

Production Efficiency of 2D Pixel Art Through 3D Model Transformation Using Node-Based Compositing

Efisiensi Produksi Pixel Art 2D melalui Transformasi Model 3D Berbasis Compositing Node

Najwa Nur Azizah Sanusi¹, Samuel Gandang Gunanto², Mohammad Arifian Rohman³

^{1,2,3}Program Studi Animasi, Institut Seni Indonesia Yogyakarta

Email: najwanura99@gmail.com¹, gandang@isi.ac.id², arifian@isi.ac.id³

Abstract

The game development industry demands high efficiency in visual asset production without compromising aesthetics. While 2D pixel art maintains consistent market appeal, its conventional manual frame-by-frame method is structurally limited in terms of consistency and scalability. This research designs and validates a deterministic hybrid pipeline for 3D-to-2D pixel art character asset production using Blender's Compositing Node, employing a Research and Development (R&D) approach combined with experimental methods. The pipeline integrates Rasterization Algorithms, Grid Snapping, and Downsampling Theory, with production time, visual quality, and reusability were evaluated objectively using screen-recording analysis. Three key findings were obtained. First, the pixel conversion formula $\text{Value} = \text{Render Resolution} \div \text{Target Resolution}$ was mathematically validated across five resolution scenarios, yielding perfect matrix partitions without pixel residue; in the critical case (1080 px $\rightarrow$ 64 px), spatial deviation was only 0.74%, well below the industry tolerance threshold of 5–10%. Second, the pipeline achieved a Break-Even Point (BEP) at $n \approx 0.97$, indicating that the reusable component investment (3.21 hours) was recovered before the first character was completed, with efficiency gains growing from 1.66% (1 asset) to 49.73% (10 assets). Third, the identified visual deviation of 1–2 pixels (1.56%–3.13%) falls below the Just Noticeable Difference (JND) threshold of 5–10%, confirming that pixel art aesthetics including outline sharpness, inter-frame proportional consistency, and silhouette readability remain fully intact. These results demonstrate that the proposed pipeline offers a reproducible, scalable, and AI-independent solution for mass production of pixel art character assets.

Keywords: *pixel art, 3d-to-2d conversion, compositing node, blender, reusability, game asset production*

Abstrak

Industri pengembangan game menuntut efisiensi tinggi dalam produksi aset visual tanpa mengorbankan estetika. Meskipun gaya *pixel art* 2D mempertahankan daya tarik pasar yang konsisten, metode produksi manual *frame-by-frame* konvensional memiliki keterbatasan struktural dalam hal konsistensi dan skalabilitas. Penelitian ini merancang dan memvalidasi *deterministic hybrid pipeline* konversi *3D-to-2D* untuk produksi aset karakter *pixel art* menggunakan *Compositing Node* Blender, dengan menerapkan metode *Research and Development* (R&D) yang dikombinasikan dengan pendekatan eksperimental. Pipeline mengintegrasikan algoritma rasterisasi (*Rasterization Algorithms*), *Grid Snapping*, dan

Downsampling Theory, sementara efisiensi waktu, kualitas visual, dan *reusability* diukur secara objektif melalui perekaman layar. Tiga temuan utama dihasilkan. Pertama, formula konversi resolusi piksel $Value = Resolusi\ Render \div Resolusi\ Target$ terbukti valid secara matematis pada lima skenario resolusi dengan menghasilkan partisi matriks sempurna tanpa *pixel residue*; pada kasus kritis (1080 px $\rightarrow$ 64 px), deviasi spasial hanya 0,74%, jauh di bawah ambang toleransi industri 5–10%. Kedua, *pipeline* mencapai *Break-Even Point* (BEP) pada $n \approx 0,97$, yang berarti investasi komponen *reusable* (3,21 jam) telah terbayar sebelum karakter pertama selesai diproduksi, dengan peningkatan efisiensi dari 1,66% (1 aset) hingga 49,73% (10 aset). Ketiga, deviasi visual 1–2 piksel yang teridentifikasi (1,56%–3,13%) berada di bawah ambang batas *Just Noticeable Difference* (JND) sebesar 5–10%, membuktikan bahwa kualitas estetika *pixel art*, meliputi ketajaman *outline*, konsistensi proporsi antar *frame*, serta keterbacaan siluet karakter tetap terjaga sepenuhnya. Temuan ini membuktikan bahwa *pipeline* yang dikembangkan merupakan solusi yang dapat direproduksi, terukur, dan tidak bergantung pada AI untuk produksi massal aset karakter *pixel art*.

Kata Kunci: *pixel art, konversi 3d-to-2d, compositing node, blender, reusability, produksi aset game*

1. PENDAHULUAN

Industri pengembangan game terus mengalami pertumbuhan pesat yang menuntut efisiensi tinggi dalam produksi aset visual tanpa mengorbankan estetika. Jumlah game indie yang dirilis melalui platform *Steam* meningkat lebih dari tiga kali lipat dalam satu dekade terakhir, memperlihatkan intensifikasi persaingan yang memerlukan solusi produksi semakin terukur dan dapat direplikasi. Di tengah tekanan tersebut, *pixel art 2D* mempertahankan popularitas dan daya tarik pasarnya secara konsisten, tidak sekadar menawarkan nilai nostalgia, tetapi juga menjadi identitas visual yang kuat bagi banyak proyek game modern karena kemampuannya mempertahankan keterbacaan visual pada resolusi rendah sekaligus meminimalkan beban teknis performa pada perangkat.

Namun dalam praktik produksinya, pembuatan aset *pixel art* secara manual menggunakan metode *frame-by-frame* menghadapi kendala struktural yang signifikan. Proses menggambar ulang setiap gerakan karakter piksel demi piksel sangat rentan terhadap inkonsistensi proporsi anatomi, distorsi visual antar *frame*, serta keterbatasan volume aset yang dapat diproduksi dalam tenggat waktu yang ketat. Ketergantungan metode ini pada kemahiran motorik individual kreator menjadikan kualitas *output* bersifat fluktuatif, serta menghambat *reusability* yang sistematis ketika diperlukan variasi karakter baru.

Upaya mengatasi hambatan tersebut mendorong optimalisasi perangkat lunak *open resources* sebagai alternatif terhadap perangkat lunak komersial yang mahal. Dalam hal ini, Blender 3D terbukti berperan sebagai katalis efisiensi teknis dan kreatif bagi pengembang game melalui dukungan ekosistemnya yang komprehensif (Lee, 2025). Secara paralel, *workflow* animasi berbasis transformasi *3D-to-2D* dalam *software game Unity* terbukti mampu menghasilkan animasi berkualitas lebih baik dengan waktu yang lebih singkat (K. Wang, 2025). Di sisi lain, tren global juga didominasi eksplorasi generasi aset grafis berbasis kecerdasan buatan (AI/*deep learning*) untuk mencapai

otomasi produksi, dengan indikator seperti SSIM, PSNR, dan *Inception Score* (IS) yang telah divalidasi sebagai metrik objektif evaluasi kualitas aset game (Ribeiro et al., 2025). Fukaya et al. (2023) bahkan memetakan seluruh spektrum metode ini secara komprehensif, mulai dari prosedural hingga generatif-AI, sebagai representasi mutakhir (*state of the art*) lanskap teknologi produksi aset game. M. Wang (2024) mengkaji bagaimana kemajuan integrasi 2D dan 3D secara fundamental mengubah paradigma produksi game menuju pengalaman yang lebih imersif.

Meskipun demikian, terdapat *research gap* yang signifikan dalam literatur terkini. Pertama, penelitian transisi *3D-to-2D* sebelumnya (K. Wang, 2025) belum menawarkan sistem yang terukur secara kuantitatif, tidak terdapat metrik waktu produksi eksplisit maupun validasi matematis resolusi piksel sebagai fondasi teknis *pipeline*. Kedua, validasi Blender sebagai instrumen konversi ke *pixel art*, khususnya melalui mekanisme *Compositing Node*, belum dilakukan (Lee, 2025). Ketiga, dominasi riset generatif berbasis AI (Fukaya et al., 2023; Ribeiro et al., 2025) serta kajian integrasi 2D-3D dari perspektif desain (M. Wang, 2024) secara struktural mengesampingkan pendekatan deterministik non-AI yang terukur berdasarkan perekaman waktu nyata.

Untuk mengisi kesenjangan penelitian tersebut, artikel ini berkontribusi melalui perancangan dan pengujian *deterministic hybrid pipeline* yang lengkap, mulai dari pemodelan 3D, transformasi melalui *Compositing Node* Blender, hingga otomasi *spritesheet* berbasis *command-line interface*. *Pipeline* mengintegrasikan algoritma *rasterisasi*, *grid snapping*, teori *downsampling*, dan mekanisme pikselasi berbasis kompresi statistik untuk merasionalkan transformasi karakter 3D menjadi *pixel art 2D* secara matematis dan terukur. Penelitian ini menyajikan: (1) validasi matematis rumus konversi resolusi piksel, (2) data empiris waktu produksi nyata berbasis rekaman *chronological timer*, serta (3) pembuktian *break-even point* yang memvalidasi skalabilitas *pipeline* untuk produksi massal.

Berdasarkan latar belakang tersebut, penelitian ini bertujuan untuk menjawab tiga pertanyaan utama: (1) Bagaimana rancangan alur kerja sistematis menggunakan pemodelan 3D untuk menghasilkan aset karakter *pixel art 2D* yang konsisten? (2) Seberapa besar efisiensi waktu yang dapat dicapai melalui metode transformasi *3D-to-2D* dibandingkan dengan pembuatan *pixel art* secara manual? (3) Bagaimana tingkat konsistensi proporsi dan kualitas visual antar *frame* animasi yang dihasilkan, serta seberapa signifikan deviasi visual yang terjadi secara perseptual?

2. METODE PENELITIAN

Penelitian ini menggunakan metode *Research and Development* (R&D) yang dikombinasikan dengan pendekatan eksperimental untuk merancang, mengembangkan, dan menguji efektivitas *pipeline* konversi otomatis dari model 3D menjadi aset pixel art 2D. Sistem konversi diimplementasikan menggunakan Blender 3D dengan mengintegrasikan empat landasan teori utama, yaitu *Rasterization Algorithms*, *Grid Snapping*, *Downsampling Theory*, serta mekanisme pikselasi berbasis kompresi statistik per blok piksel.

Data eksperimen dikumpulkan melalui teknik perekaman layar (*screen recording*)

selama proses pengerjaan guna mengukur efisiensi waktu, kualitas visual, dan *reusability* secara objektif. Pengukuran efisiensi produksi mengadopsi metode Perbandingan Waktu Absolut (*Absolute Time Comparison*), yang berfokus pada analisis *micro-tasking* di dalam perangkat lunak untuk membedakan dua komponen waktu secara tegas:

a. *Static Cost* (Biaya Waktu Tetap)

Tahapan kerja dalam *pipeline* 3D yang hanya perlu dieksekusi satu kali di awal produksi sebagai master *blueprint* atau templat biner. Komponen ini tidak lagi memerlukan pengulangan pada produksi karakter berikutnya, sehingga berkontribusi 0 jam terhadap waktu produksi variasi baru (Toftedahl & Engström, 2019)

b. *Variable Cost* (Biaya Waktu Dinamis)

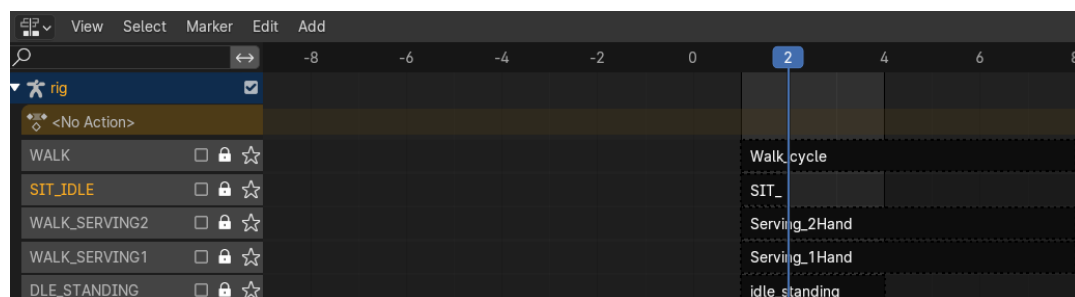
Tahapan kerja spesifik yang harus selalu dikerjakan ulang pada setiap variasi karakter baru, seperti penyesuaian anatomi rambut, desain pakaian unik, serta waktu *baking* dan *rendering output*.

Penelitian ini juga menggunakan Blender 4.0 sebagai perangkat lunak utama, dengan memanfaatkan dua fitur, yaitu *Non-Linear Animation* (NLA) untuk efisiensi animasi dan *Compositing* untuk konversi gaya visual. Eksperimen dijalankan pada perangkat keras dengan spesifikasi prosesor Intel Core i7-13650HX (20 CPU, ± 2.6 GHz), RAM 16GB, dan kartu grafis NVIDIA GeForce RTX 4050.

Konversi model 3D menjadi aset *pixel art* 2D pada penelitian ini dilakukan melalui tiga tahap utama:

a. Penyusunan *reusable animation* dengan NLA

Satu *base model* dibuat dan diberi *rigging*, lalu seluruh animasi gerak (berjalan, *idle*, makan, dan sebagainya) dibuat dan disimpan sebagai *action* pada *NLA Editor*. *Base model* ini kemudian dimodifikasi sesuai desain tiap karakter, sementara animasi tetap bisa dipakai ulang hanya dengan mengaktifkan *checkbox* pada *track* animasi yang diinginkan, tanpa perlu membuat ulang animasi untuk setiap karakter baru.

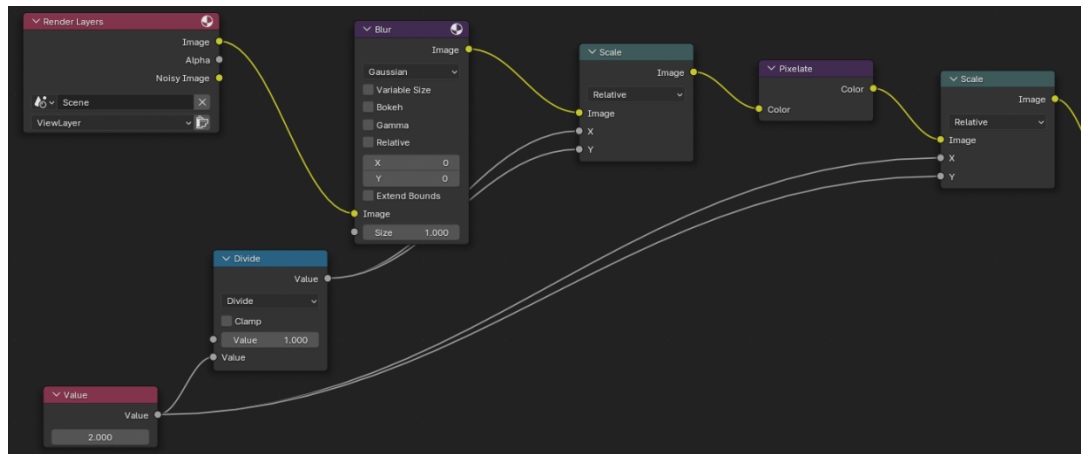


Gambar 1. Tampilan Non-Linear Animation (NLA)
(Sanusi, 2026)

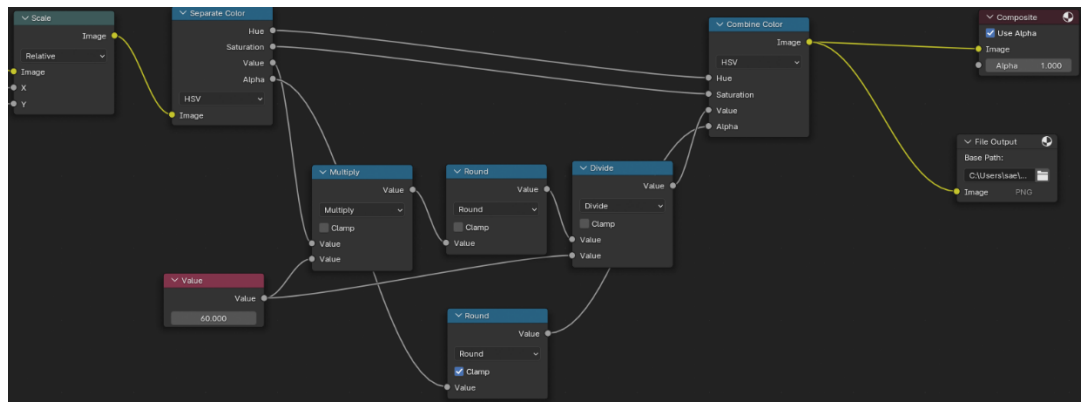
b. Konversi visual melalui *compositing node*

Hasil *render* diproses melalui tiga tahap *node*: (1) *Gaussian blur* untuk menghaluskan tepi objek, (2) Tahap *pixelation*, kombinasi *node Scale-Pixelate*–

Scale, merupakan tahap inti penentu resolusi visual aset, di mana citra hasil blur terlebih dahulu diperkecil (*downscale*) secara relatif melalui *node Scale* dengan faktor skala yang ditentukan oleh satu *node Value* yang diproses melalui *node Divide*, sehingga *node Value* inilah yang menjadi parameter tunggal penentu resolusi pixel art yang dihasilkan, dan (3) *posterize* warna untuk mengurangi gradasi warna, sehingga citra menyerupai gaya *pixel art* konvensional. Hasil akhir disimpan dalam format PNG.



Gambar 2. Compositing node (1)
(Sanusi, 2026)



Gambar 3. Compositing node (2)
(Sanusi, 2026)

c. Penyusunan *sprite sheet* dengan *ImageMagick*

Proses ini memanfaatkan *ImageMagick*, sebuah perangkat lunak berbasis *command-line tool* yang perlu dipasang terlebih dahulu pada sistem sebelum dapat digunakan. Penggabungan dilakukan melalui perintah *magick montage*, yang dijalankan pada *terminal* (Windows PowerShell) yang dibuka langsung dari folder berisi rangkaian citra hasil *render*.

Keberhasilan eksperimen diukur berdasarkan tiga variabel operasional: (1) Efisiensi Waktu, persentase penghematan durasi kerja antara metode otomatis dan manual; (2) Kualitas Visual, kemampuan hasil konversi mempertahankan esensi karakter tanpa cacat visual seperti *blurring* atau *pixel jittering*; (3) *Reusability*, fleksibilitas *pipeline* untuk diterapkan pada karakter 3D lain. Analisis data dilakukan melalui dua tahap: analisis sistem komputasi untuk mengevaluasi ketepatan sinkronisasi matematis formula nilai *Value* di dalam Blender, serta analisis komparatif untuk membandingkan data kuantitatif waktu absolut pada *micro-tasking*.

3. HASIL DAN PEMBAHASAN

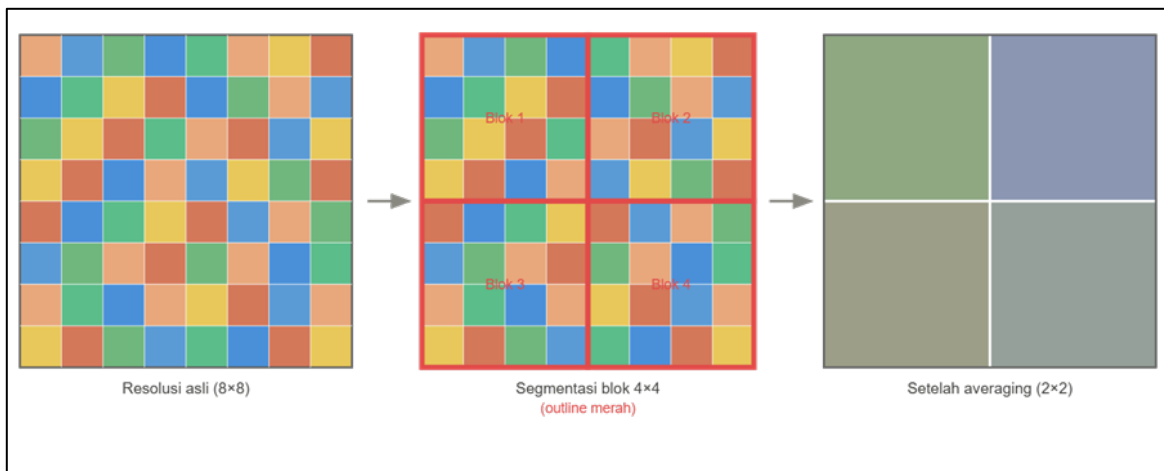
a. Fondasi Matematis Rumus Pikselasi

Dalam komputasi grafis, sebuah gambar direpresentasikan sebagai matriks dua dimensi yang tersusun atas baris dan kolom piksel. Ketika dua parameter kunci ditetapkan, Resolusi Render (resolusi tinggi sebagai masukan awal) dan Resolusi Target (resolusi rendah sebagai keluaran akhir), keduanya mendefinisikan rasio perubahan skala linear yang menjadi landasan proses kompresi visual. Rasio ini dirumuskan sebagai Faktor Skala Penurunan (*Downscaling Factor*), selanjutnya disebut parameter *Value*:

$$Value = \frac{Resolusi\ Render}{Resolusi\ Target}$$

Formula ini valid secara matematis karena menghasilkan bilangan bulat (integer) yang merepresentasikan jumlah piksel asli yang harus dikompresi menjadi satu piksel baru. Apabila hasil pembagian merupakan bilangan bulat, misalnya $256 \div 64 = 4$, partisi matriks visual terbentuk secara sempurna tanpa sisa piksel terpotong (*pixel residue*), sebagaimana diverifikasi melalui uji *Block Size Consistency* (BSC) dan uji reversibilitas (Seo et al., 2026)

Nilai parameter *Value* yang dihasilkan oleh formula tersebut dapat langsung diterapkan secara akurat pada *node Pixelate* dalam Blender. *Node* ini bekerja menggunakan mekanisme kompresi visual berbasis *grid*. Kanvas beresolusi tinggi dibagi menjadi petak-petak berukuran $Value \times Value$ piksel. Selanjutnya, sistem membaca nilai RGB seluruh piksel dalam satu petak, menghitung nilai rata-rata warna statistiknya (Jahnavi et al., 2024), kemudian menyeragamkan seluruh piksel dalam petak tersebut dalam satu warna. Proses ini menciptakan ilusi optik berupa kotak-kotak warna solid yang khas dalam estetika *pixel art* (Binninger & Sorkine-Hornung, 2024; Coutinho & Chaimowicz, 2022). Pendekatan berbasis statistik ini secara efektif mempertahankan kompleksitas visual seperti gradasi pencahayaan dan detail bayangan dari model 3D secara proporsional, sekaligus menjaga ketajaman *jagged edges* sebagai fitur identitas *pixel art*, keunggulan yang tidak dapat dicapai oleh teknik interpolasi kontinu seperti *Bilinear* maupun *Bicubic Filtering* (Jahnavi et al., 2024).



Gambar 4. Mekanisme *Downscaling Factor*
(Sanusi, 2026)

Untuk memvalidasi formula secara empiris, dilakukan pengujian terhadap lima skenario resolusi yang berbeda. Hasil pengujian disajikan pada Tabel 1 berikut.

Skenario	Resolusi Render (R)	Resolusi Target (T)	Perhitungan (R/T)	Nilai Parameter (Value)	Dimensi Blok (V×V)	Status Validitas
1	256 px	64 px	$256 \div 64$	4	4×4 px	Valid (Integer)
2	128 px	64 px	$128 \div 64$	2	2×2 px	Valid (Integer)
3	512 px	64 px	$512 \div 64$	8	8×8 px	Valid (Integer)
4	512 px	128 px	$512 \div 128$	4	4×4 px	Valid (Integer)
5	1024 px	128 px	$1024 \div 128$	8	8×8 px	Valid (Integer)

Tabel 1. Verifikasi Formula pada Lima Skenario Resolusi

Keberhasilan uji lintas skenario, dari parameter terkecil (*Value* = 2) hingga terbesar (*Value* = 8), membuktikan bahwa formula bersifat skalabel pada seluruh ekosistem resolusi target dan memiliki kepastian komputasional yang valid untuk mengeliminasi fluktuasi galat visual (*stray pixels*).

Kasus kritis muncul pada skenario konversi 1080 px → 64 px yang menghasilkan nilai non-integer pada faktor skala linear:

$$K = \frac{1080}{64} = 16,875.$$

Node Pixelate tidak dapat membagi matriks kanvas menggunakan blok berukuran desimal, sistem membulatkan nilai tersebut ke bilangan bulat terdekat, yaitu *Value* = 17. Pergeseran dari nilai teoritis (16,875) ke nilai praktis (17) inilah yang disebut sebagai deviasi spasial (Venkataramanan et al., 2021). Kalkulasi deviasi relatifnya adalah:

$$\text{Deviasi} = \left| \frac{\text{Value Praktis} - \text{Value Teoritis}}{\text{Value Teoritis}} \right| \times 100\%$$

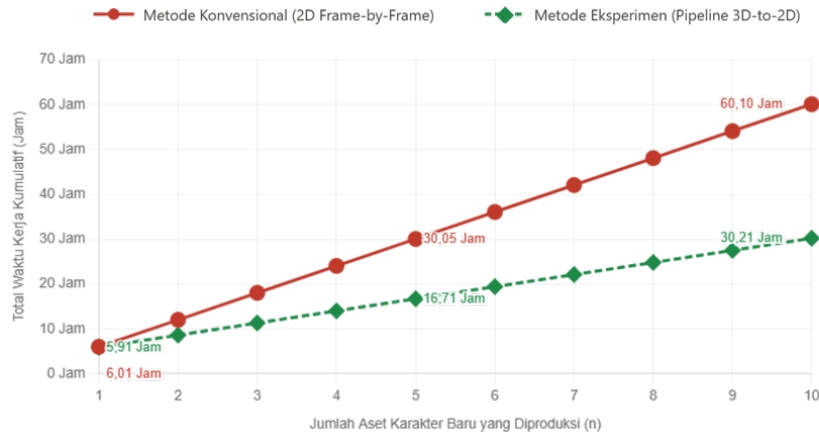
$$\text{Deviasi} = \left| \frac{17 - 16,875}{16,875} \right| \times 100\% = 0,74\%$$

Nilai ini jauh di bawah ambang toleransi industri yang berkisar 5–10% (Venkataramanan et al., 2021), sehingga secara perseptual hampir tidak dapat dideteksi oleh penglihatan manusia, kondisi *Just Noticeable Difference (sub-JND)* (Hu et al., 2023; Jiang et al., 2022; Zhang et al., 2023). Lebih lanjut, algoritma *Grid Snapping* secara otomatis mengunci koordinat pembulatan tersebut ke kisi integer terdekat secara absolut, sehingga tepi aset tetap berada pada tingkat kontras tertinggi dan pola *jagged edges* sebagai fitur estetika utama pixel art tetap terjaga (Coutinho & Chaimowicz, 2022). Selisih 8 px antara dimensi hasil ($17 \times 64 = 1.088$ px) dan resolusi awal (1.080 px) didistribusikan secara merata oleh sistem pada area *padding* tanpa mendistorsi proporsi internal objek 3D yang dikonversi (Jahnavi et al., 2024; Seo et al., 2026).

b. Matriks Pengukuran Efisiensi dan Analisis Break-Even Point

Berdasarkan data empiris dari rekaman waktu yang dilakukan selama penelitian, diperoleh durasi produksi dasar untuk satu karakter utama sebagai berikut: (a) Metode Konvensional (*2D Frame-by-Frame*): 6 jam 00 menit 40 detik, setara 6,01 jam; (b) Metode Eksperimen (*Pipeline 3D-to-2D*): Komponen *Reusable (Static Cost)* 3 jam 12 menit 43 detik (3,21 jam), Komponen *Non-Reusable (Variable Cost)* 2 jam 41 menit 47 detik (2,70 jam), dengan total produksi karakter pertama 5 jam 54 menit 30 detik (5,91 jam).

Apabila produksi hanya melibatkan satu karakter, efisiensi kedua metode tampak serupa dengan selisih tipis sekitar 6 menit. Namun, keunggulan matematis *pipeline 3D-to-2D* terlihat secara signifikan saat memasuki fase produksi massal.



Gambar 5. Proyeksi Kumulatif Waktu Produksi Berdasarkan Jumlah Aset Karakter (Sanusi, 2026)

Untuk menentukan pada volume produksi ke-n berapa investasi waktu pada *pipeline* 3D terbayar lunas, disusun fungsi linear waktu dari kedua metode:

1) Fungsi Waktu Metode Konvensional (T_2D):

$$T_2D(n) = n \cdot t_2D$$

2) Fungsi Waktu Metode Eksperimen (T_3D):

$$T^3D(n) = T_{reuse} + (n \cdot t_{non-reuse})$$

Titik impas (BEP) terjadi pada saat $T_2D(n) = T_3D(n)$, sehingga persamaan kesetimbangan dapat ditulis sebagai:

$$n \cdot t_2D = T_{reuse} + (n \cdot t_{non-reuse})$$

Substitusi nilai-nilai empiris dari rekaman waktu ke dalam persamaan kesetimbangan untuk mencari nilai n:

$$n \cdot 6,01 = 3,21 + (n \cdot 2,70)$$

$$n = \frac{3,21}{3,31} \approx 0,97$$

Berdasarkan hasil derivasi di atas, nilai $n \approx 0,97$ menghasilkan tiga kesimpulan ilmiah yang saling menguatkan:

1) Akselerasi BEP yang ekstrem

Titik impas temporal terjadi sebelum karakter pertama selesai diproduksi secara utuh, yang mengindikasikan efisiensi *pipeline 3D-to-2D* telah aktif sejak unit produksi pertama.

2) Investasi waktu yang langsung terbayar

Berbeda dengan *pipeline* baru pada umumnya yang baru mencapai BEP setelah

memproduksi 3–5 aset, waktu pengerjaan karakter pertama melalui metode 3D (5,91 jam) sudah lebih singkat dibandingkan metode 2D manual (6,01 jam).

3) Efek multiplikatif *reusability*

Nilai $n < 1$ membuktikan bahwa paradigma *reusability* bukan sekadar strategi jangka panjang, melainkan solusi yang langsung memberikan penghematan biaya produksi sejak karakter pertama dihasilkan.

Tingginya tingkat *reusability* pada metode *3D-to-2D* bersumber dari perbedaan fundamental arsitektur manipulasi data visual yang bertumpu pada prinsip *Skeletal Animation*. Dalam grafis komputer 3D, aspek visual (*surface/mesh*) dan aspek penggerak (*structure/rig*) dikelola secara independen melalui tiga komponen hierarkis:

1) Abstraksi Struktur Tulang (*Shared Skeletal Rig*)

Satu struktur rig yang telah dikonfigurasi dengan sistem *Inverse Kinematics* (IK) dapat dipasangkan secara instan ke ratusan model karakter baru yang berbeda, selama proporsi dasar topologinya serupa.

2) Otomatisasi Deformasi *Mesh* (*Skinning dan Weight Painting*)

Melalui data numerik *Weight Painting*, setiap vertex pada model 3D baru secara otomatis mengikuti pergerakan *bone* yang sesuai tanpa intervensi visual manual.

3) Pemisahan Data Animasi melalui *Non-Linear Animation* (NLA)

Data gerakan disimpan sebagai *Action Clips* yang terisolasi dari bentuk model 3D, sehingga satu *Action Clip* dapat diterapkan ke karakter baru secara instan tanpa merekam ulang setiap frame.

Aspek Teknis	Metode Konvensional 2D Frame-by-Frame	Metode Eksperimen 3D-to-2D (Skeletal)
Arsitektur Data	Raster Terikat: Informasi bentuk dan gerakan menyatu di dalam satu piksel gambar per frame.	Modular Terpisah: Informasi bentuk (<i>mesh</i>) dan gerakan (<i>rig/animation data</i>) dikelola secara terpisah.
Shared Structure	Tidak ada: Setiap frame adalah entitas baru yang berdiri sendiri. Tidak terdapat struktur internal yang dapat dibagikan ke aset lain.	Ada: Satu struktur <i>rig</i> dan data <i>keyframe</i> dapat didistribusikan secara universal ke karakter lain.
Iterasi Karakter Baru	Redrawing Total: Pembuatan kostum atau rambut baru mengharuskan kreator menggambar ulang seluruh sekuens gerakan dari awal.	Otomatisasi Proyeksi: Kreator hanya memodifikasi <i>mesh</i> 3D statis; gerakan dieksekusi secara otomatis oleh <i>rig</i> .
Ketergantungan Keahlian	Sangat Tinggi: Bergantung sepenuhnya pada konsistensi motorik manual, intuisi anatomi, dan <i>hard-skill</i> menggambar kreator.	Relatif Rendah: Bergantung pada akurasi input data numerik, manipulasi geometri dasar, dan algoritma render.

Tabel 2. Komparasi Teknis Metode Konvensional 2D Frame-by-Frame vs. Metode Eksperimen 3D-to-2D

c. Kalkulasi Persentase Peningkatan Efisiensi

Persentase peningkatan efisiensi (*efficiency gain*) dihitung menggunakan rumus:

$$\text{Efisiensi} = \left[\frac{T_{\text{konvensional}} - T_{\text{eksperimen}}}{T_{\text{konvensional}}} \right] \times 100\%$$

Dengan data operasional: waktu produksi metode 2D konvensional per karakter (t_{2D}) = 6,01 jam; investasi awal komponen *Reusable* (T_{reuse}) = 3,21 jam; waktu 3D per karakter, komponen *Non-Reusable* ($t_{non-reuse}$) = 2,70 jam. Hasil kalkulasi pada berbagai volume produksi disajikan pada Tabel 3 berikut.

Jumlah Aset (n)	Total Waktu 2D Konvensional (Jam)	Total Waktu 3D Eksperimen (Jam)	Selisih Waktu Absolut (Jam)	Keunggulan Efisiensi (%)
1 Aset	6,01	5,91	0,10	1,66%
2 Aset	12,02	8,61	3,41	28,37%
5 Aset	30,05	16,71	13,34	44,39%
10 Aset	60,10	30,21	29,89	49,73%

Tabel 3. Ringkasan Perbandingan Waktu Produksi dan Persentase Efisiensi

Data pada Tabel 3 menunjukkan tren akselerasi efisiensi yang konsisten seiring bertambahnya volume produksi. Pada $n = 10$ aset, metode *3D-to-2D* mampu memangkas hampir 50% total waktu kerja dibandingkan metode konvensional, membuktikan efektivitas paradigma *reusability* dalam skala produksi industri.

d. Evaluasi Kualitas Visual

Pengujian *pipeline* konversi *3D-to-2D* mengidentifikasi deviasi spasial sebesar 1 hingga 2 piksel pada hasil rendering isometrik. Temuan ini tidak dieliminasi maupun diminimalkan, melainkan dikaji secara ilmiah sebagai fenomena yang valid dan inheren dalam sistem pemrosesan citra digital.

Secara mekanis, deviasi tersebut berakar pada Teori Pemrosesan Citra, khususnya konsep *Sub-Pixel Rendering* dan *Aliasing* (Luo & Zhang, 2020). Objek dalam perangkat lunak 3D direpresentasikan dalam ruang koordinat kontinu berbasis bilangan desimal (*floating-point*). Ketika kamera *orthographic* memproyeksikan geometri kontinu tersebut ke kanvas target beresolusi rendah, seperti 64×64 piksel, sistem komputasi diharuskan menerjemahkan koordinat halus tersebut ke dalam kisi piksel diskrit yang bersifat kaku (bilangan bulat) (Luo & Zhang, 2020; Seo et al., 2026). Proses diskretisasi ini secara inheren memicu *rasterization artifacts* yang apabila tepi geometri berada pada koordinat desimal seperti 16,875, sistem membulatkan nilai tersebut ke piksel utuh terdekat. Dengan demikian, munculnya selisih 1–2 piksel pada resolusi rendah bukan merupakan kegagalan sistem maupun kecacatan *pipeline*, melainkan

konsekuensi matematis yang tak terhindarkan dari prinsip proyeksi citra digital (Luo & Zhang, 2020).

Untuk membuktikan bahwa deviasi tersebut tidak berdampak pada kualitas estetika, penelitian ini menggunakan pendekatan kuantitatif berbasis Teori Persepsi Visual: *Just Noticeable Difference (JND)* (Hu et al., 2023; Jiang et al., 2022; Zhang et al., 2023) sebagai pengganti penilaian subjektif semata. Pada resolusi target 64×64 piksel:

$$\text{Galat Minimum} = \frac{1}{64} \times 100\% = 1,56\%$$

$$\text{Galat Maksimum} = \frac{2}{64} \times 100\% = 3,13\%$$

Berdasarkan hukum Weber-Fechner dalam kerangka JND, sistem visual manusia baru mampu mendeteksi perubahan dimensi linear pada suatu stimulus apabila perubahan tersebut mencapai ambang batas 5%–10% dari total dimensi objek yang diamati (Jiang et al., 2022; Liu et al., 2020). Oleh karena nilai deviasi riil dalam penelitian ini (1,56%–3,13%) berada jauh di bawah ambang batas JND, secara perseptual deviasi 1–2 piksel tersebut tidak dapat dideteksi oleh mata manusia. Akibatnya, karakteristik visual *pixel art*, meliputi ketajaman garis, konsistensi outline, serta keterbacaan (*readability*) siluet karakter tetap terjaga sepenuhnya di seluruh sekuens animasi yang dihasilkan.

4. KESIMPULAN

Penelitian ini berhasil merancang dan memvalidasi *deterministic hybrid pipeline* konversi *3D-to-2D* untuk produksi aset karakter pixel art menggunakan Blender. Tiga kesimpulan utama dapat ditarik berdasarkan data empiris yang diperoleh.

Pertama, formula konversi resolusi piksel $\text{Value} = \text{Resolusi Render} \div \text{Resolusi Target}$ terbukti valid secara matematis dan skalabel di seluruh ekosistem resolusi yang diuji. Pada lima skenario resolusi integer, formula menghasilkan partisi matriks yang sempurna tanpa *pixel residue*, sementara pada kasus kritis ($1080 \text{ px} \rightarrow 64 \text{ px}$), deviasi spasial hanya sebesar 0,74%, jauh di bawah ambang toleransi industri 5–10%. Validasi ini membuktikan bahwa *pipeline* yang dikembangkan memiliki kepastian komputasional yang absolut dan bukan merupakan hasil kebetulan empiris.

Kedua, *pipeline* 3D-to-2D mencapai Break-Even Point (BEP) pada nilai $n \approx 0,97$, yang berarti investasi waktu komponen reusable (3,21 jam) telah terbayar sebelum karakter pertama selesai diproduksi. Keunggulan efisiensi tumbuh secara konsisten seiring volume produksi, dari 1,66% pada aset pertama hingga 49,73% pada aset kesepuluh. Temuan ini membuktikan bahwa paradigma *reusability* berbasis *Skeletal Animation* secara fundamental mengungguli metode konvensional *2D frame-by-frame* dalam konteks produksi massal.

Ketiga, deviasi visual sebesar 1–2 piksel yang ditemukan pada hasil rendering isometrik merupakan konsekuensi matematis yang tak terhindarkan dari proses diskretisasi koordinat kontinu ke kisi piksel diskrit. Namun, deviasi riil sebesar 1,56%–3,13% berada di bawah ambang batas JND sebesar 5–10%, sehingga secara

perseptual tidak dapat dideteksi oleh sistem visual manusia. Kualitas estetika *pixel art*, meliputi ketajaman *outline*, konsistensi proporsi antar *frame*, serta keterbacaan siluet karakter, tetap terjaga sepenuhnya di seluruh sekuens animasi yang dihasilkan.

SARAN

Berdasarkan temuan dan keterbatasan penelitian ini, beberapa rekomendasi diajukan untuk pengembangan lebih lanjut. Pertama, penelitian ini hanya menguji *pipeline* pada satu model karakter *humanoid* dengan proporsi tubuh standar; penelitian selanjutnya perlu menguji *pipeline* pada karakter dengan topologi non-*humanoid* atau proporsi anatomis yang ekstrem guna menguji generalisasi sistem *weight painting* otomatis. Kedua, validasi kualitas visual pada penelitian ini masih mengandalkan pendekatan JND secara teoretis; riset berikutnya disarankan mengintegrasikan metrik objektif berbasis *image processing* seperti SSIM (*Structural Similarity Index Measure*) dan PSNR (*Peak Signal-to-Noise Ratio*) untuk menghasilkan evaluasi yang lebih terukur. Ketiga, penelitian ini membuktikan keunggulan *pipeline deterministik non-AI*, namun studi komparatif kuantitatif terhadap pendekatan generatif berbasis *deep learning* belum dilakukan; perbandingan langsung antara kedua paradigma tersebut akan memberikan peta posisi yang lebih komprehensif bagi pengembang game dalam menentukan pilihan metodologi produksi aset.

DAFTAR PUSTAKA

- Binniger, A., & Sorkine-Hornung, O. (2024, December 3). SD- π XL: Generating Low-Resolution Quantized Imagery via Score Distillation. *Proceedings - SIGGRAPH Asia 2024 Conference Papers, SA 2024*. <https://doi.org/10.1145/3680528.3687570>
- Coutinho, F., & Chaimowicz, L. (2022). *On the Challenges of Generating Pixel Art Character Sprites Using GANs*. <https://doi.org/10.1609/aiide.v18i1.21951>
- Fukaya, K., Daylamani-Zad, D., & Agius, H. (2023). *Intelligent Generation of Graphical Game Assets: A Conceptual Framework and Systematic Review of the State of the Art*. <https://doi.org/10.1145/3708499>
- Hu, T., Yin, H., Wang, H., Sheng, N., & Xing, Y. (2023). Pixel-Domain Just Noticeable Difference Modeling with Heterogeneous Color Features. *Sensors*, 23(4). <https://doi.org/10.3390/s23041788>
- Jahnavi, M., Rajeswara Rao, D., & Sujatha, A. (2024). A Comparative Study Of Super-Resolution Interpolation Techniques: Insights For Selecting The Most Appropriate Method. *Procedia Computer Science*, 233, 504–517. <https://doi.org/10.1016/j.procs.2024.03.240>
- Jiang, Q., Liu, Z., Wang, S., Shao, F., & Lin, W. (2022). *Towards Top-Down Just Noticeable Difference Estimation of Natural Images*. <https://doi.org/10.1109/TIP.2022.3174398>
- Lee, D.-M. (2025). Blender 3D as a Catalyst for Indie Game Development. *International JOURNAL OF CONTENTS*, 21(2), 67–74. <https://doi.org/10.5392/ijoc.2025.21.2.067>

- Liu, H., Zhang, Y., Zhang, H., Fan, C., Kwong, S., Kuo, C. C. J., & Fan, X. (2020). Deep Learning-Based Picture-Wise Just Noticeable Distortion Prediction Model for Image Compression. *IEEE Transactions on Image Processing*, 29, 641–656. <https://doi.org/10.1109/TIP.2019.2933743>
- Luo, D., & Zhang, J. (2020). The #-filter anti-aliasing based on sub-pixel continuous edges. *Mathematics*, 8(10). <https://doi.org/10.3390/MATH8101655>
- Ribeiro, R., de Carvalho, A. V., & Rodrigues, N. B. (2025). Image-Based Video Game Asset Generation and Evaluation Using Deep Learning: A Systematic Review of Methods and Applications. *IEEE Transactions on Games*, 17(3), 622–630. <https://doi.org/10.1109/TG.2024.3487054>
- Seo, J. W., Lee, J. W., Lee, J. H., Kim, J. B., & Jung, J. W. (2026). Structure-Aware Pixel Art Scaling via Block Size Detection. *Applied Sciences (Switzerland)*, 16(5). <https://doi.org/10.3390/app16052314>
- Toftedahl, M., & Engström, H. (2019). *A Taxonomy of Game Engines and the Tools that Drive the Industry*. <https://doi.org/10.26503/dl.v2019i1.1077>
- Venkataramanan, A. K., Wu, C., Bovik, A. C., Katsavounidis, I., & Shahid, Z. (2021). *A Hitchhiker's Guide to Structural Similarity*. <https://doi.org/10.1109/ACCESS.2021.3056504>
- Wang, K. (2025). Leveraging Unity for 2D Pixel Game Development: Techniques and Best Practices. *ITM Web of Conferences*, 70, 03002. <https://doi.org/10.1051/itmconf/20257003002>
- Wang, M. (2024). Integrating 2D and 3D in Game Development: Design and Technology Collaboration. *Applied and Computational Engineering*, 110(1), 170–174. <https://doi.org/10.54254/2755-2721/110/2024melb0116>
- Zhang, Z., Shang, X., Li, G., & Wang, G. (2023). Just Noticeable Difference Model for Images with Color Sensitivity. *Sensors*, 23(5). <https://doi.org/10.3390/s23052634>