

Menjelajahi Efisiensi dan Aksesibilitas Godot untuk Pengembang Lokal: Studi Kasus Bilu Mela

Fajar Nuswantoro, M.Sn¹

¹Fakultas Film dan Televisi, Institut
Kesenian Jakarta, Indonesia

e-mail: fajarnuswantoro@ikj.ac.id¹

Received : Juni, 2024	Accepted : Juni, 2024	Published : Juni, 2024
-----------------------	-----------------------	------------------------

ABSTRAK

Dalam industri game, tantangan utama yang dihadapi adalah kompleksitas dalam penggabungan ilmu teknologi informasi dan seni film serta animasi. Kesenjangan teknis dan artistik ini seringkali memperlambat proses produksi dan mengurangi jumlah output yang dapat dicapai. Penelitian ini bertujuan untuk mengidentifikasi alternative metode produksi yang lebih murah untuk meningkatkan kuantitas produksi game tanpa mengorbankan kualitas.

Untuk mencapai tujuan ini, dilakukan serangkaian eksperimen pembuatan game secara langsung oleh seseorang yang belum pernah membuat game. Metode ini diharapkan dapat memberikan gambaran seberapa jauh sebuah mesin game bisa menjadi wahana pembuatan game dan seberapa jauh sebuah komunitas bisa membantu menjawab pertanyaan saat menemui kesulitan dalam pembuatan.

Hasil dari penciptaan ini menunjukkan bahwa penggunaan Godot dalam produksi game memberikan dampak signifikan terhadap efisiensi produksi. Proses yang lebih efisien ini berpotensi meningkatkan jumlah produksi game di Indonesia, memberikan kesempatan bagi industri game lokal untuk berkembang dan bersaing di pasar global. Dengan demikian, penelitian ini tidak hanya berkontribusi pada pengetahuan akademis, tetapi juga memiliki implikasi praktis yang signifikan bagi industri game. Penemuan ini diharapkan dapat mendorong lebih banyak inovasi dan estetika dalam penerapan kecerdasan buatan dalam berbagai aspek produksi game.

Keyword: Efisiensi, Game, Alih wahana, Godot, Animasi

1. PENDAHULUAN

Perfilman Indonesia masih mempunyai banyak masalah, dimulai dari pendanaan, produksi, distribusi dan pemasaran. Hampir di semua lini punya masalah yang belum terurai meski Indonesia sudah puluhan tahun merdeka. Bisnis perfilman memang belum cukup menggiurkan atau menghasilkan sehingga para pemangku kebijakan tidak tertarik untuk menggali persoalan ini lebih dalam.

Masalah semakin lebar mengingat film punya keterkaitan dengan media lain yang lebih spesifik, misalnya animasi. Permasalahan di industri animasi Indonesia juga sama peliknya dengan industri film. Para individu yang terlibat dalam produksi animasi berusaha secara mandiri untuk mengatasi masalah mereka masing-masing. Industri animasi tetap menghasilkan konten-konten yang menarik meski jauh dari perhatian pemerintah.

Permasalahan yang dihadapi industri film dan animasi sekilas memang rumit dan membuat sedih. Namun, permasalahan mereka ternyata tidak separah yang dialami oleh industri gim di Indonesia. Industri gim hampir bisa dikatakan mati suri karena industri ini seringkali lepas dari pengamatan media. Jika film atau animasi masih menyumbang kabar viral setiap tahunnya, gim Indonesia tidak pernah menyumbang kabar yang menggembirakan. Produksi gim di Indonesia bisa dikatakan sangat jarang. Bahkan mungkin sebagian penduduk Indonesia tidak sadar jika Indonesia sudah memproduksi gim lokal dan sudah dipasarkan secara internasional.

Kenapa gim dirasa penting? Jika melihat pada data yang berlaku di belahan dunia lain, seharusnya industri gim menjadi pelopor dalam hal penghasilan. Bahkan, penghasilannya seharusnya mengalahkan industri film dan animasi. Sebagai perbandingan, pendapatan game berjudul Grand Theft Auto V adalah sebesar 6 milyar dollar sedangkan pendapatan film berjudul Avatar – pemegang rekor tertinggi – adalah 2,8 milyar dollar. Tentu data ini menimbulkan pertanyaan selanjutnya: kenapa gim di Indonesia tidak bisa menghasilkan pendapatan yang cukup atau mengalahkan posisi film dan animasi?

Ada beberapa faktor yang patut diduga menjadi tersangka atas seretnya perkembangan industri gim. Salah satu faktor yang paling logis adalah sulitnya produksi gim. Produksi gim tidak hanya melibatkan ilmu seni dan desain tapi juga melibatkan ilmu teknologi yang seringkali berganti karena cepatnya revolusi di bidang teknologi informasi. Bukti bahwa produksi gim tidak mudah bisa kita lihat dari jumlah kampus yang membuka jurusan produksi gim.

Jika kita mencari ‘sekolah pembuatan game’ di internet maka kita tidak akan mendapatkan jawaban yang mutlak. Faktanya memang belum banyak tersedia jurusan produksi game di kampus-kampus Indonesia. Jurusan yang tersedia adalah jurusan yang beririsan dengan produksi game; misalnya desain komunikasi visual, teknologi informasi, dan film animasi. Penulis memilih kata beririsan karena ketiga jurusan itu tidak mengajarkan produksi game secara utuh. Lulusan ketiga jurusan itu tidak bisa membuat game secara mandiri. Mereka hanya bisa mengerjakan sebagian dari produksi game sesuai bagian masing-masing.

Faktor lain yang bisa menjadi sumber masalah adalah biaya langganan perangkat lunak yang dirasa memberatkan untuk produser gim tingkat pemula. Dua perangkat lunak pembuatan gim paling populer di Indonesia adalah Unity dan Unreal. Keduanya menawarkan biaya langganan fantastis. Ini membuat dilemma bagi para pembuat gim. Di satu sisi popularitas kedua perangkat lunak itu memudahkan pertukaran informasi antar pengguna namun di sisi lain biaya langganan dianggap memberatkan.

Lalu apa solusinya? Bagi seniman animasi yang sudah terbiasa memproduksi menggunakan Blender maka pilihan yang paling ‘dekat’ adalah perangkat lunak yang bernama Godot. Awalnya Blender mempunyai fitur pembuatan gim di dalam perangkat lunaknya. Blender memutuskan untuk mengeluarkan fitur gim karena ingin berkonsentrasi pada fitur produksi animasi. Godot lalu menangkap peluang ini dan melanjutkan pembangunan fitur gim dalam perangkat lunak yang terpisah. Blender membawa semangat bahwa perangkat lunak harus bisa diakses oleh siapa saja tanpa biaya. Semangat ini juga dibawa oleh Godot. Godot sampai kapanpun akan tersedia untuk siapa saja secara gratis.

Penelitian telah menunjukkan bahwa penerapan Godot Engine dalam mengembangkan permainan edukatif dapat secara signifikan meningkatkan literasi matematika siswa. (Aprilia, Wihardjo, Mursyidah, & Jannah, 2024) Sifat interaktif dari permainan berbasis Godot secara efektif meningkatkan minat, motivasi, dan pemahaman siswa terhadap konsep matematika. Observasi mengungkapkan bahwa siswa lebih mendalami materi saat disajikan dalam format permainan interaktif, yang mengarah pada peningkatan konsisten dalam pemahaman mereka terhadap aritmatika, geometri, dan aljabar. Perbandingan antara pre-test dan post-test menunjukkan peningkatan yang signifikan dalam pemahaman siswa terhadap prinsip-prinsip matematika. Umpan balik dari guru menekankan efektivitas permainan dalam menyederhanakan konsep-konsep yang kompleks, sementara wawancara siswa menyoroti peningkatan keterlibatan dan antusiasme. Kuesioner pengguna lebih lanjut menunjukkan potensi integrasi permainan ke dalam kurikulum sekolah, menekankan perannya sebagai alat edukatif yang berharga. Secara keseluruhan, penelitian ini mengonfirmasi bahwa Godot Engine adalah sumber daya yang kuat untuk menciptakan permainan edukatif yang membuat pembelajaran matematika lebih menarik dan efektif.

Godot juga telah digunakan untuk menciptakan pembelajaran mesin. Artikel "Deep Reinforcement Learning with Godot Game Engine" oleh Mahesh Ranaweera dan Qusay H. Mahmoud menyajikan kerangka kerja komprehensif untuk menggunakan game engine Godot dalam mengembangkan model pembelajaran penguatan mendalam (DRL), yang secara khusus menargetkan penelitian sim-to-real. (Ranaweera & Mahmoud, 2024) Para penulis merancang kerangka kerja Python yang berinteraksi dengan Godot untuk melakukan tugas-tugas DRL, memungkinkan pengaturan lingkungan virtual dengan batasan, gerakan, dan objek interaktif. Sifat open-source Godot dan sistem lingkungan berbasis node yang fleksibel diakui sebagai keuntungan signifikan, memungkinkan kustomisasi yang luas dan kebebasan pengembangan dibandingkan dengan game engine yang bersifat proprietary. Penelitian ini menunjukkan penggunaan teknik randomisasi domain yang efektif untuk meningkatkan proses pembelajaran, menjembatani kesenjangan antara simulasi virtual dan aplikasi dunia nyata. Penerapan praktis dari kerangka kerja ini divalidasi melalui pengembangan dan manipulasi platform Stewart 3-DoF, yang menunjukkan potensi untuk simulasi berfidélitas tinggi dalam penelitian DRL.

Pembahasan sebelumnya cukup menjabarkan situasi yang terjadi saat ini. Terlihat bahwa Godot sepertinya bisa membantu industri gim keluar dari kebuntuan ini. Namun, status Godot yang belum populer dan belum terbukti kemampuannya, menimbulkan pertanyaan:

1. Meskipun gratis, bagaimana dengan kemudahan penggunaan perangkat lunak Godot?
2. Bagaimana dukungan yang diberikan Godot apabila pengguna menemukan kesulitan teknis?
3. Sejauh mana Godot bisa membantu produksi gim hingga akhir?

Tujuan penelitian adalah untuk memahami bagaimana Godot bekerja. Penulis akan langsung menggunakan perangkat lunak Godot dan membuat satu gim sederhana. Gim tersebut akan diunggah sebagai gim Android. Penulis juga akan meneliti kelengkapan fitur yang ditawarkan oleh Godot berikut kemudahannya.

Jika penelitian ini berhasil maka penulis bisa menawarkan Godot sebagai alternatif pengajaran kepada kampus-kampus yang mempunyai jurusan animasi atau disain komunikasi visual. Kampus-kampus tersebut bisa membuat mata kuliah produksi gim dengan perangkat lunak yang gratis.

Hal ini akan membawa ke efek yang lebih luas dimana para alumni kampus yang menjadi kreator gim pemula bisa langsung memproduksi gim tanpa harus mengumpulkan modal besar terlebih dahulu. Mereka bisa langsung mulai menelurkan gim-gim sekaligus menyebarkannya ke platform penjualan gim. Penjualan gim diharapkan mampu menghidupi para kreator dan bisa menjadi modal untuk membuat gim berikutnya.

Jika gim-gim dari para kreator ini berhasil mendapatkan peminat maka ini akan menjadi pangsa pasar baru bagi para produser film. Para produser yang memegang hak cipta karakter atau cerita terkenal bisa menyalurkan hak cipta tersebut dan memproduksi menjadi sebuah gim. Penulis membayangkan pasar gim akan lebih bergairah jika ada gim yang bercerita tentang Dilan, Warkop, Kasino Indro atau Juki.

Dengan demikian penelitian ini diharapkan membawa dampak signifikan yang membantu kelancaran industri gim di Indonesia.

Sasaran penelitian adalah membuka segala kemungkinan dari apa yang Godot tawarkan. Penulis akan mendisain sebuah gim lalu mencari tahu bagaimana disain itu bisa diwujudkan dengan Godot. Penulis berharap bisa memahami alur kerja dari Godot sekaligus mempelajari tampilan antar muka. Penulis juga akan membaca dokumentasi daring dari Godot yang akan berfungsi sebagai buku petunjuk penggunaan.

Setelah melalui penelitian ini, penulis diharapkan mampu mengoperasikan seluruh fitur-fitur Godot secara mandiri.

Penulis akan memusatkan perhatian pada proses pembuatan gim. Penulis akan berusaha mewujudkan desain gim dengan perangkat lunak Godot. Sebisa mungkin penulis tidak menggunakan perangkat lunak yang lain terkait dengan gim.

Pemusatan ini penting karena Godot menyatakan bahwa perangkat lunaknya bisa digunakan untuk membuat gim tanpa perlu terkait dengan perangkat lunak lain. Pernyataan tersebut memberikan relevansi kepada keputusan penulis untuk fokus hanya ke Godot.

Fokus pada satu perangkat lunak ini nantinya akan memberikan hanya dua hasil: Godot bisa untuk produksi gim atau tidak sama sekali.

Penulis akan meneliti cara penggunaan perangkat lunak Godot. Untuk itu penulis memerlukan petunjuk yang komprehensif untuk dapat menyelami Godot secara mendalam. Di era sumber yang terbuka, petunjuk sebuah perangkat lunak tidak lagi disalurkan melalui buku tapi disebarkan secara online dengan tajuk 'documentation'.

Pembuat perangkat lunak menuliskan petunjuk penggunaan secara detil sehingga pengguna tidak perlu bertanya lagi. Banyak hal yang dituliskan dalam sebuah dokumentasi perangkat lunak, terlebih perangkat lunak gim. Dokumentasi resmi Godot adalah sumber daya penting bagi pengembang pemula dan berpengalaman yang menggunakan mesin untuk membuat gim dan proyek interaktif.

Dokumentasi Godot ditulis selama bertahun-tahun sejak versi pertama hingga versi terakhir. Tulisannya mencakup semua aspek fungsionalitas mesin. Ini diperbarui secara berkala untuk mengikuti perkembangan dan fitur terbaru di mesin. Dokumentasi tersedia online dan dapat diakses di situs web resmi: <https://docs.godotengine.org/>.

Jika dilihat sekilas, ada beberapa pola yang bisa terbaca sekaligus menjadi pedoman cara membaca. Pola pertama adalah Navigasi dan Struktur. Dokumentasi disusun sedemikian rupa sehingga memungkinkan navigasi yang mudah dalam mencari topik tertentu. Dokumentasi terdiri dari banyak elemen tapi dibagi menjadi beberapa bagian yang berbeda, masing-masing berfokus pada aspek tertentu dari perangkat lunak seperti skrip, fisika, animasi, jaringan, dan banyak lagi.

Pola kedua adalah bilah pencarian. Seperti dokumentasi online lainnya, Godot menyediakan bilah pencarian yang memungkinkan pengguna menemukan informasi dengan cepat tentang kata kunci atau topik tertentu. Selain itu, ada indeks yang mencantumkan semua topik yang dibahas, sehingga lebih mudah menelusuri konten yang tersedia.

Pola ketiga adalah dokumen tutorial. Seringkali penjelasan secara teori tidak mencukupi pemahaman pengguna. Untuk itu, dokumentasi juga mencakup serangkaian tutorial mendetail dan panduan langkah demi langkah untuk membantu pemula bekerja dengan Godot. Tutorial ini mencakup berbagai topik, mulai dari membuat game sederhana hingga pembuatan skrip dan teknik animasi tingkat lanjut.

Gim tidak bisa lepas dari code atau kode komputer. Sepanjang dokumentasi, Anda akan menemukan banyak contoh kode di GDScript, yang merupakan bahasa skrip utama Godot. Contoh-contoh ini membantu pengguna memahami cara mengimplementasikan fitur atau fungsi tertentu dalam proyek mereka. Godot juga menyediakan kode dalam bahasa C# yang juga digunakan di perangkat lunak gim lain.

Godot lahir dari komunitas yang ingin membangun mesin gim untuk semua orang. Atas dasar itu, Dokumentasi terbuka bagi semua orang untuk memberikan ide atau usulan. Dokumentasi Godot adalah upaya kolaboratif, dan

kontribusi dari komunitas dipersilakan. Ini memungkinkan pengguna untuk berbagi pengetahuan dan keahlian mereka, meningkatkan dokumentasi untuk semua orang.

Godot juga diterjemahkan dalam berbagai bahasa. Tujuannya untuk memudahkan pengguna yang tidak bisa berbahasa Inggris. Cara ini efektif untuk memperbanyak pengguna Godot.

Dokumentasi Godot memuat seluruh API yang bisa digunakan. API adalah singkatan dari Application Programming Interface atau "Antarmuka Pemrograman Aplikasi." API adalah seperangkat aturan, protokol, dan alat yang memungkinkan aplikasi perangkat lunak yang berbeda untuk berkomunikasi dan berinteraksi satu sama lain. Dengan kata lain, API menentukan metode dan format data yang dapat digunakan pengembang untuk berinteraksi dengan komponen, layanan, atau platform perangkat lunak tertentu.

Terakhir, Dokumentasi Godot berisi Proyek Demo Resmi. Selain contoh penggunaan bahasa GDScript, Godot juga menyertakan demo. Demo ini bisa di salin temple di mesin untuk sekedar membuktikan bahwa kode yang berada di demo bisa dijalankan.

2. METODE PENELITIAN

Penulis menggunakan praktek langsung sebagai metode penelitian. Penulis akan langsung mempraktekkan cara-cara yang tersedia dan langsung membuat gim. Penelitian ini akan melewati beberapa tahap.

Tahap pertama adalah pembuatan disain gim. Penulis akan menentukan aturan permainan, karakter yang dimainkan, obyek permainan, lingkungan permainan, level permainan dan distribusi permainan. Disain akan menjadi pedoman utama di sepanjang penelitian.

Tahap kedua adalah produksi gim. Penulis akan memproduksi gim sesuai disain. Rencana pekerjaan meliputi penempatan grafis, penulisan koding, pembuatan sampel dan finalisasi gim.

Tahap ketiga adalah pengumpulan data. Data yang dikumpulkan adalah logika gim, error atau kesalahan yang terjadi, koding yang benar, cara membaca dokumentasi dan pelajaran yang bisa dipetik.

Tahap keempat adalah penarikan kesimpulan atas berbagai data dan pengalaman yang terjadi saat produksi.

Tahap kelima adalah publikasi penciptaan ke platform Android. Tujuan akhir dari penelitian ini adalah gim yang bisa diunduh dan dimainkan oleh pengguna.

Penulis meyakini bahwa setiap tahap mempunyai kesulitan sendiri. Namun kesulitan tersebut akan membawa ke penelitian dan penyelidikan yang otomatis akan menambah wawasan dan ilmu kepada penulis. Pada akhirnya penulis berharap penelitian ini akan membuat penulis makin mahir menggunakan perangkat lunak Godot.

3. PEMBAHASAN

Penulis akan mencoba membuat game dari karakter animasi berjudul 'Bilu Mela'. Animasi ini adalah karya lokal yang sudah mendunia dan sudah terdistribusi ke 20 negara. Bilu adalah seekor ikan berwarna biru yang mempunyai karakter ekstrovert sementara Mela adalah seekor ikan merah yang berkarakter introvert. Karakter ini sudah dikenal oleh banyak orang dan mempunyai basis komunitas yang besar. Berdasarkan hal tersebut, pemilihan karakter ini sudah tepat.



Gambar 1. Karakter Bilu Mela (Dokumen Penulis)

a. Awal Mula Godot

Godot dimulai sebagai proyek pribadi oleh Juan Linietsky dan Ariel Manzur. Awalnya, mereka menciptakan mesin ini untuk kebutuhan produksi game pribadi mereka di studio game indie Argentina. Sepanjang tahun, mesin tersebut dikembangkan dan ditingkatkan berdasarkan kebutuhan proyek-proyek yang mereka kerjakan.

Pada tahun 2014, setelah melihat potensi yang lebih besar dan kemungkinan manfaat bagi komunitas pengembang game yang lebih luas, Linietsky dan Manzur memutuskan untuk merilis Godot sebagai perangkat lunak open source di bawah lisensi MIT. Keputusan ini memungkinkan komunitas global untuk berkontribusi dalam pengembangan mesin, serta menggunakan dan memodifikasinya tanpa pembatasan.

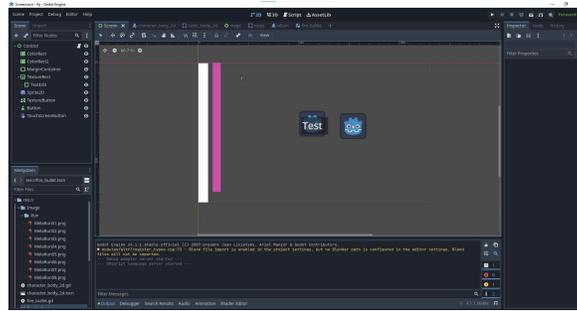
Keputusan untuk membuatnya open source terbukti sukses, dengan banyak kontributor yang bergabung dan membantu meningkatkan Godot. Sejak saat itu, mesin ini telah melihat pertumbuhan yang signifikan dalam popularitas dan fungsionalitas, dengan versi baru yang terus dikeluarkan dan fitur-fitur baru yang terus ditambahkan.

Salah satu keunggulan Godot adalah pendekatannya yang modular, yang memungkinkan pengembang untuk menyesuaikan mesin sesuai kebutuhan mereka. Selain itu, dukungan untuk berbagai platform, fleksibilitas dalam bahasa scripting, dan model lisensinya yang ramah pengembang menjadikannya pilihan populer di kalangan pengembang indie dan studio game skala kecil hingga menengah.

Sejak perilisannya sebagai open source, Godot telah menjadi salah satu mesin game open source terpopuler, dengan komunitas yang aktif dan berdedikasi yang terus berkontribusi dalam pengembangan dan pendidikan seputar mesin tersebut.

Godot memiliki sejumlah karakteristik unik yang membedakannya dari mesin game lain. Desain adegan dan node adalah inti dari pendekatan Godot. Semua hal, mulai dari karakter, kamera, hingga elemen UI, diperlakukan sebagai "node". Node-node ini dapat dikelompokkan menjadi "adegan", yang kemudian dapat diinstansiasi sebagai bagian dari adegan lain, memberikan struktur yang sangat modular dan dapat digunakan kembali.

Tampilan editor yang konsisten untuk semua platform memudahkan pengembangan lintas platform. Godot mendukung berbagai platform, termasuk Windows, macOS, Linux, Android, iOS, dan banyak lagi. Fungsionalitas drag-and-drop dan one-click deploy membuat proses penerbitan game menjadi lebih sederhana.



Gambar 2. Tampilan Muka Godot (Dokumen Penulis)

Godot memiliki sistem animasi yang kuat dan fleksibel, memungkinkan animasi hampir di setiap properti yang ada di mesin.

Meskipun ada banyak kelebihan, seperti semua perangkat, Godot mungkin tidak sempurna untuk setiap proyek atau tim. Pilihan mesin game harus didasarkan pada kebutuhan spesifik proyek, sumber daya tim, dan preferensi pengembangan.

Ketika penulis pertama kali mendownload Godot, prosesnya sangatlah mudah dan cepat. Website resmi Godot menawarkan berbagai versi dari mesin tersebut, baik yang stabil maupun versi beta. Penulis memilih untuk mendownload versi stabil terbaru. Ukuran file-nya relatif kecil jika dibandingkan dengan mesin game lainnya, jadi tidak memakan waktu lama untuk mendownload dan menginstalnya.

Salah satu hal yang penulis suka dari Godot adalah bahwa mesin ini open-source dan gratis tanpa royalti. Hal ini membuatnya sangat menarik bagi pengembang indie dan mereka yang baru memulai pembuatan game.

Setelah mendownload, penulis hanya perlu mengekstrak file dan menjalankan aplikasi Godot tanpa perlu proses instalasi tambahan. Ini sangat memudahkan, terutama bagi mereka yang ingin segera memulai tanpa harus melalui proses instalasi yang rumit.

Dokumentasi Godot adalah salah satu aspek terkuat dari mesin ini. Berdasarkan pengalaman penulis, dokumentasinya sangat lengkap, rapi, dan mudah dipahami. Dokumentasi Godot mencakup hampir semua aspek dari mesin game, mulai dari dasar-dasar, scripting, hingga fitur-fitur lanjutan. Setiap bagian dijelaskan dengan rinci dan disertai dengan contoh kode.

Karena Godot adalah proyek open-source, banyak kontributor dari komunitas yang terus-menerus memperbarui dan memperbaiki dokumentasi. Hal ini memastikan bahwa informasi yang disajikan selalu relevan dan up-to-date. Selain referensi API dan penjelasan fitur, dokumentasi Godot juga mencakup banyak tutorial dan panduan langkah demi langkah yang sangat membantu bagi pemula maupun developer berpengalaman.

Meski dokumentasi utamanya dalam bahasa Inggris, ada upaya dari komunitas untuk menerjemahkannya ke berbagai bahasa lain, memungkinkan lebih banyak pengembang dari seluruh dunia untuk memanfaatkan mesin ini. Banyak bagian dari dokumentasi yang menyertakan demo atau contoh yang interaktif, memungkinkan pengguna untuk langsung melihat hasil dari kode yang diberikan.

Namun, seperti semua dokumentasi, kadang-kadang mungkin ada bagian yang kurang jelas atau memerlukan penjelasan lebih lanjut. Di sinilah keberadaan komunitas Godot yang aktif dan forum-forum diskusi menjadi sangat berharga. Jika penulis menemui hambatan atau pertanyaan, biasanya mudah untuk menemukan jawaban atau bantuan dari komunitas.

Secara keseluruhan, penulis sangat puas dengan dokumentasi Godot dan merasa itu adalah sumber daya yang sangat berharga bagi semua pengembang yang menggunakan mesin ini.

Godot menggunakan dua bahasa pemrograman yaitu GDScript dan C#. GDScript adalah bahasa pemrograman yang dikembangkan khusus untuk mesin game Godot. Dirancang untuk mirip dengan Python dalam hal sintaks, sehingga mudah dipelajari, terutama bagi mereka yang sudah familiar dengan Python.

Tujuan utama dari GDScript adalah untuk mengoptimalkan integrasi dengan mesin Godot. Berkat desainnya yang spesifik ini, GDScript dapat bekerja dengan cepat dan efisien dalam lingkungan Godot, memastikan performa yang baik dan interaksi mulus dengan fitur-fitur mesin.

Kelebihan lain dari GDScript adalah kemudahannya dalam menulis kode. Dengan sintaks yang sederhana dan jelas, pengembang dapat fokus pada logika game tanpa harus terlalu banyak mengkhawatirkan detail-detail bahasa pemrograman yang kompleks.

Meski GDScript adalah pilihan utama untuk scripting di Godot, mesin tersebut juga mendukung bahasa lain seperti C# dan, melalui plugin, bahasa-bahasa lain seperti C++ dan Rust. Namun, bagi banyak pengembang yang baru memulai dengan Godot atau bagi mereka yang ingin prototipe cepat, GDScript sering menjadi pilihan pertama karena integrasinya yang erat dengan mesin dan kurva belajarnya yang relatif landai.

Secara keseluruhan, GDScript adalah alat yang kuat untuk pengembangan game di Godot, menawarkan kombinasi dari kemudahan penggunaan dan efisiensi yang sulit untuk ditandingi oleh bahasa scripting lainnya dalam konteks mesin ini.

b. Konsep Permainan

Pertama adalah pembuatan skenario. Skenarionya adalah Bilu Mela harus melewati sekian papan yang tersebar di udara. Bilu Mela harus melompat ke dalam papan-papan itu agar dia tidak jatuh ke dalam jurang. Bilu Mela juga harus menginjak papan dengan warna yang sama dengan tubuhnya. Jika Bilu menginjak warna biru atau Mela menginjak warna merah maka mereka akan mendapatkan poin. Setiap level mempunyai poin yang berbeda. Semakin tinggi levelnya, semakin tinggi nilai poin yang harus didapat. Bilu Mela tidak boleh menginjak papan yang mempunyai duri tajam. Jika itu terjadi, maka nilai kesehatan Bilu dan Mela akan berkurang. Nilai kesehatan Bilu Mela hanya tersedia empat poin di setiap level. Demikian skenario dari game Bilu Mela Jump. Perbedaan setiap level tergantung dari variabel jumlah papan, penempatan papan, target poin dan kecepatan karakter.

Setelah skenario tercipta, proses produksi dilanjutkan ke tahap pembuatan karakter. Style game Bilu Mela Jump adalah dua dimensi sehingga penulis membutuhkan gambar-gambar dua dimensi untuk dipasang dan dianimasikan ke dalam mesin Godot. Bilu Mela adalah karakter tiga dimensi. Penulis perlu mengubah gambar tiga dimensi menjadi gambar dua dimensi melalui proses render. Render memerlukan waktu sekitar satu menit per frame. Hasil render lalu diimpor ke dalam mesin Godot.

Penulis tidak hanya memerlukan render karakter tapi juga render papan, gambar awan dan kontur tanah yang akan dipijak oleh Bilu Mela. Penulis hanya perlu merender satu papan biru, satu papan merah, tiga gambar langit dan satu petak gambar tanah. Penulis akan menduplikasi gambar-gambar tersebut menjadi lingkungan yang utuh sesuai tuntutan skenario di setiap level.



Gambar 3. Tampilan Muka Bili Mela Jump (Dokumen Penulis)

Karakter tidak hanya terdiri dari gambar tapi juga terdiri dari logika physic atau fisika. Fisika yang dimaksud disini adalah gravitasi dan pemberian massa berat. Semua benda mempunyai massa berat yang akan ditarik oleh gravitasi. Semakin besar jumlah berat, semakin cepat gravitasi akan menarik benda tersebut. Karakter di dalam godot juga perlu diberikan massa berat dan gravitasi supaya karakter tersebut bisa loncat, jalan ataupun lari. Kecepatan dari masing-masing kegiatan itu tergantung dari seting massa berat dan gravitasi.

Perhitungan fisika dan gabungan dengan gambar akan memberikan ilusi bahwa karakter bergerak, menapak tanah, meloncat ke papan dan berjalan di atas papan dan juga jatuh ke dalam jurang.

Setelah semua gambar yang diperlukan sudah tersedia, langkah berikutnya adalah membuat logika game. Skenario telah menentukan logika yang harus dipenuhi misalnya logika Bili mendapat poin ketika menginjak papan biru. Logika ini bisa dicapai dengan pembuatan collision design atau disain tabrakan.

Menyiapkan collision atau deteksi tabrakan dalam game yang penulis kembangkan selalu menjadi proses yang menarik dan terkadang menantang. Dalam Godot, hal ini melibatkan pemahaman yang baik mengenai sistem "Physics Engine" dan node-node yang terkait dengan deteksi dan respons terhadap tabrakan.

Pertama, penulis perlu menentukan bentuk collision dari objek-objek dalam game, apakah itu karakter, musuh, atau rintangan. Godot menyediakan beberapa pilihan untuk bentuk collision, seperti "CollisionShape2D" untuk game 2D atau "CollisionShape" untuk game 3D. Bentuk-bentuk ini harus disesuaikan dengan geometri dari objek-objek dalam game untuk menciptakan deteksi tabrakan yang akurat.

Untuk karakter atau objek yang bergerak, penulis juga perlu menetapkan logika untuk bagaimana mereka bereaksi ketika tabrakan terjadi. Apakah mereka akan berhenti bergerak, memantul kembali, atau melakukan aksi tertentu? Ini melibatkan scripting menggunakan GDScript, dimana penulis menulis fungsi yang menentukan aksi yang terjadi ketika collision dilakukan.

Di beberapa kasus, misalnya untuk area yang memicu event atau zona musuh, penulis mungkin menggunakan "Area2D" (untuk game 2D) atau "Area" (untuk game 3D) untuk mendeteksi kapan karakter masuk atau keluar dari sebuah wilayah, tanpa mempengaruhi fisika karakter itu sendiri.

Salah satu tantangan yang sering dihadapi penulis adalah memastikan bahwa bentuk collision diatur dengan benar dan logika tabrakan di-script dengan efisien untuk menghindari bug atau masalah kinerja. Penulis perlu berulang kali melakukan uji coba, menyesuaikan bentuk collision dan mengoptimalkan logika scripting untuk memastikan pengalaman bermain yang lancar dan menyenangkan.

Juga, penanganan tabrakan untuk objek yang kompleks atau animasi yang melibatkan banyak bagian yang bergerak bisa menjadi cukup kompleks dan memerlukan pendekatan kreatif untuk mengatasi.

Secara keseluruhan, meskipun menetapkan dan menangani collision bisa menjadi proses yang rumit, Godot menyediakan alat yang cukup bermanfaat dan fleksibel untuk membantu pengembang melalui tantangan ini.

Godot memiliki konsep "signals" yang merupakan mekanisme efisien dan kuat untuk menghandle event dan komunikasi antar node dalam suatu scene. Signals sangat penting dalam menciptakan gameplay yang dinamis dan interaktif tanpa membuat hubungan yang terlalu erat antar objek, yang bisa mengakibatkan struktur kode yang sulit dikelola.

Sebagai contoh, dalam pengembangan suatu game, penulis mungkin memiliki karakter yang harus memberi tahu UI ketika nyawa (health points) karakter tersebut berubah. Sehingga, UI dapat memperbarui tampilannya sesuai dengan nyawa yang tersisa.

Pertama, penulis mendefinisikan sinyal pada karakter tersebut, yang biasanya dilakukan dalam skrip GDScript dengan menggunakan kata kunci "signal". Misalnya, signal bisa dinamai "health_changed" dan mempunyai parameter untuk mengirimkan nilai nyawa yang baru.

```
signal health_changed(new_health)
```

Kemudian, di dalam kode logika karakter, ketika nyawa berubah (misalnya, ketika karakter menerima kerusakan), penulis akan "emit" atau memancarkan signal tersebut, menyertakan nilai nyawa yang baru sebagai parameter.

```
emit_signal("health_changed", health)
```

Selanjutnya, dalam node UI, penulis akan "connect" atau menghubungkan signal tersebut dengan suatu metode. Metode ini akan dijalankan ketika signal dipancarkan, memungkinkan UI untuk merespons perubahan nyawa dengan cara yang diinginkan, seperti memperbarui tampilan bilah kesehatan.

```
character.connect("health_changed", self,
"_on_health_changed") func _on_health_changed(new_health):
# Kode untuk memperbarui UI berdasarkan nilai `new_health`
```

Penulis menemukan bahwa penggunaan signals sangat membantu dalam mengorganisir kode dan membuat logika game lebih modular dan reusable. Selain itu, ini memungkinkan penulis untuk lebih mudah mengisolasi dan memperbaiki bug atau melakukan iterasi pada mekanika game tertentu, karena logika yang terkait dengan suatu event terlokalisasi dalam metode yang terhubung dengan signal tersebut.

Tantangan yang penulis temui biasanya melibatkan mengatur sinyal untuk objek yang diinstansiasi secara dinamis atau pengelolaan sinyal pada scene yang kompleks. Ini mungkin memerlukan penggunaan metode yang lebih canggih, seperti penggunaan group atau pengelolaan state untuk menghubungkan dan memutuskan sinyal pada waktu yang tepat selama runtime.

Secara keseluruhan, pengalaman menggunakan signals dalam Godot telah memberikan penulis alat yang sangat kuat untuk membuat game yang interaktif dan responsif sambil mempertahankan struktur kode yang bersih dan terorganisir.

c. Tampilan Antar Muka

Penulis memasuki tahap pembuatan desain antarmuka. Di sini penulis merancang tombol untuk aktifitas loncat. Tombol ini akan membuat karakter Bilu atau Mela meloncat dan berpindah platform. Tombol ini mengirimkan signal yang akan mengeksekusi fungsi koordinat Y di karakter Bilu atau Mela. Masing-masing karakter mempunyai tombol sendiri. Penempatan tombol sesuai dengan karakter Bilu Mela. Bilu selalu berada di sebelah kiri dan Mela selalu berada di sebelah kanan layar.

Desain antarmuka selanjutnya adalah pembuatan papan skor yang menunjukkan target papan yang harus diinjak oleh Bilu dan Mela. Setiap Bilu berhasil menginjak papan warna biru maka nilai di papan skor akan berkurang. Dengan demikian, pemain tahu berapa papan lagi yang harus diinjak oleh Bilu. Hal yang sama berlaku untuk Mela.

Selanjutnya adalah pembuatan papan skor kesehatan. Papan ini menunjukkan jumlah kesempatan bagi Bilu dan Mela untuk salah menginjak papan berwarna hitam yang ada jebakannya. Skor kesehatan berjumlah empat dan berkurang setiap injakan di papan hitam. Metode skor kesehatan sama dengan metode skor target dimana keduanya mengirimkan signal ke fungsi yang mengatur penilaian.

d. Pembuatan Level

Setelah peraturan basic sudah selesai dikerjakan, penulis lalu merancang level permainan. Penulis berkeinginan untuk membuat game yang tidak menggunakan instruksi berlebihan. Penulis ingin pemain game untuk memahami cara bermain dengan cara try and error. Oleh karena itu penulis membuat level pertama dengan sangat gampang dan intuitif. Metode ini diyakini memudahkan pemain untuk paham permainan meski tanpa petunjuk tertulis. Pelajaran dengan simbol dan tanpa tulisan ini memberikan keuntungan saat penulis menjual game ke berbagai negara. Penulis tidak memerlukan alih bahasa yang banyak.

Level satu sudah tercipta dan tinggal melanjutkan ke level selanjutnya. Konsep peningkatan level adalah makin banyak target dan makin menantang. Level satu mempunyai satu target, level dua memiliki dua target, level tiga ada empat target dan seterusnya. Peningkatan level kerumitan dinilai mempunyai dampak yang memotivasi. (Cieslak, 2021)

e. Integrasi Google Play Service

Setelah pembuatan level selesai, langkah selanjutnya adalah melakukan integrasi game dengan Google Play Service. Google Play Service adalah layanan dari Google yang memberikan kemudahan bagi para produsen game untuk menghitung statistik permainan antar pengguna. Statistik yang dihitung adalah pemain dengan skor terbanyak, pangkat pemain dan klasemen pemain. Ide dasar layanan ini adalah membuat suasana kompetitif antar pemain saat memainkan game yang sama. Pemain diharapkan terpacu untuk memenangkan game lebih banyak dibanding pemain lain.

Ketika penulis pertama kali mencoba mengintegrasikan Google Play Game Service dengan Godot, ada beberapa tantangan yang dihadapi, namun dengan bantuan dokumentasi dan komunitas, prosesnya menjadi lebih mudah.

Awalnya, penulis harus mencari plugin yang sesuai untuk integrasi ini. Ada beberapa pilihan plugin yang dibuat oleh anggota komunitas, yang memudahkan integrasi layanan Google Play Game ke dalam proyek Godot. Setelah menemukan plugin yang tepat, penulis perlu memasukkannya ke dalam proyek dan mengonfigurasikannya.

Salah satu langkah yang cukup memakan waktu adalah proses konfigurasi di konsol pengembang Google Play. Penulis harus memastikan bahwa ID aplikasi, ID layanan, dan semua informasi lainnya sesuai dengan yang ada di proyek Godot. Kesalahan kecil dalam konfigurasi ini dapat mengakibatkan layanan tidak berfungsi dengan benar.

Setelah konfigurasi selesai, penulis mulai menguji fungsionalitas, seperti leaderboard dan prestasi. Di sinilah penulis menemui beberapa hambatan. Terkadang, layanan tidak merespons seperti yang diharapkan atau terjadi keterlambatan dalam pembaruan poin atau pencatatan prestasi. Namun, dengan sedikit debugging dan referensi ke dokumentasi, masalah-masalah ini dapat diatasi.

Sebuah aspek yang penulis hargai dari Godot adalah kemudahannya dalam melakukan debugging. Godot menyediakan banyak informasi log yang membantu penulis dalam melacak dan memperbaiki masalah yang mungkin muncul saat integrasi.

Keseluruhan proses memang memerlukan sedikit kesabaran dan pemahaman mendalam tentang kedua platform, yaitu Godot dan Google Play Game Service. Namun, setelah berhasil diintegrasikan, fitur-fitur yang ditawarkan oleh layanan ini memperkaya pengalaman pengguna dan meningkatkan keterlibatan mereka dengan game.

Dalam pengalaman penulis, meskipun ada tantangan di awal, hasil akhirnya sangat memuaskan. Integrasi Google Play Game Service dengan Godot membuka banyak peluang untuk meningkatkan kualitas dan daya tarik game bagi pemain.

f. Mengintegrasikan In-app Purchases (IAP) dengan Godot

Mengintegrasikan in-app purchases (IAP) dengan Godot merupakan pengalaman yang menarik dan, pada saat yang sama, memberikan beberapa tantangan. Berikut cerita tentang pengalaman penulis saat melakukan integrasi tersebut:

Ketika penulis memulai proses integrasi IAP dengan Godot, hal pertama yang dilakukan adalah mencari plugin yang tepat. Ada beberapa pilihan di luar sana, namun penulis memilih salah satu yang paling direkomendasikan oleh komunitas dan memiliki dokumentasi yang jelas.

Setelah memilih plugin, proses integrasi dimulai dengan menambahkannya ke dalam proyek Godot dan mengonfigurasikannya sesuai dengan kebutuhan game. Salah satu bagian yang cukup kompleks adalah memastikan bahwa semua item yang ingin dijual sudah dikonfigurasi dengan benar di konsol pengembang, baik itu Google Play untuk Android atau App Store untuk iOS.

Selanjutnya, penulis bekerja pada implementasi kode di sisi Godot. Ini melibatkan pemanggilan fungsi-fungsi tertentu saat pengguna memutuskan untuk melakukan pembelian atau saat perlu memvalidasi transaksi yang telah dilakukan.

Pada tahap awal pengujian, beberapa masalah muncul. Terkadang, respons dari layanan IAP tidak segera muncul atau transaksi gagal tanpa alasan yang jelas. Ini memerlukan banyak waktu debugging dan penyelidikan untuk menemukan sumber masalahnya. Penulis seringkali harus kembali ke dokumentasi atau forum komunitas Godot untuk mencari solusi atau saran terkait masalah yang dihadapi.

Namun, setelah beberapa iterasi dan perbaikan, sistem IAP mulai berfungsi dengan lancar. Hal terpenting yang penulis pelajari dari proses ini adalah pentingnya melakukan pengujian yang ekstensif, memastikan setiap skenario pembelian dicoba, dan selalu memeriksa log dan respons dari sistem untuk mendeteksi masalah apa pun.

Meskipun ada hambatan di sepanjang jalan, integrasi IAP dengan Godot akhirnya berhasil dan memberikan nilai tambah signifikan untuk game. Ini memungkinkan penulis untuk menawarkan konten premium atau fitur tambahan kepada pengguna, meningkatkan potensi pendapatan dari game tersebut.

g. Publikasi di Play Store

Mempublish game di Play Store menggunakan Godot adalah proses yang memerlukan perencanaan dan ketelitian. Berikut pengalaman penulis saat melalui proses tersebut:

Pertama-tama, penulis memastikan game sudah dalam kondisi yang siap untuk dirilis. Ini termasuk memeriksa semua bug, memastikan semua aset sudah pada tempatnya, dan melakukan pengujian ekstensif pada berbagai perangkat Android.

Kemudian, penulis mulai dengan proses ekspor dari Godot ke format APK. Godot memiliki opsi ekspor yang cukup sederhana untuk Android, tetapi memerlukan beberapa setup awal, seperti menyiapkan keystore untuk penandatanganan aplikasi dan mengonfigurasi template ekspor Android.

Setelah APK siap, langkah selanjutnya adalah mempersiapkan halaman Play Store untuk game. Ini memerlukan pembuatan ikon, screenshot, dan deskripsi yang menarik untuk menarik perhatian pengguna. Penulis juga memastikan bahwa semua informasi metadata lainnya, seperti kategori game, usia target, dan hak akses yang diperlukan, sudah diatur dengan benar.

Salah satu aspek yang cukup menantang adalah memastikan game memenuhi semua kebijakan dan panduan konten Google Play. Penulis harus membaca dan memastikan game tersebut tidak melanggar aturan apa pun agar tidak dihapus atau mendapatkan sanksi.

Setelah semuanya siap, penulis mengunggah APK ke konsol pengembang Play Store dan mengirimkannya untuk ditinjau. Ada masa menunggu selama beberapa jam hingga beberapa hari sebelum game benar-benar live di Play Store. Selama waktu ini, penulis seringkali merasa cemas dan menantikan feedback dari tim tinjauan Google.

Ketika game akhirnya live, itu adalah saat yang sangat memuaskan. Namun, pekerjaan tidak berhenti di sana. Penulis terus memonitor ulasan pengguna, feedback, dan data analitik untuk memahami bagaimana game tersebut diterima oleh pemain dan apakah ada masalah teknis atau konten yang perlu diperbaiki.

Secara keseluruhan, mempublish game di Play Store adalah proses yang memerlukan perhatian terhadap detail dan kesabaran. Namun, melihat game yang dibuat sendiri tersedia untuk diunduh dan dimainkan oleh pengguna di seluruh dunia adalah salah satu momen paling memuaskan dalam perjalanan pengembangan game.

h. Advertising The Game With Google Ads

Mengiklankan game dengan Google Ads memberikan pelajaran berharga bagi penulis tentang dunia pemasaran digital. Berikut pengalaman penulis selama menggunakan platform ini:

Awalnya, penulis memulai dengan menetapkan tujuan kampanye. Apakah itu meningkatkan kesadaran tentang game, mendapatkan unduhan lebih banyak, atau mendorong pembelian dalam aplikasi, menentukan tujuan jelas sangat penting untuk mengarahkan strategi iklan.

Selanjutnya, penulis perlu memahami audiens target. Google Ads menyediakan banyak opsi penargetan, termasuk demografi, minat, perilaku, dan lebih lagi. Memilih target yang tepat sangat penting untuk memastikan iklan dilihat oleh orang yang paling mungkin tertarik dengan game.

Pembuatan iklan itu sendiri adalah proses kreatif. Penulis memastikan untuk membuat grafik yang menarik dan menulis teks iklan yang menarik perhatian serta menjelaskan apa yang membuat game ini unik. Pemilihan kata kunci juga penting, terutama untuk kampanye berbasis pencarian, di mana iklan muncul berdasarkan istilah pencarian yang dimasukkan pengguna.

Setelah kampanye dijalankan, salah satu bagian paling penting adalah pemantauan dan optimalisasi. Google Ads menyediakan analitik yang mendalam, memungkinkan penulis untuk melihat metrik seperti jumlah klik, konversi, dan biaya per klik. Dengan informasi ini, penulis dapat menyesuaikan kampanye untuk meningkatkan efisiensi dan ROI.

Tentu saja, ada kurva belajar saat menggunakan Google Ads, terutama jika seseorang baru mengenal pemasaran digital. Penulis menghadapi beberapa hambatan, seperti menentukan anggaran harian yang tepat, memahami laporan analitik, atau menyesuaikan penargetan untuk mendapatkan hasil yang lebih baik. Namun, dengan waktu dan praktik, serta memanfaatkan sumber daya dan tutorial yang tersedia, penulis menjadi lebih mahir dalam mengelola kampanye iklan.

Keseluruhan proses memberi penulis pemahaman yang lebih baik tentang bagaimana pemasaran digital bekerja dan bagaimana mempromosikan game atau aplikasi dengan cara yang efektif. Meskipun menantang, pengalaman tersebut sangat berharga dan membantu meningkatkan visibilitas game di pasar yang sangat kompetitif.

4. SIMPULAN

Godot ternyata mudah digunakan, terutama bagi pemula, dengan antarmukanya yang dirancang secara intuitif. Platform ini menawarkan fitur drag-and-drop yang mempermudah dalam merancang adegan dan mengelola objek. Untuk penulisan skrip, Godot menggunakan GDScript, bahasa yang serupa dengan Python, membuatnya lebih mudah untuk dipelajari. Namun, perlu dicatat bahwa ada kurva belajar dalam menggunakan fitur-fitur canggih dari Godot, seperti halnya pada perangkat lunak pengembangan game lainnya.

Sebagai software open source, Godot umumnya mendapat dukungan dari komunitasnya. Pengguna dapat mengakses berbagai forum, dokumentasi, tutorial, dan diskusi grup untuk bantuan teknis. Tim pengembang Godot juga secara rutin merilis pembaruan untuk memperbaiki bug dan menambah fitur baru. Akan tetapi, dukungan langsung dari tim Godot mungkin tidak segera atau sepersonal layanan dukungan dari perangkat lunak berbayar.

Godot cukup mampu untuk handle produksi game yang lengkap, dari game indie sederhana hingga proyek yang lebih kompleks. Platform ini mendukung pengembangan game 2D dan 3D, dilengkapi dengan mesin fisika terintegrasi dan sistem animasi. Banyak game sukses telah dikembangkan dan dirilis dengan menggunakan Godot.

Pada pengolahan game dari karakter animasi Bilu Mela, penulis berhasil membuat game yang bisa dimainkan dan tanpa error yang berarti. Game ini sudah diunggah ke Play Store dan bisa diunduh oleh siapa saja. Proses pembuatan cukup lancar karena forum tanya jawab tentang Godot sudah banyak tersebar di internet.

Dari sisi komersial, Godot sudah bisa berintegrasi dengan platform Google Play Store tanpa kendala yang berarti. Google juga sudah memberikan petunjuk khusus bagi para pengembang Godot. Tentu saja ini sangat besar artinya bagi komunitas Godot.

Secara umum, Godot merupakan pilihan yang tepat untuk pengembang game yang mencari solusi tanpa biaya dengan dukungan kuat dari komunitasnya.

5. DAFTAR PUSTAKA

Tim Pusat Animasi IKJ. (2016). *Laporan FGD Sub Sektor Animasi Indonesia*. Jakarta: IKJ.

Aprilia, S. D., Wihardjo, E., Mursyidah, I. L., & Jannah, E. S. (2024). Implementation of Godot Engine in Developing Mathematical Literacy Educational Games. *Research Gate*, 1-15.

Ranaweera, M., & Mahmoud, Q. (2024). Deep Reinforcement Learning with Godot Game Engine. *Electronics*, 985. <https://doi.org/10.3390/electronics13050985>

Godot Developer. (n.d.). Godot Documentation. Retrieved September 5, 2023, from <https://docs.godotengine.org/en/stable/index.html>

Chandler, Heather & Chandler, Rafael, (2011). *Fundamentals of Game Development*. Jones & Bartlett Learning. Canada.

Myler, Caroline, (2004). *Digital Storytelling: A Creator's Guide to Interactive Entertainment*. Focal Press. New York.

M. O. Riedl and A. Zook, (2013). AI for game production. *IEEE Conference on Computational Intelligence in Games (CIG)*, Niagara Falls, ON, Canada. Doi: 10.1109/CIG.2013.6633663.

- Pramono, Andy; Pujiyanto; Puspasari, Betty Dewi; Dhanti, Nadhifa Salmatia, (2021). Character Thematic Education Game "AK@R" of Society Themes for Children with Malang-Indonesian Visualize. *International Journal of Instruction*.
- Vandewalle, A., Malliet, S., & Demoen, K. (2023). Playing with Character: A Framework of Characterization in Video Games. *Games and Culture*, 0(0). <https://doi.org/10.1177/15554120231179496>
- Bethke, E. (2003). *Game Development and Production*. United Kingdom: Wordware Pub.
- Chandler, H., Chandler, R. (2011). *Fundamentals of Game Development*. United States: Jones & Bartlett Learning.
- Game Production Studies. (2021). Netherlands: Amsterdam University Press.
- Irish, D. (2005). *The Game Producer's Handbook*. Switzerland: Thomson Course Technology.
- Lemarchand, R. (2021). *A Playful Production Process: For Game Designers (and Everyone)*. United States: MIT Press.
- Cieslak, K. (2021, September 10). Try Evidence Home. Retrieved October 21, 2023, from Try Evidence: <https://tryevidence.com/blog/game-difficulty-level-how-to-choose-it-properly/>
- Elindawati, R. (2019). DAMPAK GAME ONLINE BAGI PENDIDIKAN ANAK. *Al-Wardah*, <https://journal.iain-ternate.ac.id/index.php/alwardah/article/view/217>.
- Godot Developer. (n.d.). Godot Documentation. Retrieved September 5, 2023, from <https://docs.godotengine.org/en/stable/index.html>
- Hardiansyah, Z. (2022, 08 23). Kompas Tekno. Retrieved 06 18, 2023, from Kompas.com: <https://tekno.kompas.com/read/2022/08/22/18150027/6-kampus-di-indonesia-yang-punya-mata-kuliah-atau-jurusan-game>
- Montes, P. (2018, April 9). Hypebeast Gaming. Retrieved September 8, 2023, from Hypebeast: <https://hypebeast.com/2018/4/grand-theft-auto-v-6-billion-dollars-sales>
- Video Game Policy: Production, Distribution, and Consumption. (2015). United Kingdom: Taylor & Francis.
- Sward, J., Moore, M. E. (2007). *Introduction to the Game Industry*. United Kingdom: Pearson Prentice Hall.
- Cooper, J. (2021). *Game Anim: Video Game Animation Explained*. United States: CRC Press.
- Agarwal, Deevyankar; Sharma, Moolchand; Sharma, Prerna; (2021). *Deep Learning in Gaming and Animations: Principles and Applications*. United States: CRC Press.
- Game Production Studies. (2021). Netherlands: Amsterdam University Press.