

Integrasi Sistem *Dynamic Lighting* Berbasis *Muzzle Flash* sebagai Solusi Navigasi dalam *Game FPS Aim Trainer*

Ali Abdul Hadi^{*1}, Noven Indra Prasetya²

^{1,2} Program Studi Informatika, Fakultas Teknik, Universitas Wijaya Kusuma Surabaya
Email: ¹aah022@mhs.uwks.ac.id, ²noven@uwks.ac.id

Abstrak

Permainan bergenre *First Person Shooter (FPS)* membutuhkan kemampuan refleks dan akurasi bidikan yang tinggi untuk mencapai performa bermain yang optimal. Sebagian besar aplikasi *aim trainer* konvensional masih mengandalkan kondisi visual terang sehingga kurang mampu merepresentasikan situasi permainan *FPS modern* yang sering menghadirkan lingkungan dengan visibilitas terbatas. Penelitian ini bertujuan untuk mengembangkan sistem *dynamic lighting* berbasis *muzzle flash* sebagai solusi navigasi visual pada *game FPS Aim Trainer* dalam kondisi ruang gelap total. Metode pengembangan yang digunakan adalah *Game Development Life Cycle* model Heather Maxwell Chandler yang terdiri dari tahap *Pre-Production*, *Production*, *Testing*, dan *Post-Production*. Sistem dikembangkan menggunakan *Unity Game Engine* dan bahasa pemrograman *C#*. Implementasi dilakukan dengan memanfaatkan komponen *Point Light* dan *Particle System* yang aktif secara *real-time* saat pemain menembakkan senjata. Hasil pengujian menunjukkan bahwa efek *muzzle flash* mampu berfungsi sebagai sumber pencahayaan sementara sekaligus media navigasi visual dalam kondisi minim cahaya. Mekanisme ini juga mendorong pemain untuk memanfaatkan kemampuan *spatial memory*, refleks visuomotor, serta pengelolaan amunisi secara strategis selama permainan berlangsung.

Kata kunci: *aim trainer, dynamic lighting, FPS, GDLC, muzzle flash, spatial memory*

Abstract

Abstract First Person Shooter (FPS) games require high reflex capability and shooting accuracy to achieve optimal gameplay performance. Most conventional aim trainer applications still rely on bright visual environments, making them less capable of representing modern FPS gameplay situations with limited visibility conditions. This study aims to develop a dynamic lighting system based on muzzle flash as a visual navigation solution in an FPS Aim Trainer game within a completely dark environment. The development process applied the Game Development Life Cycle model proposed by Heather Maxwell Chandler, consisting of Pre-Production, Production, Testing, and Post-Production stages. The system was developed using Unity Game Engine and the C# programming language. The implementation utilized Point Light and Particle System components that activate in real-time whenever the player fires a weapon. The testing results indicate that the muzzle flash effect successfully functions as a temporary lighting source as well as a visual navigation medium under low-visibility conditions. Furthermore, the mechanic encourages players to utilize spatial memory, visuomotor reflexes, and strategic ammunition management during gameplay.

Keywords: *aim trainer, dynamic lighting, FPS, GDLC, muzzle flash, spatial memory.*

I. PENDAHULUAN

Kemahiran membidik (*aim*) dalam permainan FPS sangat bergantung pada refleks pemain saat mengoperasikan tetikus untuk mengarahkan *crosshair* ke sasaran. Walaupun kemampuan ini dapat diasah melalui jam terbang dalam gim, pemain pemula di ekosistem kompetitif sering kali menghadapi hambatan berupa perilaku toksik dari pemain lain, seperti *flaming*, *grieving*, pelecehan seksual, hingga *trolling* terbukti dapat menurunkan motivasi pemain untuk berlatih meningkatkan akurasi mereka. Sebagai solusi, aplikasi pihak ketiga yang dikenal sebagai *Aim Trainer* diperlukan agar pemain dapat fokus mengasah kemampuan membidik tanpa gangguan lingkungan sosial yang negatif. [1]

Implementasi mekanisme *Random Weapon Accuracy (RWA)* dalam berbagai permainan bergenre *shooter* sering diadopsi untuk mengejar aspek realisme melalui komponen proyektil yang bersifat acak saat menembakkan senjata, meskipun mampu meningkatkan efek realitas visual, penggunaan RWA beresiko mengurangi jenjang kompetensi (*skill gap*) antara pemain awam dan ahli akibat tidak adanya pola dadakan yang terukur. Sebagai alternatif, sistem *Predetermined Recoil Patterns (PRP)* menawarkan konsistensi alur proyektil, baik pada sumbu horizontal maupun vertikal, yang bersifat unik pada setiap jenis senjata. konsistensi pola ini membuka ruang bagi pemain untuk melakukan penguasaan ingatan serta adaptasi motorik yang presisi. Dengan demikian, pemain yang mendedikasikan upaya untuk mendalami pola tembakan tersebut akan memperoleh keunggulan kompetitif yang signifikan melalui kemampuan pengendalian senjata yang lebih terprediksi dibandingkan pemain lainnya. [2]

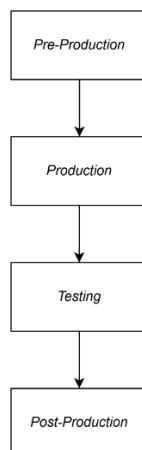
Secara mendasar, cahaya dikenal luas sebagai komponen yang sangat penting bagi keberlangsungan hidup manusia. Kehadiran sistem penerangan memegang peranan kunci yang memungkinkan individu berfungsi secara optimal di berbagai situasi, mengingat dampaknya yang bersentuhan langsung dengan segala aspek kehidupan kita. Lebih jauh lagi, pencahayaan bertindak sebagai pendorong utama bagi ketajaman penglihatan (*visual performance*), serta menjadi faktor penentu dalam menjaga kualitas kognitif maupun *mood*. [3]

Tata cahaya dalam pembuatan *game* dan *3-D environments* memiliki peran krusial yang setara dengan industri sinematografi untuk mengarahkan *visual attention*, membangun *mood*, dan menjaga visibilitas objek sesuai dengan *genre* serta *game mechanics* yang diusung. Namun, metode pencahayaan yang umum digunakan saat ini masih memisahkan tata cahaya *environment* dari sistem penerangan karakter yang bekerja secara *dynamically*. Kelemahan utama dari pendekatan konvensional tersebut adalah ketiadaan kapabilitas untuk beradaptasi secara otonom saat *runtime* guna menyeimbangkan komposisi warna dan *brightness* antara target dengan latar belakangnya, padahal harmonisasi responsif ini sangat krusial untuk merebut atensi visual pemain secara instan pada momen-momen penentu. [4]

Penggunaan *muzzle flash* di simulasi latihan menembak ini sebenarnya punya fungsi yang jauh lebih penting daripada cuma sekadar hiasan biar kelihatan keren di layar. Di dalam lingkungan yang jarak pandangannya sangat terbatas atau gelap banget, cahaya yang muncul saat menembakk ini berubah jadi alat bantu navigasi yang dinamis, dan itu adalah satu-satunya cara untuk pemain bisa mendapatkan info *visual* soal apa yang ada di sekitarnya. Situasi seperti ini dapat membuat pemain tidak bisa hanya mengandalkan pandangan mata yang jelas terus-menerus, tetapi mereka dipaksa untuk memakai cahaya singkat agar bisa tahu di mana lokasi target mereka yang sebenarnya. Secara teknis, sistem ini berfungsi agar senjata bergeser jadi semacam alat pemandu, di mana pemain harus berani mengambil risiko untuk menghabiskan peluru mereka hanya demi bisa melihat posisi target dalam sekejap saja. [5]

II. METODE PENELITIAN

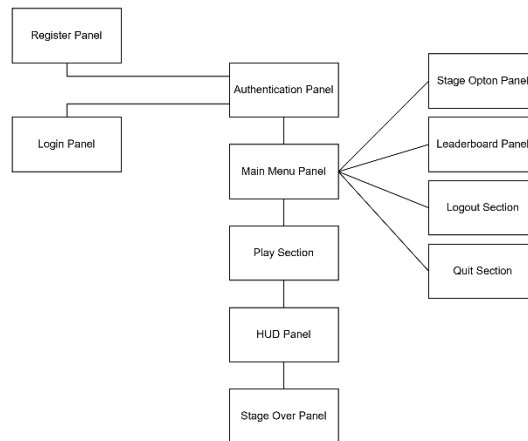
Pengembangan sistem dalam penelitian ini menggunakan metode *Game Development Life Cycle* (GDLC) yang dirumuskan oleh Heather Maxwell Chandler. Pemilihan metode GDLC didasarkan pada pendekatannya yang terstruktur dan dirancang khusus untuk pembuatan *video game*, di mana prosesnya mampu menyelaraskan kerumitan logika pemrograman dengan elemen kreatif secara interaktif. Model ini berjalan dalam empat fase siklus utama: *Pre-Production*, *Production*, *Testing*, dan *Post-Production*. Secara tahapan-tahapan tersebut diaplikasikan sebagai panduan sistematis mulai dari proses perancangan, implementasi, hingga pengujian mekanisme navigasi berbasis *muzzle flash* di dalam lingkungan *game FPS Aim Training*.



Gambar 1. Metode GDLC Heather Maxwell Chandler

2.1. Pre-Production

Tahap *Pre-Production* merupakan fase awal dalam metode *Game Development Life Cycle* (GDLC) yang digunakan sebagai proses perencanaan sebelum proses pengembangan *game* dimulai. Pada tahapan ini dilakukan perumusan konsep permainan, identifikasi kebutuhan sistem, perancangan mekanisme *gameplay*, serta pengujian awal terhadap mekanik utama yang akan diterapkan pada *game*. Tahap ini memiliki peranan penting karena menjadi landasan utama selama proses produksi berlangsung.



Gambar 3. Urutan *Screen Flow*

2.2. Production

Fase Produksi merupakan tahapan eksekusi teknis yang sangat penting guna menerjemahkan seluruh rancangan desain ke dalam bentuk perangkat lunak (*software*) yang utuh. Proses ini dicapai melalui penggabungan struktur kode pemrograman, aset visual, dan alur logika permainan di dalam *Unity Game Engine*. Langkah awal dari tahapan produksi adalah membangun struktur utama permainan (*base game*). Pengerjaan ini melibatkan penulisan *C# script* untuk mengendalikan sistem fisika, pergerakan karakter (*player movement*), deteksi benturan antar objek (*collision*), serta mekanik menembak dasar. Apabila fondasi sistem tersebut telah rampung, fokus pengembangan beralih secara mendalam pada penerapan sistem panduan visual berbasis *muzzle flash*, yang menjadi inti inovasi dari simulasi ini. Dari segi teknis, fitur tersebut beroperasi dengan memanipulasi tata cahaya secara langsung (*real-time*) melalui pemadaman seluruh sumber cahaya di sekeliling pemain (*ambient light*). Sebagai gantinya, sistem memanfaatkan komponen *Point Light* serta *Particle System* yang diletakkan secara presisi tepat di moncong senjata. Ketika sistem menerima *input* tembakan, algoritma secara otomatis akan membangkitkan kilatan cahaya berintensitas tinggi dalam kurun waktu sepersekian detik. Mekanisme ini dirancang secara khusus untuk memastikan bahwa pemain hanya memperoleh jarak pandang (*visibility*) sesaat, sebuah kondisi operasional yang sangat sejalan dengan konsep dasar dari teori pelatihan adaptasi refleks.

2.3. Testing

Fase pengujian atau *testing* dilaksanakan dalam dua tahapan utama untuk memastikan kelancaran program sekaligus menilai efektivitas *game* yang dikembangkan. Tahap pertama yaitu *Alpha Testing*, berupa evaluasi internal yang memanfaatkan metode *Black Box Testing* untuk meninjau fungsi mekanik *muzzle flash*. Fokusnya adalah memastikan kilatan cahaya beserta efek partikel dapat muncul secara presisi setiap kali sistem menerima *input* tembakan tanpa adanya kendala atau *bug*. Begitu *bug* berhasil diperbaiki, pengujian dilanjutkan ke tahap *Beta Testing* dengan melibatkan pengguna selain *developer* untuk memperoleh *feedback* teknis yang lebih objektif. Evaluasi tahap kedua ini dirancang untuk menilai *user experience* secara langsung, terutama untuk menguji apakah kondisi minim cahaya di dalam permainan benar-benar efektif dalam memaksa pemain menggunakan *spatial memory* serta melatih kecepatan refleks mereka.

2.4. Post-Production

Tahap akhir dari keseluruhan siklus pengembangan ini adalah *Post-Production*, yang menjadi fase penutup dengan fokus utama pada finalisasi produk perangkat lunak. Pada tahapan ini, pihak pengembang bertugas untuk melakukan analisis mendalam terhadap kumpulan data dan *feedback* yang berhasil didapatkan selama berjalannya sesi *Beta Testing*. Langkah evaluasi akhir ini sangat penting untuk menarik kesimpulan yang objektif mengenai tingkat kelayakan serta seberapa efektif implementasi mekanik *muzzle flash* tersebut saat dimainkan oleh pengguna. Sebagai langkah pamungkas, seluruh rekam jejak proses mulai dari perancangan awal hingga hasil temuan pengujian akan dirangkum secara sistematis ke dalam dokumentasi akhir (*post-mortem*) guna keperluan perbaikan susulan (*patching*) dan sebagai referensi pengembangan *game* di masa mendatang.

III. HASIL DAN PEMBAHASAN

3.1. Implementasi

Tahap implementasi sistem pada penelitian ini difokuskan pada pencahayaan dinamis (*dynamic lighting*) di dalam *Unity Game Engine*. Untuk menciptakan kondisi keterbatasan visual yang menjadi syarat utama pelatihan refleks, langkah pertama yang dilakukan adalah melakukan manipulasi pada *Global Lighting Controller*. Seluruh sumber pencahayaan utama di dalam mesin *game*, baik itu *ambient light* maupun *directional light*, dinonaktifkan secara total. Manipulasi ini menghasilkan lingkungan ruang simulasi yang berada dalam kondisi gelap gulita dengan tingkat visibilitas mendekati nol, sebuah kondisi ekstrem yang sengaja dirancang agar pemain tidak bisa lagi mengandalkan penglihatan konstan saat mencari posisi papan target. Sebagai solusi atas ketiadaan cahaya lingkungan tersebut, fokus implementasi dialihkan pada perakitan sistem navigasi visual berbasis *muzzle flash*. Secara teknis, mekanisme ini dibangun dengan menempatkan komponen *Point Light* berintensitas pancaran tinggi serta efek visual (*Particle System*) secara presisi tepat di ujung moncong laras senjata karakter. Komponen cahaya dinamis ini tidak menyala secara permanen, melainkan dikendalikan melalui instruksi logika *C# script*. Setiap kali algoritma mendeteksi adanya *input* klik atau tarikan pelatuk dari pemain, sistem akan secara instan membangkitkan kilatan cahaya (*real-time illumination*) yang hanya berdurasi sepersekian detik.

```
public void TriggerLight()
{
    if (!gameObject.activeInHierarchy) return;

    if (lightCoroutine != null)
        StopCoroutine(lightCoroutine);

    lightCoroutine = StartCoroutine(FlashLightRoutine());
}

// -----
private IEnumerator FlashLightRoutine()
{
    muzzleLight.enabled = true;
    muzzleLight.intensity = lightIntensity;

    float elapsed = 0f;

    // Fase menyala penuh + flicker
    while (elapsed < lightDuration)
    {
        elapsed += Time.deltaTime;

        if (enableFlicker)
        {
            float flicker = Random.Range(1f - flickerAmount, 1f);
            muzzleLight.intensity = lightIntensity * flicker;
        }

        yield return null;
    }

    // Fase fade-out
    float currentIntensity = muzzleLight.intensity;
    while (currentIntensity > 0f)
    {
        currentIntensity -= fadeSpeed * Time.deltaTime;
        muzzleLight.intensity = Mathf.Max(0f, currentIntensity);
        yield return null;
    }

    muzzleLight.intensity = 0f;
    muzzleLight.enabled = false;
    lightCoroutine = null;
}
```

Gambar 4. Fungsi *muzzle flash*



Gambar 5. Efek *muzzle flash* di kondisi terang



Gambar 6. Efek *muzzle flash* di kondisi ruang gelap

Implementasi pencahayaan yang muncul secara intermiten atau terputus-putus inilah yang menjadi inti dari simulasi permainan. Pendaran *muzzle flash* bertugas untuk memecah kegelapan ruangan sesaat, menyorot geometri ruangan dan menyingkap posisi target yang sebelumnya telah diacak (*randomize*) oleh sistem. Transisi kilat antara terang dan gelap total ini secara teknis memaksa otak pemain untuk merekam letak sasaran menggunakan ingatan lokasi (*spatial memory*) sebelum ruangan kembali tertutup kegelapan.

3.2. Mekanika Sistem

Desain sistem dalam pengembangan ini mengintegrasikan tata ruang simulasi dan mekanisme tembakan guna membentuk skenario latihan refleks yang berintensitas tinggi. Tahapan ini bermula dari rekayasa pada Global Lighting Controller yang terdapat pada Unity Game Engine. Segala bentuk pencahayaan bawaan sistem, termasuk ambient light dan directional light, dimatikan sepenuhnya guna menciptakan arena simulasi yang gelap pekat dengan jarak pandang nyaris nol. Pada situasi keterbatasan informasi penglihatan ini, pergerakan pemain maupun ketepatan membidik dikontrol melalui player controller yang disusun agar bergerak responsif tanpa jeda. Di sisi lain, pelacakan lontaran proyektil menerapkan teknik raycasting, yaitu memancarkan garis virtual dari pusat layar (crosshair) guna memastikan apakah tembakan mengenai papan target yang posisinya senantiasa diacak (*randomize*) oleh program.

Untuk mengatasi tidak adanya pencahayaan di sekitar ruangan, sistem menerapkan konsep pencahayaan dinamis (*dynamic lighting*) yang bersumber dari kilat senjata. Instrumen navigasi visual ini dikembangkan dengan menyematkan fitur Point Light berdaya sinar kuat dan efek grafis Particle System persis di ujung laras senapan karakter. Operasional fitur cahaya ini diatur secara presisi menggunakan perintah logika berformat C# script. Tiap kali algoritma mengenali input berupa tarikan pelatuk, sistem secara otomatis akan memicu kilatan muzzle flash (*real-time illumination*) yang durasinya hanya berlangsung dalam hitungan milidetik. Guna memperkuat kesan nyata atau tingkat imersi di dalam simulasi, pengembang juga menyertakan komponen AudioSource yang diatur sedemikian rupa untuk mengeluarkan efek suara tembakan (*gunshot*) tepat bertepatan dengan momen cahaya tersebut membelah kegelapan.

Penggunaan pencahayaan yang menyala secara berkala ini pada dasarnya merombak total mekanika inti (*core mechanics*) dari permainan itu sendiri. Dalam kondisi tanpa cahaya, fungsi senjata tidak lagi terbatas sebagai instrumen penghancur sasaran, tetapi juga bertransformasi menjadi sarana tunggal untuk panduan orientasi taktis. Pengguna akan dihadapkan pada konsep risk management, yang menuntut mereka untuk rela membuang sejumlah amunisi (*pre-fire*) hanya demi meraih visibilitas sejenak (*temporary visibility*). Perubahan kilat dari terang benderang kembali ke gelap pekat ini akan mendorong otak pemain untuk memetakan arsitektur ruangan serta letak sasaran lewat kemampuan ingatan lokasi (*spatial memory*). Kesimpulannya, pelepasan tembakan berikutnya tidak lagi berpatokan pada penglihatan yang konstan, tetapi lebih didasarkan pada insting serta kemampuan adaptasi refleks visuomotor dalam merespons letak objek yang baru saja terekspose sekilas.

3.3. Desain Sistem

Desain sistem dalam rancangan ini memfokuskan pada penggabungan antara efek cahaya dinamis dengan alur logika inti permainan (*core gameplay*). Desain tersebut memetakan peran vital efek *muzzle flash* yang tidak sekadar menjadi pemanis *visual*, melainkan berfungsi sebagai instrumen navigasi utama di tengah lingkungan simulasi yang gelap gulita. diawali dengan *InteractionManager* yang menangkap *input*

tembakan, lalu secara simultan memerintahkan *WeaponManager* untuk melontarkan proyektil dan *FlashSystem* untuk membangkitkan pendaran cahaya seketika (*real-time illumination*).

Kerangka teknis ini dipastikan agar setiap tembakan yang dilepaskan pemain menghasilkan umpan balik (*feedback*) berupa jarak pandang (*visibility*) sementara pada area sekitar. Lebih lanjut, arsitektur sistem mengatur hubungan fungsional di mana pengaturan durasi dan kekuatan cahaya *muzzle flash* menjadi penentu tingkat kesulitan simulasi secara langsung. Mekanika ini sengaja dikembangkan untuk memicu kemampuan manajemen risiko (*risk management*) pemain dalam mengelola amunisi, sehingga mereka dituntut jeli dalam menentukan kapan harus menembak demi menyingkap posisi papan target yang tersembunyi di balik kegelapan.

3.4. Pengujian

Tabel 1. *Blackbox* pengujian mekanika *muzzle flash* dan *dynamic lighting*

No	Skenario Pengujian	Test Case	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
1.	Validasi kondisi awal kondisi ruang simulasi	Melihat kondisi ruang sekitar yang awalnya terang lalu gelap saat waktu hitungan mundur persiapan selesai	<i>Global lighting</i> dan <i>ambient light</i> sepenuhnya nonaktif. Layar menampilkan ruangan yang gelap gulita, di mana hanya elemen HUD (<i>Heads-Up Display</i>) dan <i>crosshair</i> yang terlihat menyala	<i>Global lighting</i> dan <i>ambient light</i> sepenuhnya nonaktif. Layar menampilkan ruangan yang gelap gulita, di mana hanya elemen HUD (<i>Heads-Up Display</i>) dan <i>crosshair</i> yang terlihat menyala	VALID
2.	Memunculkan efek muzzle flash	Pemain menekan tombol kiri pada mouse untuk menembak senjata	Sistem secara instan menampilkan <i>Particle System</i> dan memancarkan <i>Point Light</i> tepat di ujung moncong senjata tanpa adanya <i>delay</i>	Sistem secara instan menampilkan <i>Particle System</i> dan memancarkan <i>Point Light</i> tepat di ujung moncong senjata tanpa adanya <i>delay</i>	VALID
3	Menguji durasi <i>Real-Time illumination</i>	Pemain menekan tombol kiri pada mouse untuk menembak senjata	Kilatan cahaya menyala dan bertahan selama hitungan milidetik sebelum secara otomatis mati dan kondisi ruang Kembali gelap gulita	Kilatan cahaya menyala dan bertahan selama hitungan milidetik sebelum secara otomatis mati dan kondisi ruang Kembali gelap gulita	VALID
4	Menguji jarak pandang sasaran	Pemain menembakkan senjatanya kearah papan target	Pantulan cahaya berhasil menangkap bentuk geometri ruangan serta memunculkan posisi papan	Pantulan cahaya berhasil menangkap bentuk geometri ruangan serta memunculkan posisi papan	VALID

			target yang berada di dalam radius pencahayaannya.	target yang berada di dalam radius pencahayaannya.	
5	Sinkronisasi Audio	Pemain menahan tombol kiri pada mouse untuk menembakan senjata secara beruntun	Efek suara senjata secara sinkron keluar dengan setiap <i>muzzle flash</i> yang muncul, tanpa ada delay	Efek suara senjata secara sinkron keluar dengan setiap <i>muzzle flash</i> yang muncul, tanpa ada delay	VALID

IV. KESIMPULAN

Pengembangan dan penelitian sistem ini menunjukkan bahwa ketiadaan pencahaya lingkungan (*ambient light*) di dalam game *First-Person Shooter (FPS)* berjenis *aim trainer* dapat diatasi secara inovatif melalui integrasi *dynamic lighting* berbasis *muzzle flash*. Penggunaan komponen *Point Light* serta *Particle System* yang menyala secara *real-time* saat senjata ditembakkan berhasil melampaui fungsinya sebagai sekadar elemen estetika visual. Fitur tersebut terbukti mampu bertransformasi menjadi satu-satunya alat navigasi yang dapat diandalkan bagi pemain untuk melihat area sekitar di tengah kondisi ruangan yang gelap total.

V. UCAPAN TERIMA KASIH

Penulis mengucapkan puji syukur kepada Tuhan Yang Maha Esa atas segala rahmat dan karunia-Nya sehingga penelitian dan penulisan tugas akhir ini dapat diselesaikan dengan baik. Penulis juga menyampaikan terima kasih yang sebesar-besarnya kepada seluruh pihak yang telah memberikan dukungan, bantuan, bimbingan, serta motivasi selama proses penyusunan penelitian ini berlangsung.

Ucapan terima kasih secara khusus penulis sampaikan kepada dosen pembimbing yang telah memberikan arahan, masukan, dan ilmu pengetahuan selama proses pengerjaan tugas akhir. Selain itu, penulis juga mengucapkan terima kasih kepada keluarga, sahabat, dan rekan-rekan yang selalu memberikan doa, semangat, serta dukungan moral sehingga penelitian ini dapat terselesaikan.

REFERENSI

- [1] M. R. Giandhani and R. Elsen, "Rancang Bangun Game Aim Trainer Untuk Pemain Game FPS Menggunakan Game Engine Unity," *Jurnal Algoritma*, vol. 22, no. 1, pp. 434–446, May 2025, doi: 10.33364/algoritma/v.22-1.1725.
- [2] S. Pylvänen and István Gyarmati, "Game feature expectations in modern first-person shooters," South-Eastern Finland University of Applied Sciences, Kouvola, 2025.
- [3] M. S. Mott, D. H. Robinson, A. Walden, J. Burnette, and A. S. Rutherford, "Illuminating the effects of dynamic lighting on student learning," *Sage Open*, vol. 2, no. 2, pp. 1–9, Apr. 2012, doi: 10.1177/2158244012445585.
- [4] M. S. El-Nasr, A. Vasilakos, C. Rao, and J. Zupko, "Dynamic intelligent lighting for directing visual attention in interactive 3-D scenes," *IEEE Trans. Comput. Intell. AI Games*, vol. 1, no. 2, pp. 145–153, Jun. 2009, doi: 10.1109/TCIAIG.2009.2024532.
- [5] L. Wilkins and R. Gray, "Effects of Stroboscopic Visual Training on Visual Attention, Motion Perception, and Catching Performance," *Percept. Mot. Skills*, vol. 121, no. 1, pp. 57–79, Aug. 2015, doi: 10.2466/22.25.PMS.121c11x0.