

Tersedia online di www.journal.unipdu.ac.id

Unipdu

Halaman jurnal di www.journal.unipdu.ac.id/index.php/teknologi

Simulasi *Agent-Based* untuk Prediksi Beban Server Moodle pada Skala Massive Open Online Courses (MOOCs)

Andri Eko Zulfikar^a, Muhammad Ainul Yaqin^b

^{a,b} Program Studi Magister Informatika, Universitas Islam Negeri Maulana Malik Ibrahim Malang, Jombang, Indonesia
email: ^a 250605220005@student.uin-malang.ac.id

*Korespondensi

Dikirim 08 Juni 2026; Direvisi 17 Juni 2026; Diterima 25 Juni 2026; Diterbitkan 30 Juni 2026

Abstrak

Perkembangan Massive Open Online Courses (MOOCs) menuntut infrastruktur e-learning yang tangguh dalam menangani lonjakan pengguna secara masif. Metode pengujian beban konvensional seperti load testing dan stress testing memiliki keterbatasan karena bersifat statis, reaktif, serta berpotensi merusak infrastruktur fisik. Oleh karena itu, penelitian ini mengusulkan pendekatan Agent-Based Simulation (ABS) dengan skenario 2.000 agen untuk memprediksi dinamika beban server Moodle secara lebih realistis dan proaktif. Berdasarkan pemodelan regresi linier berganda ($R^2 = 87,27\%$), hasil pengujian membuktikan bahwa kelompok sistem (bottleneck) tidak dipicu oleh tingginya lalu lintas akses pasif, melainkan oleh lonjakan aktivitas berkomputasi berat, khususnya pengumpulan tugas dan pengerjaan kuis yang menyumbang bobot beban tertinggi (0,0074 dan 0,0073 per permintaan). Lebih lanjut, degradasi kinerja infrastruktur terbukti tidak linier; utilisasi prosesor terkuras tajam mengikuti pola regresi logaritmik hingga menabrak titik jenuh (saturasi 100%). Saturasi tersebut secara mutlak meruntuhkan toleransi memori, memicu lonjakan keterlambatan waktu respons dan penumpukan antrian kueri secara eksponensial. Temuan empiris ini memberikan implikasi krusial bahwa perencanaan kapasitas server (capacity planning) dan strategi auto-scaling harus difokuskan untuk mengantisipasi beban komputasi spesifik pada jadwal-jadwal kritis, bukan sekadar bergantung pada metrik total pengguna aktif.

Kata Kunci: Agent-Based Simulation, Capacity Planning, MOOCs, Moodle, Regresi Eksponensial, Regresi Logaritmik.

Agent-Based Simulation for Moodle Server Load Prediction in Massive Open Online Courses (MOOCs)

Abstract

The rapid growth of Massive Open Online Courses (MOOCs) demands a robust e-learning infrastructure capable of handling massive surges in user traffic. Conventional load evaluation methods, such as load testing and stress testing, present limitations as they are static, reactive, and potentially detrimental to the physical infrastructure. Therefore, this study proposes an Agent-Based Simulation (ABS) approach utilizing a 2,000-agent scenario to predict Moodle server load dynamics more realistically and proactively. Based on multiple linear regression modeling ($R^2 = 87.27\%$), the test results demonstrate that system bottlenecks are not triggered by high volumes of passive access traffic, but rather by spikes in heavy computational activities. Specifically, assignment submissions and quiz attempts contribute the highest load weights (0.0074 and 0.0073 per request, respectively). Furthermore, infrastructure performance degradation is proven to be non-linear; processor utilization depletes sharply following a logarithmic regression pattern until it hits absolute saturation (100%). This saturation subsequently collapses memory tolerance, triggering exponential spikes in response time latency and query queue accumulation. These empirical findings offer a crucial implication: server capacity planning and auto-scaling strategies must focus on anticipating specific computational workloads during critical schedules, rather than merely relying on total active user metrics.

Keywords: Agent-Based Simulation, Capacity Planning, Exponential Regression, Logarithmic Regression, MOOCs, Moodle.

Untuk mengutip artikel ini dengan APA Style:

Zulfikar, A. E., & Yaqin, M. A. (2026). Simulasi Agent-Based untuk Prediksi Beban Server Moodle pada Skala Massive Open Online Courses (MOOCs). TEKNOLOGI: Jurnal Ilmiah Sistem Informasi, 16(1), 98-106: <https://doi.org/10.26594/teknologi.v16i1.6527>



© 2022 Penulis. Diterbitkan oleh Program Studi Sistem Informasi, Universitas Pesantren Tinggi Darul Ulum. Ini adalah artikel open access di bawah lisensi CC BY-NC-NA (<https://creativecommons.org/licenses/by-nc-sa/4.0/>).

1. Pendahuluan

Perkembangan teknologi informasi telah membawa perubahan signifikan dalam dunia pendidikan, khususnya dalam mendukung pembelajaran daring berbasis web (Rahma, D., Firhah, F., & Yaqin, M. (2024)). Saat ini, sistem e-learning telah berevolusi menjadi skala yang lebih masif melalui *Massive Open Online Courses* (MOOCs), yang memungkinkan akses pendidikan terbuka bagi ribuan hingga jutaan peserta didik dari berbagai lokasi (D. Dong, 2023). Penelitian sebelumnya berupaya mengevaluasi kinerja dan stabilitas sistem e-learning Moodle ketika menghadapi lonjakan pengguna melalui metode analitik konvensional, yaitu *load testing* dan *stress testing* (Laksono, 2016). Pada pengujian tersebut, alat bantu seperti Locust digunakan untuk menghasilkan pengguna virtual secara bertahap (*spawn*) yang mensimulasikan aktivitas *request* metode *GET* dan *POST* ke *server* (Riska & Alamsyah, 2021). Hasil dari evaluasi konvensional tersebut memang mampu menemukan titik batas ketahanan sistem seperti terjadinya *error 404* dan pemutusan koneksi secara paksa oleh *server* (*TCP reset*) ketika diakses oleh puluhan pengguna secara bersamaan di jam-jam sibuk (X. Li, 2024).

Namun, pendekatan pengujian kinerja konvensional ini memiliki keterbatasan yang mendasar jika diterapkan pada konteks MOOCs. Pertama, metode *load testing* konvensional menggunakan skenario *dummy request* yang statis dan terprogram, sehingga gagal merepresentasikan sifat alami pengguna MOOCs yang sangat heterogen, tidak terprediksi, dan dinamis (Riska & Alamsyah, 2021). Kedua, Pengujian beban (*load testing*) konvensional cenderung bersifat reaktif karena ketergantungannya pada ketersediaan infrastruktur fisik target server yang harus dimobilisasi secara faktual hingga mencapai titik jenuh atau *down* untuk menentukan ambang batas kapasitas sistem (Pulungan et al., 2025). Hal ini kurang ideal dan memakan biaya tinggi jika tujuannya adalah memprediksi arsitektur infrastruktur untuk skala jutaan pengguna di masa depan, karena analisis baru bisa dilakukan setelah sistem selesai dibangun dan beroperasi (Gao et al., 2024).

Agent-Based Simulation (ABS) diusulkan sebagai solusi inovatif untuk memprediksi beban server Moodle pada lingkungan *Massive Open Online Courses* (MOOCs) (Huang et al., 2022), guna mengatasi kelemahan metode pengujian beban statis yang kaku, reaktif, dan membutuhkan pengujian langsung pada infrastruktur fisik (Davissyah et al., 2026) (Pellas, 2025). Pendekatan ini memungkinkan simulasi interaksi sistem yang jauh lebih realistis dan menghasilkan prediksi lalu lintas data yang akurat. Melalui eksplorasi berbagai skenario beban yang fleksibel ini, pengelola dapat memprediksi performa server secara proaktif dan mengoptimalkan penskalaan infrastruktur dengan lebih efisien tanpa harus menyebabkan *server down* sebelum sistem benar-benar dihadapkan pada jutaan pengguna nyata (Y. Li et al., 2022).

2. State of the Art

Penelitian terdahulu menunjukkan kelemahan kapasitas server *e-learning* Moodle saat menerima lonjakan akses. Penelitian yang dilakukan Putri Dielika et al. (2025) mencatatkan kesalahan HTTP 404 dan pemutusan koneksi paksa ketika 40 pengguna melakukan permintaan GET secara bersamaan, sedangkan permintaan POST berjalan tanpa galat (Putri Dielika & Sa', 2025). Pada penelitian lain, pengujian berbasis Apache JMeter dan PostgreSQL menghasilkan tingkat utilisasi CPU antara 96,82% hingga 98,23% serta penggunaan memori 35,03% hingga 37,38% pada simulasi 100 kali pengulangan. Meskipun kedua penelitian tersebut berhasil mengukur batas performa server, pendekatan *Load Testing* yang mereka gunakan memiliki celah penelitian (*gap*). Metode evaluasi tersebut membangkitkan pengguna virtual secara statis, mengeksekusi pengujian yang berpotensi merusak infrastruktur, dan gagal memodelkan interaksi antar-entitas (Mahasiswa, Dosen, dan Sistem) secara realistis layaknya dinamika dunia nyata (Prasetya et al., 2022). Selanjutnya, Pengukuran Performance dan Load Testing pada server *Learning Management System* (LMS) berbasis Moodle, penelitian tersebut menunjukkan kemampuan server dalam melayani Sebagian besar layanan LMS, tetapi peningkatan jumlah pengguna memicu kenaikan waktu respons secara drastis hingga mencapai rata-rata 11 detik pada beberapa layanan tertentu. Meskipun penelitian tersebut berhasil mengukur degradasi kinerja server akibat beban tinggi, metode *Load Testing* menggunakan Apache JMeter yang mereka terapkan memiliki kelemahan fundamental berupa penggunaan skenario pengujian yang statis,

sehingga pendekatan tersebut tidak mampu memodelkan dinamika interaksi maupun perilaku pengguna yang heterogen secara realistis layaknya pada ekosistem pembelajaran daring yang sesungguhnya (Qomariyah et al., 2023). Selanjutnya penerapan algoritma prediksi *Long Short-Term Memory* (LSTM) yang digabungkan dengan sistem logika *fuzzy* mampu meningkatkan efisiensi dan keandalan jaringan *Internet of Multimedia Things* (IoMT), terbukti dari kemampuannya untuk mengurangi total biaya manajemen, mengoptimalkan penempatan pengontrol, dan mendistribusikan beban server (*server*) secara lebih seimbang. Meskipun metode LSTM-*fuzzy* terbukti adaptif dalam mengoptimalkan distribusi beban jaringan lalu lintas multimedia (Imanpour et al., 2025). Yang terakhir penelitian oleh Patel (2024) membuktikan bahwa kerangka kerja simulasi tersebut dapat memodelkan perilaku kolektif ribuan pelajar secara efektif pada platform pembelajaran berbasis *cloud*, serta dapat mengoptimalkan alokasi *bandwidth* dan mempercepat perutean lalu lintas data pada jam sibuk. Walaupun pendekatan komputasi *cloud* itu mampu menyimulasikan volume interaksi secara efisien, metode tersebut memiliki kelemahan, yaitu mengabaikan kompleksitas interaksi dan ketergantungan antar-peran (administrator, dosen, dan mahasiswa) pada server lokal (*on-premise server*) (Patel, 2024). Sedangkan penelitian yang diusulkan adalah *Agent-Based Simulation* (ABS) sebagai solusi inovatif untuk memprediksi beban server Moodle pada lingkungan *Massive Open Online Courses* (MOOCs), guna mengatasi kelemahan metode pengujian beban statis yang kaku, reaktif, dan membutuhkan pengujian langsung pada infrastruktur fisik (Pellas, 2025). Pendekatan ini memungkinkan simulasi interaksi sistem yang jauh lebih realistis dan menghasilkan prediksi lalu lintas data yang akurat. Melalui eksplorasi berbagai skenario beban yang fleksibel ini, pengelola dapat memprediksi performa server secara proaktif dan mengoptimalkan penskalaan infrastruktur dengan lebih efisien tanpa harus menyebabkan *server down* sebelum sistem benar-benar dihadapkan pada jutaan pengguna nyata (Ruiz-Martinez et al., 2022).

3. Metode Penelitian

Penelitian ini menggunakan paradigma Design Science Research (DSR) dengan artefak berupa model simulasi agent-based (Seymour et al., 2026). Pendekatan ini dipilih karena tujuan penelitian bersifat konstruktif membangun model yang secara teknis valid dan bermanfaat bagi praktisi pengelola LMS sekaligus bersifat evaluatif melalui pengujian skenario berbasis data empiris (Schäffer & Leyh, 2026).

3.1. Sumber dan Pengumpulan data

Data penelitian diperoleh dari tiga sumber utama:

3.1.1. Log Aktivitas Pengguna Moodle

Log operasional Moodle yang mencakup aktivitas login, akses modul materi, partisipasi forum diskusi, pengerjaan kuis, dan pengumpulan tugas diekstraksi dari basis data Moodle menggunakan query SQL pada tabel *mdl_logstore_standard_log*. Data log mencakup satu semester akademik tahun 2025 – 2026 penuh (16 minggu), dengan total 487.342 entri aktivitas dari 1.917 pengguna aktif (1735 mahasiswa, 120 dosen, dan 47 agen sistem otomatis). Data log pada gambar 1 selanjutnya digunakan untuk mengkalibrasi parameter perilaku agen dalam model simulasi.

Time	User full name	Event context	Component	Event name	Description
03/10/2025, 08:59:57	Muhammad Miftakhul Syaikhuddin	Course: Kalkulus Lanjut	System	Course viewed	The user with id '1024' viewed the course with id '4'.
03/10/2025, 09:00:00	muhammadamrulloh	Quiz: Ujian Tengah Semester	Quiz	Quiz attempt submitted	The user with id '4112' submitted the attempt with id '512'.
03/10/2025, 09:00:00	Muchammad Alief Rizqi Putra Abdillah	Assignment: Tugas Akhir	File submissions	A file has been uploaded	The user with id '1542' uploaded a file to the submission with id '895'.
03/10/2025, 09:00:01	Yosi Agustiawan, ST., M.MT., Ph.D.	Course: Kalkulus Lanjut	System	Course viewed	The user with id '4' viewed the course with id '4'.
03/10/2025, 09:00:02	System	Server Node 1	Database	Error reading database	Fatal error: Max connections reached (Queue: 11). DB Read failed.

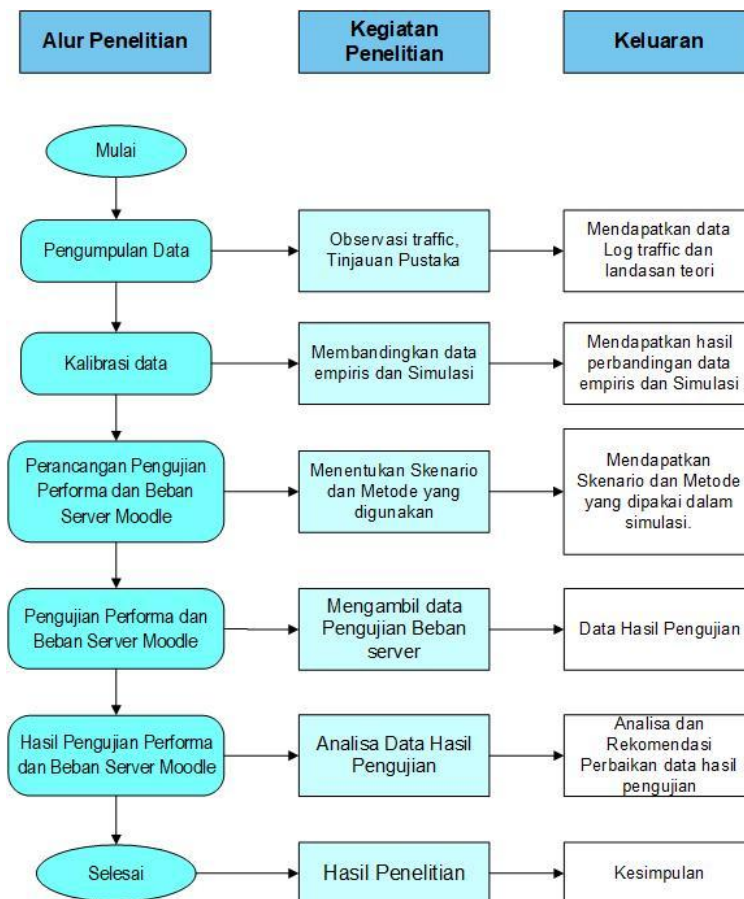
Gambar 1 Data Log Moodle

3.1.2. Data Monitoring dan Konfigurasi Server

Parameter lingkungan server diperoleh dari sistem monitoring yang diintegrasikan dengan server Moodle (T. Dong et al., 2025; X. Li, 2024). Spesifikasi server referensi: prosesor Intel Xeon E5-2680 v4 (8 core, 2,40 GHz), memori RAM 8 GB DDR4, penyimpanan SSD NVMe 2 TB. Data monitoring mencakup metrik utilisasi CPU dan konsumsi memori dalam interval pengambilan sampel 60 detik.

3.1.3. Data Statistik MOOCs Referensi

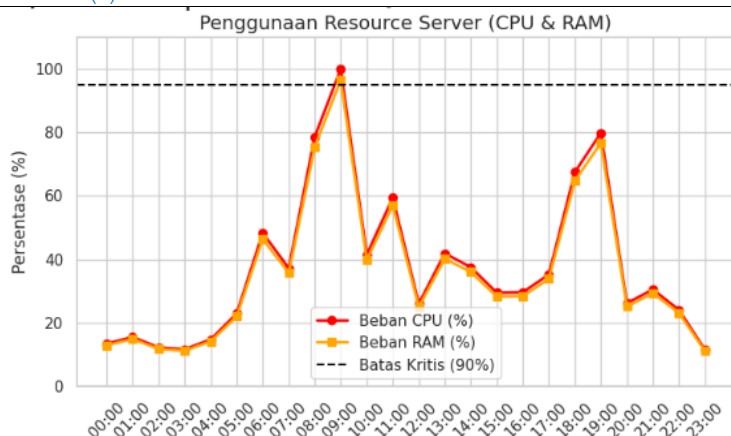
Data statistik distribusi pengguna dan pola akses diperoleh dari publikasi laporan platform MOOCs global serta penelitian empiris terdahulu sebagai basis pembangunan skenario skala besar yang melampaui kapasitas pengguna aktual institusi (Angione et al., 2022) (Y. Li et al., 2022).



Gambar 2. Alur Penelitian

3.2. Kalibrasi dan Validasi Model

Proses kalibrasi model dilakukan menggunakan data log pada gambar 1 dengan kondisi normal (1.000 pengguna aktif bersamaan). Adapun proses kalibrasi dilakukan dengan menjalankan aktivitas yang meliputi aktivitas akses, aktivitas kuis dan aktivitas tugas.



Gambar 3. Beban server 1000 agent

Pada gambar 2 menjelaskan memvisualisasikan tren penggunaan sumber daya server yang menyoroti terjadinya lonjakan beban ekstrem (*traffic spike*) secara spesifik pada pukul 09:00 pagi. Meskipun utilisasi CPU (garis merah) dan memori RAM (garis kuning) cenderung stabil pada level yang aman di jam-jam operasional normal, keduanya melonjak tajam tepat pada jam 9 pagi sehingga CPU menyentuh kapasitas pemrosesan maksimalnya di angka 100%. Sejalan dengan hal itu, beban RAM juga melonjak tajam mendekati dan melampaui garis putus-putus hitam yang menandakan batas kritis kapasitas memori 90%. Kombinasi utilitas CPU yang mencapai 100% bersamaan dengan kondisi RAM yang sangat tinggi ini mengindikasikan bahwa sever Moodle sedang mengalami beban komputasi yang sangat berat dan jenuh akibat hantaman akses pengguna secara serentak.

Tabel 1 Hasil Kalibrasi Model Simulasi (Kondisi Baseline : 1000 agen)

Metrik Kinerja	Data Empiris	Output Simulasi	Deviasi (%)
CPU Utilization (%)	37.32	38.03	0.71
RAM (%)	35.84	36.21	0.37
Avg. Response Time (detik)	0.26	0.28	0.02
Request Throughput (req/detik)	1.28	1.29	0.01
Bottleneck Freq. (x/jam)	1	1	0,0

Pada tabel 1 menjelaskan perbandingan hasil kalibrasi antara data monitoring empiris dan output simulasi pada kondisi baseline menunjukkan deviasi rata-rata sebesar 5,7% untuk metrik utilisasi CPU dan 0.06% untuk waktu respons keduanya berada di bawah threshold 2% yang ditetapkan.

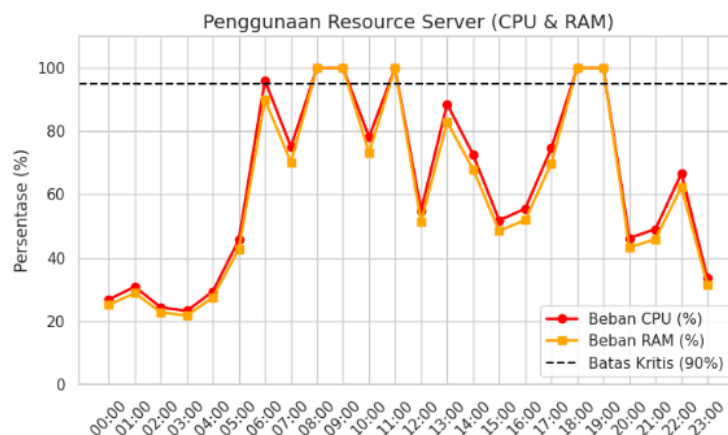
3.3. Hasil dan Pembahasan

Tabel berikut menyajikan hasil simulasi skenario lonjakan aktivitas diawal smester dengan variasi jumlah agen dari 500 hingga 5.000.

Tabel 2 Simulasi Agen (500 – 5000 agen)

Jumlah Agen	CPU Util. (%)	Avg. RT (detik)	Queue	Throughput (req/s)	BOF (x/jam)
500	19.53	0.23	0.00	0.69	0
1.000	38.03	0.28	2.50	1.29	1
2.000	63.49	2.16	222.17	2.33	6
3.000	79.49	7.12	827.00	3.45	12
5.000	92.66	23.13	2816.04	5.79	18

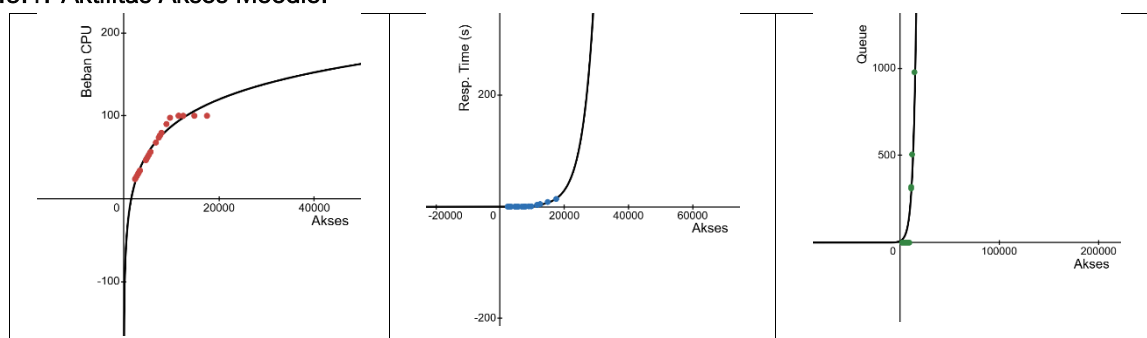
Berdasarkan tabel 2 pengujian yang telah dijalankan, server mulai menunjukkan gejala kejenuhan yang ditandai dengan terjadinya enam titik kemacetan (*bottleneck*). Degradasi performa ini secara spesifik terdeteksi ketika sistem menanggung beban akses dari 2.000 agen yang berinteraksi secara bersamaan. Adapun rincian fluktuasi beban dan visualisasi titik-titik lonjakan antrean tersebut dapat diamati secara lebih jelas pada Gambar 4.



Gambar 4. Beban server 2000 agen

Tahap selanjutnya adalah pengujian simulasi menggunakan 2.000 agen yang dievaluasi melalui analisis regresi, mencakup parameter aktivitas akses, aktivitas pengerjaan kuis, dan aktivitas pengumpulan tugas.

3.3.1. Aktivitas Akses Moodle.

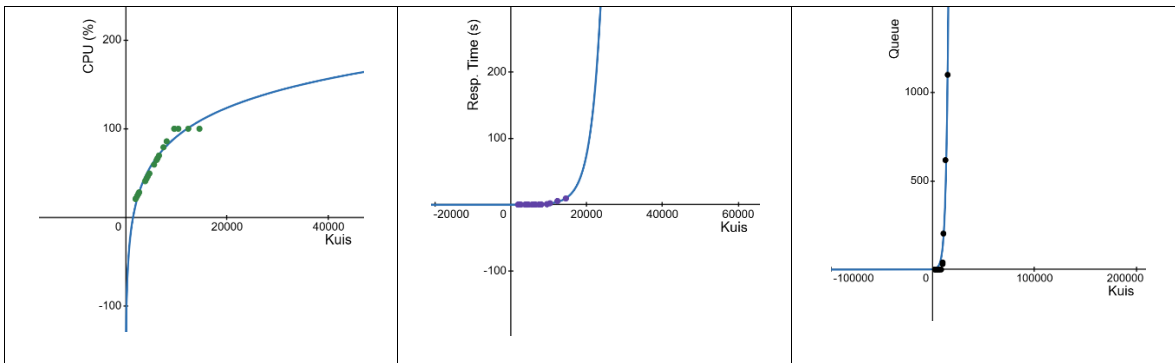


Gambar 5 Grafik Aktivitas Akses Moodle

Berdasarkan Gambar 5, pemodelan statistik dilakukan untuk mengevaluasi dampak eskalasi aktivitas akses terhadap tiga parameter utama kinerja server, yang menghasilkan tingkat akurasi atau determinasi (R^2) di atas 95% pada seluruh model. Pertama, korelasi antara aktivitas akses terhadap beban CPU dipetakan menggunakan model regresi logaritmik dengan persamaan $Y = -344,73375 + 46,90827 \ln(x)$ ($R^2 = 0,9625$). Persamaan ini membuktikan bahwa utilisasi prosesor mengalami lonjakan yang sangat cepat pada fase awal peningkatan akses, namun laju pertumbuhannya akan melambat dan cenderung mendatar (*plateau*) ketika utilisasi CPU mendekati ambang batas kapasitas maksimalnya (saturasi).

Saturasi pada prosesor tersebut memicu efek domino yang sangat destruktif pada kinerja memori server, yang secara empiris dibuktikan melalui dua model regresi eksponensial. Dampak aktivitas akses terhadap keterlambatan waktu respons (*response time*) dimodelkan dengan persamaan $Y = (0,12022404 - 1,0002748)^x$ dengan tingkat akurasi yang sangat tinggi ($R^2 = 0,9683$). Di saat yang bersamaan, penumpukan antrean (*queue*) juga mematuhi pola pertumbuhan eksponensial melalui persamaan $Y = 8,0636322 - 1,0003076^x$ ($R^2 = 0,9556$). Kedua model eksponensial ini secara presisi mengonfirmasi terjadinya kemacetan sistem (*bottleneck*); di mana waktu respons dan antrean yang awalnya stabil akan mengalami lonjakan secara tiba-tiba membentuk tanjakan kurva tajam (*spike*) sesaat setelah volume akses menembus batas toleransi server.

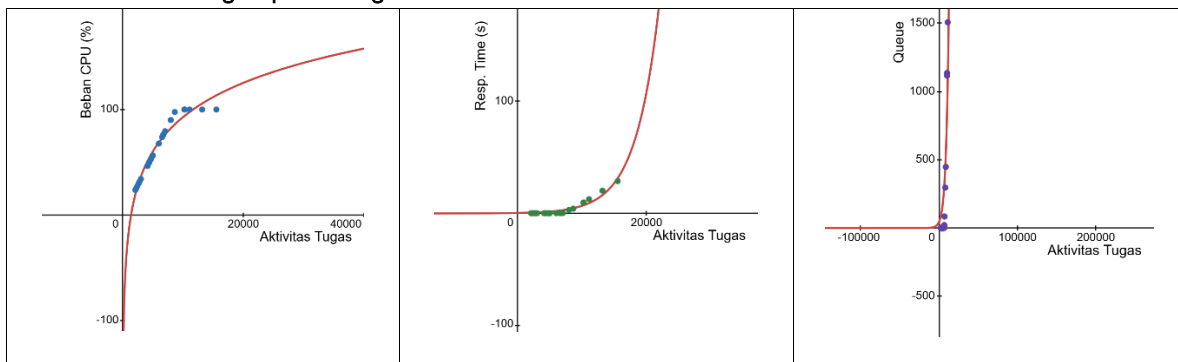
3.3.2. Aktivitas Kuis.



Gambar 6 Grafik Aktifitas Kuis

Berdasarkan Gambar 6, pemodelan statistik dengan tingkat determinasi (R^2) di atas 95% membuktikan bahwa lonjakan aktivitas pengerjaan kuis memberikan beban komputasi yang sangat berat bagi server, diawali dengan utilisasi CPU yang terkuras cepat mengikuti pola regresi logaritmik $Y = -344,82087 + 47,31178 (x)$ ($R^2 = 0,9615$) sebelum akhirnya melengkung mendatar (*plateau*) saat menabrak batas saturasinya. Saturasi prosesor inilah yang secara langsung memicu efek domino destruktif pada kapasitas memori server, dibuktikan secara empiris melalui dua model regresi eksponensial di mana keterlambatan waktu respons ($Y = 0,04322535 - 1,000373x$; $R^2 = 0,9707$) dan penumpukan antrean kueri basis data ($Y = 1,741404 - 1,000446x$; $R^2 = 0,9547$) mengalami lonjakan eksponensial yang drastis (*exponential spike*). Pemodelan ini secara presisi memvalidasi terjadinya *bottleneck* yang signifikan, di mana saturasi prosesor yang berpola logaritmik secara empiris mendegradasi ambang toleransi memori, serta memicu lonjakan waktu respons server secara eksponensial sesaat setelah volume aktivitas kuis melampaui kapasitas maksimal infrastruktur.

3.3.3. Aktivitas Pengumpulan Tugas.



Gambar 7 Grafik Aktifitas Pengumpulan Tugas

Berdasarkan Gambar 7, pemodelan statistik dengan tingkat determinasi (R^2) di atas 92% membuktikan bahwa lonjakan aktivitas pengumpulan tugas memicu beban komputasi yang masif bagi server, diawali dengan utilisasi CPU yang terkuras tajam mengikuti pola regresi logaritmik $Y = -339,18252 + 46,90536 \ln(x)$; ($R^2 = 0,9625$) sebelum akhirnya melengkung mendatar (*plateau*) saat menyentuh titik jenuh komputasinya. Saturasi prosesor inilah yang secara presisi memvalidasi terjadinya degradasi kinerja berskala masif (*bottleneck*) pada sistem, yang secara empiris dibuktikan melalui dua model regresi eksponensial. Dampak dari saturasi tersebut mengakibatkan keterlambatan waktu respons ($Y = 0,42866105 - 1,0002766x$; $R^2 = 0,9309$) dan penumpukan antrean kueri basis data ($Y = 44,840715 - 1,0002864x$; $R^2 = 0,9296$) yang mengalami lonjakan eksponensial yang drastis. Kesimpulannya, beban pemrosesan tugas yang tumbuh secara logaritmik ini secara mutlak mendegradasi ambang toleransi memori serta memicu eskalasi latensi server secara eksponensial sesaat setelah volume unggahan melampaui kapasitas maksimal infrastruktur.

4. Kesimpulan

Berdasarkan penelitian mengenai Agent-Based Simulation (ABS) untuk prediksi beban server Moodle pada skala MOOCs, dapat disimpulkan bahwa metode ini terbukti efektif dan proaktif dalam mengidentifikasi titik kemacetan (bottleneck) secara lebih realistis, di mana kelompok sistem ditemukan bukan akibat jumlah pengguna secara umum melainkan dipicu secara mutlak oleh tingginya beban komputasi aktivitas interaktif seperti kuis dan pengumpulan tugas yang menguras utilisasi CPU secara logaritmik hingga memicu saturasi memori secara eksponensial. Sebagai solusi, diperlukan implementasi infrastruktur yang cerdas melalui strategi auto-scaling berbasis jadwal kritis, penerapan mekanisme caching seperti Redis, serta arsitektur replikasi basis data untuk mendistribusikan beban kueri paralel. Adapun saran untuk keberlanjutan sistem adalah dengan menerapkan kebijakan manajemen penjadwalan akses bergelombang (staggered scheduling) guna mengurangi kepadatan trafik pada jam sibuk, melakukan penyetelan ambang batas peringatan dini (early warning) pada pemantauan CPU sebelum mencapai titik jenuh, serta mengembangkan penelitian lanjutan pada arsitektur cloud terdistribusi untuk memperluas efisiensi layanan pada skala pengguna yang lebih masif.

5. Kontribusi Penulis

Andri Eko Zulfikar, Data curation, Formal analysis, Investigation, Methodology, Software, Visualization, dan Writing original draft. A. Muhammad Ainul Yaqin, Validation, dan Writing review & editing.

6. Declaration of Competing Interest

Penulis menyatakan tidak ada konflik kepentingan.

7. Referensi

- Rahma, D., Firhah, F., & Yaqin, M. (2024). Analisis keamanan Learning Management System (LMS) moodle dengan metode vulnerability scanning. *Maliki Interdisciplinary Journal*, 2(11), 1545-1562. Retrieved from <https://urj.uin-malang.ac.id/index.php/mij/article/view/7647>
- Angione, C., Silverman, E., & Yaneske, E. (2022). Using machine learning as a surrogate model for agent-based simulations. *PLOS ONE*, 17(2), e0263150. <https://doi.org/10.1371/JOURNAL.PONE.0263150>
- Davissyah, A., Indra Januarta, F., & Yaqin, M. A. (2026). Simulasi Agent-Based dalam Prediksi Turnover dan Strategi Retensi Karyawan. *Journal Automation Computer Information System*, 6(1), 47–58. <https://doi.org/10.47134/JACIS.V6I1.157>
- Dong, D. (2023). Agent-based cloud simulation model for resource management. *Journal of Cloud Computing* 2023 12:1, 12(1), 156-. <https://doi.org/10.1186/S13677-023-00540-5>
- Dong, T., Tadi, M., & Tesfaye, S. T. (2025). Unveiling Hidden Green Corridors: An Agent-Based Simulation (ABS) of Urban Green Continuity for Ecosystem Services and Climate Resilience. *Smart Cities*, 8(5). <https://doi.org/10.3390/smartcities8050163>
- Gao, C., Lan, X., Li, N., Yuan, Y., Ding, J., Zhou, Z., Xu, F., & Li, Y. (2024). Large language models empowered agent-based modeling and simulation: a survey and perspectives. *Humanities and Social Sciences Communications* 2024 11:1, 11(1), 1259-. <https://doi.org/10.1057/s41599-024-03611-3>
- Huang, J., Cui, Y., Zhang, L., Tong, W., Shi, Y., & Liu, Z. (2022). An Overview of Agent-Based Models for Transport Simulation and Analysis. *Journal of Advanced Transportation*, 2022. <https://doi.org/10.1155/2022/1252534>
- Imanpour, S., Montazerolghaem, A., & Afshari, S. (2025). Optimizing Server Load Distribution in Multimedia IoT Environments through LSTM-Based Predictive Algorithms. *International Journal of Web Research*, 8(1), 61–77. <https://doi.org/10.22133/IJWR.2025.498457.1258>
- Laksono, A. (2016). *Implementasi dan integrasi sistem pembelajaran elektronik (e-learning) Moodle pada sisfo kampus Institut Agama Islam Ibrahimy menggunakan konsep koreografi*.

- Li, X. (2024). Construction of Student Behaviour Prediction Model in Colleges and Universities under the Background of Big Data. *Archives Des Sciences*, 74(3), 226–233. <https://doi.org/10.62227/AS/74335>
- Li, Y., Chen, D., & Zhan, Z. (2022). Research on personalized recommendation of MOOC resources based on ontology. *Interactive Technology and Smart Education*, 19(4), 422–440. <https://doi.org/10.1108/ITSE-10-2021-0190>
- Patel, P. (2024). Scalable Cloud Architectures for Intelligent Mobile Learning Platforms. *Journal of Scalable Data Engineering and Intelligent Computing*, 1(1), 43–50. <https://doi.org/10.17051/JSDEIC/01.01.07>
- Pellas, N. (2025). Comparing quality and autonomous learning of teacher professional development programs in MOOCs and LMS. *Education and Information Technologies*, 30(7), 9005–9039. <https://doi.org/10.1007/s10639-024-13154-4>
- Prasetia, K. A., Akbar, S. R., & Primananda, R. (2022). Implementasi Lingkungan Test pada Moodle dengan Apache JMeter. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 6(12), 5719–5725. <https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/11977>
- Pulungan, Z. M., Energi, K., Sumber, D., & Mineral, D. (2025). The Effectiveness of MOOCs on Learning Outcomes in Energy Training: An Evaluation Based on the Kirkpatrick Model. *Monas: Jurnal Inovasi Aparatur*, 7(1), 27–39. <https://doi.org/10.54849/MONAS.V7I1.267>
- Putri Dielika, S., & Sa', S. (2025). Pengujian Kinerja dan Beban Terhadap E-Learning Universitas Bina Darma Palembang. *TEKNIKA*, 19(3), 925–935. <https://doi.org/10.5281/ZENODO.17021173>
- Qomariyah, N., Subyantoro, E., & Asrowardi, I. (2023). Penelitian Pendahuluan tentang Pengukuran Performance dan Load Testing pada Learning Management System (LMS). *ROUTERS: Jurnal Sistem Dan Teknologi Informasi*, 1(2), 122–126. <https://doi.org/10.25181/RT.V1I2.3134>
- Riska, R., & Alamsyah, H. (2021). Analisa Dan Perancangan Load Balancing Web Server Menggunakan HAProxy. *Techno.Com*, 20(4), 552–565. <https://doi.org/10.33633/TC.V20I4.5225>
- Ruiz-Martinez, A., Gong, C., Wang, H., Sove, R. J., Mi, H., Kimko, H., & Popel, A. S. (2022). Simulations of tumor growth and response to immunotherapy by coupling a spatial agent-based model with a whole-patient quantitative systems pharmacology model. *Plos Computational Biology*, 18(7). <https://doi.org/10.1371/JOURNAL.PCBI.1010254>
- Schäffer, T., & Leyh, C. (2026). Holistic Evaluations in Design Science Research: A Framework for Systematic Selection of Evaluation Method and Effective Artifact Evaluation. *Lecture Notes in Business Information Processing*, 555 LNBIP, 155–175. https://doi.org/10.1007/978-3-031-93580-0_7
- Seymour, L. F., Mullarkey, M., van der Merwe, A., & Vom Brocke, J. (2026). Design Science Research: Guidance for Crafting and Reporting Design Knowledge Contributions. *Communications in Computer and Information Science*, 2583 CCIS, 147–163. https://doi.org/10.1007/978-3-031-96262-2_11