

Implementasi Teknik Visualisasi Dinamis pada Augmented Reality untuk Representasi Operasi Vektor

*Implementation of Dynamic Visualization Techniques in Augmented Reality to Enhance
Spatial Representation of Vector Operations*

Triana Harmini¹, Oddy Virgantara Putra², Hibatullah Agung Prihadian³

*Program Studi Teknik Informatika, Fakultas Sains dan Teknologi, Universitas Darussalam
Gontor, Ponorogo, Indonesia*

E-mail : triana@unida.gontor.ac.id¹, oddy@unida.gontor.ac.id²,

hibatullahagungprihadian10@student.cs.unida.gontor.ac.id³

Received 27 October 2025; Revised 22 November 2025; Accepted 24 November 2025

Abstrak

Konsep vektor, khususnya terkait arah dan transformasi besaran dalam ruang 3D, menuntut kemampuan visualisasi spasial yang tinggi. Media pembelajaran konvensional dan aplikasi AR statis seringkali memiliki keterbatasan dalam merepresentasikan proses terjadinya operasi vektor secara nyata. Penelitian ini bertujuan mengimplementasikan teknik Visualisasi Vektor Dinamis (*Dynamic Vector Visualization*) berbasis *Augmented Reality* (AR) untuk mensimulasikan perilaku vektor secara *real-time*. Sistem dikembangkan menggunakan *engine* Unity 3D dengan menerapkan arsitektur Visualisasi Berbasis Status (*State-Driven Visualization*). Metode ini mengintegrasikan aset adaptif dari Blender (*Keyframe Animation*) dengan logika kontrol *Finite State Machine* (FSM) untuk menjamin presisi geometris dan efisiensi komputasi pada perangkat *mobile*. Validasi dilakukan oleh ahli materi dan media, serta uji *usability* terhadap 30 mahasiswa. Hasil pengujian menunjukkan bahwa teknik visualisasi dinamis memperoleh skor validasi ahli materi sebesar 86% dan ahli media sebesar 95%. Uji penerimaan pengguna menghasilkan skor *System Usability Scale* (SUS) sebesar 92,7 yang masuk dalam kategori "*Best Imaginable*". Temuan ini mengonfirmasi bahwa integrasi mekanisme animasi prosedural dalam lingkungan AR terbukti efektif dalam membantu mahasiswa memvisualisasikan konsep abstrak operasi vektor dibandingkan representasi statis.

Kata Kunci - *Augmented Reality*; Visualisasi Dinamis; Vektor; *Finite State Machine*; Unity 3D.

Abstract

Vector concepts, particularly regarding direction and magnitude transformation in 3D space, demand high spatial visualization skills. Conventional learning media and static AR applications often have limitations in representing the process of vector operations realistically. This study aims to implement Dynamic Vector Visualization techniques based on Augmented Reality (AR) to simulate vector behavior in real-time. The system was developed using the Unity 3D engine by applying a State-Driven Visualization architecture. This method integrates adaptive assets from Blender (Keyframe Animation) with Finite State Machine (FSM) control logic to ensure geometric precision and computational efficiency on mobile devices. Validation was conducted by material and media experts, along with usability testing on 30 undergraduate students. The results showed that the dynamic visualization technique achieved a material expert validation score of 86% and a media expert score of 95%. User acceptance testing yielded a System Usability Scale (SUS) score of 92.7, categorized as "Best Imaginable." These findings confirm that the integration of procedural animation mechanisms within an AR environment is proven effective in assisting students to visualize abstract vector operation concepts compared to static representations.

Keywords - *Augmented Reality*; *Dynamic Visualization*; *Vector*; *Finite State Machine*; Unity 3D.

1. PENDAHULUAN

Perkembangan teknologi informasi dalam beberapa dekade terakhir telah membawa dampak yang signifikan dalam berbagai bidang kehidupan, termasuk dunia pendidikan. Teknologi informasi tidak hanya berfungsi sebagai alat bantu komunikasi, tetapi juga menjadi sarana utama dalam penyampaian dan pertukaran ilmu pengetahuan [1]. Pemanfaatan teknologi ini memungkinkan penyajian materi pembelajaran yang lebih dinamis, interaktif, dan dapat diakses kapan saja dan di mana saja [2]. Transformasi ini mendorong terciptanya metode pembelajaran baru yang lebih adaptif terhadap kebutuhan generasi digital, serta mempercepat proses penguasaan konsep-konsep kompleks yang sebelumnya sulit dipahami melalui metode konvensional. Dengan demikian, integrasi teknologi informasi dalam pendidikan menjadi suatu kebutuhan mutlak dalam menghadapi tantangan abad ke-21 [3].

Pembelajaran matematika terutama materi vektor sering dipersepsikan abstrak dan menantang karena menuntut kemampuan visualisasi ruang dan penalaran logis [4]. Materi vektor menuntut kemampuan visualisasi arah dan besar dalam ruang tiga dimensi; pendekatan konvensional seperti papan tulis, penjelasan verbal, serta gambar statis seringkali belum memadai untuk membantu pemahaman konsep yang bersifat abstrak [5].

Salah satu pendekatan yang relevan adalah memanfaatkan teknologi seperti *Augmented Reality*, yang tidak hanya menawarkan pengalaman belajar yang interaktif tetapi juga mampu mengakomodasi berbagai gaya belajar siswa [6]. Teknologi *Augmented Reality* (AR) menawarkan jalan tengah yang kuat: materi abstrak dapat “diangkat” ke dunia nyata sehingga mahasiswa dapat melihat dan memanipulasi objek vektor secara langsung dalam ruang 3D [7]. Definisi klasik AR menekankan integrasi dunia nyata virtual, interaksi real-time, dan pendaftaran spasial 3D tiga ciri yang relevan untuk pembelajaran vektor [8].

Penggunaan *Augmented Reality* (AR) sebagai media pembelajaran matematika telah terbukti efektif dan mendapat respons positif, khususnya untuk topik-topik yang cenderung abstrak. Studi yang dilakukan oleh [4] mengenai jaring-jaring kubus dan balok menegaskan kelayakan tinggi AR untuk visualisasi, sejalan dengan temuan [6] yang menunjukkan potensi AR dalam meningkatkan keterlibatan dan pemahaman konsep melalui visualisasi 3D. Dalam konteks implementasi teknis, pendekatan marker-based telah berhasil diterapkan pada berbagai materi, mulai dari bangun ruang [9] hingga sel biologi [10]. Keuntungan *marker-based* adalah kemampuannya mempertahankan skala dan posisi objek 3D secara konsisten mengikuti ukuran fisik *marker*, yang sangat penting untuk materi Vektor yang membutuhkan presisi tinggi dalam demonstrasi panjang dan arah. Pendekatan ini juga dipilih karena terbukti mudah diimplementasikan dan kompatibel dengan berbagai perangkat Android tanpa bergantung pada hardware Gyroscope [11], [12].

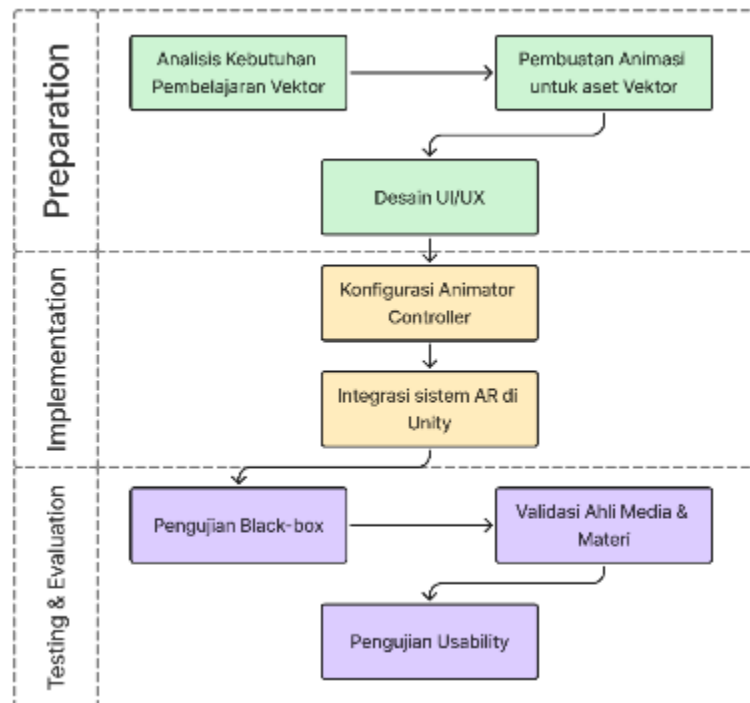
Meskipun AR memberikan visualisasi 3D yang unggul, penting untuk membandingkannya dengan pendekatan non-AR. Aplikasi seperti GeoGebra [5] dan media animasi Animaker [13] celah penelitian yang substansial masih ditemukan pada aspek metode representasi. Mayoritas penelitian AR terdahulu umumnya hanya menampilkan objek 3D secara statis (diam) untuk materi geometri bangun ruang. Padahal, konsep vektor menuntut pemahaman tentang proses perubahan (transformasi), seperti bagaimana vektor memanjang saat dikalikan skalar atau bagaimana resultan terbentuk dari pergeseran posisi. Belum banyak penelitian yang mengeksplorasi penerapan mekanisme animasi dinamis dalam AR untuk memvisualisasikan perilaku operasi vektor secara *real-time* guna mengurangi beban kognitif mahasiswa.

Merespons kebutuhan representasi tersebut, tujuan utama penelitian ini adalah mengimplementasikan teknik Visualisasi Vektor Dinamis (*Dynamic Vector Visualization*) pada media AR berbasis Android. Secara spesifik, penelitian ini menerapkan arsitektur Visualisasi Berbasis Status (*State-Driven Visualization*) menggunakan *Finite State Machine* (FSM) untuk mensimulasikan transisi operasi vektor seperti perubahan besaran dan dekomposisi komponen secara akurat dan efisien.

Ruang lingkup penelitian dibatasi pada platform Android dengan materi Aljabar Linear. Pengembangan sistem memanfaatkan integrasi Blender 3D untuk perancangan aset adaptif dan Unity 3D untuk logika kontrol animasi. Kontribusi keilmuan utama dari penelitian ini adalah penerapan desain arsitektur *State-Driven Animation* dalam lingkungan AR yang mampu merepresentasikan operasi matematis abstrak menjadi simulasi visual konkret. Penelitian ini memberikan alternatif solusi teknis untuk menghadirkan visualisasi geometris yang presisi namun tetap efisien secara komputasi pada perangkat *mobile*.

2. METODE PENELITIAN

Penelitian ini dilaksanakan menggunakan pendekatan pengembangan sistematis yang terbagi menjadi tiga tahapan utama: Preparation (Persiapan), Implementation (Implementasi), dan Testing & Evaluation (Pengujian dan Evaluasi). Alur penelitian dirancang untuk menghasilkan framework visualisasi dinamis yang teruji validitas dan efektivitasnya. Tahapan rinci penelitian ditunjukkan pada Gambar 1.



Gambar 1 Alur Penelitian

Penelitian ini dilaksanakan mengikuti alur pengembangan sistematis yang divisualisasikan pada Gambar 1. Proses dimulai dengan Tahap Persiapan (*Preparation*), di mana fokus utama kegiatan adalah melakukan analisis kebutuhan sistem untuk mengidentifikasi kendala visualisasi pada materi vektor, seperti representasi arah, besaran, dan operasi resultan. Pada tahap ini, dilakukan pula perancangan aset vektor adaptif menggunakan perangkat lunak Blender 3D. Teknik *Keyframe Animation* disiapkan pada tahap pemodelan ini untuk menangani mekanisme perubahan besaran vektor agar proporsi geometris tetap terjaga saat di-*render*. Selain aset 3D, desain antarmuka pengguna (UI/UX) dan elemen visual 2D juga dirancang pada fase ini untuk memastikan navigasi aplikasi yang intuitif.

Tahap selanjutnya adalah Tahap Implementasi (*Implementation*), yang difokuskan pada penerapan teknik Visualisasi Vektor Dinamis (*Dynamic Vector Visualization*). Pada tahap ini, logika kontrol animasi dibangun menggunakan arsitektur Visualisasi Berbasis Status (*State-Driven Visualization*) di dalam *Unity Animator Controller*. Mekanisme ini dirancang secara spesifik untuk mengatur transisi antar-kondisi vektor—mulai dari kondisi diam (*Idle*) hingga

simulasi pergerakan operasi seperti transformasi besaran dan dekomposisi komponen—secara *real-time* berdasarkan interaksi pengguna. Aset animasi adaptif yang telah dikonfigurasi logikanya kemudian diintegrasikan dengan Vuforia SDK, di mana *marker* fisik ditetapkan sebagai titik referensi koordinat dunia (*World Center*) untuk menjamin bahwa visualisasi dinamis tersebut tetap stabil dan akurat secara geometris saat diproyeksikan ke ruang nyata.

Rangkaian penelitian diakhiri dengan Tahap Pengujian dan Evaluasi (*Testing & Evaluation*) untuk memverifikasi kinerja dan kebergunaan sistem. Pengujian diawali dengan metode *Black Box* untuk memastikan fungsi logika transisi animasi berjalan sesuai input tanpa kesalahan sistem (*bug*). Setelah fungsionalitas terverifikasi, dilakukan validasi ahli yang melibatkan ahli materi dan ahli media untuk menilai kelayakan konten serta kualitas visualisasi. Terakhir, pengujian usability dilakukan terhadap mahasiswa menggunakan metode *System Usability Scale* (SUS). Pengujian ini bertujuan untuk mengukur tingkat kegunaan, efisiensi, dan kepuasan pengguna terhadap antarmuka visualisasi dinamis yang dikembangkan dalam skenario penggunaan nyata.

Evaluasi kelayakan sistem dilakukan menggunakan instrumen kuesioner dengan skala Likert. Data kuantitatif yang diperoleh dari validasi ahli dan uji pengguna dianalisis menggunakan teknik persentase kelayakan untuk mendapatkan nilai akhir yang ternormalisasi (skala 0-100). Rumus yang digunakan adalah sebagai berikut:

$$NP = \frac{R}{SM} \times 100 \quad (1)$$

Keterangan :

NP : Nilai Persentase (*Final Score*)

R : Rekapitulasi Skor Perolehan (*Jumlah jawaban responden*)

SM : Skor Maksimal ($N \times \text{Jumlah Soal} \times \text{Skor Maksimal Butir}$)

Hasil perhitungan nilai akhir (*NP*) kemudian dikonversikan ke dalam kriteria kelayakan. Khusus untuk uji pengguna, nilai akhir diinterpretasikan menggunakan grafik standar *System Usability Scale* (SUS) untuk menentukan predikat penerimaan sistem (*Adjective Rating*).

3. HASIL DAN PEMBAHASAN

3.1. Preparation

3.1.1. Kebutuhan Perangkat Keras dan Lunak.

Kebutuhan perangkat keras yang digunakan dalam merancang aplikasi ini ditunjukkan dalam tabel 1 berikut :

Tabel 1 Perangkat Keras Laptop

Perangkat Keras	Spesifikasi
Prosesor	AMD Ryzen 7 5800H
RAM	24 GB
GPU	Nvidia Geforce RTX 3060
Penyimpanan	1 TB
Resolusi	1920 x 1080 pixel

Kebutuhan perangkat uji (smartphone) yang digunakan untuk uji coba aplikasi ini ditunjukkan dalam tabel 2 berikut :

Tabel 2 Perangkat Keras Smartphone

Perangkat Keras	Spesifikasi
Brand	Xiaomi 12 Lite
RAM	8 GB
Penyimpanan	256 GB
OS	Android 14

Spesifikasi perangkat uji pada Tabel 2 dipilih untuk merepresentasikan perangkat standar yang dimiliki rata-rata mahasiswa, guna memastikan teknik visualisasi dinamis dapat berjalan optimal tanpa lag pada perangkat mid-range.

Adapun kebutuhan perangkat lunak yang digunakan dalam membangun aplikasi ini ditunjukkan dalam tabel 3 berikut :

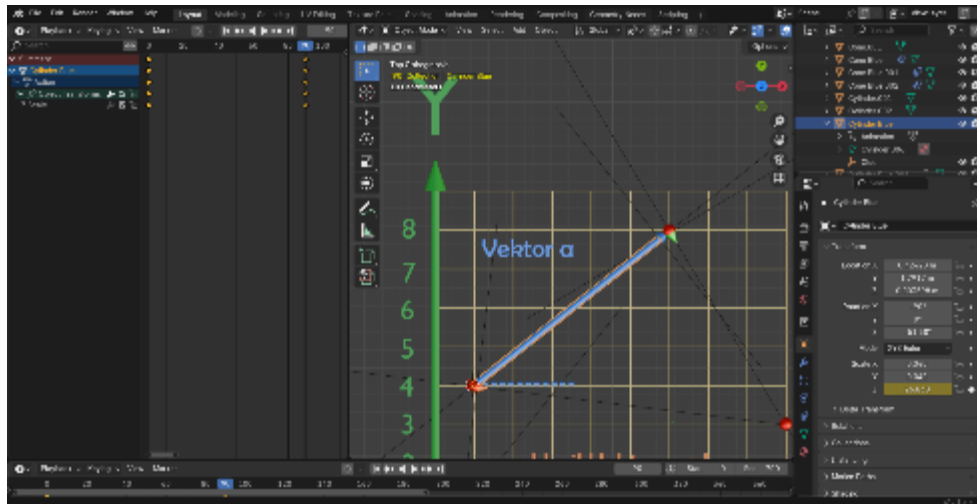
Tabel 3 Versi Perangkat Lunak

Perangkat Lunak	Versi
OS	Windows 11 24H2
Unity Hub	3.15.2 Student Version
Blender	4.4.3

Tujuan analisis ini adalah memastikan lingkungan pengembangan mampu menangani proses rendering animasi 3D dan kompilasi AR secara efisien.

3.1.2. Pembuatan Objek 3D Vektor (Perancangan Aset Adaptif)

Proses pemodelan menggunakan perangkat lunak Blender 3D difokuskan pada pembuatan aset vektor yang mendukung animasi perubahan besaran (*magnitude scaling*). Berbeda dengan pemodelan statis biasa, objek vektor dirancang dengan teknik *Keyframe Animation*.



Gambar 2 Perancangan Animasi Vektor di Aplikasi Blender

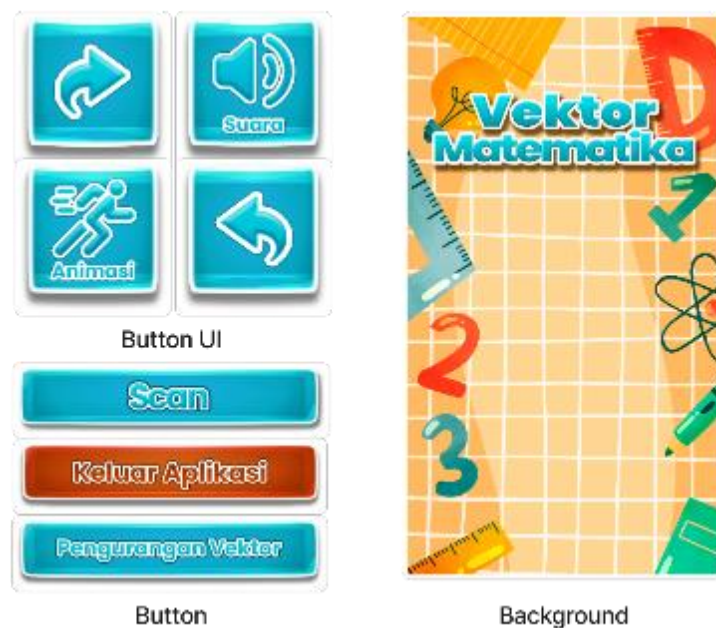
Seperti diperlihatkan pada Gambar 2, proses perancangan aset dilakukan di lingkungan Blender (versi 4.4.3). Fokus utama pada tahap ini adalah penerapan teknik *Keyframe Animation* untuk mendefinisikan perilaku vektor.

Berbeda dengan pemodelan 3D statis konvensional, aset vektor ini dirancang sebagai objek modular yang terdiri dari komponen body (Cylinder) dan head (Cone). Tantangan teknis utama dalam memvisualisasikan perubahan panjang vektor adalah mempertahankan rasio aspek

kepala panah (arrowhead) agar tidak mengalami distorsi visual (menjadi pipih atau lonjong) saat badan vektor memanjang.

Untuk mengatasi hal tersebut, penelitian ini menerapkan strategi animasi berbasis Transformasi Parsial. Pada timeline animasi (terlihat pada bagian kiri Gambar 2), dilakukan perekaman keyframe pada rentang frame 0 hingga 90. Logika animasi dirancang sedemikian rupa sehingga ketika badan vektor mengalami scaling pada sumbu longitudinal, posisi kepala panah akan mengalami translasi mengikuti ujung badan, tanpa ikut mengalami perubahan skala. Aset yang telah memuat data animasi ini kemudian diekspor (baked) ke dalam format .fbx agar dapat dibaca oleh *Unity Animator Controller* sebagai klip animasi yang utuh.

3.1.3. Pembuatan Aset 2D



Gambar 3 Aset UI/UX

Selain aset 3D, elemen antarmuka pengguna (*User Interface*) dirancang untuk mendukung interaksi kontrol animasi. Aset visual seperti tombol animasi, panel background dan ikon navigasi dibuat menggunakan perangkat lunak desain grafis Canva. Desain UI menerapkan prinsip minimalis dengan kontras warna yang jelas agar tombol navigasi mudah diakses oleh pengguna saat kamera AR sedang aktif. Seluruh aset 2D diekspor dalam format .png dengan latar belakang transparan (*transparent background*) untuk diintegrasikan ke dalam Unity.

3.2. Implementation

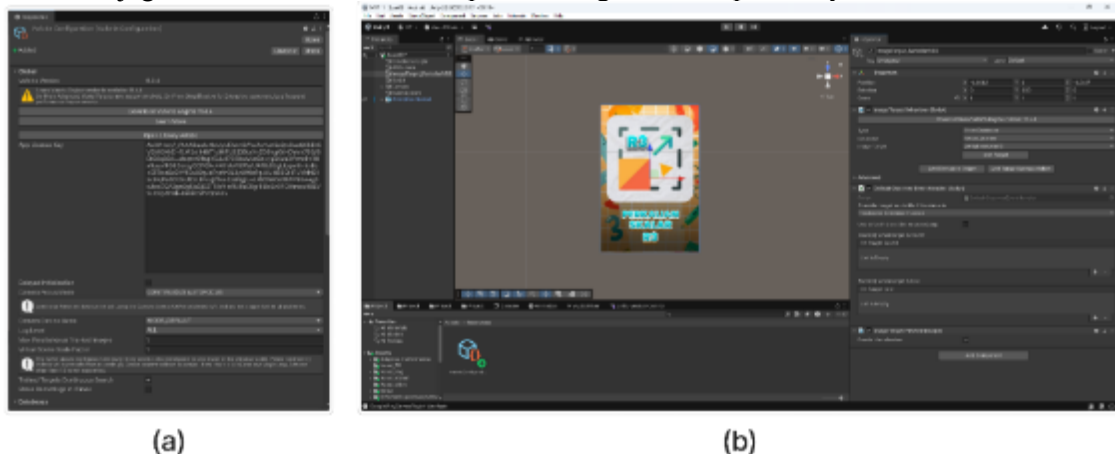
Tahap implementasi difokuskan pada konstruksi lingkungan *Augmented Reality* (AR) dan integrasi logika visualisasi ke dalam *engine* Unity 3D. Proses ini dilakukan secara bertahap mulai dari inisialisasi sesi AR, konfigurasi pelacakan marker, hingga integrasi aset animasi.

3.2.1. Inisiasi Lingkungan AR di Unity

Pengembangan logika sistem dilakukan dengan mengintegrasikan dua perangkat lunak utama: Unity 3D dan Vuforia SDK. Unity 3D berfungsi sebagai *engine* utama yang menangani

rendering grafis dan logika pemrograman C# [14], sementara Vuforia SDK berperan sebagai pustaka *Computer Vision* yang memungkinkan kamera perangkat mendeteksi *marker* fisik secara *real-time* [15]. Sinergi kedua *tools* ini memungkinkan penggabungan objek virtual dinamis dengan lingkungan nyata secara presisi.

Pada Gambar 4 (a) ditunjukkan langkah awal pengembangan dimulai dengan membangun *AR Session Origin*. Komponen ini dikonfigurasi menggunakan *AR Camera* yang berfungsi sebagai mata virtual untuk menangkap *feed* video dari kamera perangkat fisik dan menyinkronkannya dengan koordinat dunia virtual di Unity. Pada tahap ini, skrip manajemen fokus kamera (*Camera Focus Mode*) diaktifkan ke mode *CONTINUOUS_AUTO* untuk memastikan objek tetap tajam saat pengguna mengubah jarak pandang. Konfigurasi ini krusial untuk menjaga stabilitas deteksi marker pada berbagai kondisi pencahayaan.

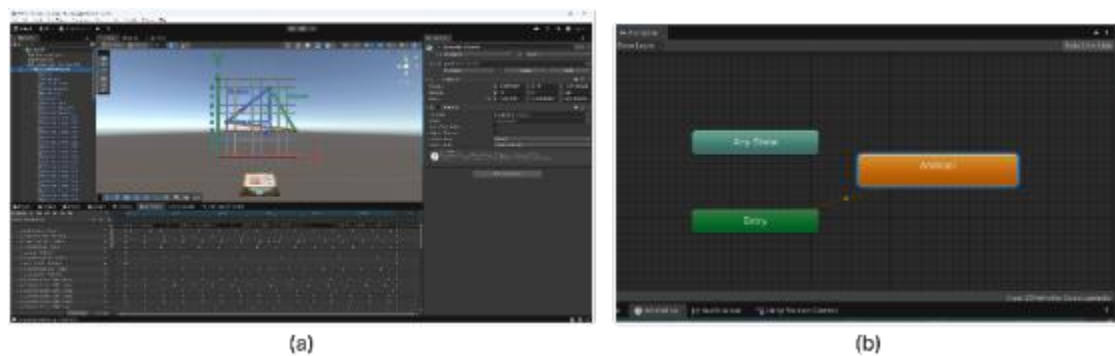


Gambar 4 Implementasi Marker-Based

Penentuan referensi spasial dilakukan menggunakan Vuforia Engine. Sebuah *Database Image Target* dibuat dengan mengunggah desain marker fisik ke dalam *Vuforia Developer Portal*. Seperti terlihat pada Gambar 4 (b), *Image Target* dikonfigurasi sebagai titik pusat koordinat dunia (*World Center Mode: DEVICE*). Sistem mengekstraksi fitur kontras tinggi (*feature points*) dari gambar marker untuk dijadikan jangkar (*anchor*). Hal ini memastikan bahwa sumbu x, y, z virtual akan selalu terkunci presisi mengikuti orientasi fisik marker, mencegah terjadinya pergeseran (*jitter*) pada visualisasi vektor nantinya.

3.2.2. Implementasi Animasi ke Unity

Setelah sistem *marker* siap, aset vektor adaptif yang telah dibuat di Blender diintegrasikan sebagai *child object* dari *Image Target*. Tahap ini berfokus pada penggabungan data animasi yang telah dibuat di Blender ke dalam arsitektur logika Unity melalui mekanisme verifikasi data *keyframe* dan pemetaan *state*. Proses diawali dengan mengimpor aset vektor berformat .fbx ke dalam *Unity Asset Pipeline*, di mana *engine* menerjemahkan data animasi yang "dibakar" (*baked animation*) menjadi *Animation Clips*. Validasi data dilakukan menggunakan jendela *Animation Dopesheet* sebagaimana ditunjukkan pada Gambar 5 (a). Pada tampilan tersebut, titik-titik *diamond* di sepanjang *timeline* merepresentasikan *keyframe* yang memuat koordinat transformasi posisi, rotasi, dan skala. Verifikasi visual ini krusial untuk memastikan bahwa interpolasi gerak yang dibaca oleh Unity berjalan presisi sesuai desain awal dan bebas dari distorsi atau *glitch* visual sebelum dipetakan ke dalam logika kontroler.

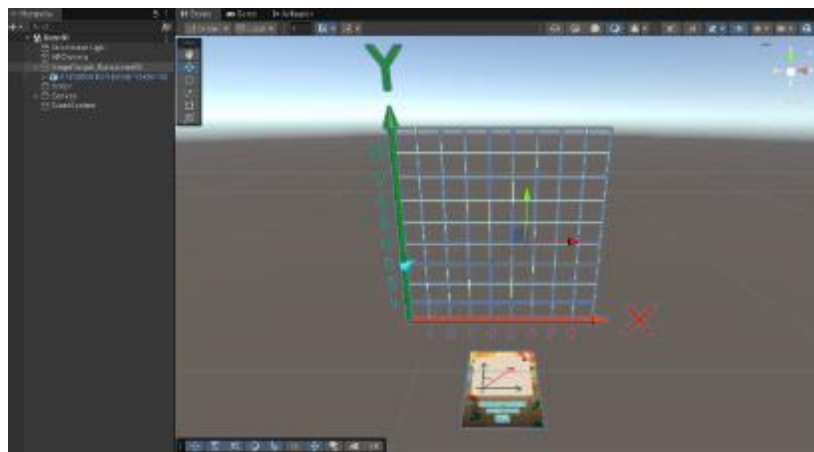


Gambar 5 Verifikasi Keyframe (a), Konfigurasi Default State pada Animation Controller (b)

Setelah integritas data animasi terverifikasi, klip tersebut dipetakan ke dalam Unity Animator Controller menggunakan arsitektur Eksekusi Linear Otomatis. Mengacu pada konfigurasi yang ditampilkan pada Gambar 5 (b), sistem menetapkan klip animasi utama sebagai *Default State* (ditandai dengan node berwarna oranye). Logika transisi dirancang menggunakan alur langsung (*Direct Transition*) dari *Entry Node* menuju *Animasi Node* tanpa memerlukan parameter persyaratan yang kompleks. Konfigurasi ini mengadopsi mekanisme pemicu berbasis kejadian (*Event-Based*), di mana deteksi *Image Target* oleh kamera Vuforia bertindak sebagai *trigger* eksternal yang secara otomatis menjalankan alur animasi. Pendekatan linear ini dipilih untuk meminimalisir latensi sistem, memberikan respons visual seketika (*zero-delay*) kepada pengguna saat materi ajar dipindai.

3.2.3. Konfigurasi Pelacakan AR (*Marker-Based Tracking*)

Setelah logika animasi berhasil dikonfigurasi, tahap selanjutnya adalah mengintegrasikan Objek Vektor dengan wadah visualisasi. Penentuan referensi spasial dilakukan menggunakan Vuforia Engine dengan mengonfigurasi *Image Target* sebagai acuan koordinat dunia (*World Center*).

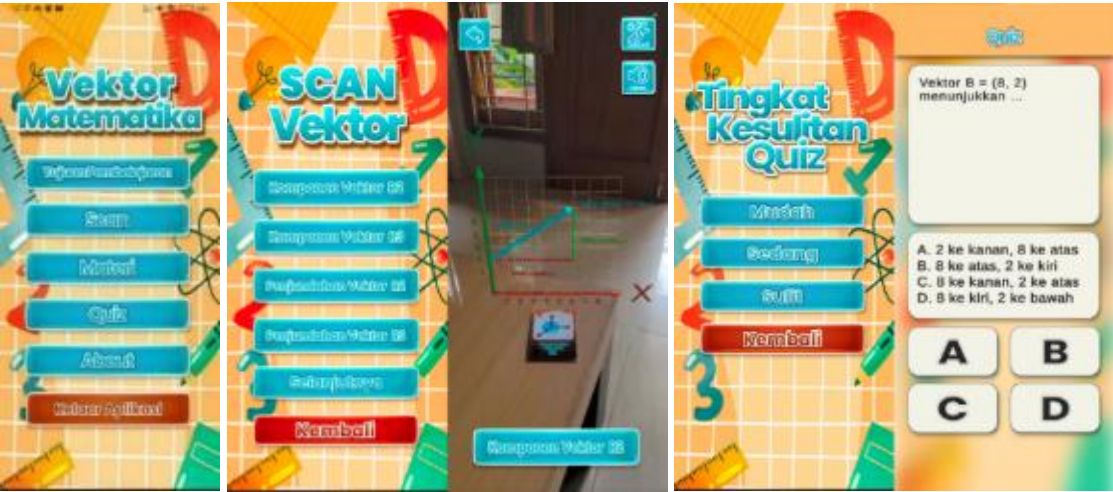


Gambar 6 Integrasi Objek Vektor pada Image Target

Mengacu pada Gambar 6, objek vektor yang telah memiliki kontrol animasi ditempatkan sebagai *child object* dari *Image Target*. Sistem dikonfigurasi menggunakan mode *DEVICE_TRACKING*, di mana Vuforia mengekstraksi fitur kontras tinggi pada marker fisik untuk dijadikan jangkar (*anchor*) koordinat (0,0,0). Konfigurasi ini memastikan bahwa sumbu x, y, z pada animasi vektor terkunci secara presisi mengikuti orientasi fisik marker, mencegah terjadinya pergeseran (*jitter*) atau *drift* yang dapat mengganggu akurasi representasi matematis saat animasi dimainkan.

3.2.4. Pengembangan Antarmuka Aplikasi (*Interface Assembly*)

Tahap akhir implementasi adalah penyusunan antarmuka pengguna (*User Interface*) dan manajemen *scene*. Struktur aplikasi dibangun menggunakan *Unity Scene Management* yang membagi aplikasi menjadi tiga modul utama: (1) Menu Utama sebagai navigasi pusat, (2) Modul AR untuk visualisasi materi, dan (3) Modul Evaluasi untuk kuis.



Gambar 7 Antarmuka Aplikasi Hasil Pengembangan

Seperti diperlihatkan pada Gambar 7, desain antarmuka disusun untuk memfasilitasi akses cepat ke fitur visualisasi. Tombol navigasi (seperti "Materi" dan "Scan AR") diimplementasikan menggunakan komponen *Unity UI Canvas* dengan *event system* yang responsif. Setelah seluruh logika dan aset terintegrasi, proyek dikompilasi (*build*) menjadi format .apk dengan target minimum Android 10 (API Level 29) untuk memastikan kompatibilitas pada perangkat *mobile* modern.

3.3. *Testing & Evaluation*

3.3.1. Pengujian Black-Box

Pengujian *black box* difokuskan untuk memverifikasi logika transisi pada *Finite State Machine* (FSM) dan responsivitas sistem AR. Pengujian dilakukan terhadap 6 skenario fungsional utama.

Tabel 4 Uji Black-box

No	Skenario Uji	Skenario Uji	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Navigasi Menu Utama	Pengguna menekan tombol fitur (Materi, Scan AR, Kuis) pada halaman utama.	Sistem merespons input sentuhan dan berpindah ke scene yang relevan tanpa delay atau force close.	Berhasil (Pindah Scene Lancar)	Valid
2	Inisialisasi AR Camera	Pengguna membuka fitur Scan AR.	Kamera perangkat aktif, autofocus berjalan, dan sistem siap memindai lingkungan.	Berhasil (Kamera Aktif)	Valid

3	Deteksi Image Target	Mengarahkan kamera ke marker fisik yang telah dicetak.	Vuforia Engine mengenali fitur marker, dan objek 3D vektor di-instantiate pada posisi koordinat (0,0,0) marker.	Berhasil (Objek Muncul)	Valid
4	Eksekusi Animasi Otomatis	Memastikan transisi dari Entry ke Scene saat marker terdeteksi.	Animasi vektor (perubahan besaran/arrah) langsung dimainkan secara otomatis (Auto-Play) segera setelah objek muncul, sesuai konfigurasi Direct Transition.	Berhasil (Animasi Jalan Otomatis)	Valid
5	Stabilitas Visualisasi	Menggerakkan kamera mengelilingi marker atau mengubah jarak pandang.	Objek vektor tetap terkunci (anchored) pada marker dan tidak mengalami pergeseran (drifting) yang signifikan.	Berhasil (Stabil)	Valid
6	Mekanisme Reset (Re-Tracking)	Menjauhkan kamera dari marker (Lost Tracking) lalu mengarahkannya kembali (Found).	Animasi di-reset dan dimainkan ulang dari awal (Replay) setiap kali marker dideteksi ulang.	Berhasil (Reset & Replay)	Valid

Berdasarkan Tabel 4, seluruh skenario pengujian menunjukkan status **Valid**. Hal ini mengonfirmasi bahwa konfigurasi *Event-Based Activation* pada Unity Animator berhasil berjalan tanpa kesalahan logika (*logic error*) atau kegagalan sistem (*crash*).

3.3.2. Validasi Ahli

Tahap validasi bertujuan untuk menguji kelayakan sistem dari dua perspektif utama: substansi keilmuan (materi) dan performa teknis (media). Data penilaian diperoleh melalui kuesioner tertutup yang diisi oleh dosen ahli matematika dan dosen ahli multimedia/teknologi pendidikan.

Validasi Ahli Materi (*Content Validity*) Evaluasi materi difokuskan untuk memastikan bahwa representasi visual yang dihasilkan oleh teknik animasi dinamis tidak menyimpang dari kaidah matematika. Indikator penilaian mencakup akurasi konsep vektor 3D, kejelasan struktur visualisasi, dan efektivitas penyajian dalam mendukung pemahaman konseptual.

Tabel 5 Uji Ahli Materi

No	Indikator Penilaian (Butir Kunci)	Skor (Skala 5)	Kategori
1	Penyajian materi vektor dalam aplikasi jelas dan sistematis	5	Sangat Baik
2	Konsep visualisasi vektor (arah, besaran, operasi) akurat	4	Baik
3	Struktur tampilan AR memudahkan visualisasi ruang	4	Baik
4	Penyajian materi dalam bentuk AR bersifat interaktif	5	Sangat Baik
5	Aplikasi efektif membantu pemahaman konsep abstrak	4	Baik
Rata-rata		4.4	Sangat Baik
Skor perolehan =		22	
Presentase		88	

Berdasarkan data pada Tabel 5, ahli materi memberikan skor rata-rata 4,4 atau setara dengan persentase kelayakan 88% (Kategori: Sangat Baik). Temuan penting dari validasi ini terlihat pada indikator ke-2 dan ke-3, di mana validator mengonfirmasi bahwa visualisasi vektor yang ditampilkan memiliki akurasi geometris yang baik. Hal ini membuktikan bahwa manipulasi *vertex* pada animasi *keyframe* berhasil mempertahankan proporsi arah dan besaran vektor secara presisi, sehingga aman digunakan sebagai media ajar Aljabar Linear yang menuntut ketelitian konsep.

Validasi Ahli Media (*Technical Validity*) Evaluasi media difokuskan pada aspek teknis implementasi, meliputi kualitas *rendering* objek, kestabilan sistem pelacakan (*tracking*), dan

responsivitas animasi. Pengujian ini memverifikasi keberhasilan penerapan arsitektur *Finite State Machine* (FSM) dalam menjaga performa aplikasi.

Tabel 6 Uji Ahli Media

No	Indikator Penilaian (Butir Kunci)	Skor (Skala 5)	Kategori
1	Visual vektor (panah, sumbu, komponen) terlihat tajam	5	Sangat Baik
2	Skala dan orientasi objek 3D akurat terhadap marker	5	Sangat Baik
3	Kestabilan tracking AR (Objek tidak bergeser/driftng)	4	Baik
4	Kelancaran kinerja aplikasi (Smoothness/Tanpa Lag)	5	Sangat Baik
5	Responsivitas animasi terhadap interaksi pengguna	4	Baik
6	Navigasi antarmuka jelas dan konsisten	4	Baik
Rata-rata		4.5	Sangat Baik
Skor perolehan =		27	
Presentase		90	

Hasil validasi media pada Tabel 6 menunjukkan pencapaian skor rata-rata 4,8 atau setara dengan persentase 90% (Kategori: Sangat Baik). Sebagaimana ditunjukkan pada indikator ke-3 dan ke-4, aspek kestabilan *tracking* dan kelancaran kinerja memperoleh skor sempurna (5/5). Hasil ini mengindikasikan bahwa penggunaan teknik *Event-Based Activation* dengan transisi linear pada *Animator Controller* sangat efektif dalam meminimalisir latensi. Sistem terbukti mampu menyajikan visualisasi 3D yang kompleks dengan pergerakan yang halus (*smooth*) dan stabil di atas *marker* fisik, tanpa membebani sumber daya komputasi perangkat *mobile*.

3.3.3. Pengujian Usability

Evaluasi tahap akhir dilakukan untuk mengukur tingkat kebergunaan (*usability*) dan akseptabilitas sistem dari perspektif pengguna akhir. Pengujian melibatkan 30 responden mahasiswa yang telah mencoba aplikasi dalam simulasi pembelajaran mandiri. Instrumen pengukuran menggunakan kuesioner adaptasi yang terdiri dari 11 butir pernyataan persepsi dengan skala Likert 5 poin [16].

Data penilaian dihitung dengan mengakumulasi skor jawaban responden, kemudian dikonversi ke dalam skala indeks 0-100 untuk disetarakan dengan standar interpretasi kelayakan sistem (*System Usability Scale*) [17].

Tabel 7 Uji Pengguna

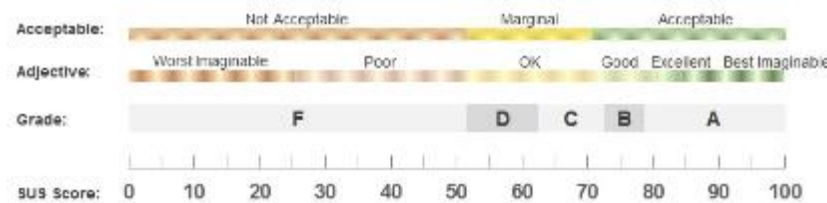
No	Soal Pertanyaan	Skor Kepuasan					Skor
		1	2	3	4	5	
1	Saya senang menggunakan aplikasi AR Vektor.	1	0	2	4	23	138
2	Aplikasi ini membuat pembelajaran vektor menjadi lebih menarik dan menyenangkan.	1	0	0	4	25	142
3	Saya tertarik untuk mencoba kembali aplikasi ini di waktu lain.	0	1	0	5	24	142
4	Saya ingin materi matematika lain juga diajarkan dengan cara seperti aplikasi AR Vektor.	0	2	0	4	24	140
5	Tampilan dan desain aplikasi terlihat menarik dan mudah dipahami.	2	0	1	4	23	136
6	Objek vektor 3D yang ditampilkan melalui AR terlihat jelas dan mudah diamati.	0	1	1	5	23	140
7	Warna, tulisan, dan tampilan keseluruhan membuat saya lebih semangat belajar.	1	0	1	6	22	138
8	Elemen UI penting (tombol utama, petunjuk) menonjol dan mudah ditemukan.	1	1	2	1	25	138
9	Aplikasi ini membantu saya memahami konsep arah dan besar vektor dengan lebih mudah.	1	1	0	3	25	140

10	Saya jadi lebih memahami operasi vektor (penjumlahan, pengurangan, dan komponen) setelah menggunakan aplikasi ini.	0	1	0	5	24	142
11	Belajar menggunakan AR Vektor lebih mudah dipahami dibandingkan hanya membaca buku atau slide.	2	0	2	4	22	134
Skor perolehan							1530
Persentase							92.7%

Perhitungan skor akhir dilakukan menggunakan rumus persentase untuk menormalisasi data kuesioner ke dalam skala indeks 0-100:

$$NP = \frac{1.530}{1.650} \times 100 = 92.7 \text{ (2)}$$

Hasil perhitungan pada Rumus (2) menunjukkan bahwa aplikasi memperoleh total skor 1.530 dari skor maksimal ideal 1.650. Jika dikonversi, nilai akhir usability sistem mencapai **92,7**. Merujuk pada grafik skala kelayakan (*Acceptability Range*) pada Gambar 7, skor ini menempatkan aplikasi pada kategori "**Acceptable**" dengan predikat "**Best Imaginable**" (*Grade A*). Hal ini mengindikasikan bahwa secara umum, sistem dapat diterima dengan sangat baik oleh pengguna target tanpa hambatan operasional yang berarti.



Gambar 7 Interpretasi System Usability Scale

Tingginya tingkat intuitivitas sistem, yang ditunjukkan oleh predikat '*Best Imaginable*', menjamin pengalaman pengguna yang mulus dan alami. Interaksi yang lancar pada fitur-fitur kunci membuktikan bahwa aplikasi ini berhasil menghilangkan hambatan teknis dalam pembelajaran, sehingga implementasi Visualisasi Vektor Dinamis dapat berfungsi optimal sebagai media pendukung pemahaman konsep Vektor bagi mahasiswa.

4. KESIMPULAN DAN SARAN

Berdasarkan hasil implementasi dan analisis data pengujian, penelitian ini menyimpulkan bahwa teknik Visualisasi Vektor Dinamis berhasil diimplementasikan pada media *Augmented Reality* berbasis Android untuk materi vektor. Penerapan arsitektur Eksekusi Linear menggunakan *Unity Animator Controller* terbukti efektif dalam mensimulasikan perilaku operasi vektor seperti transformasi besaran dan dekomposisi komponen secara *real-time* dan stabil.

Secara spesifik, pendekatan hibrida yang menggabungkan Aset Adaptif (Blender Keyframe) dengan logika Finite State Machine (Unity FSM) mampu menghasilkan visualisasi 3D yang presisi secara geometris tanpa membebani kinerja perangkat. Keberhasilan teknis ini dibuktikan melalui validasi ahli media yang memberikan skor 95% (Sangat Baik) dan ahli materi sebesar 86% (Sangat Baik). Selain validitas teknis, teknik visualisasi dinamis juga mendapatkan penerimaan yang sangat tinggi dari pengguna akhir. Hasil uji *usability* terhadap 30 mahasiswa menghasilkan skor konversi 92,7 yang masuk dalam kategori "*Best Imaginable*". Analisis indikator menunjukkan bahwa animasi prosedural secara signifikan membantu mahasiswa memvisualisasikan proses abstrak arah dan operasi vektor yang sulit dipahami melalui media statis. Dengan demikian, integrasi mekanisme animasi prosedural dalam lingkungan AR

terkonfirmasi sebagai solusi efektif untuk meningkatkan representasi spasial pada pembelajaran matematika tingkat perguruan tinggi.

Meskipun telah terbukti efektif, penelitian ini masih memiliki ruang untuk pengembangan lebih lanjut. Dari sisi teknis pelacakan, sistem saat ini masih bergantung pada metode *Marker-Based Tracking* yang membutuhkan pencahayaan optimal dan marker fisik, sehingga penelitian selanjutnya disarankan untuk mengeksplorasi metode *Markerless* (seperti *Plane Detection* atau *Mid-Air*) agar visualisasi dapat dilakukan di ruang bebas secara lebih fleksibel.

DAFTAR PUSTAKA

- [1] A. Maritsa, U. Hanifah Salsabila, M. Wafiq, P. Rahma Anindya, and M. Azhar Ma'shum, "Pengaruh Teknologi Dalam Dunia Pendidikan," *Al-Mutharahah: Jurnal Penelitian dan Kajian Sosial Keagamaan.*, vol. 18, no. 2, pp. 91–100, Dec. 2021, doi: 10.46781/al-mutharahah.v18i2.303.
- [2] I. Febrianti *et al.*, "Pengaruh Penggunaan Teknologi Informasi Dalam Manajemen Perencanaan Pendidikan Untuk Meningkatkan Efisiensi Pendidikan," *Academy of Education Journal.*, vol. 14, no. 2, pp. 506–522, Jul. 2023, doi: 10.47200/aoej.v14i2.1763.
- [3] S. R. Balmes, "Technology Integration and Transformative Innovation in Education," *International Journal of Research Publications.*, vol. 106, no. 1, Jul. 2022, doi: 10.47119/IJRP1001061820223743.
- [4] N. I. Ikawati., Dimpudus, A., Greas, K. E .A., & Ramadani. M., Pakpahan, Y. L. E., Ramadhani, "Pengembangan media pembelajaran matematika berbasis Android menggunakan Augmented Reality pada materi jaring-jaring kubus dan balok," *Primatika : Jurnal Pendidikan Matematika.*, vol. 13, no. 2, pp. 175–188, Oct. 2024, doi: <https://doi.org/10.30872/primatika.v13i2.4381>.
- [5] R. B. Ryandi and D. D. Santri, "Geogebra untuk Pembelajaran Vektor," *PRISMA*, vol. 10, no. 1, p. 77, Jun. 2021, doi: 10.35194/jp.v10i1.1084.
- [6] M. U. Gusteti *et al.*, "Penggunaan Augmented Reality dalam Pembelajaran Matematika: Sebuah Analisis Berdasarkan Studi Literatur," *EDUKATIF : JURNAL ILMU PENDIDIKAN.*, vol. 5, no. 6, pp. 2735–2747, Dec. 2023, doi: 10.31004/edukatif.v5i6.5963.
- [7] T. Harmini, F. Reza Pradhana, D. Nur Suryanita, and K. Warniasih, "Implementasi Game Edukasi 3d Pada Materi Aljabar Melalui Pendekatan Game Development Life Cycle (GDLC)," *Riemann: Research of Mathematics and Mathematics Education.*, vol. 6, no. 1, pp. 23–38, Apr. 2024, doi: 10.38114/rksfhj44.
- [8] F. Arena, M. Collotta, G. Pau, and F. Termine, "An Overview of Augmented Reality," *Computers*, vol. 11, no. 2, p. 28, Feb. 2022, doi: 10.3390/computers11020028.
- [9] R. Rahmat and N. Noviyanti, "Augmented Reality untuk Materi Bangun Ruang Menggunakan Unity 3D, Vuforia SDK dan Aplikasi Blender," *Jurnal TIKA*, vol. 5, no. 3, pp. 86–92, Jan. 2021, doi: 10.51179/tika.v5i3.59.
- [10] R. Oktrilani, V. I. Delianti, B. R. Fajri, and A. D. Samala, "Rancang Bangun Media Pembelajaran Berbasis Augmented Reality pada Materi Sistem Pernapasan Mata Pelajaran Biologi Kelas XI MIPA Tingkat SMA," *JAVIT : Jurnal Vokasi Informatika.*, pp. 79–86, Jun. 2023, doi: 10.24036/javit.v3i2.156.
- [11] B. Arifitama, A. Syahputra, and K. B. Y. Bintoro, "Analisis Perbandingan Efektifitas Metode Marker dan Markerless Tracking pada Objek Augmented Reality," *JURNAL INTEGRASI.*, vol. 14, no. 1, pp. 1–7, Apr. 2022, doi: 10.30871/ji.v14i1.3985.
- [12] Y. Abdurrahman and M. Azrino Gustalika, "Aplikasi Augmented Reality dengan Marker Based dan Markerless Tracking sebagai Pengenalan Budaya Candi Mendut," *Remik : Riset dan E-Jurnal Manajemen Informatika Komputer.*, vol. 7, no. 2, pp. 859–871, Apr. 2023, doi: 10.33395/remik.v7i2.12137.

- [13] N. A. L. Sidabutar and R. Reffina, “Pengembangan Media Pembelajaran Matematika SMA dengan Aplikasi Animaker pada Materi Vektor,” *Jurnal Cendekia : Jurnal Pendidikan Matematika*, vol. 6, no. 2, pp. 1374–1386, Apr. 2022, doi: 10.31004/cendekia.v6i2.1362.
- [14] J. Pernas-Álvarez and D. Crespo-Pereira, “Open-source 3D discrete event simulator based on the game engine unity,” *Journal of Simulation*, vol. 18, no. 5, pp. 886–902, Sep. 2024, doi: 10.1080/17477778.2024.2314166.
- [15] B. O. Tafakkur, L. P. I. Kharisma, A. A. Rizal, and A. Abdurahim, “Implementasi Augmented Reality Sebagai Media Promosi Pada Lesehan Kalisari Dengan Metode Based Marker Tracker,” *JTIM : Jurnal Teknologi Informasi dan Multimedia*, vol. 5, no. 1, pp. 10–21, May 2023, doi: 10.35746/jtim.v5i1.331.
- [16] A. Joshi, S. Kale, S. Chandel, and D. Pal, “Likert Scale: Explored and Explained,” *British Journal of Applied Science & Technology*, vol. 7, no. 4, pp. 396–403, Jan. 2015, doi: 10.9734/bjast/2015/14975.
- [17] R. Y. Manurung, D. Krisbiantoro, and D. A. B. Utami, “Usability Evaluation of Tokopedia Application Version 3.242 Using System Usability Scale (SUS) Method,” *Sinkron*, vol. 9, no. 1, pp. 366–374, Jan. 2024, doi: 10.33395/sinkron.v9i1.13191.