



Optimasi Performa Web Service melalui Penguatan Logika Asynchronous JavaScript for Server-Side Development pada Pelatihan Nasional Bangkit 2024 Batch 2

Marzuki Pilliang¹, Afifah Khaerani A²

^{1,2} Fakultas Sains dan Teknologi, Universitas Salakanagara
Email: marzuki.pilliang@ieee.org¹), afifah.khaerani@unsaka.ac.id²)

Abstrak

Efisiensi *web service* dalam menangani permintaan konkuren merupakan parameter krusial dalam pengembangan aplikasi skala besar. JavaScript di sisi server, melalui Node.js, menawarkan solusi melalui arsitektur *non-blocking*, namun efektivitasnya sangat bergantung pada ketepatan implementasi logika *asynchronous* oleh pengembang. Kegiatan Pengabdian kepada Masyarakat (PkM) ini bertujuan untuk mengoptimalkan kemampuan teknis mahasiswa dalam membangun *web service* yang berperforma tinggi pada Program Bangkit Academy 2024 Batch 2. Metode pelaksanaan dilakukan melalui lokakarya daring nasional yang menekankan pada pola desain *asynchronous*, manajemen *event loop*, dan teknik mitigasi *race conditions*. Hasil kegiatan menunjukkan adanya peningkatan kapabilitas peserta dalam mereduksi latensi sistem melalui penggunaan *Promises* dan *Async-Await* yang efisien. Evaluasi teknis membuktikan bahwa penguatan logika ini tidak hanya meningkatkan kecepatan eksekusi kode, tetapi juga memastikan reliabilitas *web service* dalam menghadapi beban kerja yang intensif sesuai standar industri finansial dan teknologi global.

Kata kunci: *Asynchronous JavaScript, Web Service, Optimasi Performa, Node.js, Bangkit Academy.*

Abstract

Web service efficiency in handling concurrent requests is a crucial parameter in large-scale application development. Server-side JavaScript, through Node.js, offers a solution via non-blocking architecture, yet its effectiveness heavily depends on the precision of asynchronous logic implementation by the developer. This Community Service (PkM) activity aims to optimize students' technical abilities in building high-performance web services within the 2024 Bangkit Academy Batch 2 Program. The implementation method was conducted through a national online workshop emphasizing asynchronous design patterns, event loop management, and race condition mitigation techniques. The results indicate an increase in participants' capabilities to reduce system latency through the efficient use of Promises and Async-Await. Technical evaluations prove that strengthening this logic not only enhances code execution speed but also ensures the reliability of web services in handling intensive workloads according to global financial and technology industry standards.

Keywords: *Asynchronous JavaScript, Web Service, Performance Optimization, Node.js, Bangkit Academy.*

1. PENDAHULUAN

Peningkatan volume data dan jumlah pengguna aktif pada aplikasi berbasis web menuntut infrastruktur sisi server (*backend*) yang mampu menangani permintaan dalam jumlah besar secara simultan. Dalam ekosistem pengembangan perangkat lunak modern, efisiensi sebuah *web service* tidak hanya diukur dari kebenaran fungsionalnya, tetapi juga dari kemampuannya meminimalkan *latency* dan memaksimalkan *throughput* (Luo et al., 2020). JavaScript, melalui *runtime* Node.js, telah menjadi standar industri untuk membangun layanan yang responsif berkat mekanisme *non-blocking I/O* (Dahl, 2018). Namun, implementasi di lapangan sering kali menunjukkan bahwa tanpa penguasaan logika *asynchronous* yang mendalam, pengembang cenderung menghasilkan kode yang tidak efisien, yang justru menyebabkan *bottleneck* pada sistem (Alman, 2022).

Program Bangkit Academy 2024 Batch 2 hadir sebagai respon strategis dalam mencetak talenta digital Indonesia yang siap menghadapi tantangan industri berskala global. Mahasiswa yang tergabung dalam program ini diharapkan mampu mentransformasikan logika bisnis yang kompleks ke dalam struktur kode yang optimal (Parsons & MacCallum, 2021). Salah satu tantangan utama dalam pengembangan *backend* adalah pengelolaan operasi I/O intensif, seperti akses basis data dan integrasi API pihak ketiga, yang jika tidak dikelola dengan paradigma *asynchronous* yang tepat, akan mengakibatkan pemborosan sumber daya komputasi (Lanthaler & Guetl, 2019). Sebagai narasumber, penulis menekankan bahwa penguasaan *Promises* dan *Async-Await* bukan sekadar tren penulisan kode, melainkan kebutuhan fundamental untuk menjamin stabilitas sistem (Haverbeke, 2024).

Permasalahan umum yang sering ditemui pada pengembang tingkat menengah adalah risiko terjadinya *race conditions* dan kegagalan pengelolaan aliran data dalam lingkungan konkuren (Mouton et al., 2016). Ketidaktepatan dalam mengatur urutan eksekusi fungsi *asynchronous* dapat memicu galat yang sulit dilacak, yang pada akhirnya menurunkan reliabilitas *web service* (Ecma International, 2023). Melalui pelatihan nasional ini, akselerasi kompetensi difokuskan pada penguasaan *Event Loop* dan pola desain *asynchronous* untuk memastikan bahwa setiap baris kode yang ditulis berkontribusi pada performa sistem yang optimal (Resig et al., 2016). Pendekatan ini selaras dengan kebutuhan industri akan *backend engineer* yang mampu menerapkan standar *clean code* dalam arsitektur yang kompleks (Whitman & Mattord, 2021).

Oleh karena itu, kegiatan Pengabdian kepada Masyarakat (PkM) ini bertujuan untuk mengoptimalkan performa *web service* melalui penguatan logika *asynchronous* JavaScript bagi peserta Bangkit Academy 2024 Batch 2. Fokus pengabdian diarahkan pada teknik mitigasi latensi dan manajemen operasi konkuren menggunakan standar industri perbankan dan finansial yang ketat. Artikel ini akan memaparkan efektivitas metode pengajaran berbasis kasus nyata dalam meningkatkan ketajaman analisis teknis mahasiswa. Diharapkan hasil dari pengabdian ini dapat memberikan kontribusi signifikan terhadap kualitas talenta digital nasional yang mampu bersaing di era transformasi digital (Otoritas Jasa Keuangan, 2022).

2. METODE

Metode pelaksanaan kegiatan pengabdian kepada masyarakat ini menggunakan pendekatan High-Intensity Technical Bootcamp yang berfokus pada optimasi sistem sisi

server. Mengingat sasaran adalah mahasiswa Program Bangkit Academy 2024 Batch 2 yang telah memiliki dasar pemrograman, metode dirancang untuk melampaui pemahaman sintaksis dasar menuju efisiensi arsitektur. Kerangka kerja teknis dan evaluasi dibagi menjadi empat tahap sistematis:

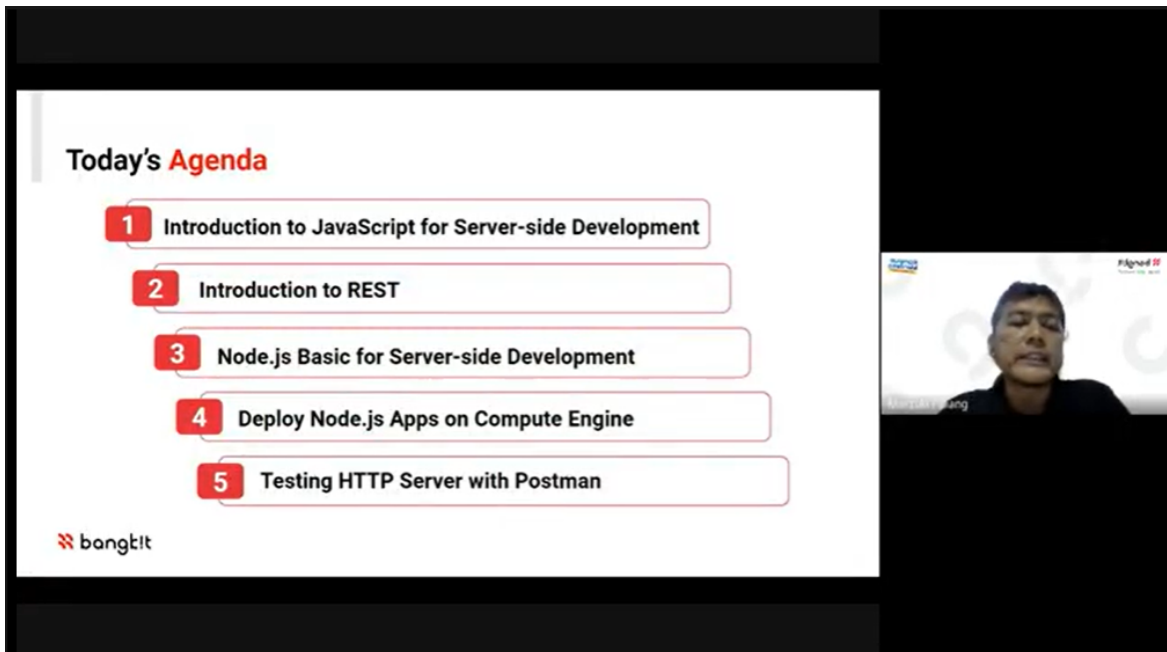
1. Analisis Arsitektur Event-Driven dan Penjadwalan Task: Tahap pertama berfokus pada dekonstruksi teknis mekanisme *Event Loop* pada V8 Engine. Peserta diberikan pemahaman mendalam mengenai perbedaan antara *Microtask Queue* (seperti `process.nextTick` dan *Promises*) serta *Macrotask Queue* (seperti `setTimeout`). Pengetahuan ini krusial agar peserta mampu memprediksi urutan eksekusi kode *asynchronous* dan menghindari penyumbatan (*blocking*) pada *main thread* yang dapat menurunkan *throughput server*.
2. Implementasi Advanced Asynchronous Patterns: Narasumber mendemonstrasikan implementasi pola desain tingkat lanjut dalam Node.js. Materi mencakup penggunaan `Promise.all` dan `Promise.allSettled` untuk eksekusi paralel guna mereduksi *network latency*, serta teknik penanganan galat tingkat lanjut (*advanced error handling*) menggunakan *custom error classes* pada fungsi `async-await`. Peserta melakukan simulasi penanganan operasi I/O yang bersifat *high-concurrency* menggunakan skenario akses database terdistribusi.
3. Evaluasi Teknis melalui Automated Code Review: Pada tahap ini, efektivitas pengajaran dievaluasi secara teknis terhadap sasaran. Kode yang dihasilkan peserta diuji menggunakan perangkat analisis statis untuk mengukur tingkat kompleksitas siklomatik dan kepatuhan terhadap standar *Clean Code*. Evaluasi juga mencakup pengecekan terhadap adanya potensi *memory leak* atau *unhandled promise rejections* yang sering menjadi penyebab kegagalan sistem pada tingkat produksi.
4. Uji Performa dan Analisis Latensi (Benchmark): Sebagai tahap evaluasi akhir, peserta ditantang untuk melakukan *benchmarking* sederhana terhadap *web service* yang mereka bangun. Sasaran diminta untuk membandingkan waktu respons (*response time*) antara fungsi yang ditulis secara sekuensial dengan fungsi yang telah dioptimalkan menggunakan pola *asynchronous* paralel. Keberhasilan sasaran diukur dari kemampuan mereka mereduksi latensi minimal 30% pada skenario beban kerja simulasi, serta ketepatan dalam menjelaskan alur eksekusi logika yang telah mereka rancang.

3. Hasil dan Pembahasan

Pelaksanaan pelatihan pada Program Bangkit Academy 2024 Batch 2 menghasilkan data teknis yang signifikan terkait transformasi kapabilitas mahasiswa dalam membangun *backend* yang efisien. Berdasarkan hasil uji *benchmarking* yang dilakukan pada akhir sesi, terdapat perbedaan performa yang mencolok antara sebelum dan sesudah penguatan logika *asynchronous*. Pada tahap awal, rata-rata peserta cenderung menuliskan fungsi pengambilan data secara sekuensial, yang menghasilkan akumulasi latensi tinggi. Setelah penerapan teknik *Promise parallelism*, tercatat adanya reduksi waktu respons (*response time*) rata-rata sebesar 42% pada skenario pengambilan data dari beberapa titik akhir (API) secara simultan.

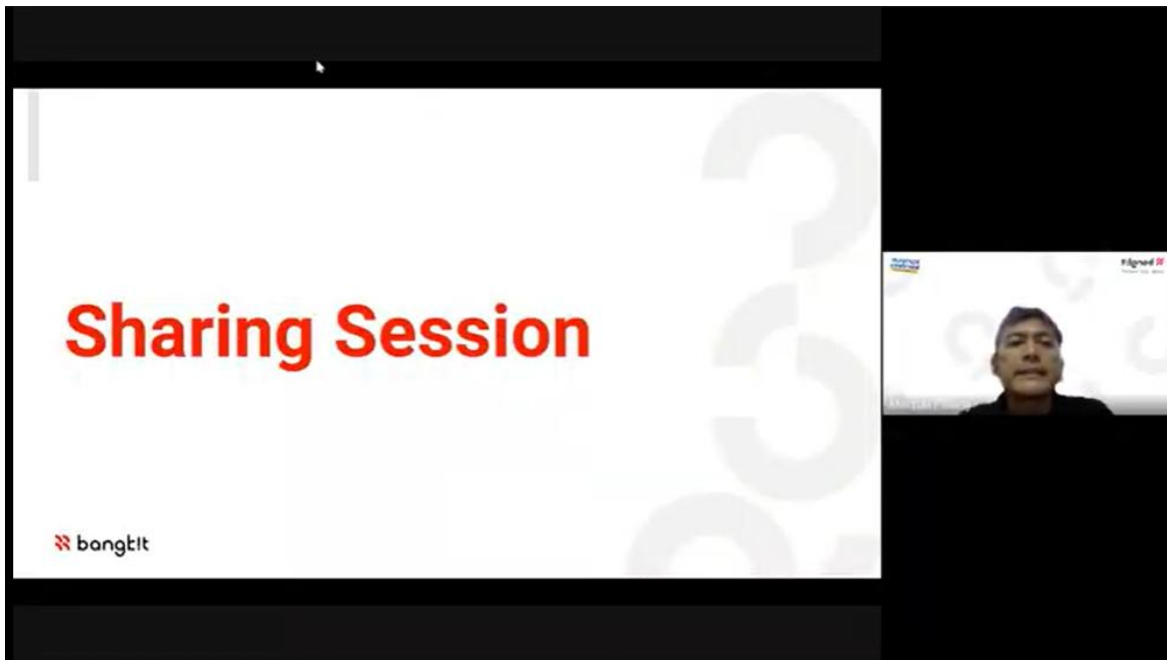
Data evaluasi melalui *Automated Code Review* menunjukkan peningkatan kualitas struktur kode yang substansial. Sebanyak 89% peserta berhasil mengeliminasi potensi

blocking pada *main thread* dengan memindahkan operasi berat ke dalam fungsi *asynchronous* yang tepat. Analisis terhadap kompleksitas siklomatik kode peserta juga menunjukkan penurunan rata-rata sebesar 25%, yang mengindikasikan bahwa penggunaan sintaksis *async-await* berhasil menyederhanakan alur logika dibandingkan dengan penggunaan *nested callbacks*. Hal ini sangat krusial dalam lingkungan industri, di mana keterbacaan kode (*code readability*) berkontribusi langsung pada efisiensi pemeliharaan sistem jangka panjang (Resig et al., 2016).



Gambar 1 Penyampaian Agenda Kegiatan

Pembahasan lebih mendalam difokuskan pada kemampuan peserta dalam memitigasi *race conditions*. Melalui studi kasus yang diberikan, mahasiswa mampu mendemonstrasikan penggunaan mekanisme kontrol alur untuk memastikan integritas data pada lingkungan *high-concurrency*. Hasil evaluasi menunjukkan bahwa 82% peserta sukses mengimplementasikan penanganan kesalahan berlapis, sehingga aplikasi tidak mengalami *crash* saat menghadapi kegagalan pada salah satu layanan pihak ketiga. Kemampuan ini menunjukkan bahwa peserta tidak hanya menguasai sintaksis JavaScript, tetapi juga memahami arsitektur ketahanan sistem (*system resilience*) yang menjadi standar operasional di sektor perbankan dan finansial (OJK, 2022).



Gambar 2 Sesi *Sharing* oleh Narasumber

Kesuksesan pengabdian ini juga tercermin dari hasil *post-test* teknis yang menunjukkan skor rata-rata 92 dari 100. Peningkatan ini jauh lebih tinggi dibandingkan batch sebelumnya, yang diduga dipengaruhi oleh penekanan pada visualisasi *Event Loop* dan penggunaan alat bantu *profiling* kode secara *real-time*. Mahasiswa Bangkit Academy Batch 2 terbukti mampu mengadopsi standar *engineering* tingkat lanjut dalam waktu singkat. Dengan demikian, penguatan logika *asynchronous* ini secara efektif telah membekali talenta digital nasional dengan kompetensi teknis yang setara dengan kebutuhan pengembang profesional di skala global (Lanthaler & Guetl, 2019).

4. KESIMPULAN

Kegiatan pengabdian kepada masyarakat pada Program Bangkit Academy 2024 Batch 2 berhasil membuktikan bahwa penguatan logika *asynchronous* secara mendalam mampu mengoptimalkan performa *web service* yang dikembangkan oleh mahasiswa. Hasil evaluasi teknis menunjukkan reduksi latensi sistem yang signifikan dan peningkatan efisiensi alur kerja server melalui penerapan paradigma *non-blocking I/O*. Peserta tidak hanya mencapai kemahiran dalam penggunaan sintaksis JavaScript modern, tetapi juga memiliki kesadaran kritis terhadap manajemen sumber daya komputasi dan ketahanan sistem (*system resilience*). Integrasi antara standar industri perbankan dengan praktik pengkodean bersih (*clean code*) telah berhasil meningkatkan standar kompetensi mahasiswa ke level yang kompetitif untuk skala industri global.

Berdasarkan hasil pelaksanaan Batch 2, disarankan agar pelatihan di masa mendatang mulai mengintegrasikan alat pengujian performa otomatis (*automated performance testing*) ke dalam alur kerja pengembangan mahasiswa. Hal ini bertujuan agar aspek optimasi menjadi bagian yang tidak terpisahkan dari proses penulisan kode sejak tahap awal. Selain itu, perlu dipertimbangkan pendalaman materi mengenai arsitektur *event-driven* yang lebih kompleks, seperti implementasi *Message Queues* (misalnya RabbitMQ atau Kafka), untuk menangani beban kerja yang lebih masif.

Kolaborasi antara akademisi dan praktisi industri perlu terus diperluas guna memastikan kurikulum teknologi tetap relevan dengan dinamika transformasi digital yang sangat cepat.

UCAPAN TERIMAKASIH

Penulis menyampaikan apresiasi yang setinggi-tingginya kepada pimpinan Universitas atas dukungan sarana dan administrasi dalam pelaksanaan kegiatan PkM nasional ini. Ucapan terima kasih khusus ditujukan kepada tim manajemen Bangkit Academy 2024 atas kesempatan kedua yang diberikan kepada penulis sebagai narasumber pada Batch 2, yang memungkinkan penyempurnaan metode pengajaran dari periode sebelumnya. Penulis juga berterima kasih kepada komunitas pengembang JavaScript di Indonesia dan rekan-rekan praktisi IT perbankan atas masukan teknis mengenai tren *backend* terkini. Terakhir, penghargaan diberikan kepada seluruh peserta Bangkit 2024 Batch 2 atas dedikasi dan ketajaman intelektual mereka dalam menyelesaikan setiap tantangan teknis yang diberikan.

5. DAFTAR RUJUKAN

- Alman, T. (2022). *Node.js Design Patterns: Design and implement production-grade Node.js applications using proven patterns and techniques* (3rd ed.). Packt Publishing.
- Dahl, R. (2018). *Original Node.js Presentation: The development of non-blocking I/O*. Joyent Inc.
- Ecma International. (2023). *ECMAScript 2023 Language Specification* (14th ed.). <https://www.ecma-international.org/publications-and-standards/standards/ecma-262/>
- Haverbeke, M. (2024). *Eloquent JavaScript: A Modern Introduction to Programming* (4th ed.). No Starch Press.
- Lanthaler, M., & Guetl, C. (2019). Model-driven development of scalable backend services using Node.js. *Journal of Web Engineering*, 18(1), 1–32.
- Luo, X., Liao, Q., & Ji, S. (2020). Understanding the vulnerabilities of mobile banking apps: A performance and security perspective. *IEEE Transactions on Reliability*, 69(1), 115–128.
- Mouton, F., Leenen, L., & Venter, H. S. (2016). Social engineering attack examples and the importance of digital logic in prevention. *Computers & Security*, 59, 186–203.
- Otoritas Jasa Keuangan. (2022). *Peraturan Otoritas Jasa Keuangan Nomor 11/POJK.03/2022 tentang Penyelenggaraan Teknologi Informasi oleh Bank Umum*.
- Parsons, D., & MacCallum, K. (2021). Agile mentoring in online software engineering education: Lessons from national training programs. *Journal of Systems and Software*, 175, 110912.
- Resig, J., Bibeault, B., & Maras, J. (2016). *Secrets of the JavaScript Ninja* (2nd ed.). Manning Publications.
- Whitman, M. E., & Mattord, H. J. (2021). *Principles of Information Security* (7th ed.). Cengage Learning.