ISSN PRINT : 2302-500X
ISSN ONLINE : 2477-3786

A Framework for Developing Augmented Reality Applications Based on the Multimedia Development Life Cycle (MDLC)

Rickman Roedavan[1], Abdullah Pirus Leman[2], Sritenaya Geovani Putri[3]

[1] Multimedia Engineering Technology Program, Faculty of Applied Science, Telkom University, Jl. Telekomunikasi No. 1, Terusan Buah Batu Bojongsoang, Bandung, West Java 40257, Indonesia
[2] Faculty of Information Sciences & Engineering, Management & Science University, University Drive, Off Persiaran Olahraga, 40100 Shah Alam, Selangor, Malaysia
[3] Department of Entrepreneurship and Strategy, Lancaster University Management School, Lancaster University, Bailrigg, Lancaster, LA1 4YX, England, United Kingdom
Email :rikman@telkomuniversity.ac.id [1], abdullah_pirus@msu.edu.my [2], s.g.putri@lancaster.ac.uk[3]

ABSTRACT

Augmented Reality (AR) applications have experienced significant growth in recent years. Despite this growth, the development methodologies for AR applications remain underexplored in academic research. Most existing approaches rely heavily on the general Multimedia Development Life Cycle (MDLC) This gap highlights the need for a more tailored development framework to effectively guide the creation of AR applications. To address this gap, this study proposes a specialized framework for developing AR applications. The framework refines and extends the MDLC methodology by integrating key elements tailored to AR-specific requirements. The proposed framework was tested through an extensive empirical study involving 10 development teams over the course of 2024. Each team utilized the framework to design and build an AR application tailored to their respective domains. The usability of the resulting applications was evaluated using the System Usability Scale (SUS), a widely recognized metric for assessing user experience. The results were highly encouraging, with the average SUS score across all applications exceeding 70%, indicating a "good" level of usability. This outcome demonstrates the effectiveness of the framework in enabling the development of high-quality AR applications that meet user needs.

Keywords : *Augmented Reality (AR), Multimedia Development Life Cycle (MDLC), Development Framework, System Usability Scale (SUS).*

1. Introduction

Augmented Reality (AR) has experienced rapid growth, transforming industries such as education, tourism, and healthcare (Nayak, 2024; Juarez, 2024; Kepceoglu, 2024). By seamlessly integrating digital content into the real world, AR creates interactive and immersive experiences that were once considered futuristic. This significant progress has been supported by advancements in game engines and AR application development frameworks such as Vuforia, ARCore and ARKir, which have simplified the creation and deployment of AR solutions (Tomic, 2023). These tools empower developers to create sophisticated AR applications with relative ease, contributing to the widespread adoption of AR technology across various domains.

Despite this progress, the development of dedicated frameworks for AR has largely been overlooked by researchers. Much of the existing research in AR focuses on exploring novel interaction methods and pushing the boundaries of AR's technical capabilities, often relying on the Multimedia Development Life Cycle (MDLC) as the primary development methodology (Aryani, 2024; Anugrah, 2024; Fahmizer, 2023, Hidayatullah, 2022). While MDLC provides a robust and systematic approach for general multimedia development, it lacks the specificity required to address the unique characteristics of AR applications. AR development involves challenges such as real-time interaction, spatial awareness, and user immersion, which necessitate a tailored approach to ensure high-quality outcomes (Para, 2024; Yoneyama, 2024).

Recognizing this gap, this paper aims to propose a framework specifically designed for AR application development. The proposed framework will be adapted using the Multimedia Development Life Cycle (MDLC) version developed by the author (Roedavan, 2022), which consists of 5 stages, differing from the 6-stage version proposed by Luther-Sutopo. This adaptation integrates AR-specific considerations to better align with the habits and unique specifications of AR development. By providing a systematic, step-by-step methodology, this framework addresses the unique challenges of AR while maintaining a strong focus on usability and efficiency.

The structure of this paper is as follows: the introduction outlines the background and rationale for this study, followed by a literature review that explores existing research and methodologies related to AR development. The proposed method section presents the details of the framework, including its design and key components. Results and discussion analyze the framework's effectiveness through empirical testing and usability evaluation, while the conclusion highlights the contributions and potential future directions of this work.

Article history:
Received 25 Januari 2025
Revised 14 February 2025
Accepted 15 February 2025

ISSN PRINT : 2302-500X
ISSN ONLINE : 2477-3786

2. Literature Review

2.1. The Multimedia Development Life Cycle (MDLC)

The Multimedia Development Life Cycle (MDLC) is one of the most widely adopted methodologies for multimedia application development. It provides a structured approach for systematically managing the creation and delivery of multimedia content, ensuring high-quality outputs.

Among the various versions of MDLC, the model proposed by Luther and later adapted by Sutopo is the most prevalent in research and practice. This version consists of six stages: Concept, Design, Material Collecting, Assembly, Testing, and Distribution. However, finding the original sources for Luther's or Sutopo's MDLC framework has proven to be a challenge, as much of the documentation exists in secondary references or interpretations by subsequent authors (Wiguna, 2024; Apryani, 2024; Atun, 2024).

The latest version of MDLC, developed by the author, consists of five steps: Initialization, Blueprint Design, Assets Preparation, Product Development, and Testing and Validation. This methodology has gained significant recognition, with over 11,000 researchers having accessed it and more than 30 studies citing it in their work. This widespread acknowledgment strengthens the argument for using the author's MDLC as the foundational framework for developing the AR Framework in this study. Table 1 highlights the comparison between the Luther-Sutopo MDLC version and the author's version.

Table1. MDLC Version Comparison

	Luther-Sutopo's MDLC	Roedavan's MDLC
No	Stages	Stages
1	Concept	Initialization
2	Design	Blueprint Design
3	Material Collecting	Assets Preparation
4	Assembly	Product Development
5	Testing	Testing & Validation
6	Distribution	

The primary distinction between the two MDLC models lies in their focus and approach to development. In Luther's model, the Material Collecting and Assembly stages stand out as key strengths. These stages streamline the multimedia development process by allowing developers to utilize pre-existing assets, thus speeding up production without requiring all multimedia elements to be created from scratch. However, one limitation of this model is its lack of explicit emphasis on interactivity, which is crucial for non-linear multimedia products.

On the other hand, author's MDLC introduces the Blueprint Design stage, which adopts the concept of "blueprint" commonly used in game engines, also referred to as Prefabrication. This stage focuses on creating designs with built-in interactivity as a default feature, ensuring that the resulting media is inherently interactive. This makes author's MDLC particularly suited for developing frameworks for interactive and immersive applications such as Augmented Reality (AR). By addressing interactivity from the outset, this model provides a more comprehensive foundation for AR development.

2.2. Augmented Reality Development Type

The rapid advancement of AR technology has led to the emergence of numerous AR applications and platforms that simplify the process of creating AR content, even for non-programmers. Platforms like Spark AR and Lens Studio empower users to design AR experiences using user-friendly interfaces and pre-built templates, eliminating the need for coding expertise (Hidayatullah, 2022; Grieger, 2023).

However, this raises an important question: does developing an AR application mean building an AR solution from scratch, adding content to an existing AR application, or merely assembling AR content using libraries provided by existing frameworks? Answering this question requires a closer look at how AR development frameworks are structured and the varying levels of abstraction they offer.

The author categorizes AR development into three distinct types of frameworks, each designed to serve different purposes and cater to varying levels of expertise. These categories not only define the amount of creative control and technical effort required but also shape the scope, functionality, and flexibility of the resulting AR applications. Below is an overview of these three types of AR development frameworks:

a. High-Level AR Development Type

High-level AR development type are designed for users with little to no programming experience, allowing them to create AR content with minimal technical expertise. Examples include platforms like Spark AR or Lens Studio, which provide user-friendly interfaces and pre-built templates. These tools simplify the AR development process by enabling users to focus on creating content rather than dealing with technical complexities (Aqviriyoso, 2022; Bostanica, 2024).

However, their accessibility comes with limitations: creativity is often constrained as users are restricted to predefined functionalities within the platform. Essentially, these frameworks are best suited for creating content on top of someone else's AR application rather than building unique and customized solutions from the ground up.

Furthermore, the purpose of AR applications developed with these tools tends to be highly restricted, as they are bound by the capabilities and features provided by the platform. Additionally, support for platforms is often limited to the services offered by the provider, which can further constrain the scalability and flexibility of AR projects.

From a monetization perspective, the implications vary between single developers and enterprise developers. Single developers may face challenges in achieving meaningful revenue due to the limited scope

ISSN PRINT : 2302-500X
ISSN ONLINE : 2477-3786

of the applications they can create. On the other hand, enterprise developers may require more robust solutions that these high-level tools cannot provide, forcing them to look for alternative platforms or custom development approaches. For service providers offering these platforms, maintaining the service comes with its own challenges, particularly the cost of server infrastructure required to sustain the platform's operations and ensure its availability for users. Balancing accessibility with these operational costs becomes a critical consideration for long-term sustainability.

b. Medium-Level AR Development Type

Medium-level AR development type strike a balance between ease of use and flexibility, making them the most commonly used approach for AR development. These frameworks leverage powerful game engines like Unity or Unreal Engine in combination with AR SDKs such as ARCore or ARKit (Tomic, 2023; Sokolov, 2024).

They enable developers to design custom AR applications with greater control over interactivity and functionality while utilizing the robust features of game engines. Middle-level type are particularly popular because they allow developers to create tailored AR solutions without the need for building complex systems from scratch.

This category offers much greater freedom compared to high-level AR tools, as it can support a wide range of industries looking to integrate AR into their solutions. From retail and healthcare to education and manufacturing, middle-level AR development provides the flexibility to address diverse use cases and requirements. Additionally, monetization options are entirely at the discretion of the developer, giving them the freedom to choose how they generate revenue, whether through one-time purchases, subscriptions, or ad-based models.

However, there are challenges associated with medium-level AR development. Each AR service or SDK comes with its own licensing terms, which can vary significantly depending on the options selected. For instance, platforms like Vuforia offer different licenses depending on the deployment method—such as device-based or cloud-based targeting—as well as limitations on the number of markers that can be detected. These licensing costs and constraints can influence the overall project budget and must be carefully considered by developers when selecting the appropriate tools and services for their AR applications.

c. Low-Level AR Development Type

Low-level AR development type are aimed at developers seeking complete control over the AR internal process. These frameworks are typically used for highly specialized or innovative projects that go beyond the capabilities of high-level or middle-level tools. Development with low-level frameworks involves either building AR solutions from scratch or extending existing frameworks by integrating additional tools, such as OpenCV for computer vision or machine learning algorithms for advanced interactions (Yang, 2021; Rajeswari, 2024).

This approach is particularly suitable for teams that need to implement additional interactions in AR applications. While basic functionalities like object tracking and 3D object placement are commonly supported in standard AR SDKs, there are more complex interactions that these SDKs do not directly provide. Examples include custom features built with OpenCV, such as advanced image processing, machine learning-based gesture recognition, or voice detection for enhanced interactivity. Furthermore, immersive effects such as dynamic soundscapes and the creation of native filter masks (e.g., for facial tracking or AR filters) often require custom development at the low level before being integrated into an application.

Developers in this realm tend to create AR plugins or extensions that enhance the capabilities of leading game engines such as Unity and Unreal Engine. These plugins enable other developers to leverage specialized features without delving into low-level programming themselves. As such, low-level AR development not only fosters innovation but also contributes to the broader AR ecosystem by providing tools that can be reused and adapted for diverse projects.

Overall, understanding these AR Development type distinctions is crucial when defining AR application goals. The choice of development type significantly impacts the degree of creative freedom, technical complexity, and interactivity achievable in the final application. By identifying the specific requirements of a project, developers can select the most appropriate development type to ensure both usability and innovation in their AR applications.

3. Proposed Method

The proposed framework is an adaptation of the author's version of the Multimedia Development Life Cycle (MDLC), combined with essential components of AR development types. This framework is structured into three layers: Phase, Development Stage, and Component, each interrelated to guide the AR development process comprehensively and systematically. Figure 1 illustrates the proposed method for the AR development framework.

The framework begins with the Pre-Production Phase, which focuses on planning and conceptualization. This phase includes two main sub-components: Segmentation and Type. Segmentation is further divided into project’s purposes, platform, and pricing strategy, all of which play critical roles in establishing the foundation of the project.

The project’s purposes sub-component defines the primary objectives and goals of the AR application. The platform sub-component identifies the specific hardware and software environment where the AR application will operate. For example, mobile platforms like ARKit or ARCore are widely accessible and suitable for general

ISSN PRINT : 2302-500X
ISSN ONLINE : 2477-3786

users, while AR glasses like Microsoft HoloLens may be chosen for specialized industrial or enterprise solutions requiring hands-free functionality (Waghmare, 2023).

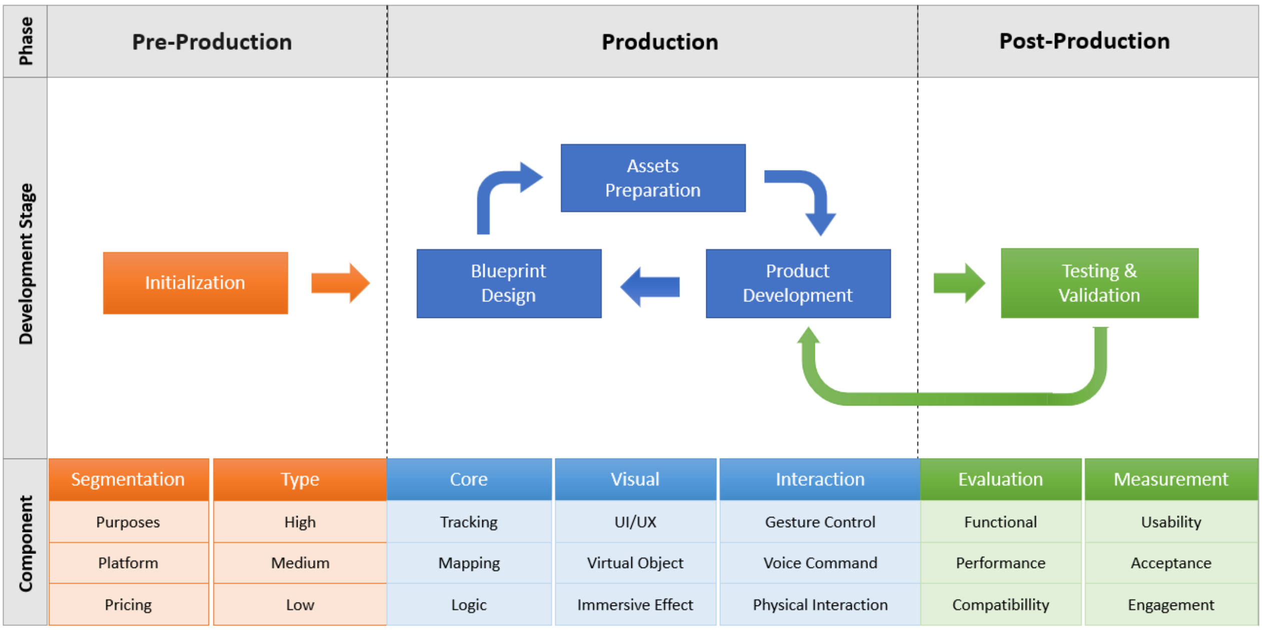


Figure 1. Augmented Reality Development Framework

The pricing strategy sub-component outlines how the AR application will be monetized or distributed. This includes decisions on whether the application will be free, subscription-based, sold as a one-time purchase, or monetized through in-app purchases or advertisements. Pricing strategy also accounts for development and operational costs, ensuring a balance between affordability for users and profitability for developers.

The Production Phase focuses on the actual development of the AR application and is divided into three main components: Core, Visual, and Interaction. The Core component covers essential AR functionalities that serve as the foundation for any AR application. One of the key functionalities is tracking, which ensures the accurate placement and alignment of virtual objects within the real-world environment.

Tracking in AR encompasses several types, each suited to different applications and environments (Sulistiyono, 2024; Alexandros, 2023). Marker-based tracking uses visual markers like QR codes or images to anchor virtual objects, offering stability and reliability in controlled settings. Marker-less tracking eliminates the need for physical markers by leveraging a device's camera and sensors to map the environment, enabling more flexible and dynamic interactions. Geo-based tracking relies on GPS and location data to position objects in specific geographical locations, ideal for outdoor applications like navigation or location-based games.

The Visual component plays a crucial role in creating engaging and immersive AR experiences, with a significant focus on the virtual objects that are overlaid in the real-world environment. These virtual objects can either be preloaded into the device or downloaded dynamically from the cloud, each approach offering distinct advantages and challenges.

When virtual objects are preloaded into the device, the application ensures offline functionality and reduces reliance on external factors like internet connectivity. This approach significantly increases the size of the application, which can lead to storage issues and impact the user's willingness to download and install the app, especially on mobile devices. On the other hand, downloading virtual objects from the cloud allows for a smaller app size and the flexibility to update or add new objects without requiring a full app update. This method depends heavily on a fast and stable internet connection, and the object format must be compatible with runtime importing.

While traditional formats like FBX are commonly used in game engines, they cannot be imported at runtime, which makes newer formats like GLTF and its binary counterpart, GLB, preferable due to their runtime compatibility and efficiency (Lam, 2024). The GLTF (GL Transmission Format) is an open standard 3D file format designed to efficiently transmit and load 3D models in web and real-time applications. It focuses on delivering compact file sizes and fast loading times by utilizing modern compression techniques. The GLB format, a binary version of GLTF, encapsulates all the 3D model's data—geometry, textures, animations, and materials—into a single file, making it more convenient for runtime use and eliminating the need for external dependencies like separate texture files. Both formats are widely supported across modern game engines and AR platforms, ensuring seamless integration and performance optimization, particularly in applications that require dynamic loading of 3D assets during runtime.

Additionally, advances in AI technology have introduced new possibilities for generating virtual

ISSN PRINT : 2302-500X
ISSN ONLINE : 2477-3786

objects. Developers may no longer need to rely solely on traditional 3D modeling software like 3DMax or Blender. AI tools, such as Stable Diffusion or similar generative AI models, can create lightweight and functional 3D objects tailored to the application's needs. This reduces the development time and cost, especially for projects requiring a large variety of simple objects. The choice of approach—preloaded, cloud-based, or AI-generated—ultimately depends on the application's requirements, target audience, and technical constraints, but all contribute to delivering a visually compelling and interactive AR experience.

The Post-Production Phase emphasizes rigorous testing, evaluation, and validation to ensure the quality, compatibility, and usability of the AR application before it is released to users. This phase includes two primary sub-components: Evaluation and Measurement. The Evaluation sub-component primarily focuses on assessing the application's compatibility and performance across various devices. Compatibility testing includes verifying that the AR application functions correctly on different screen sizes, camera qualities, and hardware capabilities. Performance testing aims to identify and address issues like lag, excessive resource usage, or overheating on devices, which could negatively impact the user experience.

The Measurement sub-component evaluates the final product's success in terms of usability, user acceptance, and engagement. For usability assessment, standardized tools like the System Usability Scale (SUS) can be employed to gather user feedback and rate the application's ease of use (Hamid, 2024). To evaluate user experience, the User Experience Questionnaire (UEQ) can provide insights into user perceptions of the application's attractiveness, efficiency, and dependability (Saleh, 2022). These survey tools offer quantitative data that help developers understand whether the application meets user expectations and provides a satisfying and engaging experience.

4. Results and Discussion

The proposed AR framework was tested with 10 teams, each consisting of three third-year Telkom University students enrolled in the Augmented Reality course. They provided with a brief document outlining the requirements. Figure 2 illustrates some pages of the project-based learning document for this AR project.

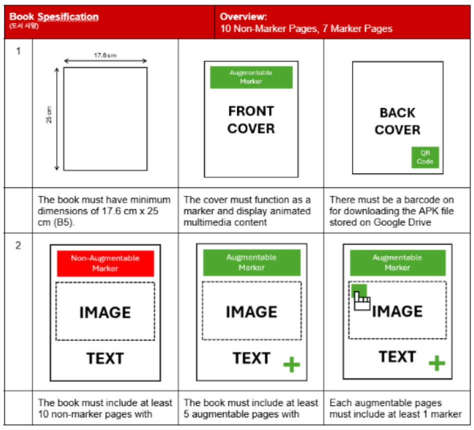


Figure 2. AR Project-Based Document

The AR markers were designed as part of a book with a minimum of 17 pages, consisting of at least 7 marker-enabled pages and 10 non-marker pages. Additionally, each team was required to include at least 10 marker cards as supplementary elements. All teams were given a duration of four months to develop an medium-level AR application using Unity and Vuforia, following a marker-based pattern. The application themes were focused on educational domains, such as history, interplanetary science, or mythology.

In the implementation phase, each team was also allowed to utilize AI-generated resources to enhance their AR projects. These resources included the creation of visual markers, 3D objects, and animations, enabling teams to streamline the development process and focus more on integrating their content into the application. For scripting needs, specifically with Unity and Vuforia, a dedicated module was provided containing pre-written C# script snippets designed to address common AR functionalities. This module served as a practical toolkit for the students, simplifying the development process. Figure 3 show some examples of these snippets.

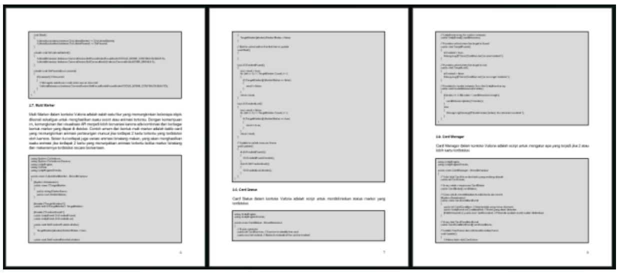
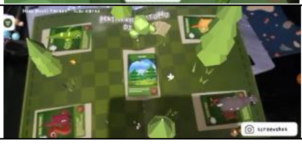


Figure 3. Example of C# Snippets

To ensure diverse and engaging experiences, the book markers were required to incorporate varying mechanics. Furthermore, combining book markers with card markers unlocked additional mechanics, enriching the overall AR experience. These variations added depth and interactivity to the projects. Table 2 showcases examples of these AR mechanics variations.

ISSN PRINT : 2302-500X
ISSN ONLINE : 2477-3786

Table 2. AR Mechanic Examples

ScreenShot	Mechanic
	Show virtual object, environment and standard information
	Show character, make it move with walk animation using gesture control
	Combining book and card to show another different visual and mechanic

Overall, all teams successfully completed their AR applications, each with unique titles and themes that showcased creativity and diversity in their projects. The applications developed include titles such as Lost, The Garage, The Moon, Olympus, Wild Wonders, The Battlefield of Muhammad Shallallahu Alaihi Wassalam, Dinopedia, Granjil Forte, Candi Yogya, and Bangun Ruang. Figure 4 illustrates a compilation of their works, highlighting the unique approaches and visual styles adopted by each team.

Figure 4. Example of Students Work

Once the applications were completed, the teams conducted user testing with 100 students from the Telkom University community. The students interacted with the AR book applications and provided feedback by completing the System Usability Scale (SUS) survey. The results showed that 9 out of 10 teams achieved a SUS score above 70, indicating that their applications were generally well-received and considered usable.

Table 3. SUS Recapitulation

No	Title	SUS Score
1	Lost	73.75
2	The Garage	83.54
3	The Moon	84.67
4	Olympus	87.06
5	Wild Wonders	80.45
6	The Battlefield of Muhammad Shallalaahu Alaihi Wassalam	78.79
7	Dinopedia	82.58
8	Granjil Forte	55.88
9	Candi Yogya	76.53
10	Bangun Ruang	80.64

Further analysis revealed that the team with a SUS score below 70 developed an application with minimal interactivity, offering only basic rotation gestures for AR object manipulation. This highlighted the importance of interactivity and diverse mechanics in creating engaging AR experiences. While the framework proved effective and applicable, certain teams required additional guidance to implement it successfully. This was particularly evident in cases where team dynamics played a significant role in the development process, an aspect that lies beyond the scope of this research.

Although the framework standardized the development steps and provided a clear methodology for AR application creation, the varying levels of technical expertise and collaboration within teams underscored the need for occasional mentoring and intervention. These challenges point to the importance of accommodating not only technical but also organizational factors in future iterations of the framework.

5. Conclusion

The study demonstrated that the proposed AR development framework is effective and applicable, enabling teams to successfully create engaging and functional AR applications across diverse educational themes. The standardized methodology provided clear guidance and structure, ensuring that most teams achieved satisfactory results, as evidenced by SUS scores above 70 in 9 out of 10 cases.

However, the analysis also highlighted certain challenges. Teams with minimal interactivity in their applications struggled to achieve high user satisfaction, underscoring the critical role of diverse and engaging mechanics in AR experiences. Additionally, team dynamics and varying levels of technical expertise played a significant role in the development process, pointing to the need for more tailored mentoring and support to address these non-technical factors.

Future iterations of the framework should consider integrating strategies to address team dynamics challenges and provide additional resources for fostering team collaboration and interactivity in AR design. Despite these challenges, the results affirm the framework's potential as a practical and adaptable tool for AR application development, particularly in educational settings. With further refinement, the framework can be extended to support more complex AR technologies and broader use cases, enhancing its versatility and impact.

References

Alexandros, K., Moustaka, M., & Evagelou, A. (2023). Location-based augmented reality for cultural heritage education: Creating educational, gamified location-based AR applications for the prehistoric

ISSN PRINT : 2302-500X
ISSN ONLINE : 2477-3786

lake settlement of Dispilio. Digital, 3(1), 18–45. https://doi.org/10.3390/digital3010002

Anugrah, M. (2024). Pengembangan media pembelajaran digital berbasis AR game-based learning dengan metode MDLC studi kasus: Pembelajaran IPA kelas 5 SD.

Apryani, A., Rohman, M., & Metandi, F. (2024). Implementasi animasi 3D pada video profil UKM jurnalistik POLNES menggunakan metode MDLC. JATI (Jurnal Mahasiswa Teknik Informatika, 8(6), 11483–11490. https://doi.org/10.36040/jati.v8i6.11521

Aqviriyoso, R., Rachman, A., & Nurhadi, A. (2022). Edukasi peran hak asasi manusia di era digital lewat filter augmented reality (AR) Instagram. Journal of Computer Science and Visual Communication Design, 7(1), 17–25. https://doi.org/10.55732/jikdiskomvis.v7i1.497

Aryani, D., Noviandi, N., Fatonah, N., & Akbar, H. (2024). Implementation of the multimedia development life cycle (MDLC) in solar system application design. International Journal of Science, Technology & Management, 5(4), 726–732. https://doi.org/10.46729/ijstm.v5i4.1123

Atun, Y., Pambudi, B., & Maulana, B. (2024). Interactive multimedia development of Indonesian endemic plants: Species uniqueness with MDLC approach. Jurnal Indonesia Sosial Teknologi, 5(7), 3203–3214. https://doi.org/10.59141/jist.v5i7.1122

Bostănică, E., Apostol, I.-G., Prelipcean, M., Goga, E., & Orzan, M. (2024). Hedonic and utilitarian motivations of social AR filters usage. https://doi.org/10.24818/ICESS/2024/087

Fahmizher, M., Ruuhwan, R., & Hartono, R. (2023). Pembuatan aplikasi denah berbasis augmented reality (AR) model marker basic tracking menggunakan metode MDLC. Jurnal Informatika dan Teknik Elektro Terapan, 11(3), 3550. https://doi.org/10.23960/jitet.v11i3s1.3550

Grieger, J., Botelho-Francisco, R., & Oliveira, L. (2023). Image dysmorphia on digital platforms: An analysis of the policies and terms of use on Instagram and Spark AR. https://doi.org/10.1007/978-3-031-47281-7_35

Hamid, K., Muhammad, H., & Nazir, Z. (2024). SUS-based usability evaluation of Windows 10 OS: An analytical survey. Journal of Interdiciplinary Innovational Leters. 1(1), 1–7.

Hidayatullah, M., Nurhayati, O., & Prasetijo, A. (2022). Perancangan game edukatif anti narkoba berbasis augmented reality (AR) menggunakan Spark AR dengan metode multimedia development life cycle (MDLC). Jurnal Ilmu Teknik dan Komputer, 6(1), 34–41. https://doi.org/10.22441/jitkom.v6i1.005

Kepceoglu, I., & Serin, M. (2024). Augmented reality applications in mathematics education. https://doi.org/10.4018/979-8-3693-6030-9.ch007

Lam, K., Croci, S., Haslbauer, P., & Smolic, A. (2024). VVGLTF: Efficient streaming of volumetric video with GLTF. Conference: IBC 2024.

Nayak, D. S., & Shivarudraswamy, R. (2024). A review on augmented reality and virtual reality technologies in the field of healthcare. https://doi.org/10.1002/9781394275472.ch3

Para, R. (2024). Intent prediction in AR shopping experiences using multimodal interactions of voice, gesture, and eye tracking: A machine learning perspective. Journal of Artificial Intelligence General Science, 7(1), 52–62. https://doi.org/10.60087/jaigs.v7i01.295

P, R., & Sasi, S. (2024). Design and development augmented reality using OpenCV for primary education. International Journal of Science and Research Archive, 13(2), 2653–2663. https://doi.org/10.30574/ijsra.2024.13.2.2432

Roedavan, R., Pudjoatmodjo, B., & Sujana, A. P. (2022). Multimedia development life cycle (MDLC). Multimedia Engineering Technology Research Laboratory . http://dx.doi.org/10.13140/RG.2.2.16273.92006

Saleh, A., Abu Addous, H., Alansari, I., & Enaizan, O. (2022). The evaluation of user experience on learning management systems using UEQ. International Journal of Emerging Technologies in Learning (iJET), 17, 145–162. https://doi.org/10.3991/ijet.v17i07.29525

Sánchez-Juárez, I., & Paredes-Xochihua, M. (2024). Augmented reality as a support for tourism promotion. ECORFAN Journal Republic of Paraguay, 10(18), 1–10. https://doi.org/10.35429/EJROP.2024.10.18.1.10

Sokolov, A., & Avrunin, O. G. (2024). Evaluation of ARCore library capabilities for determining the distance to objects in the frame. Optoelectronic Information-Power Technologies, 47(1), 58–65. https://doi.org/10.31649/1681-7893-2024-47-1-58-65

Sulistiyono, M., Hasyim, J., Bernadhed, B., Liantoni, F., & Sidauruk, A. (2024). Comparative study of marker-based and markerless tracking in augmented reality under variable environmental conditions. Journal of Soft Computing Exploration, 5(4), 413–422. https://doi.org/10.52465/joscex.v5i4.503

Tomić, D., & Galzina, V. (2023). Comparing the features and capabilities of ARCore, ARKit, and Vuforia: An analysis of their potential for creating augmented reality experiences. Interdisciplinary Management Research XIX.

Wiguna, W. (2024). Simulasi parkir mobil menggunakan metode MBL dan MDLC berbasis Android. Jurnal Nasional Komputasi dan Teknologi Informasi (JNKTI), 7(4), 635–644. https://doi.org/10.32672/jnkti.v7i4.7730

Waghmare, S., Kadam, M., & Babbar, A. (2023). Oriental institute Microsoft HoloLens: A revolutionary augmented reality device. Journal Of the Oriental Institute.

Yang, S., Pang, X., & He, X. (2021). A novel mobile application for medication adherence supervision based on AR and OpenCV designed for elderly

ISSN PRINT : 2302-500X
ISSN ONLINE : 2477-3786

patients. https://doi.org/10.1007/978-3-030-78111-8_23

Yoneyama, J., Fujimoto, Y., Okazaki, K., Sawabe, T., Kanbara, M., & Kato, H. (2024). Augmented conversations: AR face filters for facilitating comfortable in-person interactions. Journal on Multimodal User Interfaces. https://doi.org/10.1007/s12193-024-00446-9