

Implementasi Markerless Augmented Reality Berbasis Web untuk Simulasi Penataan Ruang Interior Menggunakan 8th Wall dan Three.js

Aditya Rahman Syach*, Diaz Adriansyah, Fahad Muhammad Alfarizi
Department of Informatics, UIN Sunan Gunung Djati Bandung, Indonesia

Article Info

Article history:

Received June 22, 2026
Revised October 05, 2026
Accepted October 25, 2025

Keywords:

interior design
markerless AR
SLAM
Three.js
WebAR

ABSTRACT

The interior design process often faces challenges when consumers struggle to visualize how furniture will look in their physical space before purchasing. Traditional Augmented Reality (AR) applications typically require native app installations, which creates accessibility barriers. This study implements a Web-based Augmented Reality (WebAR) application for interior design simulation using the 8th Wall engine and Three.js framework. Utilizing Simultaneous Localization and Mapping (SLAM) technology, the application enables markerless tracking, allowing users to project 3D furniture models directly onto their physical floors via a standard mobile web browser. The system features an integrated 3D model catalog and multi-gesture interactions, including single-finger drag for translation, two-finger pinch for scaling and rotation, and a raycasting-based tap mechanism for object selection and deletion. Black-box testing on various mobile browsers confirmed that the system successfully fulfilled 9 out of 10 designed functional test scenarios. While the SLAM tracking and gesture controls function accurately, the evaluation identified a technical limitation in the web-based wall collision detection. The findings demonstrate that WebAR provides a practical, installation-free alternative for immersive interior space planning.

This is an open access article under the [CC BY-SA](#) license.



Corresponding Author:

Aditya Rahman Syach
Department of Informatics, Faculty of Science and Technology,
UIN Sunan Gunung Djati Bandung Jl. A. H. Nasution No. 105, Cibiru, Bandung, Indonesia. 40614
Email: 124050069@student.uinsgd.ac.id

1. PENDAHULUAN

Penataan ruang interior merupakan aspek penting dalam menciptakan hunian yang nyaman dan fungsional. Namun dalam praktiknya, proses ini kerap menjadi tantangan tersendiri bagi banyak orang. Seseorang sangat sulit membayangkan bagaimana tampilan furnitur di dalam ruangan sebelum benar-benar ditempatkan secara fisik, sehingga proses dekorasi dan penataan menjadi memakan waktu serta tidak efisien [1]. Kondisi ini semakin diperparah di era belanja digital, di mana konsumen harus membuat keputusan pembelian hanya berdasarkan gambar statis dua dimensi. Data menunjukkan bahwa 62% pembeli mengaku kesulitan membayangkan bagaimana furnitur akan terlihat di rumah mereka, dan ketidakpastian ini menimbulkan keraguan yang berujung pada penurunan kepercayaan diri konsumen untuk melakukan pembelian [2].

Augmented Reality (AR) hadir sebagai solusi yang menjanjikan untuk menjawab permasalahan tersebut. Dengan memanfaatkan AR, pengguna dapat secara virtual menempatkan furnitur di dalam ruang hunian mereka, mengonseptualisasikan desain, dan membuat keputusan yang tepat berdasarkan visualisasi interior yang akurat [3]. Salah satu pendekatan yang semakin banyak dikembangkan adalah teknik *markerless AR*, yang memanfaatkan metode SLAM untuk mendeteksi permukaan ruangan tanpa memerlukan penanda

fisik. Melalui teknik ini, objek virtual furnitur dapat ditampilkan dan dimodifikasi secara real-time sehingga pengguna memiliki pengalaman interaktif di lingkungan dunia nyata [4]. Namun, sebagian besar implementasi AR yang ada masih bergantung pada aplikasi native yang mengharuskan proses instalasi, sehingga mengurangi aksesibilitas bagi pengguna umum.

Untuk mengatasi hambatan aksesibilitas tersebut, pendekatan *Web-based AR* (WebAR) menawarkan solusi yang lebih inklusif. 8th Wall merupakan platform AR berbasis web yang memungkinkan pengalaman AR berkualitas tinggi diakses langsung melalui browser standar tanpa memerlukan unduhan aplikasi, dengan dukungan fitur *world tracking* dan *markerless AR* untuk penempatan objek virtual di ruang nyata [5]. Sementara itu, Three.js sebagai framework 3D berbasis web dapat diintegrasikan langsung dengan sistem SLAM milik 8th Wall, memungkinkan rendering konten AR yang berjalan sepenuhnya di atas teknologi web standar seperti WebGL dan WebAssembly [6]. Kombinasi kedua teknologi ini membuka peluang untuk menghadirkan simulasi penataan ruang interior yang imersif dan mudah diakses hanya melalui tautan di peramban *smartphone*. Oleh karena itu, penelitian ini bertujuan mengimplementasikan sistem *Markerless Augmented Reality* berbasis web menggunakan 8th Wall dan Three.js sebagai solusi simulasi penataan ruang interior yang praktis dan *accessible* bagi pengguna.

2. KAJIAN TERKAIT

2.1. Markerless AR dan Teknologi SLAM

Pelacakan lingkungan fisik pada teknologi *Augmented Reality* (AR) umumnya terbagi menjadi metode *marker-based* dan *markerless tracking* [8]. Pendekatan *marker-based* mengandalkan penanda cetak dua dimensi, seperti kode QR berkontras tinggi, sebagai jangkar untuk merender objek tiga dimensi (3D) [14]. Penerapan metode ini pada simulasi interior membatasi mobilitas pengguna, sebab visualisasi objek virtual akan langsung terputus apabila penanda tersebut terhalang atau berada di luar jangkauan pandang kamera [13].

Sebagai alternatif, *markerless AR* menghilangkan keharusan penggunaan penanda fisik dengan menganalisis geometri serta fitur alami ruangan secara langsung [15]. Karakteristik tersebut sangat sesuai untuk simulasi dekorasi interior karena memfasilitasi penempatan furnitur berskala nyata secara dinamis dan fleksibel pada area lantai yang kosong [13].

Fondasi utama pembentuk kapabilitas *markerless AR* adalah algoritma *Simultaneous Localization and Mapping* (SLAM) [13]. Memanfaatkan sensor kamera perangkat, SLAM mengeksekusi dua proses komputasi spasial secara serentak, yakni melacak kedudukan serta orientasi kamera di ruang 3D (lokalisasi) dan membangun representasi visual dari struktur fisik sekitar (pemetaan) [12].

Pada aplikasi tata letak ruang, kemampuan SLAM sangat dibutuhkan untuk mengidentifikasi bidang datar horizontal sebagai acuan lantai [14]. Selama proses pemindaian berjalan, algoritma ini terus mendeteksi elemen visual alami seperti tekstur ubin atau batas dinding, yang kemudian diekstrak menjadi sekumpulan titik fitur (*feature points*) guna menyusun jaring *point cloud* [12].

2.2. Perkembangan WebAR dan Keunggulannya Dibandingkan AR Native

Kemunculan *Web-based Augmented Reality* (WebAR) merombak mekanisme distribusi konten imersif. Pada sistem tradisional, pengguna diwajibkan mengunduh aplikasi *native* berukuran besar dari platform distribusi digital [15]. Kebutuhan instalasi ini memicu hambatan adopsi (*friction*) akibat tingginya konsumsi ruang penyimpanan dan beban komputasi awal perangkat [10]. Khusus pada sektor desain interior, keengganan konsumen untuk memasang aplikasi baru hanya demi menguji coba kecocokan beberapa perabot menjadi kendala bisnis yang merugikan.

Sebagai alternatif, WebAR memperluas aksesibilitas teknologi imersif dengan mengeksekusi objek tiga dimensi (3D) langsung di dalam peramban web bawaan perangkat seluler [11]. Karakteristik lintas platform (*cross-platform*) serta bobot sistem yang ringan menjadi keunggulan utamanya. Lingkungan AR dapat langsung dimuat tanpa proses instalasi tambahan hanya dengan mengakses tautan URL atau memindai kode QR [15]. Kemudahan penyebaran (*easy spread*) melalui tautan semacam ini menjadikan WebAR sebagai solusi visualisasi yang amat praktis bagi konsumen [11].

Pada awal perkembangannya, performa WebAR memang sempat terhambat oleh keterbatasan kecepatan pemrosesan peramban seluler dan minimnya akses menuju sensor kamera tingkat lanjut [10]. Kendala teknis tersebut berhasil diatasi seiring lahirnya kerangka kerja grafis web modern [16]. Integrasi WebGL memanfaatkan pustaka seperti Three.js memfasilitasi perenderan model 3D sekaligus kalkulasi pencahayaan secara mulus di dalam peramban [16]. Di samping itu, kehadiran mesin pelacak seperti 8th Wall menyuntikkan kemampuan deteksi kedalaman ruang dan algoritma SLAM ke dalam web—sebuah fitur yang sebelumnya eksklusif beroperasi stabil pada aplikasi *native*. Perpaduan inovasi ini memungkinkan WebAR masa kini menyaingi kualitas visual maupun interaksi sistem *native* dengan tingkat aksesibilitas yang jauh lebih tinggi.

2.3. Pemanfaatan Three.js dalam Rendering 3D

Keberhasilan implementasi WebAR sangat bergantung pada efisiensi pemrosesan dan perenderan grafis tiga dimensi (3D) secara langsung di peramban seluler. Pustaka Three.js dipilih untuk memenuhi kebutuhan komputasi tersebut berkat kemampuannya mengabstraksi kompleksitas antarmuka WebGL tingkat rendah menjadi sintaksis JavaScript yang lebih modular [16]. Pada ekosistem web, pustaka ini terbukti andal dalam memuat aset model furnitur berekstensi .glb secara seketika (*real-time*). Three.js juga mengelola tata cahaya dinamis dan mengalkulasi proyeksi bayangan realistis pada permukaan lantai fisik menggunakan material khusus seperti ShadowMaterial [16].

Secara arsitektur, performa visual Three.js ditopang oleh hubungan hierarkis empat komponen fundamental: *Scene*, *Camera*, *Renderer*, dan *Mesh*. *Scene* berfungsi sebagai wadah "dunia virtual" tempat seluruh objek furnitur dan sumber cahaya disusun ke dalam satu struktur grafik terpadu. Di dalam ruang tersebut, *Camera* (khususnya tipe *PerspectiveCamera*) bertugas menetapkan koordinat dan sudut pandang visual selayaknya mata manusia, yang transformasinya terus disinkronkan dengan pergerakan fisik kamera gawai melalui modul SLAM [14].

Elemen virtual yang ditangkap oleh kamera disebut *Mesh*, yakni wujud fisik objek 3D hasil kombinasi dari *Geometry* (kerangka matematis pembentuk struktur) dan *Material* (penentu respons visual permukaan terhadap pantulan cahaya). Seluruh proses komputasi komponen-komponen ini bermuara pada *Renderer* (umumnya *WebGLRenderer*), yang mengeksekusi kalkulasi data dari *Scene* berdasarkan perspektif *Camera*. Hasil perhitungan tersebut kemudian diterjemahkan menjadi matriks piksel dinamis dan digambar pada elemen <canvas> HTML dengan kecepatan bingkai (*frame rate*) tinggi guna menghasilkan transisi pergerakan yang mulus di layar pengguna [16].

3. METODE PENELITIAN

Pengembangan aplikasi WebAR dalam penelitian ini menerapkan metode *Multimedia Development Life Cycle* (MDLC). Pendekatan tersebut dipilih karena dinilai paling sesuai untuk memandu rekayasa perangkat lunak berbasis elemen visual dan multimedia interaktif secara sistematis melewati enam tahapan yang terstruktur.

3.1. Concept

Tahap konsep merumuskan fondasi rancang bangun aplikasi simulasi tata ruang interior berbasis WebAR. Sistem didefinisikan untuk beroperasi secara penuh di sisi klien (*client-side*) melalui peramban seluler guna mengeliminasi kebutuhan instalasi perangkat lunak pihak ketiga. Fokus pengembangan dititikberatkan pada penciptaan kapabilitas pelacakan tanpa penanda (*markerless tracking*) untuk memproyeksikan perabot tiga dimensi dengan presisi tinggi. Secara spesifik, perangkat lunak ini ditargetkan bagi pengguna yang memerlukan alat bantu visual instan guna memfasilitasi eksplorasi tata letak dekorasi ruangan secara interaktif.

3.2. Design

Tahap perancangan mendefinisikan arsitektur interaksi sistem dan tata letak antarmuka pengguna. Skenario operasional disusun secara spesifik untuk memandu pengguna memindai permukaan lantai, memilih aset dari katalog visual, serta memanipulasi furnitur. Manipulasi tersebut mencakup translasi kedudukan objek melalui sentuhan satu jari, modifikasi skala dan rotasi menggunakan dua jari, hingga instruksi penghapusan berbasis komputasi pancaran sinar. Guna memaksimalkan visibilitas tangkapan kamera pada layar gawai, antarmuka dibangun dengan pendekatan minimalis memanfaatkan kelas utilitas dari kerangka kerja Tailwind CSS dan DaisyUI, memastikan area kontrol navigasi tetap responsif sekaligus ergonomis.

3.3. Material Collecting

Tahap pengumpulan material difokuskan pada penyiapan serta optimasi aset digital. Komponen visual utama berupa model tiga dimensi untuk berbagai perabot interior, meliputi meja, tempat tidur, lemari, dan elemen dekorasi. Agar proses pemuatan di peramban web berjalan efisien, seluruh model diseragamkan ke dalam format biner glTF (.glb). Ekstensi ini dipilih karena kemampuannya memadatkan data jaring geometri beserta detail tekstur menjadi satu berkas berukuran ringan. Sebagai langkah akhir, aset-aset tersebut diunggah ke layanan awan Cloudinary agar sistem dapat memanggilmnya secara dinamis.

3.4. Assembly

Fase perakitan menggabungkan rancangan logika dan aset material menjadi kode sumber yang siap dieksekusi. Struktur web dibangun menggunakan *bundler* Vite untuk memastikan manajemen modul berjalan efisien. Pada aspek visual, perenderan grafis tiga dimensi diprogram menggunakan pustaka Three.js yang mengelola pencahayaan lingkungan sekaligus mengalkulasi proyeksi bayangan pada lantai melalui penggunaan Shadow Material. Guna mengakomodasi pelacakan ruang nyata, logika program diintegrasikan dengan mesin 8th Wall. *Pipeline* khusus dari mesin ini bertugas mengekstrak fitur ruangan, mendeteksi

permukaan datar secara otomatis, serta menerjemahkan masukan gestur layar sentuh menjadi kalkulasi transformasi pada objek virtual.

3.5. Testing

Tahap pengujian difokuskan secara eksklusif pada verifikasi fungsionalitas internal melalui pengujian kotak hitam (*black-box testing*) oleh pengembang, tanpa melibatkan pengujian penerimaan pengguna eksternal. Evaluasi ini dirancang secara komprehensif untuk memvalidasi respons sistem terhadap berbagai input dan kondisi operasional melalui sepuluh skenario uji utama. Secara spesifik, pengujian memvalidasi proses inisialisasi *Augmented Reality*, kewajiban protokol keamanan HTTPS, serta kemampuan algoritma SLAM dalam mendeteksi dan mengunci koordinat lantai dasar sekaligus penolakannya terhadap permukaan non-datar seperti dinding. Selain itu, evaluasi juga menguji keandalan interaksi pengguna, termasuk mekanisme *tap* berbasis *raycasting* untuk memunculkan furnitur, gestur translasi dengan satu jari, rotasi objek menggunakan dua jari, hingga simulasi benturan objek virtual terhadap batas fisik ruangan atau dinding nyata. Untuk memastikan stabilitas aplikasi pada berbagai situasi, pengujian turut mencakup skenario operasional pada kondisi lingkungan dengan pencahayaan rendah, penanganan kesalahan pada perangkat seluler tanpa kelengkapan sensor giroskop, serta keandalan sistem saat harus memuat aset 3D dari layanan awan di bawah kondisi koneksi jaringan yang lambat.

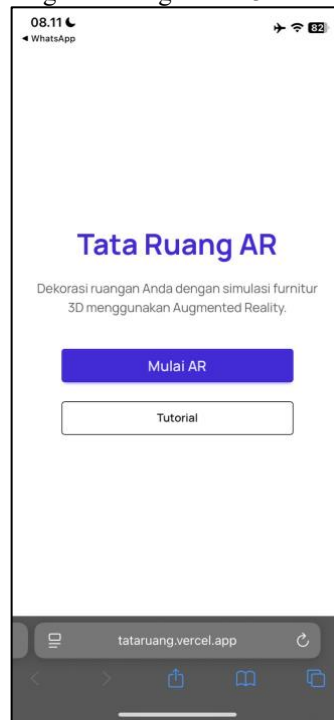
3.6. Distribution

Tahap distribusi merupakan langkah final untuk meluncurkan aplikasi ke lingkungan produksi. Berbeda dengan perangkat lunak konvensional yang bergantung pada platform toko aplikasi *native*, berkas akhir proyek WebAR ini dikompilasi melalui Vite dan didistribusikan langsung menggunakan layanan *hosting* awan. Metode ini menghasilkan tautan publik tunggal beserta kode QR. Pengguna cukup mengklik tautan atau memindai kode tersebut untuk langsung memuat pengalaman simulasi tata ruang di dalam peramban seluler mereka.

4. HASIL DAN PEMBAHASAN

4.1. Analisis Sistem

Aplikasi Tata Ruang AR telah berhasil diimplementasikan dan dapat diakses langsung melalui tautan URL peramban (Gambar 1). Antarmuka utama memvisualisasikan lingkungan tata letak ruangan secara *real-time* yang digabungkan dengan menu navigasi katalog model 3D di bagian bawah layar.



Gambar 1. Tampilan antarmuka utama aplikasi

4.2. Spesifikasi Sistem

Aplikasi ini dikompilasi dan di-*hosting* menggunakan spesifikasi teknologi yang dirangkum pada Tabel 1.

Tabel 1. Spesifikasi Teknologi Sistem

Komponen	Keterangan
Framework	Vite + Three.js v0.183.2
AR Engine	8th Wall SLAM (via CDN)
UI Library	Tailwind CSS v4 + DaisyUI v5
Bahasa	JavaScript (77%), HTML (22%), CSS (1%)
Build Tool	Vite v8
Deployment	Vercel

Karena implementasi pelacakan AR sangat bergantung pada sensor *Inertial Measurement Unit* (IMU), khususnya giroskop, untuk memetakan lingkungan, aplikasi ini memiliki batasan aksesibilitas perangkat keras. Rincian kebutuhan spesifikasi minimum perangkat seluler disajikan pada Tabel 2.

Tabel 2. Spesifikasi Perangkat Seluler

Kriteria	Spesifikasi Minimum
Sistem Operasi	iOS 14+ / Android 8.0+
Browser	Safari 14+ (iOS) / Chrome 84+ (Android)
Kamera	Kamera belakang dengan autofokus
Sensor	Giroskop + Akselerometer (IMU)
Koneksi	HTTPS wajib (syarat akses kamera browser)
RAM	Minimum 3 GB
Prosesor	ARM64 (Apple A12+ / Snapdragon 730+)

4.3. Implementasi Sistem

Bagian ini memaparkan hasil implementasi dari setiap fitur utama yang telah dikembangkan, meliputi deteksi permukaan AR dan penempatan objek 3D.

4.3.1. Deteksi Permukaan (SLAM)

Mekanisme pelacakan diimplementasikan melalui *script tag* dengan konfigurasi `data-preload-chunks="slam"` guna mempercepat inisialisasi pengenalan ruang. Tahap deteksi beroperasi stabil pada permukaan horizontal (lantai). Namun, hasil implementasi menunjukkan kelemahan di mana sistem belum mampu mengenali bidang vertikal maupun permukaan miring. Selain itu, karena ketiadaan mekanisme *collision detection*, objek virtual tidak merespons keberadaan dinding nyata dan masih dapat menembusnya.



Gambar 2. Proses deteksi permukaan menggunakan SLAM

4.3.2. Penempatan Objek 3D

Pengguna dapat memproyeksikan furnitur dan memanipulasi kedudukannya melalui input layar sentuh (Gambar 3 dan 4). Pada aspek visual, sistem merender objek menggunakan pencahayaan statis yang intensitas dan arahnya disandikan langsung (*hardcoded*) di dalam program. Oleh karena itu, bayangan dan

tingkat kecerahan perabot virtual belum dapat beradaptasi dengan kondisi cahaya lingkungan yang sebenarnya (*light estimation*).



Gambar 3. Penempatan objek 3D virtual pada permukaan



Gambar 4. Contoh visualisasi tata letak furnitur dalam Mode AR

4.4. Hasil Black-Box Testing

Evaluasi fungsionalitas sistem melalui *black-box testing* menunjukkan capaian keberhasilan pada 9 dari 10 skenario pengujian utama (Tabel 3).

Tabel 3. Hasil pengujian Black-Box pada seluruh skenario interaksi

No	Skenario Uji	Input	Output yang Diharapkan	Hasil
1	Inisialisasi AR	Buka aplikasi, arahkan kamera ke lantai	Sistem mendeteksi lantai dan menampilkan indikator SLAM siap	Sesuai
2	Penempatan objek	Tap pada area lantai terdeteksi	Objek 3D muncul di posisi tap	Sesuai
3	Memindahkan objek	Drag objek ke posisi baru	Objek berpindah sesuai arah drag	Sesuai
4	Merotasi objek	Gesture putar pada objek	Objek berputar sesuai input rotasi	Sesuai
5	Pencahayaannya rendah	AR dijalankan di ruangan gelap	Sistem tetap berjalan, objek tampil (pencahayaannya statis)	Sesuai
6	Permukaan non-datar	Kamera diarahkan ke dinding	Lantai tidak terdeteksi, tidak ada penempatan	Sesuai
7	Collision dinding	Objek dipindah melewati dinding	Objek menembus dinding (collision belum diimplementasi)	Tidak sesuai
8	Buka di peramban non-HTTPS	Akses via HTTP	Kamera ditolak browser, AR tidak berjalan	Sesuai
9	Perangkat tanpa IMU	Buka di perangkat tanpa giroskop	Halaman landing 8th Wall menampilkan pesan tidak kompatibel	Sesuai
10	Load aset dari CDN	Inisialisasi dengan koneksi lambat	Aset 3D tetap termuat, terdapat loading state	Sesuai

Berdasarkan Tabel 3, seluruh fitur navigasi, manipulasi objek, dan penanganan akses sensor berjalan sesuai harapan, kecuali pada skenario simulasi benturan (*collision*) objek terhadap dinding nyata (Skenario 7).

4.5. Pembahasan

Hasil pengujian dan implementasi secara keseluruhan membuktikan bahwa aplikasi Tata Ruang AR telah berhasil mencapai tujuan utamanya sebagai media visualisasi penataan ruang yang praktis, stabil, dan imersif. Pendekatan *Web-based Augmented Reality* (WebAR) yang diterapkan sukses mendobrak batasan aksesibilitas aplikasi *native*, di mana pengguna terbukti dapat memuat simulasi AR secara instan langsung melalui peramban seluler mereka tanpa hambatan instalasi. Antarmuka sistem yang dibangun dengan kerangka kerja Tailwind CSS dan DaisyUI juga tampil sangat responsif, menyajikan tata letak visual navigasi yang konsisten di berbagai ukuran layar perangkat.

Dari segi fungsionalitas teknis, integrasi antara kerangka kerja Three.js dan mesin pelacak 8th Wall (SLAM) menunjukkan performa yang sangat memuaskan. Seperti yang didemonstrasikan pada visual simulasi penempatan objek, sistem secara akurat dan cepat mampu mendeteksi bidang datar horizontal pada lantai ruangan fisik dan menjadikannya sebagai titik jangkar (*origin*) yang sangat stabil untuk memproyeksikan furnitur 3D secara *real-time*. Keberhasilan interaksi ini didukung kuat oleh data pengujian *black-box*, di mana aplikasi terbukti sukses menjalankan 9 dari 10 skenario fungsionalitas utama dengan sempurna. Pengguna dapat dengan lancar menempatkan perabot virtual, melakukan translasi posisi melalui usapan satu jari, hingga merotasi objek secara presisi, yang keseluruhannya mampu dirender tanpa jeda yang berarti. Selain itu, strategi manajemen aset menggunakan ekosistem *Content Delivery Network* (CDN) eksternal terbukti sangat efektif dalam mendistribusikan model 3D berkualitas tinggi sambil tetap menjaga ukuran aplikasi agar tetap ringan dan cepat dimuat.

Meskipun sistem telah menunjukkan keandalan tingkat tinggi dalam memfasilitasi pengalaman dekorasi yang utuh, terdapat peluang kecil untuk penyempurnaan di masa mendatang guna mencapai tingkat realisme absolut. Berdasarkan satu-satunya skenario uji yang belum terpenuhi, integrasi fitur *collision detection* lanjutan dapat dipertimbangkan pada iterasi berikutnya agar objek virtual mampu merespons batas fisik seperti dinding secara natural. Selain itu, transisi dari pencahayaan statis menuju komputasi *light estimation* dinamis juga dapat menjadi nilai tambah untuk mengadaptasi bayangan objek dengan kondisi cahaya nyata. Terlepas dari ruang pengembangan tersebut, aplikasi ini telah secara meyakinkan berhasil menawarkan solusi *markerless* AR yang andal, sangat mudah diakses, dan siap memfasilitasi kebutuhan desain interior pengguna secara optimal.

5. KESIMPULAN

Berdasarkan hasil implementasi dan pengujian yang telah dilakukan, dapat disimpulkan bahwa aplikasi Tata Ruang AR berhasil dikembangkan sebagai media visualisasi tata letak ruangan berbasis Augmented Reality yang dapat diakses langsung melalui peramban pada perangkat mobile. Pemanfaatan Three.js sebagai rendering engine dan 8th Wall Engine SLAM sebagai teknologi pelacakan lingkungan memungkinkan pengguna menempatkan objek 3D secara real-time tanpa memerlukan instalasi aplikasi native.

Teknologi SLAM yang diterapkan mampu mendeteksi permukaan lantai horizontal secara stabil dan menjadikannya sebagai bidang referensi untuk penempatan objek virtual. Selain itu, penggunaan aset 3D yang di-hosting melalui CDN berhasil menjaga ukuran repositori proyek tetap ringan sekaligus mempertahankan kualitas visual objek yang ditampilkan kepada pengguna.

Hasil pengujian black box menunjukkan bahwa sistem telah berfungsi sesuai dengan spesifikasi yang dirancang. Dari sepuluh skenario pengujian yang dilakukan, sembilan skenario dinyatakan berhasil, sedangkan satu skenario terkait collision detection pada dinding belum memenuhi hasil yang diharapkan. Secara keseluruhan, sistem telah mampu memenuhi kebutuhan fungsional utama dan memberikan pengalaman pengguna yang baik dalam melakukan simulasi penataan ruang menggunakan teknologi Augmented Reality.

Meskipun sistem telah berjalan dengan baik, masih terdapat beberapa aspek yang dapat dikembangkan untuk meningkatkan kualitas dan realisme aplikasi. Salah satu pengembangan yang disarankan adalah penerapan mekanisme collision detection yang lebih akurat menggunakan teknik seperti raycasting atau bounding volume, sehingga objek virtual tidak dapat menembus dinding maupun batas fisik ruangan.

Selain itu, sistem pencahayaan dapat ditingkatkan melalui integrasi teknologi light estimation agar pencahayaan objek 3D mampu menyesuaikan kondisi lingkungan nyata secara dinamis. Pengembangan kemampuan deteksi bidang juga perlu dipertimbangkan sehingga sistem tidak hanya mengenali lantai, tetapi juga dapat mendeteksi permukaan lain seperti meja, rak, dan dinding untuk mendukung penempatan objek yang lebih fleksibel.

Di sisi performa, proses pemuatan aset 3D dapat dioptimalkan melalui penerapan teknik lazy loading atau progressive loading. Pendekatan tersebut diharapkan dapat mengurangi waktu tunggu pengguna, terutama pada kondisi jaringan yang lambat, sekaligus meningkatkan kenyamanan penggunaan aplikasi secara keseluruhan. Dengan berbagai pengembangan tersebut, aplikasi Tata Ruang AR berpotensi memberikan pengalaman visualisasi ruang yang lebih realistis, akurat, dan interaktif pada masa mendatang.

REFERENSI

- [1] S. Revathy, A. Harini, and S. Sruthika, "Augmented Reality in Interior Design," *Journal of Innovative Image Processing*, vol. 6, no. 3, pp. 305–313, Aug. 2024, doi: 10.36548/jiip.2024.3.007.
- [2] 3D Cloud and Provoke Insights, "3D Cloud Furniture Shopping Trends Study 2026," 3D Cloud, 2026. [Online]. Available: <https://3dcloud.com/3d-cloud-furniture-shopping-trends-study/>
- [3] R. Liu, B. Balakrishnan, and E. M. Saari, "The Impact of Augmented Reality (AR) Technology on Consumers' Purchasing Decision Processes," *Frontiers in Business, Economics and Management*, vol. 13, no. 2, pp. 181–185, 2024.
- [4] Y. Sahria, B. Santoso, and T. Kuat, "Innovative Learning Media for Interior Design using Augmented Reality at Vocational High Schools," *IOTA*, vol. 4, no. 2, 2024, doi: 10.31763/iota.v4i2.724.
- [5] E. Murphy-Chutorian, "How We Engineered AR for the Mobile Browser with 8th Wall Web," *8th Wall Engineering Blog*, 8th Wall, 2019. [Online]. Available: <https://www.8thwall.com/blog/post/41173258896>
- [6] 8th Wall, "8th Wall Products," *8th Wall*, Niantic, 2023. [Online]. Available: <https://www.8thwall.com/products>
- [7] G. T. Suryasamudra, V. Valenthresia, N. F. B. M. Idres, and G. Wang, "Utilizing Augmented Reality to Improve Customers' Purchase Intention in Retail Furniture Applications in Indonesia," in *2024 2nd International Conference on Technology Innovation and Its Applications (ICTIIA)*, 2024.
- [8] K. Sato, R. Sasage, T. Soejima, N. Kamioka, and Y. Kawai, "Development of Augmented Reality-Based Application to Search for Electric Appliances and Furniture," in *2021 IEEE 3rd Global Conference on Life Sciences and Technologies (LifeTech)*, 2021.
- [9] F. F. Hidayat, M. Stefano, S. Adam, and F. I. Maulana, "Co-Designing Immersive Eco-Centric Living Spaces: Augmented Reality Integration in Furniture Retailing (AR-InFuRe)," in *2024 International Conference on Informatics, Multimedia, Cyber and Information System (ICIMCIS)*, 2024.
- [10] Y. Pei, Y. Wu, H. Jiang, S. Wang, S. Xu, F. Wang, and J. Zhou, "WA VIS: A Web-based Augmented Reality Text Data Visual Analysis Tool," in *2019 International Conference on Virtual Reality and Visualization (ICVRV)*, 2019.
- [11] Z. Li, W. Song, and L. Yang, "Design and implementation of traditional kite art platform based on human-computer interaction and WebAR technology," in *2021 2nd International Conference on Intelligent Computing and Human-Computer Interaction (ICHCI)*, 2021.
- [12] J. Motsch, S. Benammar, and Y. Bergeon, "Interior Mapping of a Building: a Real-life experiment with Microsoft Kinect for Windows v1 and RGBD-SLAM," in *2017 International Conference on Military Technologies (ICMT)*, 2017.
- [13] S. Sharma, Y. Kaikini, P. Bhodia, and S. Vaidya, "Markerless Augmented Reality based Interior Designing System," in *2018 International Conference on Smart City and Emerging Technology (ICSCET)*, 2018.
- [14] S. Sridhar and S. Sanagavarapu, "Instant Tracking-Based Approach for Interior Décor Planning with Markerless AR," in *2020 Zooming Innovation in Consumer Technologies Conference*, 2020.
- [15] N. R. R., R. M., and S. Sultana, "Markerless Augmented Reality Application for Interior Designing," in *2022 Second International Conference on Advanced Technologies in Intelligent Control, Environment, Computing & Communication Engineering (ICATIECE)*, 2022.
- [16] P. Gouthami, S. Darun, and K. Rathivarman, "DecoraAI: AI-powered Augmented Reality (AR) Interior Design Web Application," in *2026 International Conference on Emerging Technologies and Future Innovations (ETFI)*, 2026.