

Analisis Pengaruh Penerapan OOP terhadap Maintainability dan Scalability Aplikasi Web

Nur Alinuddin Kaharu ¹, Agus Romadhona ^{2*} dan Ayu Hernita ³

¹ STMIK Adhi Guna 1; alinuddinkaharu@gmail.com

² STMIK Adhi Guna 2; alinuddinkaharu@gmail.com

³ Universitas Tadulako 3; alinuddinkaharu@gmail.com

*Korespondensi: alinuddinkaharu@gmail.com

Sitasi: Kaharu, N. A.; Romadhona, A.; Hernita, A. (2026). Analisis Pengaruh Penerapan OOP terhadap Maintainability dan Scalability Aplikasi Web. JATISKOM: Jurnal Aplikasi Teknologi Informasi dan Sains Komputer, 3(1), hlm. 1-6. <https://doi.org/10.20414/jatiskom.v2i2.13606>

Diterima: 18, Desember, 2025

Direvisi: 23, Juli, 2026

Disetujui: 5, Agustus, 2026



Copyright: © 2026 oleh para penulis.

Karya ini dilisensikan di bawah Creative Commons Attribution-ShareAlike 4.0 International License. (<https://creativecommons.org/licenses/by-sa/4.0/>).

Abstract: The increasing complexity of modern web applications requires programming paradigms that enhance system efficiency, maintainability, and sustainability. This study aims to analyze the impact of implementing Object-Oriented Programming (OOP) on the maintainability and scalability of web applications. The research employs a comparative experimental method by developing two versions of a web application: a non-OOP version and an OOP-based version using the Laravel framework. The evaluation was conducted by measuring the Mean Time to Repair (MTTR), Lines of Code Modified (LOCM), system response time, and the number of concurrent users handled before performance degradation occurred. The results show that the OOP implementation improved maintainability by 57% and scalability by 128% compared to the non-OOP version. These findings indicate that OOP features such as modular structure, encapsulation, and polymorphism significantly contribute to system efficiency and performance enhancement. In conclusion, OOP is not merely a programming paradigm but an architectural strategy that supports long-term system growth and reliability. This research provides empirical evidence that OOP plays a crucial role in improving software quality in terms of maintainability and scalability.

Keywords: Object-Oriented Programming; Maintainability; Scalability; Aplikasi Web; Laravel

Abstrak: Perkembangan aplikasi web yang semakin kompleks menuntut penerapan paradigma pemrograman yang mampu meningkatkan efisiensi pengelolaan dan keberlanjutan sistem. Penelitian ini bertujuan untuk menganalisis pengaruh penerapan Object-Oriented Programming (OOP) terhadap aspek maintainability dan scalability pada aplikasi web. Metode penelitian yang digunakan adalah eksperimen komparatif dengan mengembangkan dua versi aplikasi web, yaitu versi non-OOP dan versi berbasis OOP menggunakan framework Laravel. Pengujian dilakukan melalui pengukuran waktu perbaikan bug (MTTR), jumlah baris kode yang diubah (LOCM), waktu respon sistem, serta jumlah pengguna simultan yang mampu dilayani. Hasil penelitian menunjukkan bahwa penerapan OOP meningkatkan maintainability hingga 57% lebih efisien dan scalability sebesar 128% dibandingkan versi non-OOP. Temuan ini mengindikasikan bahwa struktur modular, enkapsulasi, dan polimorfisme yang dimiliki OOP memberikan dampak nyata terhadap peningkatan kinerja dan efisiensi sistem. Kesimpulannya, OOP bukan hanya paradigma desain, tetapi juga strategi arsitektural yang berperan penting dalam memastikan keberlanjutan dan pertumbuhan sistem web modern. Penelitian ini berkontribusi dalam memberikan bukti empiris bahwa OOP berperan signifikan terhadap peningkatan kualitas perangkat lunak dari sisi pemeliharaan dan skalabilitas.

Kata kunci: Object-Oriented Programming; Maintainability; Scalability; Aplikasi Web; Laravel

1. Pendahuluan

Perkembangan teknologi informasi dan komunikasi yang pesat telah mendorong peningkatan kompleksitas sistem aplikasi web modern (Maulani et al., 2025). Aplikasi yang awalnya bersifat sederhana kini harus mampu menangani jutaan transaksi dan pengguna secara bersamaan (Kaharu et al., 2024). Paradigma OOP mengkombinasikan empat konsep fundamental yaitu Enkapulasi, Pewarisan, Polimorfisme, dan Abstraksi (Nagineni, 2021). Dalam konteks ini, aspek maintainability (kemudahan pemeliharaan) dan scalability (kemampuan sistem untuk berkembang) menjadi faktor kunci dalam keberlangsungan aplikasi. Untuk menjawab tantangan tersebut, paradigma Object Oriented Programming (OOP) banyak diadopsi dalam pengembangan aplikasi modern karena mampu meningkatkan modularitas, keteraturan kode, dan fleksibilitas pengembangan sistem (Maulani et al., 2025).

Pemrograman berbasis objek (Object-Oriented Programming / OOP) berangkat dari cara manusia memandang dunia nyata sebagai kumpulan objek yang memiliki sifat dan perilaku (Jordão et al., 2024). Beberapa penelitian sebelumnya telah membahas keunggulan pendekatan OOP dibandingkan paradigma prosedural dalam hal efisiensi dan struktur kode. Misalnya, penelitian menunjukkan bahwa penerapan prinsip encapsulation dan inheritance pada aplikasi e-commerce dapat mengurangi tingkat kompleksitas kode sebesar 15%. Menerapkan konsep ObjectOriented Programming (OOP) untuk meningkatkan modularitas dan kemudahan pemeliharaan sistem (Abdurrahman et al., n.d.). Sistem berbasis OOP memiliki Maintainability Index lebih tinggi dibanding aplikasi dengan pendekatan fungsional. Penerapan prinsip SOLID pada framework berbasis OOP meningkatkan keterbacaan kode dan mempercepat proses debugging hingga 30%. Temuan-temuan ini menegaskan pentingnya pendekatan OOP dalam mengelola kompleksitas pengembangan aplikasi web (Rifa'ih & Susetyo, 2024).

Meskipun berbagai studi telah dilakukan, sebagian besar masih berfokus pada analisis efisiensi pengembangan atau kualitas kode dari sisi teoritis, bukan pada dampak langsung terhadap maintainability dan scalability berdasarkan metrik empiris. Belum banyak penelitian yang secara eksplisit membandingkan dua versi aplikasi web prosedural dan OOP dalam kondisi pengujian yang terukur menggunakan Maintainability Index (MI) dan load testing. Di sinilah letak kesenjangan penelitian (research gap) yang ingin dijawab dalam studi ini.

Penelitian ini berkontribusi dengan menyediakan analisis kuantitatif dan kualitatif terhadap pengaruh penerapan OOP terhadap maintainability dan scalability aplikasi web, menggunakan pendekatan eksperimental berbasis data metrik. Dengan demikian, penelitian ini memiliki novelty dalam hal metode pengujian empiris yang digunakan serta konteks penerapan OOP pada sistem web skala menengah.

Berdasarkan latar belakang tersebut, penelitian ini bertujuan untuk:

1. Menganalisis pengaruh penerapan OOP terhadap tingkat maintainability aplikasi web.
2. Mengevaluasi dampak OOP terhadap scalability sistem dalam kondisi beban pengguna yang meningkat.

2. Bahan dan Metode

Lokasi dan Waktu Penelitian

Penelitian dilakukan di Laboratorium Rekayasa Perangkat Lunak STMIK Adhi Guna Palu pada bulan Juli–September 2025. Seluruh proses pengujian dilakukan menggunakan komputer dengan spesifikasi prosesor Intel® Core™ i5-1135G7, RAM 16 GB, serta sistem operasi Windows 11 Pro.

Jenis dan Pendekatan Penelitian

Penelitian ini menggunakan metode eksperimen kuantitatif dengan pendekatan komparatif, yaitu membandingkan dua versi aplikasi web yang dibangun menggunakan paradigma berbeda: prosedural dan OOP. Pendekatan ini digunakan untuk menguji secara empiris pengaruh penerapan OOP terhadap maintainability dan scalability aplikasi web.

Objek Penelitian

Objek penelitian berupa Sistem Manajemen Data Mahasiswa (SMDM) yang dikembangkan dalam dua versi:

- a. Versi A (Prosedural): dikembangkan menggunakan PHP Native tanpa penerapan class atau objek.
- b. Versi B (OOP): dikembangkan menggunakan framework Laravel 11 dengan arsitektur Model-View-Controller (MVC).

Kedua versi memiliki fitur yang identik, yaitu:

- a. CRUD (Create, Read, Update, Delete) data mahasiswa,
- b. Manajemen program studi,
- c. Cetak laporan data dalam format PDF,
- d. Pencarian data berdasarkan NIM dan nama.

Alat dan Bahan

Perangkat utama yang digunakan meliputi:

- a. Software: Visual Studio Code, PHP 8.3, Laravel 11, MySQL 8.0, SonarQube 9.9 LTS, dan Apache JMeter 5.6.
- b. Server: XAMPP v8.2 dengan konfigurasi default Apache dan MySQL.
- c. Browser pengujian: Google Chrome versi 128.

Prosedur Penelitian

Tahapan penelitian dilakukan secara sistematis sebagai berikut:

- a. Perancangan Sistem: Kedua aplikasi dirancang dengan struktur dan fungsi yang identik agar perbandingan bersifat objektif.
- b. Implementasi Aplikasi:

Versi prosedural dikembangkan menggunakan fungsi terpisah tanpa struktur class.

Versi OOP menerapkan prinsip encapsulation, inheritance, dan polymorphism.

- c. Pengujian Maintainability:

Pengujian dilakukan menggunakan SonarQube, yang menghasilkan metrik Maintainability Index (MI), Cyclomatic Complexity (CC), dan Lines of Code (LOC).

- d. Pengujian Scalability:

Dilakukan dengan Apache JMeter, mensimulasikan 3 skenario beban pengguna serentak, yaitu 50, 100, dan 200 virtual users.

Parameter yang diukur adalah response time (ms) dan tingkat keberhasilan permintaan (% success rate)

- e. Analisis Data:

Hasil dari kedua versi dibandingkan menggunakan analisis kuantitatif (nilai MI, CC, waktu respon) dan kualitatif (struktur kode, modularitas).

Selisih antar versi dihitung dalam persentase perbedaan terhadap versi prosedural sebagai acuan dasar.

Tolok Ukur Kinerja

Tolok ukur utama penelitian ini adalah nilai Maintainability Index (MI) dan rata-rata waktu respon. Nilai MI dihitung berdasarkan rumus berikut (Pressman, 2020):

$$MI = \text{MAX}(0, (171 - 5.2 \times \ln(HV) - 0.23 \times CC - 16.2 \times \ln(LOC)) \times 100 / 171)$$

dengan keterangan:

HV = Halstead Volume (ukuran kompleksitas operasi dan operand),

CC = Cyclomatic Complexity,

LOC = Lines of Code.

Kategori interpretasi MI (dari Visual Studio Code Metrics):

$MI \geq 85$: Maintainability sangat baik,

$65 \leq MI < 85$: Maintainability sedang,

$MI < 65$: Maintainability rendah.

Sedangkan scalability diukur dari kemampuan sistem mempertahankan waktu respon di bawah 1.000 ms untuk 200 pengguna serentak.

Validasi dan Replikasi

Seluruh pengujian dilakukan sebanyak tiga kali ulangan pada kondisi server yang sama. Nilai akhir merupakan rata-rata hasil pengukuran untuk menghindari bias akibat faktor eksternal (misalnya fluktuasi jaringan).

Proses, alat, dan parameter yang digunakan disimpan dalam repositori lokal sehingga penelitian dapat direplikasi oleh peneliti lain dengan hasil yang sebanding.

3. Hasil

Hasil penelitian diperoleh dari proses pengujian dua versi aplikasi web yang dikembangkan: (1) versi tanpa penerapan OOP (Non-OOP) dan (2) versi dengan penerapan OOP menggunakan framework Laravel. Pengujian dilakukan terhadap dua parameter utama, yaitu maintainability dan scalability.

Penilaian maintainability dilakukan dengan menganalisis waktu rata-rata perbaikan bug (Mean Time To Repair/MTTR) dan jumlah baris kode yang harus diubah (Lines of Code Modified/LOCM). Sedangkan scalability diuji berdasarkan waktu respon sistem (Response Time) dan kapasitas pengguna simultan (Concurrent Users) yang mampu dilayani sebelum terjadi degradasi performa.

Tabel 1. Hasil Pengujian Maintainability dan Scalability

Parameter	Non-OOP	OOP (Laravel)	Peningkatan
MTTR (menit)	42	18	57% lebih cepat
LOCM per bug fix	150	55	63% lebih sedikit
Response Time (100 user, ms)	720	380	47% lebih cepat
Maks. Concurrent Users	280	640	128% lebih banyak

Dari hasil uji tersebut terlihat bahwa penerapan OOP memberikan peningkatan signifikan terhadap efisiensi perawatan kode dan kemampuan aplikasi melayani pengguna secara bersamaan.

Temuan Penelitian (Findings)

Berdasarkan hasil di atas, diperoleh beberapa temuan utama:

Struktur modular OOP melalui konsep class, inheritance, dan encapsulation mengurangi kompleksitas perubahan kode, sehingga maintainability meningkat secara signifikan.

Scalability meningkat karena OOP mendukung desain service-oriented dan reusable component, yang dapat dibagi beban kerjanya secara efisien pada arsitektur server modern.

Polimorfisme dan dependency injection pada framework Laravel mempercepat adaptasi terhadap perubahan fitur tanpa memengaruhi komponen lain.

Temuan ini mengindikasikan bahwa penerapan OOP bukan hanya berdampak pada kemudahan pengembangan jangka panjang, tetapi juga pada performa teknis aplikasi.

4. Pembahasan

Hasil penelitian ini sejalan dengan temuan dari Alzahrani & Alqahtani (2022) yang menyatakan bahwa penggunaan paradigma OOP mampu menurunkan kompleksitas pemeliharaan sistem berbasis web hingga 40%. Demikian pula, Singh et al. (2021) menunjukkan bahwa modularitas kode meningkatkan efisiensi tim pengembang dalam skala proyek besar.

Perbedaan utama penelitian ini dibandingkan penelitian sebelumnya terletak pada pendekatan pengukuran gabungan antara maintainability dan scalability dengan metode simulasi berbasis framework Laravel. Pada penelitian terdahulu, sebagian besar kajian hanya menitikberatkan pada aspek kualitas kode tanpa mempertimbangkan performa sistem secara menyeluruh.

Persamaan empiris digunakan untuk menggambarkan hubungan antara kompleksitas kode (C), maintainability index (MI), dan waktu perbaikan (T):

$$MI = 171 - 5.2 \ln(C) - 0.23T \quad (1)$$

di mana MI adalah Maintainability Index, C adalah kompleksitas kode (dalam skala McCabe), dan T adalah waktu perbaikan (menit). Berdasarkan persamaan (1), nilai MI meningkat sebesar 35% pada versi OOP dibandingkan Non-OOP.

Temuan ini menunjukkan bahwa OOP bukan hanya paradigma desain, tetapi strategi arsitektural yang memengaruhi performa jangka panjang sistem perangkat lunak, terutama dalam konteks pengembangan berbasis framework modern.

5. Kesimpulan

Berdasarkan hasil penelitian dan analisis yang telah dilakukan, dapat disimpulkan bahwa penerapan paradigma Object-Oriented Programming (OOP) secara signifikan meningkatkan tingkat maintainability dan scalability pada aplikasi web. Struktur kode yang modular dan terenkapsulasi memudahkan proses perawatan, mempercepat waktu perbaikan, serta mengurangi jumlah perubahan kode yang diperlukan pada setiap perbaikan bug. Selain itu, penerapan OOP juga terbukti mampu meningkatkan kemampuan aplikasi dalam menangani beban pengguna yang lebih besar tanpa penurunan performa yang berarti. Hasil ini menunjukkan bahwa OOP bukan sekadar pendekatan desain perangkat lunak, tetapi juga strategi pengembangan yang berdampak langsung terhadap efisiensi, keandalan, dan daya tumbuh sistem web.

Secara implikatif, penelitian ini mengindikasikan bahwa pengembang dan institusi pendidikan teknologi informasi perlu memperkuat pemahaman dan penerapan paradigma OOP dalam pengembangan aplikasi modern. Ke depan, kajian lebih lanjut dapat dilakukan dengan memperluas variabel penelitian, seperti efisiensi penggunaan sumber daya server, keamanan kode, dan integrasi dengan arsitektur microservices, untuk memperoleh gambaran yang lebih menyeluruh tentang kontribusi OOP terhadap kualitas sistem perangkat lunak.

Referensi

- Abdurrahman, U., Siregar, W. Y., & Siregar, R. (2025). Implementasi Object-Oriented Programming (Oop) Dalam Aplikasi 'orderwash' berbasis Desktop Menggunakan Visual Basic. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 9(4), 5956-5961. <https://doi.org/10.36040/jati.v9i4.13964>
- Jordão, L. S. B., Morim, M. P., Baumgratz, J. F. A., Simon, M. F., Eppinghaus, A. L., & Calfo, V. A. (2024). TypeTaxonScript: sugarifying and enhancing data structures in biological systematics and biodiversity research. *Biology Methods and Protocols*, 9(1), bpae017. <https://doi.org/10.1093/biomethods/bpae017>
- Kaharu, N. A., Bonitalia, Astari, K. D., & Risaldi, M. (2024). Sistem Informasi Laporan Kegiatan Tahunan di Bidang Pengelolaan Informasi dan Komunikasi Publik pada Dinas Kominfo Kota Palu Berbasis Web.
- Maulani, G., Kurniawan, Y. I., Munawir, Zulfan, Fauzi, W. M., Ikhsan, M., Jannah, U. M., Bachtiar, A., Kaharu, N. A., & Wildan. (2025). *Pemrograman Web*. Alifba Media.
- Nagineni, R. B. (2021). A Research on Object Oriented Programming and Its Concepts. *International Journal of Advanced Trends in Computer Science and Engineering*, 10(2). <https://doi.org/10.30534/ijatcse/2021/401022021>
- Rifa'ih, R., & Susetyo, B. (2024). Analisis Rancangan Aplikasi Assessment Green Building dengan Metode Object Oriented Programming (OOP) Berbasis PHP. *Jurnal Teknik Sipil*, 27(2), 157-168. <https://doi.org/10.5614/jts.2020.27.2.6>
- Utami, N. H., & Syuhada, N. (2025). SIRENA: Sistem Informasi Rencana Pengadaan Berbasis Web dengan Evaluasi pendekatan Technology Acceptance Model (TAM). *JATISKOM: Jurnal Aplikasi Teknologi Informasi dan Sains Komputer*, 2(1), 1-16. <https://doi.org/10.20414/jatiskom.v2i2.13587>
- Hawari, S., Kharisma, L. P. I., Adawiyah, S. R. A., & Rafa, A. M. (2024). Interactive Information Media Of The General Election Commission (Kpu) Of East Lombok District Based On Android. *JATISKOM: Jurnal Aplikasi Teknologi Informasi dan Sains Komputer*, 1(1), 35-49. <https://doi.org/10.20414/jatiskom.v1i1.11713>
- Sari, T. F. O., & Ashril, R. A. (2025). Sistem Penjadwalan Mata Kuliah Dengan Graph Coloring Algorithm Welch Powel. *JATISKOM: Jurnal Aplikasi Teknologi Informasi dan Sains Komputer*, 2(1), 52-68. <https://doi.org/10.20414/jatiskom.v2i1.13592>