

Design and implementation of a reusable component-based architecture for learning media

Muhammad Amin Rais¹, Mutiara Akbar Nasution²

^{1,2}Program Studi Magister Teknik Informatika, Universitas Sumatera Utara, Indonesia

Email: gamirai@yahoo.com; mutiaraakbamst03@gmail.com

ABSTRAK

Pengembangan media pembelajaran digital sering kali memerlukan implementasi ulang fitur-fitur serupa seperti sistem kuis, pemutar video, navigasi modul, dan pelacak progres pada aplikasi yang berbeda. Penelitian ini mengusulkan reusable prefab-based architecture pada platform Unity dengan menerapkan prinsip Component-Based Software Engineering (CBSE). Arsitektur yang dikembangkan terdiri dari tujuh komponen inti reusable yaitu Lobby, Module Layout, Material, Video Player, Quiz, Progress Tracker, dan Main Setting Component. Penelitian menggunakan pendekatan Research and Development (R&D) dengan model ADDIE yang dimodifikasi dan diimplementasikan menggunakan Unity 2022 LTS serta bahasa pemrograman C#. Evaluasi dilakukan melalui implementasi component library pada lima aplikasi media pembelajaran yang berbeda. Hasil penelitian menunjukkan bahwa sebagian besar komponen memiliki reuse ratio yang tinggi, maintainability index di atas 70, serta coupling dependencies yang rendah sehingga mendukung modularitas dan reusability sistem. Penelitian ini menunjukkan bahwa reusable prefab-based architecture dapat menjadi solusi efektif dalam pengembangan media pembelajaran digital yang modular, fleksibel, dan scalable.

Kata Kunci: rekayasa perangkat lunak berbasis komponen; Unity; prefab yang dapat digunakan kembali; media pembelajaran; penggunaan kembali perangkat lunak

ABSTRACT

The development of digital learning media often requires repeated implementation of similar features such as quiz systems, video players, module navigation, and progress trackers in different applications. This study proposes a reusable prefab-based architecture on the Unity platform by applying the principles of Component-Based Software Engineering (CBSE). The developed architecture consists of seven core reusable components: Lobby, Module Layout, Material, Video Player, Quiz, Progress Tracker, and Main Setting Component. The study uses a Research and Development (R&D) approach with a modified ADDIE model and is implemented using Unity 2022 LTS and the C# programming language. Evaluation is carried out through the implementation of component libraries in five different learning media applications. The results show that most components have a high reuse ratio, a maintainability index above 70, and low coupling dependencies, thus supporting system modularity and reusability. This study shows that a reusable prefab-based architecture can be an effective solution in the development of modular, flexible, and scalable digital learning media.

Keywords: component-based software engineering; unity; reusable prefab; learning media; software reuse

Corresponding Author:

Muhammad Amin Rais,
Universitas Sumatera Utara,
Jalan Dr. T. Mansur No.9, Padang Bulan, Kec. Medan Baru, Kota Medan,
Sumatera Utara 20222, Indonesia
Email: gamirai@yahoo.com



1. PENDAHULUAN

Pesatnya perkembangan teknologi Informasi saat ini telah memicu transformasi yang besar, khususnya dalam lanskap pendidikan. Saat ini, media pembelajaran digital bukan lagi sekadar pelengkap,

melainkan kebutuhan penting dalam ekosistem belajar-mengajar modern, baik di sekolah maupun perguruan tinggi (Arkorful & Abaidoo, 2014). Karakteristik pengembangan media pembelajaran harus sejalan dengan kegunaan yang praktis, membimbing peserta didik untuk memecahkan dan menjawab masalah, dan variasi pembelajaran (Dwijayani, 2019).

Berbagai platform interaktif telah hadir, mulai dari solusi berbasis web hingga aplikasi desktop canggih yang memanfaatkan *game engine* seperti Unity (Niemela & Hyyro, 2019). Namun, permasalahan mendasar yang dihadapi para pengembang media pembelajaran adalah setiap aplikasi dibuat dari awal (from scratch) untuk setiap mata pelajaran atau mata kuliah yang berbeda.

Inefisiensi ini berdampak langsung pada pemborosan waktu, biaya, dan tenaga yang signifikan (Salas, 2020). Terlebih lagi, banyak pengembang media pembelajaran sebenarnya adalah praktisi pendidikan (guru atau dosen) yang memiliki keterbatasan sumber daya. Fitur-fitur dengan fungsionalitas serupa seperti sistem kuis, pemutar video, atau pelacak progres belajar terus-menerus dibangun ulang secara terpisah di setiap proyek karena tidak adanya mekanisme penggunaan kembali (*reuse*) yang sistematis (Frakes & Kang, 2005).

Dalam dunia rekayasa perangkat lunak, konsep *Software Reuse* telah lama diakui sebagai jawaban atas problematika ini. Krueger, (1992) menegaskan bahwa esensi dari *software reuse* adalah membangun sistem dari komponen-komponen yang sudah ada, alih-alih memulainya dari dasar. Paradigma ini diperkuat oleh *Component-Based Software Engineering* (CBSE) yang mengedepankan pembangunan sistem melalui komponen yang bersifat *loosely coupled*, dapat digunakan kembali, dan memiliki antarmuka yang terdefinisi dengan jelas (Chiao et al., 2017). Sebagai salah satu *game engine* terpopuler, Unity memiliki fitur 'Prefab' yang secara teknis sangat selaras dengan prinsip CBSE. Prefab memungkinkan setiap objek dikonfigurasi sebagai unit mandiri yang bisa diinstansiasi berulang kali (Unity, 2024), menjadikan Unity platform yang sangat ideal untuk arsitektur berbasis komponen.

Meskipun konsep reusable learning object dan software reuse telah banyak dibahas pada penelitian sebelumnya, sebagian besar penelitian masih berfokus pada reuse konten pembelajaran dan belum secara spesifik membahas reusable prefab-based architecture pada platform Unity untuk pengembangan media pembelajaran modular. Selain itu, evaluasi teknis terhadap modularitas dan maintainability komponen reusable pada media pembelajaran berbasis Unity masih relatif terbatas.

Penelitian ini bertujuan untuk merancang serta mengimplementasikan arsitektur komponen yang reusable khusus untuk media pembelajaran di Unity. Kami merumuskan tujuh komponen inti sebagai unit fungsional utama: *Quiz*, *Video Player*, *Progress Tracker*, *Material*, *Module Layout*, *Lobby*, dan *Main Setting Component*. Melalui arsitektur ini, pengembang tidak perlu lagi membangun sistem dari nol, melainkan cukup 'merakit' aplikasi baru dari komponen yang telah teruji. Secara spesifik, penelitian ini akan mengeksplorasi tiga hal utama: (1) Bagaimana merancang struktur komponen yang mendukung fleksibilitas di Unity?, (2) Apa saja unit fungsional yang paling krusial untuk dijadikan komponen reusable?, (3) Sejauh mana efektivitas arsitektur ini berpotensi meningkatkan efisiensi pengembangan jika ditinjau dari sisi modularitas dan kemudahan penggunaan?

Diharapkan, hasil dari perancangan dan evaluasi terhadap ketujuh komponen ini dapat memberikan solusi yang skalabel bagi para pendidik dan pengembang dalam menciptakan media pembelajaran yang berkualitas secara lebih cepat.

2. KAJIAN PUSTAKA

A. *Software Reuse dan Component-Based Software Engineering*

Software reuse adalah proses penggunaan kembali aset perangkat lunak yang telah ada termasuk kode, desain, dan komponen dalam membangun sistem baru. Krueger, (1992) membagi pendekatan reuse ke dalam tiga tingkatan: reuse kode, desain, dan arsitektur. Component-Based Software Engineering (CBSE) adalah paradigma yang menekankan perancangan sistem dari komponen loosely coupled yang memiliki antarmuka terdefinisi, berperilaku sebagai black-box yang dapat digantikan tanpa mempengaruhi komponen lain. Pengembangan CBSE dapat menentukan bagaimana gaya arsitektur dari Model yang dikembangkan (Wei et al., 2009).

Szyperski et al., (2002) mendefinisikan komponen perangkat lunak sebagai unit komposisi dengan antarmuka yang dispesifikasikan secara kontraktual dan dependensi konteks yang eksplisit, sehingga dapat di-deploy secara independen. Kim et al., (2017) membuktikan bahwa arsitektur berbasis komponen reusable pada sistem IoT dapat mengurangi biaya pengembangan secara signifikan, dengan error rate evaluasi energi hanya 6.83%. Frakes & Kang (2005) menegaskan bahwa meski manfaat software reuse sudah diakui, tantangan utama adalah kurangnya standar arsitektur dan dukungan tooling.

B. *Media Pembelajaran Digital*

Media pembelajaran adalah segala sesuatu yang dapat digunakan untuk menyampaikan pesan dalam proses belajar-mengajar sehingga merangsang perhatian dan minat belajar (Jauza & Albina, 2025). Arkorful & Abaidoo, (2014) menunjukkan bahwa media pembelajaran digital memberikan keunggulan dalam fleksibilitas,

aksesibilitas, dan akomodasi gaya belajar individual, namun tantangan utamanya adalah biaya dan waktu pengembangan yang tinggi.

Pengembangan media pembelajaran umumnya memuat beberapa komponen yang harus ada seperti Materi, Quiz, Menu Utama, dan Panduan, desain yang interaktif dan di deploy pada perangkat Mobile seperti Android. (Farhani et al., 2026; Purnama & Putra, 2026). Media berbasis Mobile telah menjadi solusi inovasi untuk menghadapi tantangan pembelajaran modern. Media interaktif mengkombinasikan beragam model yang relevan untuk menghasilkan penyampaian yang menarik dan memiliki fitur pendukung yang tepat. untuk menghasilkan Pengembangan Media yang baik harus melalui validasi dan efektivitas (Hidayah et al., 2025).

Konsep Reusable Learning Objects (RLO) telah diusulkan sebagai solusi, namun sebagian besar penelitian RLO berfokus pada reuse konten, bukan pada reuse komponen arsitektur software yang menjadi fokus penelitian ini. (Salas, 2020) menunjukkan bahwa pendekatan agile dalam pengembangan RLO dapat meningkatkan fleksibilitas dan kecepatan pembuatan materi pembelajaran.

C. Unity sebagai Platform Pengembangan Interaktif

Unity adalah game engine lintas platform yang mendukung pengembangan untuk Windows, Android, iOS, dan WebGL dari satu basis kode tunggal (Hasaniyah et al., 2023). Sistem Prefab dalam Unity merupakan mekanisme bawaan untuk membuat, mengkonfigurasi, dan menyimpan GameObject sebagai aset reusable yang dapat diinstansiasi berkali-kali dalam berbagai scene (Juárez-González et al., 2025; Unity Technologies, n.d.). Sistem ini secara alamiah mendukung prinsip CBSE, komponen dapat diperluas tanpa mengubah implementasi dasarnya (Open/Closed Principle).

Scriptable Objects melengkapi sistem Prefab sebagai mekanisme penyimpanan data yang terpisah dari logika aplikasi, memungkinkan konfigurasi komponen tanpa mengubah kode sumber, selaras dengan pemisahan concern (Separation of Concerns) dalam CBSE.

D. Metrik Evaluasi Kualitas Perangkat Lunak

Evaluasi kualitas komponen perangkat lunak dalam penelitian ini menggunakan empat metrik utama. Pertama, Reuse Ratio mengukur proporsi komponen yang berhasil digunakan kembali terhadap total komponen dalam sistem baru. Kedua, Coupling dependencies digunakan untuk menilai ketergantungan antar komponen. Ketiga, Maintainability Index (MI) adalah metrik gabungan yang menghitung kemudahan pemeliharaan kode berdasarkan Halstead Volume, Cyclomatic Complexity, dan Lines of Code. Keempat, Cyclomatic Complexity yang dikembangkan McCabe mengukur kompleksitas logis program secara kuantitatif.

3. METODE PENELITIAN

Penelitian ini menggunakan pendekatan Research and Development (R&D) dengan model ADDIE (Analysis, Design, Development, Implementation, Evaluation) yang dimodifikasi untuk pengembangan perangkat lunak (Sugiyono, 2013).



Gambar 1. Alur Metodologi Penelitian ADDIE

A. Analisis Kebutuhan

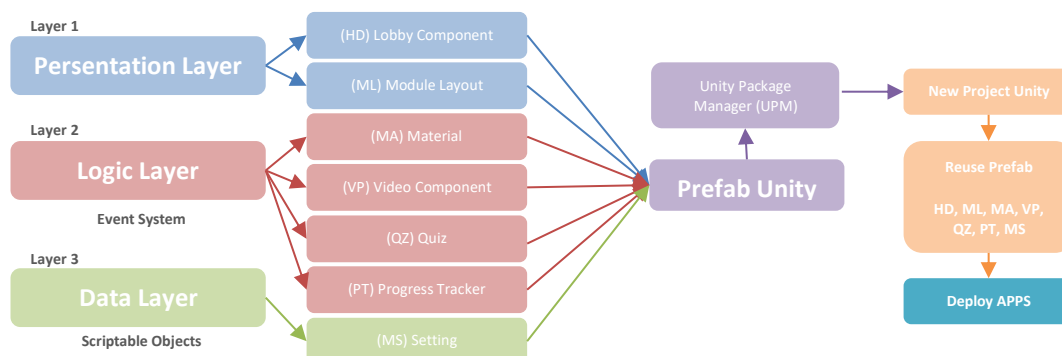
Tahap analisis dilakukan melalui studi literatur dan observasi terhadap pola umum pengembangan media pembelajaran. Identifikasi komponen dilakukan dengan menganalisis kesamaan fungsional yang muncul berulang di berbagai aplikasi media pembelajaran. Hasilnya adalah tujuh komponen yang disajikan dalam Tabel 1.

Tabel 1. Identifikasi Komponen reusable untuk media pembelajaran

Code	Nama Component	Layer	Fungsi
HD	Lobby Design	<i>Presentation</i>	Halaman Utama Aplikasi: Menampilkan Daftar Menu utama dari sistem dan profil pengguna.
MA	Material	<i>Logic</i>	Menampilkan Material Teks., Gambar, dan Dokumen dalam format yang terstruktur.
ML	Module Layout	<i>Presentation</i>	Mengatur Tata Letak dan Navigasi antar Modul dan Bab dalam satu Aplikasi Media Pembelajaran
PT	Progress Tracker	<i>Logic</i>	Melacak dan Menampilkan Progress atau Kemajuan Belajar secara Real Time
VP	Video Player	<i>Logic</i>	Memutar Konten Video Pembelajaran lengkap dengan Navigasi
QZ	Quiz	<i>Logic</i>	Mengelola Soal, Pilihan Jawaban, serta Penilaian
ST	Main Setting	<i>Data</i>	Mengelola MainEvent system.

B. Desain Arsitektur

Arsitektur yang diusulkan mengikuti prinsip CBSE dengan tiga lapisan utama: (1) Presentation Layer (Lobby Component, Module Layout Component) bertanggung jawab atas navigasi dan tampilan; (2) Logic Layer (Quiz, Video Player, Material, Progress Tracker Component) menangani inti fungsionalitas pembelajaran; (3) Data & Configuration Layer (Main Setting Component) mengelola preferensi dan data persisten. Gambar 2 menunjukkan ilustrasi arsitektur lapisan reusable prefab component yang diusulkan.



Gambar 2. Arsitektur Lapisan Prefab Reuse Component

Pola desain yang digunakan dalam implementasi meliputi: (1) Singleton Pattern untuk komponen Manager (Main Setting, Progress Tracker) yang perlu diakses secara global; (2) Observer Pattern (Event System) untuk komunikasi antar komponen tanpa ketergantungan langsung; (3) Scriptable Object Pattern untuk pemisahan data dari logika komponen, meningkatkan kemampuan konfigurasi tanpa mengubah kode; dan (4) Factory Pattern untuk instansiasi komponen secara dinamis berdasarkan kebutuhan konten.

C. Implementasi

Implementasi dilakukan menggunakan Unity 2022 LTS dengan bahasa pemrograman C#. Setiap komponen diimplementasikan sebagai Prefab mandiri yang disimpan dalam Component Library, sebuah Unity Package Manager (UPM) yang dapat diimpor ke proyek media pembelajaran baru. Struktur direktori dirancang untuk memudahkan pengelolaan dan penemuan komponen.

Untuk mendemonstrasikan reusability, arsitektur diuji dengan mengimplementasikan reusable component pada lima aplikasi media pembelajaran yang berbeda. Kelima aplikasi dibangun tanpa menulis ulang logika komponen inti.

D. Evaluasi

Evaluasi dilakukan pada dua dimensi utama. Pertama, evaluasi teknis mengukur tingkat reusability komponen menggunakan metrik, Evaluasi ini akan meninjau dari 5 aplikasi yang telah dikembangkan dengan menggunakan ulang komponen-komponen yang diajukan.:

1) Reuse Ratio

Metrik ini digunakan untuk mengukur bagaimana komponen yang dibuat benar benar di gunakan kembali pada proyek berbeda. Perbandingan yang dilakukan pada Bagian ini adalah melihat Komponen apa yang digunakan pada Aplikasi A, Aplikasi B, dan Aplikasi C (Total 5 Aplikasi). Dengan rumus perhitungan sebagai berikut :

$$Reuse\ Ratio\ (RR) = \left(\frac{Jumlah\ Komponen\ Reuse}{Total\ Komponen\ dalam\ Sistem\ Baru} \right) \quad (1)$$

2) Coupling Dependencies

Dalam penelitian Software Engineering, tingkat keterikatan antar komponen yang rendah menjadi salah satu karakteristik penting dalam pengembangan sistem modular. Evaluasi pada penelitian ini menggunakan coupling dependencies untuk mengukur jumlah ketergantungan antar komponen pada reusable architecture yang dikembangkan.

Nilai coupling dependencies menunjukkan berapa banyak hubungan atau dependensi yang dimiliki suatu komponen terhadap komponen lainnya. Semakin rendah nilai coupling dependencies, maka semakin independen komponen tersebut dan semakin mudah untuk digunakan kembali pada aplikasi lain.

Evaluasi coupling dilakukan melalui analisis hubungan antar script dan komunikasi antar komponen pada sistem berbasis prefab di Unity (Mihale-Wilson et al., 2022).

3) Maintainability Index (MI)

MI adalah metrik gabungan yang digunakan untuk mengukur tingkat kemudahan dalam memelihara, memahami, dan memodifikasi kode program (Heitlager et al., 2007; Oman & Hagemeister, 1992). Menurut (Mikejo5000, 2025), maintainability index digunakan untuk mengukur kemudahan pemeliharaan kode. Metrik ini dihitung berdasarkan, Halstead Volume, Cyclomatic Complexity, dan Lines of Code (LoC).

$$MI = 171 - 5.2\ln(V) - 0.23(G) - 16.2\ln(LOC) \quad (2)$$

Keterangan:

- V = Halstead Volume
- G = Cyclomatic Complexity
- LOC = Lines of Code
- 4) Cyclomatic Complexity

Metode pengukuran Metrik yang dikembangkan oleh Thomas J. McCabe untuk pengukuran kuantitatif dari kompleksitas logis sebuah program. Nilai Cyclomatic Complexity dikategorikan berdasarkan standar McCabe, yaitu:

- 1–10 (low risk),
- 11–20 (moderate risk),
- dan >20 (high risk).

4. HASIL DAN PEMBAHASAN

A. *Arsitektur Komponen yang diusulkan*

Arsitektur yang diusulkan terdiri dari tiga lapisan dengan tujuh komponen inti yang saling berinteraksi melalui event system berbasis Observer Pattern. Diagram arsitektur menunjukkan bahwa setiap komponen bersifat independen dan berkomunikasi melalui antarmuka yang terdefinisi, bukan melalui ketergantungan langsung. Lapisan Presentation (Lobby Component, Module Layout Component) bertanggung jawab atas navigasi dan tampilan utama aplikasi. Lapisan Logic (Quiz Component, Video Player Component, Material Component, Progress Tracker Component) menangani inti fungsionalitas pembelajaran. Lapisan Data dan Konfigurasi (Main Setting Component) mengelola preferensi pengguna dan data persisten.

B. *Implementasi Komponen dalam Unity*

Setiap komponen diimplementasikan sebagai Unity Prefab dengan tiga elemen utama: (1) MonoBehaviour script yang mendefinisikan logika komponen, (2) Scriptable Object yang menyimpan data konfigurasi yang dapat diubah tanpa mengubah kode, dan (3) UI prefab yang mendefinisikan tampilan visual komponen.

Sebagai contoh, Quiz Component dirancang dengan antarmuka IQuizComponent yang mendefinisikan method LoadQuestions(QuizData data), StartQuiz(), dan OnQuizComplete(QuizResult result). Pengembang yang menggunakan komponen ini hanya perlu menyediakan data soal dalam format QuizData (Scriptable Object) dan mendaftarkan handler untuk event OnQuizComplete, tanpa perlu memahami detail implementasi internal.

Video Player Component menggunakan Unity Video Player API sebagai fondasi, dengan lapisan abstraksi yang menambahkan fitur bookmark, subtitle, dan tracking kemajuan yang terintegrasi dengan Progress Tracker Component melalui event system. Komunikasi ini terjadi secara loosely coupled sehingga Video Player Component dapat digunakan tanpa Progress Tracker jika diperlukan.

C. *Demonstrasi Reusability pada Berbagai Aplikasi Pembelajaran*

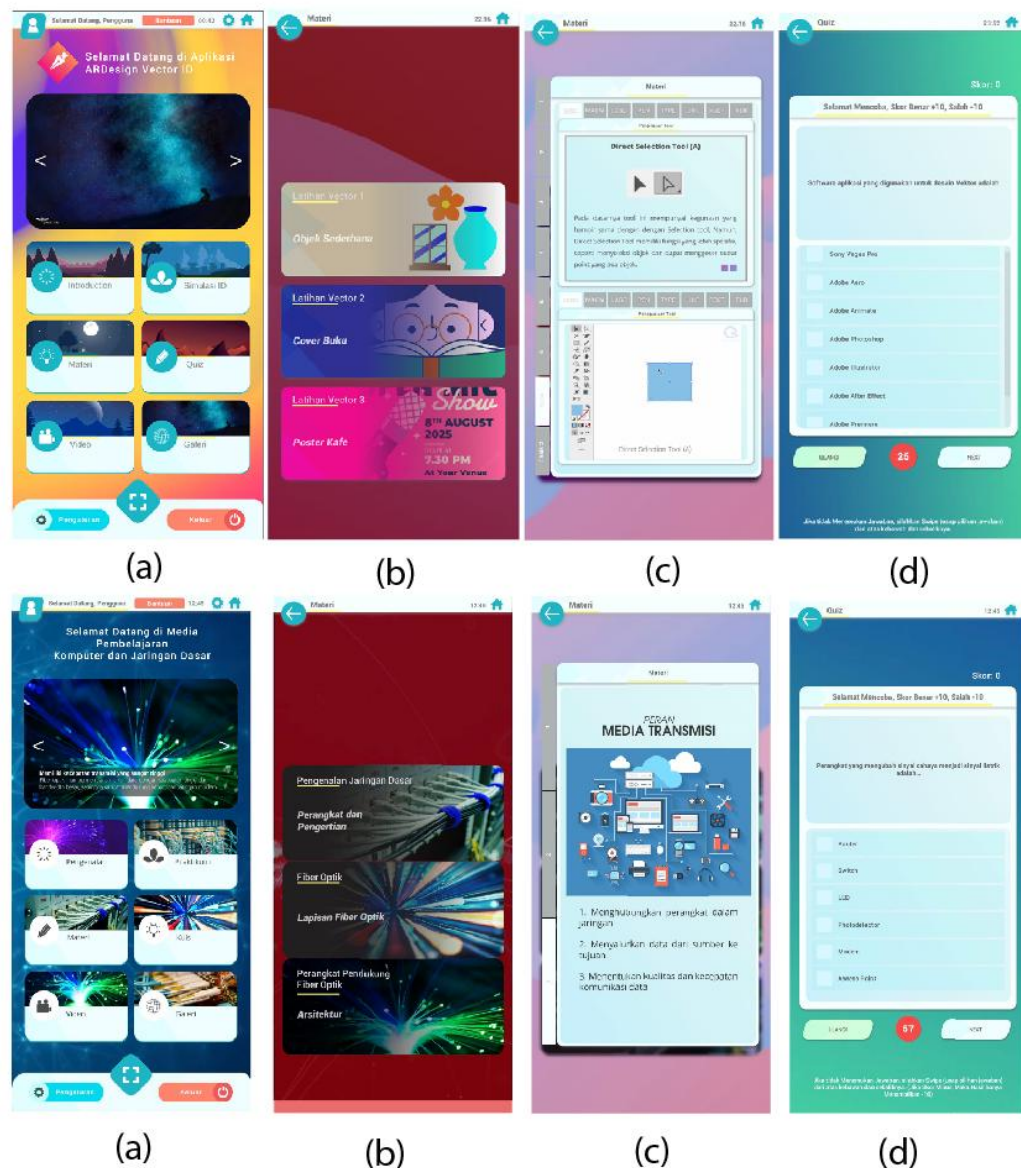
Untuk menguji tingkat reusability dari arsitektur yang dikembangkan, component library diimplementasikan pada lima aplikasi media pembelajaran yang berbeda, yaitu:

1. Media Pembelajaran Desain Grafis, (Quiz Component, Video Player Component, Material Component, Module Layout Component, Lobby Component, dan Main Setting Component, Progress Tracker Component)
2. Media Pembelajaran Jaringan Dasar, (Quiz, Video Player Component, Material Component, Module Layout Component, Lobby Component, dan Main Setting Component, Progress Tracker Component)
3. Komik AR, (Main Setting Component, Modul Layout, dan Video Player, Quiz, Materi Component, Progress Tracker Component)
4. Komik Tennis Meja, (Modul Kamera AR, Video Player, Main Setting, dan Modul Component)
5. Network ID Learning Media. (Video Player, Main Setting, Materi, dan Modul Layout Component, Progress Tracker Component)

Hasil implementasi menunjukkan bahwa seluruh aplikasi berhasil dibangun menggunakan reusable component yang sama tanpa melakukan penulisan ulang terhadap logika utama sistem. Perbedaan antar aplikasi hanya terletak pada konten pembelajaran seperti materi, video, gambar, dan soal kuis yang dikonfigurasi menggunakan Scriptable Object.

Media Pembelajaran Desain Grafis dan Jaringan Dasar menggunakan hampir seluruh komponen inti seperti Quiz Component, Material Component, Video Player Component, Module Layout Component, Lobby Component, dan Main Setting Component. Sementara itu, aplikasi Komik AR dan Komik Tennis Meja menggunakan kombinasi komponen yang lebih spesifik sesuai kebutuhan aplikasi augmented reality. Hasil ini menunjukkan bahwa arsitektur reusable berbasis prefab mampu mendukung pengembangan berbagai jenis media pembelajaran secara fleksibel dan scalable.

Selain meningkatkan fleksibilitas pengembangan, pendekatan reusable component juga membantu pengembang mengurangi proses implementasi berulang pada fitur-fitur dasar pembelajaran. Pengembang tidak perlu lagi membuat ulang sistem kuis, video player, ataupun navigasi modul pada setiap proyek karena seluruh fungsi inti telah tersedia dalam bentuk reusable prefab.



Gambar 3. Demonstrasi Komponen Reuse pada Aplikasi Media Pembelajaran Desain Grafis (Atas) dan Jaringan Dasar (Bawah)

Gambar di atas mendemonstrasikan bagaimana penggunaan reuse Component pada pengembangan Media Belajar yang dapat mempercepat pembuatan Media, Seperti pada bagian a. Dapat dilihat tampilan Dashboard yang mudah di gunakan dan kustomisasi, b. Bagian Pembukaan Materi, c. Bagian isi konten Materi, dan d. Bagian Quiz. Pemanfaatan prefab dalam pengembangan software dapat memberikan dampak yang signifikan dalam pengerjaan dan efisiensi waktu pengembangan.

D. Evaluasi Teknis Reusable Component

Evaluasi teknis dilakukan menggunakan beberapa software engineering metrics yang meliputi Reuse Ratio, Maintainability Index, Coupling Dependencies, dan Cyclomatic Complexity. Evaluasi dilakukan terhadap seluruh reusable component yang dikembangkan. Metrik Maintainability Index, Cyclomatic Complexity, dan Coupling Dependencies diperoleh menggunakan fitur Code Metrics pada Visual Studio 2022 yang terintegrasi dengan proyek Unity berbasis C#.

Tabel 2. Hasil Metrik Teknis Per Komponen

Nama Component	Reuse Ratio %	Maintainability Index (i)	Coupling Dependencies	Cyclomatic Complexity
Quiz Component	80%	91	8	9
Video Player Component	60%	76	3	7
Material Component	80%	70	3	1
Progress Tracker Component	80%	83	11	11
Module Layout Component	100%	81	1	4
Lobby Component	80%	73	3	1
Main Setting Component	100%	86	4	5

Berdasarkan hasil evaluasi, Module Layout Component dan Main Setting Component memperoleh reuse ratio tertinggi sebesar 100%. Hal ini menunjukkan bahwa kedua komponen digunakan secara konsisten pada seluruh aplikasi pembelajaran yang dikembangkan. Tingginya reuse ratio menunjukkan bahwa kedua komponen memiliki tingkat fleksibilitas dan generic functionality yang baik sehingga dapat diterapkan pada berbagai jenis aplikasi pembelajaran tanpa memerlukan modifikasi besar.

Quiz Component dan Progress Tracker Component memperoleh nilai maintainability index yang tinggi, yaitu masing-masing sebesar 91 dan 83. Nilai tersebut menunjukkan bahwa struktur kode pada kedua komponen relatif mudah dipelihara dan dikembangkan kembali. Penggunaan Scriptable Object dan modular architecture membantu mengurangi kompleksitas implementasi sehingga perubahan konfigurasi dapat dilakukan tanpa memengaruhi keseluruhan sistem.

Dari sisi coupling dependencies, sebagian besar komponen memiliki tingkat ketergantungan yang rendah. Module Layout Component memiliki coupling dependencies terendah sebesar 1, yang menunjukkan independensi komponen yang sangat baik. Sebaliknya, Progress Tracker Component memiliki coupling dependencies tertinggi sebesar 11 karena komponen ini berinteraksi dengan beberapa komponen lain melalui event system untuk melakukan sinkronisasi progres pembelajaran. Meskipun demikian, komunikasi antar komponen tetap dilakukan menggunakan Observer Pattern sehingga hubungan antar komponen masih bersifat loosely coupled.

Evaluasi Cyclomatic Complexity menunjukkan bahwa sebagian besar komponen berada pada kategori low risk. Material Component dan Lobby Component memperoleh nilai complexity sebesar 1, menunjukkan bahwa logika program relatif sederhana dan mudah dipelihara. Progress Tracker Component memperoleh complexity tertinggi sebesar 11 karena menangani beberapa alur logika terkait sinkronisasi data progres pengguna. Namun nilai tersebut masih berada pada kategori moderate risk dan masih dapat dikelola dengan baik dalam pengembangan perangkat lunak modular.

Secara keseluruhan, hasil evaluasi menunjukkan bahwa reusable prefab-based architecture yang diusulkan berhasil memenuhi karakteristik utama reusable software component, yaitu modularity, maintainability, low coupling, dan reusability. Pendekatan ini juga membuktikan bahwa penerapan CBSE pada Unity dapat menjadi solusi efektif dalam pengembangan media pembelajaran digital yang lebih scalable dan efisien. Evaluasi dilakukan pada lima aplikasi yang dikembangkan oleh tim yang sama sehingga hasil reuse ratio berpotensi lebih tinggi dibandingkan implementasi pada pengembang yang berbeda.

5. PENUTUP

Penelitian ini berhasil merancang dan mengimplementasikan reusable prefab-based architecture untuk pengembangan media pembelajaran menggunakan Unity dengan menerapkan prinsip Component-Based Software Engineering (CBSE). Arsitektur yang dikembangkan terdiri dari tujuh komponen inti yaitu Quiz Component, Video Player Component, Progress Tracker Component, Material Component, Module Layout Component, Lobby Component, dan Main Setting Component yang diimplementasikan sebagai Unity Prefab independen dan dapat digunakan kembali pada berbagai aplikasi pembelajaran.

Hasil implementasi menunjukkan bahwa reusable component yang dikembangkan berhasil digunakan pada lima aplikasi media pembelajaran yang berbeda tanpa melakukan penulisan ulang terhadap logika utama sistem. Perbedaan antar aplikasi hanya terletak pada konfigurasi konten pembelajaran seperti materi, video, gambar, dan soal kuis yang diatur menggunakan Scriptable Object. Hal ini menunjukkan bahwa arsitektur yang diusulkan mampu mendukung pengembangan media pembelajaran secara modular, fleksibel, dan scalable.

Berdasarkan hasil evaluasi teknis, sebagian besar komponen memperoleh nilai reuse ratio yang tinggi dengan Module Layout Component dan Main Setting Component mencapai reuse ratio sebesar 100%. Selain itu, nilai Maintainability Index pada sebagian besar komponen berada di atas 70 yang menunjukkan bahwa struktur kode relatif mudah dipelihara dan dikembangkan kembali. Dari sisi coupling, sebagian besar komponen memiliki tingkat ketergantungan yang rendah sehingga mendukung karakteristik loosely coupled

pada arsitektur CBSE. Evaluasi Cyclomatic Complexity juga menunjukkan bahwa sebagian besar komponen berada pada kategori low risk hingga moderate risk sehingga kompleksitas logika program masih dapat dikelola dengan baik.

Secara keseluruhan, penelitian ini membuktikan bahwa penerapan reusable prefab-based architecture pada Unity dapat meningkatkan modularitas, maintainability, dan reusability dalam pengembangan media pembelajaran digital. Pendekatan ini memungkinkan pengembangan aplikasi pembelajaran secara lebih efisien karena fitur-fitur inti dapat digunakan kembali tanpa implementasi ulang pada setiap proyek baru.

Penelitian ini masih memiliki keterbatasan karena evaluasi difokuskan pada aspek teknis perangkat lunak dan belum mencakup pengujian usability pengguna akhir maupun pengukuran performa sistem pada skala enterprise. Oleh karena itu, penelitian selanjutnya dapat dikembangkan dengan menambahkan evaluasi pengguna, integrasi cloud database, adaptive learning system, serta pengembangan reusable architecture untuk lingkungan pembelajaran berbasis multiplayer dan cross-platform learning environment.

REFERENSI

- Arkorful, V., & Abaidoo, N. (2014). The role of e-learning, the advantages and disadvantages of its adoption in higher education. *International Journal of Education and Research*, 2(12), 397–410.
- Chiao, H.-M., Huang, W.-H., & Doong, J.-L. (2017). Innovative research on development game-based tourism information service by using component-based software engineering. In *2017 International Conference on Applied System Innovation (ICASI)* (pp. 1565–1567). IEEE. <https://doi.org/10.1109/ICASI.2017.7988227>
- Dwijayani, N. M. (2019). Development of circle learning media to improve student learning outcomes. *Journal of Physics: Conference Series*, 1321(2), Article 022099. <https://doi.org/10.1088/1742-6596/1321/2/022099>
- Farhani, V. C., Santika, S., & Mulyani, E. (2026). Development of Android-based interactive learning media to facilitate the flipped classroom learning model. *Kognitif: Jurnal Riset HOTS Pendidikan Matematika*, 6(1), 158–171. <https://doi.org/10.51574/kognitif.v6i1.3281>
- Frakes, W. B., & Kang, K. (2005). Software reuse research: Status and future. *IEEE Transactions on Software Engineering*, 31(7), 529–536. <https://doi.org/10.1109/TSE.2005.85>
- Hasaniyah, N., Arsyad, B., & Hasan, A. A. (2023). Development of augmented reality-based Arabic Uslub materials to improve daily language expressing ability. *ELOQUENCE: Journal of Foreign Language*, 2(1), 241–254. <https://doi.org/10.58194/elquence.v2i1.672>
- Heitlager, I., Kuipers, T., & Visser, J. (2007). A practical model for measuring maintainability. In *6th International Conference on the Quality of Information and Communications Technology (QUATIC 2007)* (pp. 30–39). IEEE. <https://doi.org/10.1109/QUATIC.2007.8>
- Hidayah, M., Nisa, A. F., Khosiyono, B. H. C., Masjid, A. A., & Cahyani, B. H. (2025). Analysis of Android-based science learning media to improve motivation and learning outcomes of elementary school students: A systematic literature review. *Jurnal Penelitian Pendidikan IPA*, 11(3), 86–96. <https://doi.org/10.29303/jppipa.v11i3.10236>
- Jauza, N. A., & Albina, M. (2025). Penggunaan media pembelajaran kreatif dan inovatif dalam meningkatkan kualitas pembelajaran. *Jurnal IHSAN: Jurnal Pendidikan Islam*, 3(2), 15–23. <https://doi.org/10.61104/ihsan.v3i2.886>
- Juárez-González, P., Cano-Iglesias, M. J., Cebrián-Robles, D., & Franco-Mariscal, A. J. (2025). *Elemental Home*: A video game to explore chemistry in everyday life. *Journal of Chemical Education*, 102(8), 3716–3724. <https://doi.org/10.1021/acs.jchemed.5c00168>
- Kim, D., Choi, J.-Y., & Hong, J.-E. (2017). Evaluating energy efficiency of Internet of Things software architecture based on reusable software components. *International Journal of Distributed Sensor Networks*, 13(1), Article 1550147716682738. <https://doi.org/10.1177/1550147716682738>
- Krueger, C. W. (1992). Software reuse. *ACM Computing Surveys*, 24(2), 131–183. <https://doi.org/10.1145/130844.130856>
- Mihale-Wilson, C., Felka, P., Hinz, O., & Spann, M. (2022). The impact of strategic core-component reuse on product life cycles: The case of location-based games. *Business & Information Systems Engineering*, 64(2), 223–237. <https://doi.org/10.1007/s12599-021-00706-y>
- Mikejo5000. (2025, October 31). *Code metrics—Maintainability index range and meaning—Visual Studio (Windows)*. Microsoft Learn. <https://learn.microsoft.com/en-us/visualstudio/code-quality/code-metrics-maintainability-index-range-and-meaning?view=visualstudio>
- Niemela, P., & Hyyro, H. (2019). Migrating learning management systems towards microservice architecture. In *Summer School on Software Maintenance and Evolution* (Vol. 2520, pp. 10–20). CEUR Workshop Proceedings. <https://ceur-ws.org/Vol-2520/paper2a.pdf>
- Oman, P., & Hagemester, J. (1992). Metrics for assessing a software system's maintainability. In *Proceedings of the Conference on Software Maintenance 1992* (pp. 337–344). IEEE. <https://doi.org/10.1109/ICSM.1992.242525>
- Pumama, N., & Putra, M. A. P. (2026). Enhancing teachers' competence through training on Android-based mathematics learning media: Peningkatan kompetensi guru melalui pelatihan media pembelajaran matematika berbasis Android. *Room of Civil Society Development*, 5(1), 24–33. <https://doi.org/10.59110/rcsd.828>
- Salas, R. P. (2020). Reusable learning objects: An agile approach. In *2020 IEEE Frontiers in Education Conference (FIE)* (pp. 1–6). IEEE. <https://doi.org/10.1109/FIE44824.2020.9273947>
- Sugiyono. (2013). *Metode penelitian pendidikan: Pendekatan kuantitatif, kualitatif, dan R&D*. Alfabeta.
- Szyperski, C., Gruntz, D., & Murer, S. (2002). *Component software: Beyond object-oriented programming*. Pearson Education.
- Unity Technologies. (n.d.). *Unity manual: Prefabs*. Retrieved May 18, 2026, from <https://docs.unity3d.com/6000.4/Documentation/Manual/Prefabs.html>
- Wei, G., Zhong-Wei, X., & Ren-Zuo, X. (2009). Metrics of graph abstraction for component-based software architecture. In *2009 WRI World Congress on Computer Science and Information Engineering* (pp. 518–522). IEEE. <https://doi.org/10.1109/CSIE.2009.697>