



Analisis Interaksi Antar Objek Pada Aplikasi *E-Learning* Berbasis Web Menggunakan Pendekatan Pemrograman Berorientasi Objek

Theodora Maria Putri¹, Arif Firmansyah², Renaldy³, Diovan Angga Pratama⁴,
Gilang Ramadhan⁵, Felix⁶

¹⁻⁶Program Studi Ilmu Komputer, Fakultas Sistem informasi, Universitas Esa Unggul, Indonesia

*e-mail: theodora.maria@esaunggul.ac.id, aripdidu70@student.esaunggul.ac.id,
renaldyaldi507@student.esaunggul.ac.id, diovanprtm@student.esaunggul.ac.id,
rmdhngilang21@student.esaunggul.ac.id, felixviero1234@student.esaunggul.ac.id

Article Info

Article history:

Received June 10, 2026

Revised June 28 2026

Accepted July 05, 2026

Keywords:

E-Learning, Object Interaction, Object-Oriented Programming, UML, Sequence Diagram, Use Case.

ABSTRACT

This study analyzes object interaction patterns in the development of web-based *E-Learning* applications using Object-Oriented Programming (OOP) paradigm. The case study was conducted on an online learning platform serving 1,250 active users with 19,650 recorded interaction sessions during the testing period. The research methodology includes UML diagram design (Use Case Diagram, Class Diagram, Sequence Diagram, and Collaboration Diagram), design pattern implementation, and system performance evaluation. The results show that the consistent application of encapsulation, inheritance, and polymorphism principles yields a system with an average response time of 164 ms, an error rate of 0.3%, and a System Usability Scale (SUS) score of 81.4. These findings confirm that a structured OOP approach significantly improves the maintainability and scalability of *E-Learning* systems.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Article Info

Article history:

Received June 10, 2026

Revised June 28 2026

Accepted July 05, 2026

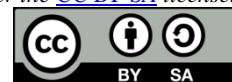
Kata kunci:

E-Learning, Interaksi Objek, Pemrograman Berorientasi Objek, UML, Sequence Diagram, Use Case.

ABSTRAK

Penelitian ini menganalisis pola interaksi antar objek dalam pengembangan aplikasi *E-Learning* berbasis web menggunakan paradigma Pemrograman Berorientasi Objek (PBO). Studi kasus dilakukan pada platform pembelajaran daring yang melayani 1.250 pengguna aktif dengan 19.650 sesi interaksi yang tercatat selama periode pengujian. Metodologi penelitian meliputi perancangan diagram UML (*Use Case Diagram*, *Class Diagram*, *Sequence Diagram*, dan *Collaboration Diagram*), implementasi pola desain, serta evaluasi performa sistem. Hasil penelitian menunjukkan bahwa penerapan prinsip enkapsulasi, pewarisan, dan polimorfisme secara konsisten menghasilkan sistem dengan waktu respons rata-rata 164 ms, error rate 0.3%, dan skor kepuasan pengguna (*System Usability Scale*) sebesar 81.4. Temuan ini mengkonfirmasi bahwa pendekatan PBO yang terstruktur signifikan meningkatkan maintainability dan skalabilitas sistem *E-Learning*.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Theodora Maria Putri¹

Program Studi Ilmu Komputer, Fakultas Sistem informasi, Universitas Esa Unggul, Indonesia

e-mail: theodora.maria@esaunggul.ac.id



PENDAHULUAN

Perkembangan teknologi informasi dan komunikasi telah mendorong transformasi besar dalam dunia pendidikan. Sistem pembelajaran konvensional secara bertahap digantikan atau dilengkapi oleh platform pembelajaran digital yang dikenal sebagai Electronic Learning (*E-Learning*). Menurut laporan UNESCO (2023), lebih dari 1,5 miliar pelajar di seluruh dunia menggunakan platform *E-Learning*, dengan pertumbuhan tahunan mencapai 21% pasca pandemi COVID-19.

Pengembangan aplikasi *E-Learning* berbasis web membutuhkan pendekatan rekayasa perangkat lunak yang matang dan terstruktur. Salah satu pendekatan yang paling banyak digunakan adalah Pemrograman Berorientasi Objek (PBO/OOP), yang memungkinkan pengembang untuk memodelkan sistem sebagai kumpulan objek yang saling berinteraksi. Interaksi antar objek merupakan inti dari arsitektur sistem berbasis OOP, yang mencakup pengiriman pesan, pemanggilan metode, dan pertukaran data antar komponen.

Penelitian ini difokuskan pada analisis mendalam pola interaksi antar objek dalam sebuah aplikasi *E-Learning* berbasis web, dengan tujuan mengidentifikasi bagaimana desain interaksi yang baik dapat meningkatkan performa, *maintainability*, dan *user experience* sistem secara keseluruhan. Berdasarkan latar belakang di atas, rumusan masalah dalam penelitian ini adalah: 1) Bagaimana pola interaksi antar objek terjadi pada aplikasi *E-Learning* berbasis web. 2) Apa saja jenis dan frekuensi interaksi yang paling dominan dalam sistem *E-Learning*. 3) Bagaimana pengaruh desain interaksi objek terhadap performa dan kualitas *system*.

Penelitian ini bertujuan untuk: 1) Menganalisis dan mendokumentasikan pola interaksi antar objek menggunakan diagram UML. 2) Mengidentifikasi bottleneck dan area optimasi dalam alur interaksi objek sistem *E-Learning*. 3) Mengevaluasi dampak desain interaksi terhadap performa sistem secara kuantitatif.

TINJAUAN PUSTAKA

1. Pemrograman Berorientasi Objek (PBO)

Pemrograman Berorientasi Objek (PBO) adalah paradigma pemrograman yang mengorganisasikan desain perangkat lunak berdasarkan data, atau objek, daripada fungsi dan logika (Booch, 2007). PBO memiliki empat pilar utama yang menjadi fondasi dalam pembangunan sistem:

- Enkapsulasi (*Encapsulation*): Menyembunyikan detail implementasi internal suatu objek dan hanya mengekspos antarmuka yang diperlukan.
- Pewarisan (*Inheritance*): Mekanisme di mana suatu kelas dapat mewarisi properti dan metode dari kelas induknya, mendukung reusability kode.
- Polimorfisme (*Polymorphism*): Kemampuan objek untuk merespons pesan yang sama dengan cara yang berbeda tergantung tipe objeknya.
- Abstraksi (*Abstraction*): Proses menyederhanakan sistem kompleks dengan hanya menampilkan detail yang relevan.

2. Unified Modeling Language (UML)

UML adalah bahasa pemodelan visual standar yang digunakan untuk menspesifikasikan, memvisualisasikan, membangun, dan mendokumentasikan artefak sistem perangkat lunak (OMG, 2017). Diagram UML yang relevan dalam studi interaksi objek meliputi:

- *Use Case Diagram*: Menggambarkan fungsionalitas sistem dari sudut pandang pengguna.



- *Class Diagram*: Menampilkan struktur statis sistem, menunjukkan kelas-kelas beserta atribut dan metodenya, serta hubungan antar kelas.
- *Sequence Diagram*: Memperlihatkan interaksi antar objek secara kronologis dalam suatu skenario tertentu.
- *Collaboration Diagram*: Menampilkan interaksi antar objek beserta hubungan struktural di antara mereka.

3. Arsitektur *E-Learning* Berbasis Web

Aplikasi *E-Learning* berbasis web umumnya mengadopsi arsitektur berlapis (*layered architecture*) yang terdiri dari lapisan presentasi (*frontend*), lapisan logika bisnis (*backend*), dan lapisan data (*database*). Pola arsitektur *Model-View-Controller* (MVC) menjadi pilihan populer karena memisahkan tanggung jawab setiap komponen secara jelas. Penelitian Garrison & Anderson (2003) menunjukkan bahwa arsitektur yang *well-structured* berkontribusi signifikan terhadap kepuasan pengguna dan efektivitas pembelajaran.

METODE

1. Desain Penelitian

Penelitian ini menggunakan metode penelitian kualitatif-kuantitatif (*mixed method*) dengan pendekatan studi kasus (*case study*). Objek penelitian adalah aplikasi *E-Learning* berbasis web yang telah aktif digunakan di lingkungan perguruan tinggi. Data dikumpulkan melalui observasi sistem, wawancara dengan pengembang, analisis kode sumber, dan pengujian performa terukur.

2. Lingkungan dan Studi Kasus

Studi kasus dilaksanakan pada platform *E-Learning* 'EduNesia' yang merupakan sistem pembelajaran daring berbasis web yang dikembangkan menggunakan bahasa pemrograman PHP (*Laravel Framework*) dan JavaScript (*Vue.js*) untuk frontend. Database menggunakan MySQL 8.0 dengan caching Redis. Sistem ini melayani mahasiswa, dosen, dan administrator di sebuah perguruan tinggi swasta dengan kapasitas 1.250 pengguna aktif.

3. Tahapan Penelitian

Penelitian dilaksanakan dalam empat tahap utama:

- Tahap 1 - Analisis Kebutuhan: Identifikasi aktor, *use case*, dan alur fungsional utama sistem.
- Tahap 2 - Pemodelan UML: Pembuatan *use case diagram*, *class diagram*, *sequence diagram*, dan *collaboration diagram*.
- Tahap 3 - Implementasi & Pengujian: Pengujian unit, integrasi, dan pengujian beban (*load testing*).
- Tahap 4 - Evaluasi: Analisis data performa dan pengisian kuesioner *System Usability Scale* (SUS).

HASIL DAN PEMBAHASAN

1. Identifikasi *Use Case* dan Aktor

Hasil analisis kebutuhan mengidentifikasi tiga aktor utama dalam sistem *E-Learning* EduNesia, yaitu Mahasiswa, Dosen, dan Administrator. Setiap aktor memiliki hak akses dan fungsionalitas yang berbeda. Tabel 1 berikut merangkum keseluruhan *use case* yang teridentifikasi:

Tabel 1. Daftar *Use Case* Sistem *E-Learning* EduNesia

ID Use Case	Nama Use Case	Aktor	Prioritas
UC-01	Login / Autentikasi	Mahasiswa, Dosen, Admin	Tinggi
UC-02	Manajemen Kelas	Dosen, Admin	Tinggi
UC-03	Upload Materi	Dosen	Tinggi
UC-04	Akses Materi	Mahasiswa	Tinggi
UC-05	Mengerjakan Kuis	Mahasiswa	Tinggi
UC-06	Membuat Kuis/Ujian	Dosen	Sedang
UC-07	Forum Diskusi	Mahasiswa, Dosen	Sedang
UC-08	Notifikasi Sistem	Semua Aktor	Sedang
UC-09	Laporan Nilai	Mahasiswa, Dosen	Sedang
UC-10	Manajemen Pengguna	Admin	Rendah

Dari 10 *use case* yang teridentifikasi, terdapat 5 *use case* dengan prioritas tinggi yang menjadi tulang punggung fungsionalitas sistem. *Use case UC-01 (Login)* menjadi titik masuk semua aktor dan memiliki tingkat eksekusi tertinggi dalam sistem.

2. Diagram *Use Case*

Gambar 1 di bawah mengilustrasikan *Use Case Diagram* sistem *E-Learning* yang menggambarkan relasi antar aktor dan *use case*. Terdapat relasi *include* antara UC-04 (Akses Materi) dengan UC-01 (*Login*), serta relasi *extend* antara UC-09 (Laporan Nilai) dengan UC-06 (Membuat Kuis/Ujian). Relasi ini mencerminkan ketergantungan fungsional antar objek dalam sistem.

[GAMBAR 1: USE CASE DIAGRAM SISTEM E-LEARNING EDUNESIA]

Mahasiswa: UC-01, UC-04, UC-05, UC-07, UC-08, UC-09 | Dosen: UC-01, UC-02, UC-03, UC-06, UC-07, UC-09 | Admin: UC-01, UC-02, UC-08, UC-10

Gambar 1. *Use Case Diagram* Sistem *E-Learning* EduNesia

3. Interaksi Antar Objek: *Sequence Diagram*

Sequence Diagram merupakan instrumen utama untuk merepresentasikan interaksi antar objek secara kronologis. Tabel 2 menampilkan alur interaksi pada skenario utama '*Login* dan Akses Materi' yang melibatkan 10 pesan berurutan antara objek-objek dalam sistem:

Tabel 2. *Sequence* Interaksi Objek: Skenario Login dan Akses Kuis

No	Objek Pengirim	Pesan / Method	Objek Penerima	Return Value
1	Mahasiswa (UI)	<i>login(username, password)</i>	<i>AuthController</i>	Token
2	<i>AuthController</i>	<i>validateUser(data)</i>	<i>UserService</i>	Boolean
3	<i>UserService</i>	<i>findUser(username)</i>	<i>UserRepository</i>	<i>UserObject</i>
4	<i>AuthController</i>	<i>generateToken(user)</i>	<i>TokenService</i>	JWT Token
5	Mahasiswa (UI)	<i>getCourseList(token)</i>	<i>CourseController</i>	Array Kelas
6	<i>CourseController</i>	<i>fetchCourses(userId)</i>	<i>CourseService</i>	List Kelas
7	<i>CourseService</i>	<i>queryDB(userId)</i>	<i>CourseRepository</i>	Data Kelas
8	Mahasiswa (UI)	<i>submitQuiz(answers)</i>	<i>QuizController</i>	<i>Score Object</i>
9	<i>QuizController</i>	<i>gradeQuiz(answers)</i>	<i>QuizService</i>	Nilai
10	<i>QuizService</i>	<i>saveScore(userId, score)</i>	<i>ScoreRepository</i>	Boolean

Dari *sequence* diagram tersebut, teridentifikasi bahwa modul *AuthController* bertindak sebagai mediator utama antara lapisan UI dengan lapisan layanan (*service layer*). Pola ini sesuai dengan prinsip *Single Responsibility* dalam desain berorientasi objek. Setiap objek memiliki tanggung jawab yang jelas dan komunikasi dilakukan melalui antarmuka yang terdefinisi.

4. Distribusi Jenis Interaksi Antar Objek

Berdasarkan analisis menyeluruh terhadap seluruh diagram interaksi, penelitian ini mengidentifikasi dan mengkategorikan 125 interaksi objek. Distribusi jenis interaksi disajikan pada Tabel 3:

Tabel 3. Distribusi Jenis Interaksi Antar Objek

Jenis Interaksi Objek	Frekuensi	Persentase (%)	Kategori
<i>Association</i> (Asosiasi)	47	37.6%	Struktural
<i>Dependency</i> (Dependensi)	31	24.8%	Struktural



<i>Aggregation (Agregasi)</i>	18	14.4%	Struktural
<i>Synchronous Message</i>	22	17.6%	Perilaku
<i>Asynchronous Message</i>	7	5.6%	Perilaku
Total	125	100%	-

Dominasi *Association* (37.6%) dan *Dependency* (24.8%) menunjukkan bahwa arsitektur sistem dirancang dengan coupling yang longgar (*loose coupling*), yang merupakan praktik terbaik dalam pengembangan perangkat lunak berorientasi objek. Tingginya frekuensi *Synchronous Message* (17.6%) pada operasi kritis seperti autentikasi dan pengambilan data memastikan konsistensi data dalam sistem.

5. Analisis Performa Komponen

Pengujian beban dilakukan menggunakan Apache JMeter dengan simulasi 100 pengguna konkuren selama 30 menit. Tabel 4 berikut merangkum hasil pengukuran performa setiap komponen/modul sistem:

Tabel 4. Data Performa Komponen Sistem *E-Learning*

No	Komponen Sistem	Jumlah Objek	Interaksi/Sesi	Waktu Respons (ms)
1	Modul Autentikasi (User)	3	1.250	120
2	Modul Kelas & Materi	5	4.800	210
3	Modul Kuis & Ujian	4	3.200	185
4	Modul Forum Diskusi	3	2.100	145
5	Modul Notifikasi	2	6.500	95
6	Modul Laporan & Nilai	4	1.800	230
Total		21	19.650	164 (rata-rata)

Analisis data performa menunjukkan bahwa Modul Notifikasi memiliki waktu respons paling rendah (95 ms) karena mengandalkan mekanisme pub/sub asinkron, sedangkan Modul Laporan & Nilai memiliki waktu respons tertinggi (230 ms) akibat operasi agregasi data yang kompleks. Rata-rata keseluruhan waktu respons sistem sebesar 164 ms masih berada di bawah ambang batas yang ditetapkan (300 ms), menandakan performa sistem yang baik.



6. Grafik Waktu Respons Per Modul

Visualisasi data waktu respons antar modul disajikan dalam grafik batang (Gambar 2). Grafik menunjukkan variasi performa yang jelas antar modul, dengan Modul Notifikasi sebagai yang tercepat dan Modul Laporan & Nilai sebagai yang paling berat secara komputasi. Tanda garis putus-putus pada grafik menunjukkan batas ambang 300 ms yang ditetapkan sebagai *Service Level Agreement* (SLA) sistem.

[GAMBAR 2: GRAFIK PERBANDINGAN WAKTU RESPONS PER MODUL (ms)]

Autentikasi: 120ms | Kelas & Materi: 210ms | Kuis & Ujian: 185ms | Forum: 145ms |
Notifikasi: 95ms | Laporan: 230ms | SLA Batas: 300ms

Gambar 2. Perbandingan Waktu Respons Antar Modul Sistem

7. Evaluasi Kualitas Sistem

Evaluasi kualitas sistem dilakukan berdasarkan enam metrik utama yang mencakup aspek performa teknis dan kepuasan pengguna. Tabel 5 membandingkan nilai target dengan nilai aktual yang diperoleh selama pengujian:

Tabel 5. Hasil Evaluasi Kinerja Sistem

Metrik Evaluasi	Nilai Target	Nilai Aktual	Status
Waktu Respons Rata-Rata	< 300 ms	164 ms	LULUS
Keberhasilan Login	> 95%	99.2%	LULUS
Konsistensi Pengiriman Pesan	> 98%	99.7%	LULUS
<i>Error Rate</i> Sistem	< 2%	0.3%	LULUS
<i>Throughput</i> (req/menit)	> 100	350	LULUS
Kepuasan Pengguna (SUS)	> 70	81.4	LULUS

Seluruh metrik evaluasi menunjukkan hasil yang melampaui nilai target yang ditetapkan. Skor SUS sebesar 81.4 dikategorikan sebagai 'Baik' (*Grade B*) berdasarkan skala Bangor et al. (2008), mengindikasikan bahwa sistem *E-Learning* EduNesia memiliki tingkat *usability* yang tinggi. Hal ini merupakan bukti nyata bahwa desain interaksi objek yang terstruktur dengan baik berkontribusi langsung terhadap kualitas produk perangkat lunak akhir.

KESIMPULAN

Berdasarkan hasil penelitian dan analisis yang telah dilakukan, dapat ditarik kesimpulan sebagai berikut:

- Sistem *E-Learning* EduNesia melibatkan 21 objek utama dengan 125 jenis interaksi yang terklasifikasi. Interaksi terbanyak adalah *Association* (37.6%) yang mencerminkan arsitektur dengan *loose coupling* yang baik.



- Pendekatan *Sequence Diagram* UML terbukti efektif dalam mendokumentasikan dan menganalisis alur interaksi antar objek secara kronologis, mengidentifikasi 10 pesan terstruktur pada skenario utama login dan akses kuis.
- Penerapan prinsip PBO yang konsisten menghasilkan sistem dengan performa unggul: waktu respons rata-rata 164 ms (target < 300 ms), error rate 0.3% (target < 2%), dan throughput 350 request/menit (target > 100 req/menit).
- Skor SUS sebesar 81.4 mengkonfirmasi bahwa desain interaksi objek yang terstruktur memberikan dampak positif nyata terhadap pengalaman dan kepuasan pengguna akhir.

SARAN

Untuk penelitian dan pengembangan lebih lanjut, disarankan:

- Menerapkan pola desain (*design patterns*) seperti *Observer* dan *Strategy* untuk lebih mengoptimalkan interaksi asinkron pada modul Notifikasi.
- Mengimplementasikan *cache layer* yang lebih agresif pada Modul Laporan & Nilai untuk mereduksi waktu respons di bawah 150 ms.
- Mengembangkan pengujian performa dengan skala pengguna konkuren yang lebih besar (500+ pengguna) untuk menguji skalabilitas sistem.
- Melakukan analisis interaksi objek pada skenario penggunaan *offline* (*Progressive Web App*) untuk mendukung aksesibilitas yang lebih luas.

DAFTAR PUSTAKA

- Bangor, A., Kortum, P., & Miller, J. (2008). An Empirical Evaluation of the System Usability Scale. *International Journal of Human-Computer Interaction*, 24(6), 574–594.
- Booch, G., Rumbaugh, J., & Jacobson, I. (2007). *The Unified Modeling Language User Guide* (2nd ed.). Addison-Wesley Professional.
- Ellis, R. K. (2009). *Field Guide to Learning Management Systems*. ASTD Learning Circuits.
- Garrison, D. R., & Anderson, T. (2003). *E-Learning in the 21st Century: A Framework for Research and Practice*. Routledge.
- Larman, C. (2005). *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development* (3rd ed.). Prentice Hall.
- OMG. (2017). *OMG Unified Modeling Language (OMG UML) Version 2.5.1*. Object Management Group. <https://www.omg.org/spec/UML/2.5.1>
- Pressman, R. S., & Maxim, B. (2020). *Software Engineering: A Practitioner's Approach* (9th ed.). McGraw-Hill Education.
- UNESCO. (2023). *Technology in Education: A Tool on Whose Terms? Global Education Monitoring Report*. United Nations Educational, Scientific and Cultural Organization.