UNIVERSITAS
MUHAMMADIYAH
SIDOARJO

UMSIDA
DARI SINI PENCERAHAN BERSEMI

Implementasi dan Analisis Pendekatan Cloud-Native untuk Meningkatkan Skalabilitas dan Performa Sistem Informasi Bengkel Dwi Jaya Understel

Oleh:

Moch bagus tri cahyo

Progam Studi Informatika

Universitas Muhammadiyah Sidoarjo

Juli, 2025

Pendahuluan

- Perkembangan teknologi cloud computing mendorong efisiensi sistem informasi.
- Sistem informasi bengkel tradisional memiliki keterbatasan skalabilitas dan performa.
- Cloud-native architecture menawarkan solusi untuk meningkatkan keandalan dan fleksibilitas.

Rumusan Masalah

Rumusan Masalah:

- Bagaimana memanfaatkan cloud-native untuk meningkatkan skalabilitas sistem bengkel?
- Apa dampak implementasi terhadap kinerja aplikasi?

Tujuan:

- Menerapkan arsitektur cloud-native pada sistem informasi bengkel.
- Melakukan analisis perbandingan kinerja sebelum dan sesudah implementasi.

Metode

- Pendekatan: Eksperimen dan analisis komparatif.Arsitektur:
- Backend: AWS Lambda
- Database: RDS
- Frontend: S3 hosting
- Monitoring: CloudWatch & grafana
- Uji coba: Perbandingan dengan arsitektur tradisional berbasis VPS.

Implementasi Sistem

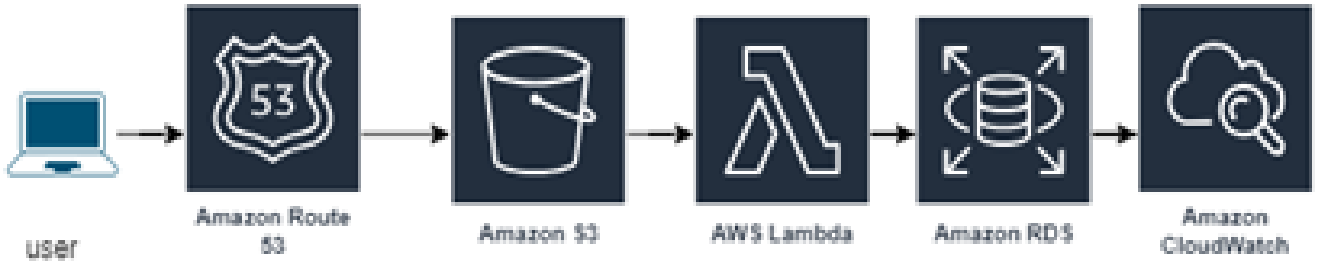
Penerapan sistem informasi bengkel berbasis cloud-native.
Integrasi komponen:

- Lambda untuk fungsi backend.
- DynamoDB untuk penyimpanan data transaksi dan sparepart.
- S3 sebagai hosting statis frontend.

CloudWatch dan grafana digunakan untuk memantau performa.

System Architecture Design

Cloud-Native Architecture



Monolithic Architecture

System Architecture Comparison

Aspect	Monolithic (Before)	Cloud-Native (AWS Lambda)	Aspect	Monolithic (Before)
Hosting	VPS (Single Server)	AWS Lambda (Serverless)	Hosting	VPS (Single Server)
Frontend Hosting	Stored on VPS	Amazon S3 (Static Hosting)	Frontend Hosting	Stored on VPS
Backend	Flask on VPS	Flask via AWS Lambda + API Gateway	Backend	Flask on VPS
Database	MySQL on the server	Amazon RDS (MySQL) or DynamoDB	Database	MySQL on the server

Hasil & Analisis

Performa: Respon sistem lebih cepat dengan Lambda.Skalabilitas: Sistem lebih mudah menyesuaikan dengan jumlah pengguna.
Efisiensi Biaya: Penggunaan sumber daya sesuai kebutuhan (pay-as-you-go).
Monitoring: CloudWatch mempermudah deteksi error.

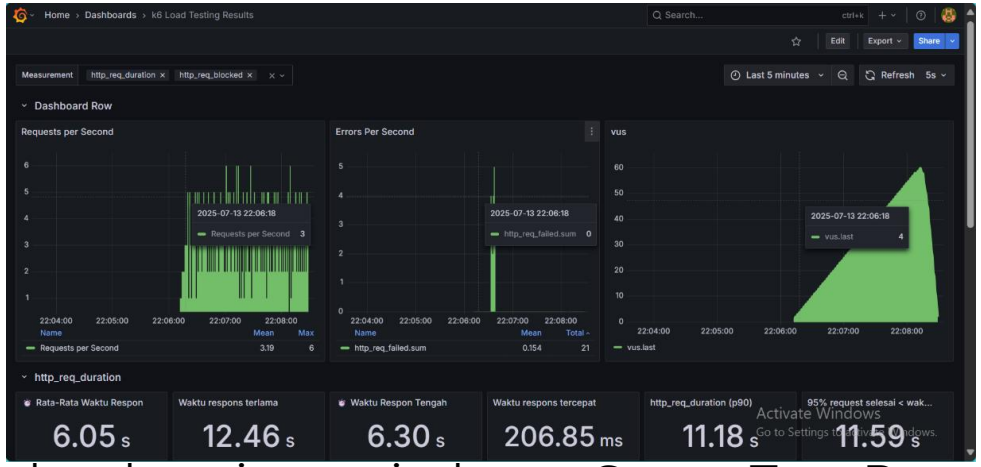
Hasil & Analisis

Stress Test Results

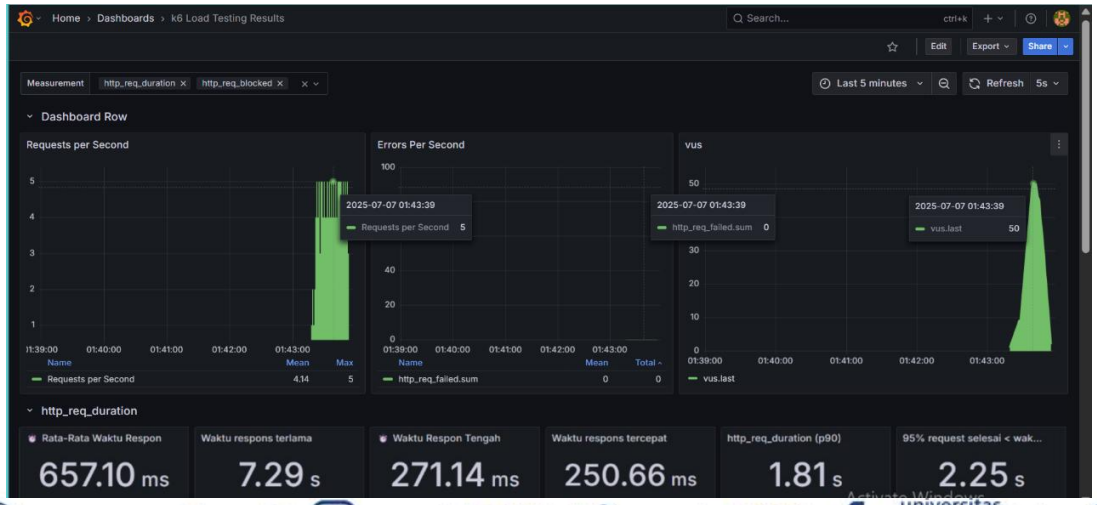
Test Case	Scenario	Virtual Users (VU)	SLA Target
TC-K01	Normal Load (open page, search spare parts) duration 1 minute	10 VU	Response < 1500ms, Error < 1%
TC-K02	Medium Load (input service transaction) duration 1 minute	20 VU	Response < 1000ms, Error < 1%
TC-K03	Peak Load (register simulation 1–60 VU) duration 1 minute	1-60 VU	Response < 1000ms, Error < 2%

Hasil & Analisis

Monolithic Architecture Stress Test Results



cloud-native arcitekture Stress Test Results



Hasil & Analisis

Monolithic Architecture Stress Test Results

Test Case	AVG Response	MAX Response	Error Rate	Status
TC-K01	1.25s	2.95s	0%	Failed
TC-K02	271.62ms	1.16s	0%	Passed
TC-K03	6.05s	12.46s	0.154%	Failed

cloud-native arcitekture Stress Test Results

Test Case	AVG Response	MAX Response	Error Rate	Status
TC-K01	2.14s	2.56s	0%	Failed
TC-K02	871.62ms	1.16s	0%	Passed
TC-K03	657.10ms	72.9s	0%	Passed

Pembahasan

Monolitik

- Baik untuk beban ringan–sedang.
- Performanya menurun signifikan saat *peak load*.

Cloud-Native

- Lebih stabil & tangguh di berbagai kondisi beban.
- Skalabilitas tinggi, error **0%** pada *peak load* .
- Tantangan utama: *cold start latency* AWS Lambda .

Keseluruhan

- Cloud-native unggul dalam skalabilitas, reliabilitas, dan fleksibilitas.
- Cocok untuk sistem dengan trafik tinggi & bervariasi.

Kesimpulan

- Arsitektur cloud-native meningkatkan skalabilitas dan kinerja sistem informasi bengkel.
- Sistem lebih fleksibel, efisien, dan mudah dikelola.
- Cocok untuk UMKM yang ingin beralih ke solusi digital berbasis cloud.

Saran

- Penelitian lanjutan dapat membandingkan layanan cloud provider lain.
- Menambahkan aspek keamanan data pada implementasi.
- Pembahsan lebih dalam mengenai cara menangani cloud strat
- Perbandingan biyaya cloud natif dan system konfensional

Referensi

[1] S. Henning and W. Hasselbring, “A configurable method for benchmarking scalability of cloud-native applications,” Empir Softw Eng, vol. 27, no. 6, Nov. 2022, doi: 10.1007/s10664-022-10162-1.

[2] K. Maydiva Heriansaputri and M. A. Romli, “Application for Mental Health Consultation with Scheduling Function at the Counseling Guidance of Universitas Teknologi Yogyakarta,” Journal of Information and Technology) Accredited Sinta, vol. 4.

[3] J. Lin, D. Xie, J. Huang, Z. Liao, and L. Ye, “A multi-dimensional extensible cloud-native service stack for enterprises,” *Journal of Cloud Computing*, vol. 11, no. 1, Dec. 2022, doi: 10.1186/s13677-022-00366-7.

[4] B. H. Lavenia, W. Hayuhardhika, N. Putra, and B. T. Hanggara, “Pengembangan Sistem Informasi Point of Sales untuk Bengkel berbasis Cloud Computing (Studi Kasus: Bengkel Mas Pur Baturaja),” 2019. [Online]. Available: http://j-ptiik.ub.ac.id

[5] D. Desmulyati and M. R. Perdana Putra, “Load Balance Design of Google Cloud Compute Engine VPS with Round Robin Method in PT. Lintas Data Indonesia,” *SinkrOn*, vol. 3, no. 2, p. 147, Mar. 2019, doi: 10.33395/sinkron.v3i2.10064.

[6] A. A’fa Nafasha, I. Putu, E. Indrawan, and G. I. Setiawan, “ANALISIS PERBANDINGAN BIAYA DAN SERVERLESS COMPUTING PADA GOOGLE CLOUD PLATFORM,” *Jurnal Manajemen dan Teknologi Informasi (JMTI)*, vol. 14, pp. 1–09, doi: 10.59819.

[7] Y. Christian and R. Bisma, “Studi Perbandingan Performa Aplikasi Web Monolitik Dan Microservice Berbasis Apache Kafka,” *Journal of Informatics and Computer Science*, vol. 03, 2021.

[8] G. Blinowski, A. Ojdowska, and A. Przybylek, “Monolithic vs. Microservice Architecture: A Performance and Scalability Evaluation,” *IEEE Access*, vol. 10, pp. 20357–20374, 2022, doi: 10.1109/ACCESS.2022.3152803.

Referensi

[9] D. Nurlaila and H. Mulyono, “Sistem Informasi Manajemen Bengkel Berbasis Web pada Bengkel Ikhsan Jaya Motor,” 2023.

[10] R. Rakhman, M. Shadiq, D. Syaddad, and V. R. Sesa, “Implementasi Cloud Native dan Multi Cloud Pada Sistem Operasi Windows Dengan Menggunakan Docker dan Cara Penggunaannya,” vol. 2, no. 5, pp. 819–826, 2024, doi: 10.5281/zenodo.13119907.

[11] S. Deng *et al.*, “Cloud-Native Computing: A Survey From the Perspective of Services,” *Proceedings of the IEEE*, vol. 112, no. 1, pp. 12–46, Jan. 2024, doi: 10.1109/JPROC.2024.3353855.

[12] A. M. Ștefan, N. R. Rusu, E. Ovreiu, and M. Ciuc, “Empowering Healthcare: A Comprehensive Guide to Implementing a Robust Medical Information System—Components, Benefits, Objectives, Evaluation Criteria, and Seamless Deployment Strategies,” *Applied System Innovation*, vol. 7, no. 3, Jun. 2024, doi: 10.3390/asi7030051.

[13] J. S. Patel, “Cloud-Native Performance Testing: Strategies for Scalability and Reliability in Modern Applications,” *European Journal of Computer Science and Information Technology*, vol. 13, no. 8, pp. 32–49, Mar. 2025, doi: 10.37745/ejcsit.2013/vol13n83249.

[14] D. Narsina, “Microservices vs. Monoliths: Comparative Analysis for Scalable Software Architecture Design”, doi: 10.18034/ei.v11i2.734.

[15] M. Bagir *et al.*, “Migrasi Cloud Native Architecture API Development untuk Memaksimalkan 5G Monetization pada Jaringan AXIATA.” [Online]. Available: https://maklumatika.i-tech.ac.id/index.php/reklamasi

[16] T. Bodner, T. Radig, D. Justen, D. Ritter, and T. Rabl, “An Empirical Evaluation of Serverless Cloud Infrastructure for Large-Scale Data Processing,” Jan. 2025, [Online]. Available: http://arxiv.org/abs/2501.07771

Referensi

[17] X. Jia, “The Role and Importance of Software Testing in Software Quality Management.” [Online]. Available: http://www.stemmpress.com

[18] S. Eismann *et al.*, “A Review of Serverless Use Cases and their Characteristics,” Jan. 2021, [Online]. Available: http://arxiv.org/abs/2008.11110

[19] C. F. Fan, A. Jindal, and M. Gerndt, “Microservices vs serverless: A performance comparison on a cloud-native web application,” in *CLOSER 2020 - Proceedings of the 10th International Conference on Cloud Computing and Services Science*, SciTePress, 2020, pp. 204–215. doi: 10.5220/0009792702040215.

[20] J. M. Menéndez, J. E. L. Gayo, E. R. Canal, and A. E. Fernández, “A comparison between traditional and Serverless technologies in a microservices setting,” May 2023, [Online]. Available: http://arxiv.org/abs/2305.13933

[21] A. Brasoveanu, M. Moodie, and R. Agrawal, “Textual evidence for the perfunctoriness of independent medical reviews,” in *CEUR Workshop Proceedings*, CEUR-WS, 2020, pp. 1–9. doi: 10.1145/nnnnnnn.nnnnnnn.

[22] A. Brasoveanu, M. Moodie, and R. Agrawal, “Textual evidence for the perfunctoriness of independent medical reviews,” in *CEUR Workshop Proceedings*, CEUR-WS, 2020, pp. 1–9. doi: 10.1145/nnnnnnn.nnnnnnn.


UMSIDA
DARI SINI PENCERAHAN BERSEMI