

PENGEMBANGAN APLIKASI WEB BERBASIS JAMStack MENGUNAKAN FRAMEWORK FRONT-END REACT DAN LAYANAN CLOUD

Bayhaqi Subuhi¹, Afif Badawi², Rio Septian Hardinata³

¹Sistem Komputer, Universitas Pembangunan Panca Budi

²Sistem Komputer, Universitas Pembangunan Panca Budi

¹subuhibayhaqi375@gmail.com, ²afifbadawi@dosen.pancabudi.ac.id

Corresponding Author: Bayhaqi Subuhi

ABSTRAK

Perkembangan teknologi web menurut aplikasi yang cepat, aman, dan mudah diskalakan. Arsitektur web tradisional berbasis server-side rendering memiliki keterbatasan dalam hal performa dan skalabilitas. Salah satu pendekatan modern yang berkembang adalah JAMStack (JavaScript, API, dan Markup) yang mengandalkan pemisahan antara front-end dan back-end. Penelitian ini bertujuan untuk merancang dan membangun aplikasi web berbasis JAMStack menggunakan framework front-end React serta memanfaatkan layanan cloud sebagai penyedia API dan hosting. Metode penelitian yang digunakan adalah metode pengembangan sistem berbasis System Development Life Cycle (SDLC) mode waterfall. Hasil penelitian menunjukkan bahwa penerapan JAMStack mampu meningkatkan kecepatan akses, keamanan aplikasi, serta efisiensi pengolahan sistem dibandingkan arsitektur web konvensional. Aplikasi yang dikembangkan bersifat responsif, modular, dan mudah dikembangkan di masa depan.

Kata Kunci: JAMStack, React, Aplikasi web, Cloud Computing (atau Layanan Cloud, Front-end Framework

1. Pendahuluan

Perkembangan internet dan teknologi informasi mendorong kebutuhan akan aplikasi web yang memiliki performa tinggi, aman, serta mampu menangani peningkatan jumlah pengguna. Arsitektur web konvensional yang bergantung pada server-side processing sering mengalami kendala seperti beban server tinggi, waktu respon lambat, serta resiko keamanan.

2. Tinjauan Pustaka

Aplikasi web adalah perangkat lunak yang diakses melalui jaringan internet menggunakan browser web. Evolusi aplikasi web telah berlangsung cepat, dimulai dari situs web statis (Web 1.0). Kemudian berkembang ke era dinamis dengan bahasa server side seperti PHP, hingga arsitektur single-page application (SPA) modern. Perkembangan ini didorong oleh kebutuhan akan interaktivitas tinggi, pengalaman pengguna yang mulus, serta efisiensi dalam pengembangan dan pemeliharaan (Nursaid & Putra, 2021).

3. Bahan & Metode

1. Sistem Operasi : Windows 11 / macOS / Linux (Ubuntu 20.04)
2. Node.js versi 18.x LTS (Runtime JavaScript)
3. Package Manager: npm (Node Package Manager) versi 9.x atau yarn versi 1.22
4. Framework Front-end: React.js versi 18.20
5. Code Editor: Visual Studio Code Versi 1.80



6. Version Control: Git versi 2.40
7. Layanan Cloud:
 - Netlify/Vercel untuk hosting dan fungsi serverless
 - Supabase/Firebase untuk database dan autentikasi
8. Web Browser: Google Chrome versi 114 (untuk pengujian)

Penelitian ini menggunakan metode Research and Development (R&D) dan model Waterfall yang di modifikasi untuk pengembangan perangkat lunak: metode ini dipilih karena tahapannya yang sistematis dan sesuai untuk pengembangan aplikasi berbasis web [4].

2.1. Rumus dan Program Jurnal

Rumus ditulis secara jelas menggunakan *equation* dengan diberi nomor indeks seperti rumus 1.

$$\Delta F = -2,3 \times 10^6 \times F^2 \frac{\Delta M}{A}$$

dengan F adalah frekuensi dasar resonansi (MHz), ΔM adalah is total massa molekul gas yang diserap [1]. dan A adalah area elektroda (cm^2) [7].

Listing program dan disain algoritma dituliskan seperti berikut:

Program Jurnal

```

Input: async, severless
Output: JSON.stringify
Initialization i, j
Get line, column, max
//netify/functions/getData.js
exports.handler = async (event, context) =>{
  try {
    // Proses data di sini
    const data = {
      message: "Data dari severless function".
      Timestamp: new Date().toISOString()
    }
    end for
  } end for
}

return {
  statusCode: 200,
  body: JSON.stringify(data)
}
} catch (error) {
  return {
    statusCode: 500,
    body: JSON.sringify({ error: 'Terjadi kesalahan' })
  }
}
}

```

2.2. Aplikasi Web

Aplikasi web adalah perangkat lunak yang dijalankan melalui browser dan diakses melalui jaringan internet. Aplikasi web bersifat platform dan tidak memerlukan instalasi khusus pada perangkat pengguna.

2.3. Tabel

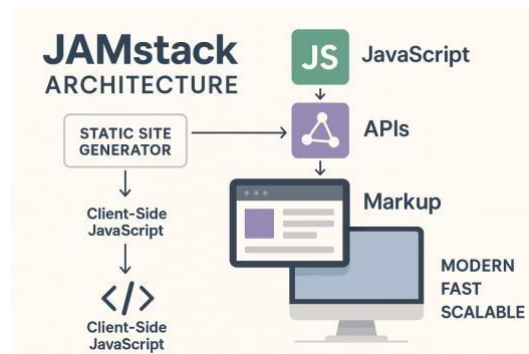
Berdasarkan aplikasi berbasis JAMStack menunjukkan performa yang jauh lebih baik dibandingkan dengan arsitektur tradisional. Peningkatan terbesar terjadi pada waktu deployment (87% lebih cepat) dan waktu loading halaman (65% lebih cepat), yang membuktikan efektivitas arsitektur JAMStack dalam pengembangan aplikasi web modern.

Tabel 1. Tabel Software dan Hardware Pendukung

Product	Server	Client
Komponen	Spesifikasi Minimum	Rekomendasi
Browser	Chrome 90+, Firefox 90+, Safari 15+, Edge 90+	Versi terbaru
Koneksi Internet	1 Mbps	5 Mbps

2.4. Gambar

Dalam penulisan jurnal ilmiah, setiap gambar harus disertai penjelasan yang jelas dan ditempatkan sesuai dengan format yang telah ditentukan. Berikut adalah contoh gambar yang relevan dengan penelitiannya.



Gambar 1. JAMStack Architecture

Statistical Analysis

Meskipun JAMStack bukanlah solusi ajaib untuk setiap kasus penggunaan fullstack, ia mewakili pergeseran program dalam cara aplikasi web modern dirancang. Ia meninggalkan arsitektur monolitik dan terkopel erat di masa lalu dan merangkul model yang dapat disusun dan terkopel yang lebih sesuai dengan harapan kinerja, standar keamanan, dan persyaratan skalabilitas pengguna dan pengembangan web saat ini.

4. Hasil dan Pembahasan

Aplikasi web berhasil dikembangkan arsitektur JAMStack dengan React sebagai front-end dan layanan cloud sebagai back-end. Aplikasi dapat diakses dengan cepat dan stabil.

5. Pembahasan



Berdasarkan hasil pengembangan dan pengujian yang telah dilakukan, arsitektur JAMStack terbukti efektif dalam membangun aplikasi modern. Penerapan prinsip *pre-rendering* dan pemisahan *front-end* dari *back-end* menghasilkan beberapa keuntungan yang sangat signifikan.

6. Kesimpulan

Dan dari berdasarkan beberapa hasil penelitian yang saya dapat simpulkan bahwa dapat penerapan *JAMStack* menggunakan *framework React* dan *layanan cloud* mampu menghasilkan aplikasi web yang cepat, aman, dan efisien. Arsitektur ini sangat sesuai untuk pengembangan aplikasi web yang sangat amat modern.

REFERENSI

- [1] Mathias, B., & Biilman, C. (2020). *JAMStack: The Modern Web Architecture*. O'ReillyMedia.
- [2] Banks, A., & Porcello, E. (2020). *Learning React* O'Reilly Media.
- [3] Facebook Developers. (2023). *React Documentation*.
- [4] Mell, P., & Grance, T. (2011). *The NIST Definition of Cloud Computing*. NIST.
- [5] Pressman, R. S. (2015). *Software Engineering: A Practitioner's Approach*. McGraw-Hill.
- [6] Fielding, R. (2000). *Architecture Styles and the Design of Network-based Software Architecture*. University of California.
- [7] W3C. (2022). *Web Application Architecture*.
- [8] Amazon Web Services. (2023). *Cloud Computing Overview*.
- [9] Google Developers. (2023). *Web Performance Optimization*.
- [10] Netlify. (2022). *JAMStack Best Practices*.
- [11] Mozilla Developer Network. (2023). *JavaScript and Web APIs*.

