rinci bagaimana sistem berbasis hypertext ini dapat digunakan untuk menghubungkan dokumen, informasi, dan sumber daya secara global.

### **Isi Proposal:**

- **Tujuan**: Untuk membuat sistem yang memungkinkan pengguna untuk mengakses informasi secara lebih mudah dan terorganisir, memfasilitasi kolaborasi antar peneliti di berbagai belahan dunia.
- **Desain Sistem**: Berners Lee mengusulkan system berbasis **hyper text** yang akan menghubungkan berbagai dokumen yang disimpan di server berbeda dengan menggunakan **link**.
- **Protokol dan Teknologi yang Digunakan**: Proposal ini mencakup konsep penggunaan HTTP untuk transfer data antar komputer dan HTML untuk membuat halaman web yang dapat diakses oleh pengguna.

Berikut adalah **link ke proposal lengkap** yang diajukan oleh Tim Berners-Lee: [Information Management: A Proposal - Tim Berners-Lee \(1989\)](http://info.cern.ch/hypertext/WWW/TheProject.html)

**Perkembangan HTML dari Versi Pertama hingga HTML5**

## 1. HTML 1.0 (1991)

adalah versi pertama dari HTML yang diciptakan oleh **Tim Berners-Lee** di CERN. HTML 1.0 sangat sederhana dan hanya mencakup elemen-elemen dasar seperti teks, hyperlink, dan gambar. Dampaknya pada dunia web adalah memberikan dasar bagi struktur dasar halaman web, memungkinkan penulis dan ilmuwan untuk saling berbagi informasi secara global melalui **hypertext**. Meskipun fungsinya terbatas, HTML 1.0 membuka pintu bagi komunikasi berbasis web.

### Fitur Utama:

- Hanya mendukung tag-tag dasar seperti `<html>`, `<head>`, `<title>`, dan `<body>`.
- **Penggunaan hyperlink:** Menyediakan elemen `<a>` untuk membuat tautan antar halaman.
- **Penggunaan teks dasar:** HTML 1.0 mendukung elemen untuk teks seperti `<p>` untuk paragraf, `<h1>` hingga `<h6>` untuk heading, dan `<ul>`, `<ol>`, dan `<li>` untuk daftar.

### Keterbatasan:

- HTML 1.0 tidak mendukung banyak elemen untuk mendesain halaman dengan lebih baik, seperti tabel atau gambar secara interaktif.
- Tidak ada dukungan untuk multimedia atau skrip interaktif.

## 2. HTML 2.0 (1995)

merupakan versi pertama yang disetujui sebagai standar. Versi ini memperkenalkan elemen formulir seperti `<form>` dan `<input>`, memungkinkan pengumpulan data dari pengguna secara langsung di halaman web. HTML 2.0 juga menyempurnakan struktur tabel untuk menyusun data dengan lebih baik.

Dampaknya sangat besar karena memungkinkan pengembangan web yang lebih interaktif dan berbasis data, serta menjadi dasar bagi perkembangan web dinamis.

### Fitur Utama:

- Menambahkan elemen untuk tabel dengan tag `<table>`, `<tr>`, `<th>`, dan `<td>`.  
Menambahkan atribut untuk mengontrol atribut elemen seperti **align** dan **bgcolor**.
- Memperkenalkan formulir interaktif dengan tag `<form>`, `<input>`, `<textarea>`, `<select>`, dan `<option>`, yang memungkinkan pengumpulan data dari pengguna.

### Keterbatasan:

- Masih belum ada dukungan untuk grafis atau multimedia, dan desain halaman masih sangat terbatas.

## 3. HTML 3.2 (1997)

melanjutkan evolusi HTML dengan menambahkan dukungan untuk elemen presentasi seperti font dan pengaturan teks yang lebih canggih. HTML 3.2 juga memperkenalkan dukungan untuk JavaScript, yang mengubah web menjadi lebih dinamis dan interaktif. Dampaknya sangat signifikan, karena web menjadi lebih menarik dan aplikasi berbasis web mulai berkembang.

### Fitur Utama:

- Memperkenalkan elemen **presentasi** seperti `<font>` dan `<center>`, meskipun banyak yang tidak direkomendasikan lagi di versi yang lebih baru karena lebih berfokus pada desain CSS.
- Menambahkan dukungan untuk elemen **tabel** lebih lanjut dan fitur seperti tabel yang lebih kompleks

dan pemformatan yang lebih fleksibel.

- Memperkenalkan tag **scripting** seperti `<script>` untuk memungkinkan pengembangan JavaScript.

#### Keterbatasan:

- Tentu saja, masih belum ada penyempurnaan untuk multimedia atau kemampuan interaksi yang lebih canggih.

#### 4. HTML 4.01 (1999)

adalah versi yang sangat penting karena memisahkan struktur konten (HTML) dan presentasi (CSS). HTML 4.01 memperkenalkan elemen semantik seperti `<div>` dan `<span>` untuk membantu struktur halaman dengan lebih baik. Dengan dukungan untuk **CSS**, desain halaman web menjadi lebih fleksibel dan lebih mudah dikelola. HTML 4.01 menetapkan standar yang mendasari banyak situs web hingga saat ini, memfasilitasi perkembangan web yang lebih terstruktur dan lebih mudah dipelihara.

#### Fitur Utama:

- **Penggunaan CSS (Cascading Style Sheets):** HTML 4.01 memberikan lebih banyak dukungan untuk CSS untuk pengaturan layout dan desain halaman.
- **Elemen Semantik:** Menambahkan elemen seperti `<div>` dan `<span>` untuk membantu strukturisasi halaman dengan lebih fleksibel.
- **Dukungan untuk multimedia:** HTML 4.01 mulai memungkinkan penggunaan elemen untuk mengintegrasikan multimedia seperti gambar dan video melalui tag `<img>`, `<audio>`, dan `<video>` (meskipun elemen `<audio>` dan

`<video>` lebih lengkap di HTML5).

- **Formulir yang lebih kuat:** Memperkenalkan atribut baru dalam elemen formulir untuk memberikan kontrol lebih pada pengumpulan data.

#### Keterbatasan:

- HTML 4.01 tidak menyediakan dukungan langsung untuk elemen-elemen dinamis atau multimedia yang lebih canggih seperti animasi atau interaksi lanjutan.
- 

#### 5. XHTML (2000)

adalah versi yang lebih ketat dan berbasis XML dari HTML 4.01. Dengan sintaksis yang lebih terstruktur, XHTML memaksa pengkodean halaman web yang lebih konsisten dan dapat diprediksi. Meskipun tidak begitu populer, XHTML memberikan dasar bagi pengembangan web yang lebih konsisten, terutama dalam hal kompatibilitas antar browser dan perangkat.

#### Fitur Utama:

- Lebih ketat dalam aturan sintaksis (misalnya, tag harus ditutup dengan benar dan atribut harus diberi tanda kutip).
- Diperkenalkan untuk memastikan web lebih konsisten dan kompatibel di seluruh perangkat dan browser.

#### Keterbatasan:

- sifatnya yang lebih ketat membuat beberapa pengembang web merasa terbatas dalam mengembangkan kode yang lebih fleksibel.

#### 6. HTML5

HTML5 adalah evolusi terbesar dalam sejarah HTML. HTML5 memperkenalkan berbagai fitur baru yang mengubah cara kita membuat dan menggunakan aplikasi web. Beberapa inovasi utama termasuk **media embedding** dengan tag `<audio>` dan `<video>`, elemen grafis seperti `<canvas>` untuk menggambar secara dinamis di browser, dan **lokasi geografis** dengan API geolocation. HTML5 juga membawa kemampuan untuk aplikasi web bekerja secara **offline** melalui **localStorage** dan **service workers**, serta memperkenalkan form input yang lebih canggih untuk interaksi yang lebih baik dengan pengguna. Dampaknya sangat besar, karena HTML5 memungkinkan pengembangan aplikasi web yang lebih kaya dan interaktif tanpa memerlukan plugin eksternal seperti Flash, serta memberikan kemudahan bagi pengembang untuk membuat aplikasi web yang dapat berjalan di berbagai perangkat

#### Fitur Utama:

- **Dukungan untuk multimedia:** HTML5 memperkenalkan tag `<audio>` dan `<video>`, yang memungkinkan pemutar audio dan video langsung di dalam halaman web tanpa memerlukan plugin eksternal seperti Flash.
- **Canvas dan SVG:** HTML5 memperkenalkan elemen `<canvas>`, yang memungkinkan pembuatan grafis dinamis menggunakan JavaScript, serta dukungan lebih baik untuk grafik berbasis vektor dengan SVG.
- **Local Storage dan Web Storage:** HTML5 menggantikan cookies untuk penyimpanan data sisi klien dengan menggunakan **localStorage** dan **sessionStorage**, memungkinkan aplikasi web menyimpan data lebih besar secara lokal.
- **Geolocation API:** HTML5 memungkinkan penggunaan **geolocation** untuk melacak posisi geografis

pengguna dan memberikan informasi berbasis lokasi.

- **Offline Web Apps:** HTML5 memungkinkan aplikasi web berfungsi offline menggunakan **Application Cache** dan **Service Workers**.
- **Formulir Lebih Canggih:** HTML5 memperkenalkan elemen formulir baru seperti `<input type="email">`, `<input type="date">`, `<input type="range">`, dan lainnya untuk membantu pembuatan formulir yang lebih interaktif dan canggih.
- **Dukungan untuk API Web:** HTML5 memungkinkan penggunaan berbagai API seperti **Web Workers**, **WebSockets**, dan **Drag and Drop**.

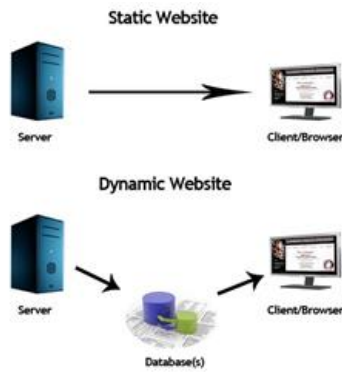
#### Keterbatasan:

- Meskipun HTML5 sangat kuat, beberapa browser lama mungkin tidak mendukung semua fitur HTML5, meskipun sekarang banyak browser modern yang sudah sepenuhnya kompatibel.

#### Munculnya Web Dinamis dan REST API

Setelah era web statis yang hanya menampilkan konten tetap menggunakan HTML, dunia web mulai berkembang ke arah web dinamis. Web dinamis adalah jenis website yang kontennya bisa berubah-ubah secara otomatis berdasarkan input pengguna atau data dari database.

Berbeda dengan web statis yang harus diubah secara manual jika ingin mengganti isi, web dinamis bisa menampilkan konten yang berbeda untuk setiap pengunjung. Misalnya E-commerce yang menampilkan daftar produk sesuai kategori, situs login yang menampilkan nama pengguna setelah berhasil masuk. Untuk contoh web dinamis sendiri ada Facebook, Blog WordPress, Amazon dll.

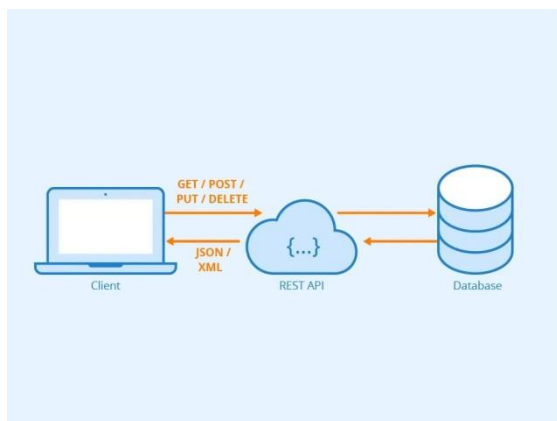


**Gambar 2** gambar perbedaan web dinamis dan static

Untuk membangun web dinamis, digunakan bahasa pemrograman server-side seperti PHP, Python (Django/Flask), Ruby on Rails, atau Node.js. Semua ini terhubung ke database, lalu menghasilkan halaman HTML secara otomatis.

### REST API: Jembatan Antara Client dan Server

Seiring berkembangnya kebutuhan aplikasi web yang lebih kompleks dan responsif, muncul konsep REST API (Representational State Transfer Application Programming Interface). REST API adalah metode yang digunakan untuk menghubungkan antara tampilan depan (front-end) dan logika di belakang (back-end/server) agar data bisa ditukar dengan lebih efisien dan terstruktur.



**Gambar 3** Rest Api Model

REST pertama kali diperkenalkan oleh Roy Fielding pada tahun 2000 dalam disertasinya di University of California,

Irvine. Ia merupakan salah satu arsitek utama dalam pengembangan protocol HTTP, sehingga REST dirancang agar dapat bekerja dengan sempurna di atas HTTP.

REST API bekerja menggunakan HTTP methods (metode permintaan) seperti:

- **GET** Digunakan untuk mengambil data dari server. Contoh: menampilkan daftar produk.
- **POST** Digunakan untuk mengirim data baru ke server. Contoh: mengirim data form pendaftaran pengguna.
- **PUT** Digunakan untuk memperbarui data yang sudah ada. Contoh: memperbarui data profil pengguna.
- **DELETE** Digunakan untuk menghapus data dari server. Contoh: Menghapus Komentar di sebuah postingan

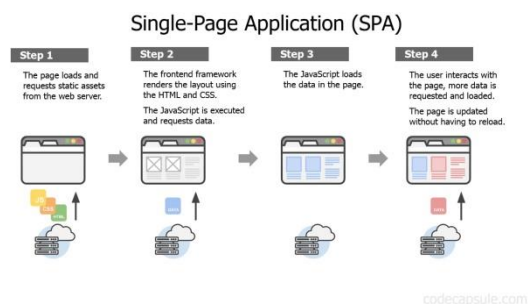
REST API umumnya mengirim dan menerima data dalam format JSON (JavaScript Object Notation), karena ringan dan mudah dibaca baik oleh manusia maupun mesin. Selain JSON, REST juga bisa menggunakan format lain seperti XML, namun JSON menjadi standard paling populer saat ini.

### Single Page Application (SPA)

Setelah munculnya REST API yang memungkinkan pertukaran data secara efisien antara client dan server, lahirlah pendekatan baru dalam pengembangan web modern yang disebut dengan Single Page Application (SPA). SPA adalah jenis aplikasi web yang hanya memuat satu halaman HTML utama, lalu mengubah kontennya secara dinamis menggunakan JavaScript tanpa harus memuat ulang halaman setiap kali pengguna berpindah menu atau fitur. Pendekatan ini sangat berbeda dengan web tradisional yang harus me-refresh seluruh halaman setiap kali terjadi perubahan tampilan.

Dengan SPA, saat pertama kali pengguna mengakses website, browser akan memuat semua komponen utama seperti HTML, CSS, dan JavaScript. Setelah itu, ketika pengguna melakukan interaksi seperti membuka halaman baru atau melihat detail data, aplikasi akan menggunakan REST API untuk mengambil data dari server, lalu menampilkan nya langsung di dalam halaman yang sama. Hal ini membuat navigasi menjadi sangat cepat dan pengalaman pengguna lebih mulus, mirip seperti menggunakan aplikasi mobile atau desktop.

Beberapa kelebihan SPA antara lain adalah kecepatan navigasi yang tinggi, beban server yang lebih ringan setelah pemuatan awal, serta pengalaman pengguna yang lebih modern. Namun, SPA juga memiliki kekurangan seperti waktu loading awal yang cenderung lebih lama dan tantangan dalam hal SEO, karena konten tidak langsung dimuat di halaman awal. SPA juga sangat bergantung pada JavaScript; jika JavaScript dinonaktifkan, aplikasi bisa tidak berfungsi sebagaimana mestinya.



**Gambar 4** Single Page Application (SPA) cara kerja

Contoh populer dari aplikasi SPA adalah Gmail, di mana saat membuka email atau berpindah antar folder tidak terjadi reload halaman. Google Drive, Facebook, dan Twitter versi modern juga menggunakan pendekatan SPA untuk memberikan interaksi yang cepat dan responsif. Dalam pengembangannya, SPA banyak dibangun menggunakan framework JavaScript seperti React.js yang dikembangkan oleh Facebook, Angular

dari Google, dan Vue.js yang populer di komunitas open source karena kesederhanaan dan fleksibilitasnya.

### Progressive Web App (PWA).

Setelah hadirnya konsep Single Page Application (SPA), dunia pengembangan web terus berkembang dengan munculnya teknologi baru bernama Progressive Web App (PWA). PWA adalah jenis aplikasi web yang dirancang agar memiliki tampilan dan pengalaman layaknya aplikasi mobile, tetapi tetap dijalankan melalui browser tanpa perlu diunduh dari toko aplikasi seperti Play Store atau App Store. Teknologi ini diperkenalkan oleh Google sekitar tahun 2015 sebagai solusi untuk menggabungkan keunggulan website dan aplikasi mobile dalam satu platform yang ringan, cepat, dan bias diakses kapan saja.



**Gambar 5** Progressive Web App (PWA)

Salah satu fitur utama dari PWA adalah kemampuannya untuk diakses secara offline menggunakan teknologi yang disebut Service Worker. Service Worker memungkinkan browser menyimpan sebagian data dan halaman website ke dalam cache, sehingga pengguna tetap bisa membuka website meskipun koneksi internet terputus. Selain itu, PWA mendukung push notification, instalasi langsung ke layar utama, dan desain yang responsif, sehingga tampil optimal di berbagai ukuran layar — mulai dari desktop hingga smartphone.

Dengan PWA, pengembang bisa membangun satu website yang memiliki



performa tinggi seperti aplikasi native tanpa perlu membuat dua versi terpisah untuk web dan mobile. Contoh penerapan PWA yang sukses adalah aplikasi Twitter Lite, Forbes, dan Pinterest, yang semuanya berhasil meningkatkan kecepatan akses, keterlibatan pengguna, hingga penghematan data. Teknologi ini sangat cocok digunakan di wilayah dengan koneksi internet lambat atau terbatas, karena lebih hemat data dan ringan diakses.

Secara umum, PWA mengandalkan kombinasi dari beberapa teknologi web modern seperti HTML5, CSS3, JavaScript, serta pemanfaatan Web App Manifest dan Service Worker untuk membuat pengalaman web lebih interaktif, cepat, dan andal. Dengan kata lain, PWA adalah salah satu tonggak besar yang membawa website ke level selanjutnya: cepat seperti aplikasi, ringan seperti website.

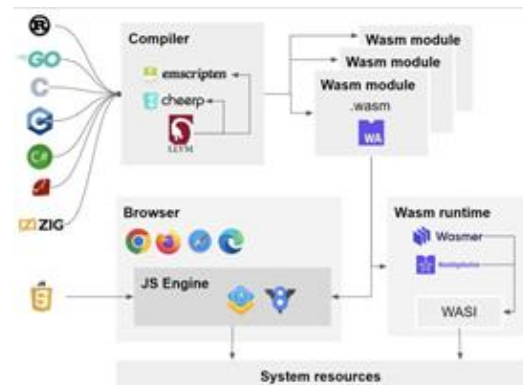
### WebAssembly (WASM)

WebAssembly, atau disingkat Wasm, adalah teknologi terbaru dalam pengembangan web yang dirancang untuk menghadirkan kinerja tinggi di dalam browser, mendekati performa aplikasi native seperti yang biasa ditemukan di desktop. Web Assembly pertama kali diumumkan pada tahun 2015 sebagai hasil kolaborasi beberapa raksasa teknologi seperti Mozilla, Google, Microsoft, dan Apple, dan kini telah menjadi standar resmi yang didukung oleh W3C (World Wide Web Consortium).

WebAssembly memungkinkan pengembangan menulis kode dalam berbagai bahasa pemrograman seperti C, C++, Rust, dan kemudian mengompilasinya ke dalam format biner (.wasm) yang bisa dijalankan di browser. Format biner ini sangat ringan dan efisien, sehingga proses eksekusinya jauh lebih cepat dibandingkan JavaScript, terutama untuk aplikasi yang memerlukan komputasi berat seperti game 3D, editor grafis, simulasi, atau bahkan software CAD langsung di browser.

Keunggulan utama WebAssembly terletak pada kecepatannya, portabilitas, serta keamanannya. Wasm berjalan di

dalam sandbox browser (lingkungan yang aman), sehingga tetap menjaga privasi dan keamanan pengguna. Selain itu, WebAssembly tidak menggantikan JavaScript, tetapi justru bisa bekerja berdampingan. Artinya, kita bisa menggunakan JavaScript untuk menangani logika UI dan interaksi pengguna, sementara bagian komputasi berat ditangani oleh Wasm.



**Gambar 6** Cara Kerja WebAssembly

Contoh nyata penggunaan WebAssembly bisa ditemukan di AutoCAD Web App, Figma, dan berbagai game berbasis web seperti yang dikembangkan dengan Unity atau Unreal Engine. Teknologi ini telah membuka pintu bagi banyak aplikasi desktop untuk berpindah ke web tanpa harus mengorbankan performa.

### PENUTUP

Perjalanan perkembangan teknologi web menunjukkan transformasi luar biasa, di mulai dari halaman statis berbasis HTML hingga munculnya teknologi modern seperti REST API, SPA, PWA, dan WebAssembly. REST API memungkinkan komunikasi efisien antara client dan server, sementara SPA menyederhanakan pengalaman pengguna dengan tampilan yang lebih cepat dan dinamis. PWA membawa pengalaman aplikasi mobile ke web dengan fitur offline dan notifikasi, sedangkan WebAssembly memungkinkan aplikasi kompleks berjalan cepat langsung di browser. Semua inovasi ini menunjukkan bahwa web bukan hanya alat untuk menyajikan informasi, tetapi telah berkembang menjadi platform aplikasi canggih



yang terus beradaptasi dengan kebutuhan pengguna masa kini.

## UCAPAN TERIMA KASIH

Bagian ini berisi ucapan terima kasih kepada sponsor atau pendonor dana, atau kepada pihak-pihak yang secara penting berperan dalam pelaksanaan penelitian.

## DAFTAR PUSTAKA

- Berners-Lee T. The World Wide Web project [Internet]. 1991 [cited 2025 Mar 27]. Available from: <https://info.cern.ch/hypertext/WWW/TheProject.html>
- Petani Kode. Sejarah HTML: Dari HTML 1.0 hingga HTML5 [Internet]. 2020 [cited 2025 Mar 27]. Available from: <https://www.petanikode.com/sejarah-h-html/>
- Raggett D, Le Hors A, Jacobs I. HTML 4.01 Specification [Internet]. 1999 [cited 2025 Mar 27]. Available from: <https://www.w3.org/TR/html4/>
- W3C. Extensible Hypertext Markup Language (XHTML) 1.0: The Extensible Markup Language (XML) [Internet]. 2000 [cited 2025 Mar 27]. Available from: <https://www.w3.org/TR/xhtml1/>
- W3C. HTML5: A vocabulary and associated APIs for HTML and XHTML [Internet]. 2014 [cited 2025 Mar 27]. Available from: <https://www.w3.org/TR/html5/>
- Alam-Naylor, S. (2022). SPAs, MPAs, the history of Twitter tech, & a failed project. Retrieved from <https://whitep4nth3r.com/blog/twitter-tech-history-spa/>
- Begemann, O. (2018). Roy Fielding's REST dissertation. Retrieved from <https://oleb.net/2018/rest/>
- Facebook. (n.d.). React – A JavaScript library for building user interfaces. Retrieved April 8, 2025, from <https://reactjs.org/>
- Fielding, R. T. (2000). Architectural styles and the design of network-based software architectures (Doctoral dissertation, University of California, Irvine). Retrieved from [https://www.ics.uci.edu/~fielding/pubs/dissertation/fielding\\_dissertation.pdf](https://www.ics.uci.edu/~fielding/pubs/dissertation/fielding_dissertation.pdf)
- Google. (n.d.). Angular. Retrieved April 8, 2025, from <https://angular.io/>
- Google Developers. (n.d.). Progressive Web Apps. Retrieved April 8, 2025, from <https://developers.google.com/web/progressive-web-apps>
- Haas, A., Rossberg, A., Schuff, D., Titzer, B. L., Gohman, D., Wagner, L., ... & Holman, M. (2017). Bringing the web up to speed with WebAssembly. *Proceedings of the 38th ACM SIGPLAN Conference on Programming Language Design and Implementation*, 185–200.
- IBM. (n.d.). What is a REST API?. Retrieved April 9, 2025, from <https://www.ibm.com/think/topics/rest-apis>
- Khandelwal, S. (2015). WebAssembly — New Standard for Powerful and Faster Web Apps. *The Hacker News*. Retrieved from <https://thehackernews.com/2015/06/webassembly-fastest-browser.html>
- Mozilla Developer Network. (n.d.). SPA (Single-page application). Retrieved April 9, 2025, from <https://developer.mozilla.org/en-US/docs/Glossary/SPA>

*Mozilla. (n.d.). MDN Web Docs. Retrieved April 8, 2025, from <https://developer.mozilla.org/>*

*ResearchGate. (2019). Impact of Progressive Web Apps on Web App Development. Retrieved from [https://www.researchgate.net/publication/330834334\\_Impact\\_of\\_Progressive\\_Web\\_Apps\\_on\\_Web\\_App\\_Development](https://www.researchgate.net/publication/330834334_Impact_of_Progressive_Web_Apps_on_Web_App_Development)*

*Richardson, L., & Amundsen, M. (2013). RESTful Web APIs. O'Reilly Media.*

*Robbins, J. (2018). Learning web design: A beginner's guide to HTML, CSS, JavaScript, and web graphics (5th ed.). O'Reilly Media.*

*Spiceworks. (2023). What Is WebAssembly (Wasm)? Working, Uses, and Importance. Retrieved from <https://www.spiceworks.com/tech/devops/articles/what-is-webassembly/>*

*Vue.js. (n.d.). The Progressive JavaScript Framework. Retrieved April 8, 2025, from <https://vuejs.org/>*