Jurnal Komputer, Informasi dan Teknologi Vol: 5, No 1, 2025, Page: 1-16

Implementasi *Drop Rate Item* menggunakan *Algoritma Weighted Random Sampling* dalam *Game Action Platformer*

Zada Alfath Maulana*, Mutaqin Akbar

Program Studi Informatika, Fakultas Teknologi Informasi, Universitas Mercu Buana Yogyakarta

Abstrak: Penelitian ini bertujuan untuk mengimplementasikan algoritma Weighted Random Sampling dalam sistem drop rate item pada game bergenre action platformer berbasis Godot Engine. Tujuan utamanya adalah menciptakan distribusi hadiah yang lebih adil dan dapat dikendalikan berdasarkan tingkat kelangkaan item seperti common, rare, dan epic. Sistem ini dirancang agar peluang kemunculan item sesuai dengan bobot yang telah ditentukan, sehingga dapat mengatasi ketidakseimbangan yang sering terjadi pada metode acak seragam. Metode penelitian meliputi studi literatur, perancangan sistem drop item, implementasi kode menggunakan bahasa GDScript, serta pengujian sistem secara visual dan fungsional. Implementasi dilakukan dalam Godot Engine versi 4.4.1 dengan menggunakan struktur data berbasis dictionary untuk mengelola bobot item. Proses pengujian dilakukan melalui sepuluh simulasi drop item, masing-masing menghasilkan antara 15 hingga 20 item yang dikategorikan berdasarkan rarity dan dikonversikan menjadi poin untuk peningkatan status karakter. Hasil penelitian menunjukkan bahwa algoritma Weighted Random Sampling mampu menghasilkan distribusi item yang mendekati rasio teoritis, dengan proporsi kemunculan item common sebesar 58,8%, rare 29,3%, dan epic 11,9%. Sistem juga mampu menampilkan item secara visual dengan layout yang responsif, border warna sesuai rarity, serta animasi yang meningkatkan persepsi keadilan. Dengan demikian, sistem ini efektif dalam memberikan pengalaman bermain yang seimbang dan sesuai prinsip desain game modern.

Kata kunci: *Weighted Random Sampling, Drop Rate, Game Platformer, Godot Engine, Distribusi Item*

DOI:
https://doi.org/10.53697/jkomitek.v5i1.2712
*Correspondence: Zada Alfath Maulana
Email:
211110019@student.mercubuana-yogya.ac.id

Received: 25-04-2025
Accepted: 25-05-2025
Published: 25-06-2025

Abstract: *This research aims to implement the Weighted Random Sampling algorithm in the item drop rate system in the Godot Engine-based action platformer game. The main goal is to create a fairer and more controllable reward distribution based on the rarity level of items such as common, rare, and epic. This system is designed so that the chance of an item appearing is in accordance with the weight that has been determined, so that it can overcome the imbalance that often occurs in uniform random methods. Research methods include literature study, designing item drop system, code implementation using GDScript language, and visual and functional testing of the system. Implementation is done in Godot Engine version 4.4.1 by using dictionary-based data structure to manage item weights. The testing process was conducted through ten simulated item drops, each resulting in between 15 to 20 items that were categorized based on rarity and converted into points for character status upgrades. The results show that the Weighted Random Sampling algorithm is able to produce an item distribution that is close to the theoretical ratio, with the proportion of common items occurring at 58.8%, rare 29.3%, and epic 11.9%. The system is also able to visually display items with a responsive layout, color borders according to rarity, and animations that increase the perception of fairness. Thus, the system is effective in providing a sei playing experience. As such, the system is effective in providing a balanced gaming experience that conforms to modern game design principles..*

Keywords: *Weighted Random Sampling, Drop Rate, Platformer Game, Godot Engine, Item Distribution*

Pendahuluan

Game adalah suatu bentuk hiburan interaktif berbasis perangkat digital yang dirancang untuk memberikan pengalaman menyenangkan kepada pemain melalui serangkaian tantangan, aturan, dan tujuan tertentu. Sejak kemunculan game pertama seperti Pong dan Tetris, industri game telah berkembang pesat baik dari segi teknologi, kompleksitas mekanik, hingga bentuk penyajian visual dan naratif. Game kini tidak hanya sekadar hiburan, tetapi juga menjadi media edukasi, simulasi, hingga ladang industri ekonomi kreatif global yang bernilai miliaran dolar.

Secara umum, genre game diklasifikasikan menjadi beberapa kategori seperti action, adventure, role-playing game (RPG), simulation, strategy, sports, hingga platformer. Masing-masing genre memiliki ciri khas dalam gameplay dan struktur interaksinya. Salah satu subgenre populer adalah action platformer, yang menggabungkan elemen aksi seperti pertarungan dengan tantangan platforming berupa loncatan, navigasi rintangan, dan timing yang presisi. Game jenis ini menuntut ketangkasan serta refleks pemain yang cepat, sambil tetap memberikan ruang eksplorasi dan peningkatan kemampuan karakter. Komponen utama dalam game action platformer umumnya terdiri dari: Karakter utama yang dapat bergerak, melompat, dan menyerang, Musuh (enemy units) dengan berbagai perilaku dan pola serangan, Lingkungan (environment) seperti platform, jebakan, dan objek interaktif, Sistem item dan power-up, yang menjadi hadiah atau dukungan kemampuan tambahan, UI/UX game seperti health bar, score counter, dan menu interaksi(Arrazzaq, Sasmito, & Zahro' 2023) (Octodinata, Pragantha, & Haris 2023).

Dalam proses desain game, salah satu mekanisme penting adalah sistem drop item, yaitu sistem yang mengatur bagaimana dan kapan pemain menerima hadiah dari musuh atau peti harta. Mekanisme ini kerap menggunakan algoritma acak (random) agar hasil yang diperoleh pemain terasa tidak monoton dan menambah replayability. Metode acak sederhana seperti uniform random digunakan untuk memilih item dari kumpulan item secara merata, tanpa memperhatikan nilai atau kelangkaan item tersebut. Contoh game yang menggunakan pendekatan sederhana ini adalah Super Mario Bros atau Castlevania awal, di mana musuh menjatuhkan item seperti jamur atau senjata tambahan secara acak namun seragam.

Namun demikian, metode acak seragam memiliki kelemahan seperti hasil distribusi yang tidak konsisten—item langka bisa saja muncul terlalu sering, atau sebaliknya, tidak muncul sama sekali. Hal ini menimbulkan persepsi ketidakadilan dalam permainan, yang dalam beberapa kasus telah disamakan dengan praktik perjudian digital melalui mekanik loot box(Nielsen & Grabarczyk, 2019) (Spicer & Close, 2022). Oleh karena itu, dalam pengembangan game modern digunakan metode Weighted Random Sampling, yakni algoritma acak berbobot yang memberikan peluang lebih besar bagi item tertentu untuk muncul, sesuai bobot yang ditentukan oleh desainer game. Penggunaan algoritma ini dianggap sebagai solusi efektif untuk mengatasi bias distribusi acak dan telah banyak digunakan pada sistem loot berbasis backend modern seperti LootLocker dan Unity (Cox, 2024) (Stokholm, 2019). Dengan sistem ini, desainer bisa menetapkan bahwa item "Common" memiliki bobot lebih besar dibandingkan item "Rare" atau "Epic", sehingga peluang kemunculannya lebih tinggi dan distribusi lebih terkendali. Penggunaan algoritma

berbobot untuk loot drop seperti ini sejalan dengan prinsip desain game yang memprioritaskan keadilan dan kepuasan pemain (Bateman et al, 2010) (Hunckie, LeBlance, & Zubek, 2004), di mana pemain lebih termotivasi saat probabilitas hadiah sesuai ekspektasi dan effort mereka (Hamari and Lehdonvirta 2010). Beberapa game modern seperti Genshin Impact, Honkai Star Rail, atau Borderlands telah menerapkan sistem loot berbobot ini untuk menciptakan pengalaman yang lebih adil, terstruktur, dan tetap memuaskan pemain dalam jangka panjang. Dengan landasan teoritis yang kuat (Efraimidis, 2015) dan perhatian terhadap etika distribusi loot (Nielsen & Grabarczyk, 2019), sistem ini menjadi alternatif yang layak untuk menciptakan pengalaman bermain yang seimbang dan bertanggung jawab. Hal ini juga sesuai dengan praktik terbaik dalam industri game, yang menyarankan distribusi loot berbasis bobot untuk menjaga keterlibatan dan kepuasan pemain("Loot Drop Best Practices" 2023). Dalam konteks skripsi ini, saya menerapkan algoritma Weighted Random Sampling dalam sistem drop item pada game action platformer berbasis Godot Engine, dengan tujuan menghasilkan sistem distribusi hadiah yang seimbang dan dapat dikendalikan secara matematis. Penelitian oleh Khandelwal dan Gupta (Khandelwal & Gupta, 2023) juga menekankan pentingnya penggunaan model probabilistik untuk merancang sistem reward yang adil, guna meningkatkan motivasi dan retensi pemain dalam jangka panjang.

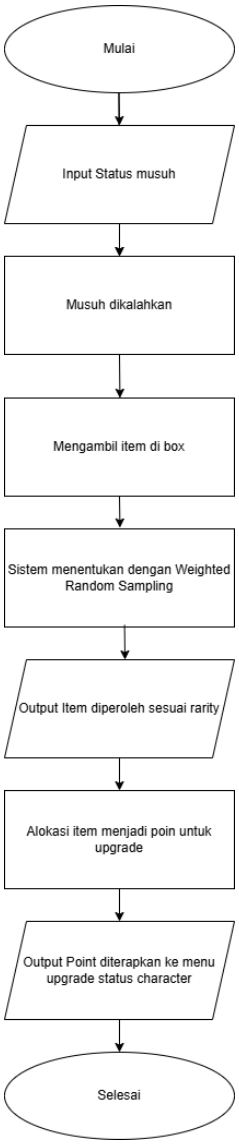
Dalam pengembangan game, terdapat berbagai platform populer seperti Unity, Unreal Engine, GameMaker Studio, dan Godot Engine. Masing-masing memiliki keunggulan tersendiri—Unity dan Unreal unggul untuk game 3D skala besar, sedangkan GameMaker dan Godot lebih banyak digunakan untuk game 2D karena ringan dan mudah dipelajari. Godot Engine adalah platform open-source yang berkembang pesat sejak dirilis pada 2014. Engine ini mendukung pengembangan game 2D dan 3D, menggunakan bahasa scripting GDScript yang mirip Python, serta memiliki sistem node dan scene yang fleksibel. Kelebihan Godot meliputi: gratis tanpa royalti, ringan, mendukung multi-platform, dan komunitas yang aktif (Bradfield, 2018). Namun, kekurangannya antara lain kemampuan 3D yang masih terbatas dan ekosistem plugin yang belum sebesar engine lain. Godot dipilih dalam penelitian ini karena cocok untuk proyek 2D dan mudah diintegrasikan dengan algoritma seperti Weighted Random Sampling.

Berdasarkan latar belakang tersebut, penelitian ini difokuskan pada penerapan algoritma Weighted Random Sampling dalam sistem drop item untuk game action platformer. Tujuannya adalah untuk menciptakan mekanisme pemilihan item yang lebih adil, di mana peluang munculnya item langka dapat diatur secara proporsional, tanpa membuat pemain harus mengulang permainan berkali-kali hanya untuk mendapatkan item tertentu—seperti yang kerap terjadi pada sistem acak di game modern seperti Genshin Impact atau Borderlands. Selain memberikan keadilan distribusi, algoritma ini juga dirancang agar efisien secara komputasi dan dapat diintegrasikan dengan mudah ke dalam sistem game berbasis Godot Engine.

Metodologi

Penelitian ini dilakukan melalui serangkaian tahapan yang mencakup studi literatur, perancangan sistem, implementasi kode program, dan pengujian sistem. Semua proses dirancang secara sistematis agar hasil penelitian dapat direplikasi oleh peneliti lain. Seluruh kode sumber, struktur data, dan algoritma yang digunakan dalam pengembangan sistem ini tersedia dan dapat dibuka aksesnya untuk keperluan verifikasi atau pengembangan lanjutan. yang mencakup studi literatur, perancangan sistem, implementasi kode program, dan pengujian sistem. Setiap tahap dirancang untuk menghasilkan sistem distribusi drop item yang adil, dapat dikendalikan berdasarkan bobot, serta memberikan pengalaman pengguna yang menarik dan fungsional.

Tahapan awal dilakukan dengan mengumpulkan berbagai referensi yang relevan dengan topik penelitian. Studi literatur ini mencakup kajian terhadap teori algoritma Weighted Random Sampling dan penerapannya dalam berbagai konteks, termasuk data stream dan sistem rekomendasi, serta bagaimana algoritma ini mulai digunakan dalam sistem loot atau drop item dalam game modern. Selain itu, dilakukan pula pendalaman terhadap dokumentasi resmi Godot Engine, buku-buku pengembangan game 2D, serta jurnal terkait pengaruh sistem reward terhadap motivasi pemain. Studi literatur ini memberikan fondasi teoretis dan teknis dalam menyusun metode pemilihan item secara berbobot yang diterapkan dalam game action platformer. Dalam perancangan sistem drop item, konsep player engagement dan retention berperan penting (Yee, 2006), sehingga implementasi Weighted Random Sampling bisa membantu menciptakan pengalaman bermain lebih imersif dan meminimalisasi kejenuhan pemain (Schell, 2023) (Fullerton & Zimmerman, 2019). Selain aspek teknis, evaluasi kreativitas dalam desain reward juga perlu diperhatikan agar pemain tidak cepat bosan (Cook & Colton, 2012), dan implementasi harus memperhatikan prinsip procedural generation agar tetap natural dan organik (Shiffman, 2012). Untuk referensi penggunaan Godot Engine secara praktis dan penerapan GDScript dalam proyek nyata merujuk pada (Lin & Lin 2021).



Gambar 1. Diagram alur Sistem Drop Item

Setelah memahami teori yang relevan, tahap desain dilakukan untuk menentukan struktur dan alur sistem drop item. Desain sistem mencakup pembuatan struktur data dalam bentuk dictionary, yang berisi item-item seperti *"werewolf feet"*, *"red claw"*, dan *"broken scythe"* dengan masing-masing nilai bobot dan rarity. Selanjutnya dirancang alur logika untuk memilih item berdasarkan bobot menggunakan fungsi probabilistik. Desain visual juga menjadi fokus penting, termasuk penempatan grid item di layar, pemberian border berdasarkan tingkat kelangkaan, serta penambahan label informasi seperti "You've Obtained" dan instruksi klaim. Posisi item ditentukan secara dinamis agar selalu muncul di tengah layar dan tetap responsif pada berbagai resolusi.

Untuk menggambarkan alur sistem secara umum, Gambar 1 menampilkan diagram alur proses distribusi item. Diagram ini menjelaskan urutan kejadian mulai dari melihat status musuh *(alive or dead)* musuh dikalahkan, pemain membuka box item, sistem menjatuhkan item secara acak menggunakan algoritma berbobot, dan akhirnya pemain menerima poin dari item tersebut.

Pada pengembangan game modern, sistem drop item menjadi salah satu komponen kunci untuk meningkatkan kepuasan dan keterlibatan pemain. Jika distribusi drop sepenuhnya acak dan seragam (uniform random), hasilnya sering kali tidak sesuai harapan — item langka bisa terlalu sering muncul, atau justru nyaris tidak pernah didapatkan. Oleh sebab itu, digunakanlah algoritma Weighted Random Sampling, yakni algoritma acak berbobot yang memberikan peluang lebih besar kepada item tertentu agar proporsional terhadap bobot kelangkaannya (Cohen and Kaplan 2008; Li 2018). Pendekatan ini banyak diadopsi dalam desain game modern, seperti pada Genshin Impact dan Borderlands, agar distribusi hadiah lebih adil dan memotivasi pemain untuk terus bermain (Hamari and Lehdonvirta 2010).Pemilihan item menggunakan formula sebagai berikut:

$$pick = rand(0, \sum_{i=1}^{n} w_1 - 1)$$

Item yang terpilih akan ditentukan oleh:

$$selected\ item = \begin{cases} x_1, jika\ pick < w_1 \\ x_2, jika\ pick < w_1 + w_2 \\ x_n, jika\ pick < \sum_{i=1}^{n} w_i \end{cases}$$

Formula ini memungkinkan item dengan bobot lebih tinggi memiliki peluang lebih besar untuk terpilih dibandingkan item berbobot rendah. Langkah-langkah perhitungan drop item pertama adalah menjumlahkan seluruh bobot item yang pada penelitian ini terdapat 3 item yang digunakan dalam sistem ini masing-masing memiliki bobot sebagai berikut: Werewolf feet (Common): 50, Red claw (Rare): 25, Broken scythe (Epic): 10.

$$total\ weight = 50 + 25 + 10 = 85$$

Ambil angka acak dalam rentang 0 hingga total bobot dikurangi satu.

$$pick = rand(0{,}84)$$

Selanjutnya adalah menentukan item berdasarkan rentang kumulatif. Jika pick < 50, maka item = “Werewolf Feet”, Jika pick 50 ≤ pick < 75, maka item = “Red Claw”, Jika pick ≤ 75 pick <84, maka item = “Broken Scythe”. Dengan demikian, probabilitas masing-masing item dapat dihitung sebagai:

$$P(werewolf\ feet): \frac{50}{85} \approx 58.82\%$$

$$P(red\ claw): \frac{25}{85} \approx 29.41\%$$

$$P(broken\ scythe): \frac{10}{85} \approx 11.76\%$$

Perhitungan ini diimplementasikan dalam fungsi get_rarity() yang akan secara otomatis menentukan item berdasarkan nilai pick yang dihasilkan dari *RandomNumberGenerator.* Setelah pemain mendapatkan sejumlah item dari proses drop, setiap item memiliki nilai tertentu yang kemudian akan dikonversikan menjadi poin. Poin-poin ini selanjutnya digunakan untuk melakukan upgrade status karakter pemain dalam game, seperti peningkatan kekuatan, kecepatan, atau atribut lainnya yang relevan dengan mekanisme permainan. Hal ini bertujuan untuk memberikan feedback yang berarti terhadap hasil drop dan menjaga keberlanjutan progres pemain. yang akan secara otomatis menentukan item berdasarkan nilai *pick* yang dihasilkan dari *RandomNumberGenerator*. Pengambilan nilai acak dilakukan menggunakan *RandomNumberGenerator*, modul internal

dari Godot Engine yang memberikan fleksibilitas dalam pengaturan seed dan distribusi acak (Dhule, 2022).

Proses implementasi dilakukan di Godot Engine versi 4.4.1 menggunakan bahasa GDScript. Godot cocok untuk pemula maupun pengembang menengah karena kesederhanaan struktur node dan scripting berbasis Python-like (Calin, 2020). Struktur proyek dibangun dalam satu scene utama bernama DropRateItem, yang terdiri dari node-node seperti CanvasLayer, GridContainer, dan ItemIcon. Pada tahap ini, peneliti menuliskan fungsi get_rarity() untuk memilih item berdasarkan bobot yang dihitung dari total keseluruhan nilai rarity, kemudian memilih secara acak dengan pengurangan kumulatif terhadap bilangan acak tersebut. Fungsi _ready() mengatur proses randomisasi, menentukan jumlah item yang akan di-drop (antara 15 hingga 20), serta memanggil scene item untuk ditampilkan dalam layout grid dengan animasi transisi menggunakan tween. Border dan ikon diatur berdasarkan rarity item yang di-drop. Seluruh elemen visual seperti label dan posisi item ditentukan secara relatif terhadap ukuran layar agar tampak rapi dan terpusat. Fungsi _input() digunakan untuk mendeteksi klik pengguna yang akan men-trigger proses claim_rewards(), yaitu perhitungan total status poin yang dikumpulkan dari semua item yang diperoleh. Hasil distribusi item yang mendekati rasio teoritis menunjukkan bahwa algoritma Weighted Random Sampling berfungsi optimal dan sesuai prinsip sampling berbobot (Cohen & Kaplan 2008) (Li, 2018). Selain itu, visualisasi feedback berbasis rarity, seperti border warna dan animasi, meningkatkan kepuasan dan persepsi fairness terhadap hasil drop (Lee, 2013).

Pengujian dilakukan untuk memastikan bahwa sistem drop item yang dikembangkan dengan algoritma Weighted Random Sampling mampu menghasilkan distribusi item yang seimbang sesuai dengan tingkat kelangkaan yang telah ditentukan. Tujuan dari pengujian ini adalah menciptakan pengalaman bermain yang adil dan tidak membosankan, dengan memastikan bahwa proses peningkatan status karakter berjalan secara proporsional terhadap usaha pemain. Dengan pendekatan ini, pemain tidak akan mengalami progres yang terlalu cepat hingga mengurangi tantangan, atau terlalu lambat hingga menyebabkan frustrasi. Prinsip ini sejalan dengan teori motivasi dalam desain game, di mana sistem hadiah harus memberikan umpan balik yang setimpal dan menjaga keseimbangan antara tantangan dan pencapaian pemain (Hamari & Lehdonvirta, 2010) (Hunckie, LeBlance & Zubek, 2004).

Selain itu, pendekatan berbobot dalam distribusi drop juga mendukung konsep player engagement dan retention, yaitu menjaga keterlibatan pemain dengan memastikan bahwa hasil yang diperoleh terasa adil dan layak (Schell 2023; Hamari and Lehdonvirta 2010). Pengujian ini juga bertujuan memastikan bahwa implementasi algoritma pada Godot Engine berjalan sesuai ekspektasi secara fungsional dan visual, serta mendukung pengembangan sistem upgrade karakter yang terukur dan tidak eksploitatif terhadap mekanik progresi (Fullerton & Zimmerman 2019) (Cook & Colton, 2012).

Hasil dan Pembahasan

Pada pengujian yang dilakukan terdapat 3 item dengan karakteristik berdasarkan rarity nya dan pada setiap item memiliki valuenya tersendiri, untuk membedakan item berdasarkan rarity dan value pada setiap item, seperti dapat dilihat pada Tabel 1.

Tabel 1. Tabel icon dan karakteristik

Nama Item	Icon	Karakteristik	Rarity	Value
Werewolf feet			Common	5
Red claw			Rare	10
Broken scythe			Epic	25

Untuk item pertama yaitu werewolf feet dengan rarity common memiliki karakteristik memiliki border berwarna hijau untuk menandakan rarity nya dan memiliki value sebanyak 5 untuk dikonversikan kedalam poin di dalam game. Item selanjutnya adalah red claw dengan rarity rare memiliki karakteristik memiliki border berwarna biru untuk menandakan rarity nya dan memiliki value 10 untuk dikonversikan menjadi poin di dalam game. Dan terakhir broken scythe dengan rarity epic memiliki karakteristik memiliki border berwarna ungu untuk menandakan rarity nya dan memiliki value 25 untuk dikonversasikan menjadi poin di dalam game.

Item-item tersebut bisa didapatkan setelah menyelesaikan stage dan dapat diambil dengan cara interaksi dengan chest yang ada setelah stage itu selesai. Dan item-item ini tidak memiliki kondisi untuk mendapatkan item tersebut, item bisa didapatkan dengan cara membuka chest setelah stage berhasil ditaklukan.

Gambar 1. Tampilan dalam game

Tampilan chest pada in game adalah seperti pada gambar 3. Setelah stage selesai akan ada area untuk mendapatkan item yang dapat diperoleh melalui interaksi dengan chest. Pada area ini tidak akan ada musuh yang akan mengganggu karakter saat mendapatkan item.

Gambar 2. Tampilan setelah item diperoleh

Setelah membuka chest yang ada item-item yang diperoleh akan ditampilkan di tengah game dan untuk menutup layer item, setelah didapatkan player bisa klik dimana saja untuk menutup lalu claim item dan mengkonfersi item-item yang didapatkan ke poin untuk upgrade status character pada game.

Pengujian dilakukan 10 simulasi yang berbeda untuk menghitung distribusi drop rate item sudah seimbang dan pastinya tidak membuat proses upgrade status character tidak terlalu cepat *(overpower)* maupun lambat *(underpower)* supaya alur permainan tetap seimbang dengan status musuh yang sudah di tentukan. Visual dari 10 simulasi pengujian drop item menggunakan algoritma Weighted Random Sampling. Gambar berikut menunjukkan hasil visualisasi item drop lengkap dengan border warna untuk menandai rarity:

Gambar 3. Hasil Simulasi Drop Item ke-1

Pada simulasi pertama item yang diperoleh sebanyak 15 item, pada simulasi pertama ini item yang didapatkan adalah werewolf feet dengan rarity common(border berwarna hijau) sebanyak 6 item, red claw dengan rarity rare(border berwarna biru) sebanyak 3 item, dan terakhir broken scythe dengan rarity epic(border berwarna ungu) sebanyak 6 item.

Gambar 4. Hasil Simulasi Drop Item ke-2

Pada simulasi kedua item yang diperoleh sebanyak 19 item dan item yang didapatkan adalah werewolf feet dengan rarity common(border berwarna hijau) sebanyak 15 item, red claw dengan rarity rare(border berwarna biru) sebanyak 3 item, dan terakhir broken scythe dengan rarity epic(border berwarna ungu) sebanyak 1 item.

Gambar 5. Hasil Simulasi Drop Item ke-3

Pada simulasi ketiga, item yang diperoleh sebanyak 19 item dan item yang didapatkan adalah werewolf feet dengan rarity common(border berwarna hijau) sebanyak 9 item, red claw dengan rarity rare(border berwarna biru) sebanyak 8 item, dan terakhir broken scythe dengan rarity epic(border berwarna ungu) sebanyak 2 item.

Gambar 6. Hasil Simulasi Drop Item ke-4

Pada simulasi keempat, item yang diperoleh sebanyak 17 item dan item yang didapatkan adalah werewolf feet dengan rarity common(border berwarna hijau) sebanyak 7 item, red claw dengan rarity rare(border berwarna biru) sebanyak 9 item, dan terakhir broken scythe dengan rarity epic(border berwarna ungu) sebanyak 1 item.

Gambar 7. Hasil Simulasi Drop Item ke-5

Pada simulasi kelima, item yang diperoleh sebanyak 16 item dan item yang didapatkan adalah werewolf feet dengan rarity common(border berwarna hijau) sebanyak 9 item, red claw dengan rarity rare(border berwarna biru) sebanyak 4 item, dan terakhir broken scythe dengan rarity epic(border berwarna ungu) sebanyak 3 item.

Gambar 8. Hasil Simulasi Drop Item ke-6

Pada simulasi keenam item yang diperoleh sebanyak 19 item dan item yang didapatkan adalah werewolf feet dengan rarity common(border berwarna hijau) sebanyak 11 item, red claw dengan rarity rare(border berwarna biru) sebanyak 8 item, dan terakhir broken scythe dengan rarity epic(border berwarna ungu) sama sekali tidak drop.

Gambar 9. Hasil Simulasi Drop Item ke-7

Pada simulasi ketujuh item yang diperoleh sebanyak 15 item dan item yang didapatkan adalah werewolf feet dengan rarity common(border berwarna hijau) sebanyak 7 item, red claw dengan rarity rare (border berwarna biru) sebanyak 5 item, dan terakhir broken scythe dengan rarity epic (border berwarna ungu) sebanyak 3 item.

Gambar 10. Hasil Simulasi Drop Item ke-8

Pada simulasi kedelapan item yang diperoleh sebanyak 17 item dan item yang didapatkan adalah werewolf feet dengan rarity common(border berwarna hijau) sebanyak 7 item, red claw dengan rarity rare(border berwarna biru) sebanyak 9 item, dan terakhir broken scythe dengan rarity epic (border berwarna ungu) sebanyak 1 item.

Gambar 11. Hasil Simulasi Drop Item ke-9

Pada simulasi kesembilan item yang diperoleh sebanyak 18 item dan item yang didapatkan adalah werewolf feet dengan rarity common(border berwarna hijau) sebanyak 10 item, red claw dengan rarity rare(border berwarna biru) sebanyak 6 item, dan terakhir broken scythe dengan rarity epic(border berwarna ungu) sebanyak 2 item.

Gambar 12. Hasil Simulasi Drop Item ke-10

Pada simulasi kesepuluh item yang diperoleh sebanyak 15 item dan item yang didapatkan adalah werewolf feet dengan rarity common(border berwarna hijau) sebanyak 9 item, red claw dengan rarity rare(border berwarna biru) sebanyak 4 item, dan terakhir broken scythe dengan rarity epic(border berwarna ungu) sebanyak 2 item.

Pengujian dilakukan sebanyak 10 kali simulasi drop item menggunakan sistem yang telah diimplementasikan. Setiap simulasi menghasilkan antara 15 hingga 20 item, yang kemudian dikategorikan berdasarkan jenis dan rarity-nya lalu jumlah total item akan dihitung dan dialokasiakan menjadi poin yang dapat digunakan untuk meng upgrade stat pada character pada game.

Tabel 2. Tabel pengujian

Simulasi	Common	Rare	Epic	Total Poin
1	6	3	6	210
2	15	3	1	130
3	9	8	2	175
4	7	9	1	150
5	9	4	3	160
6	11	8	0	135
7	7	5	3	135
8	7	9	1	150
9	10	6	2	160
10	9	4	2	135

Rata-rata per jenis item item yang didapatkan dari sepuluh kali pengujian adalah Common: 9.1 item (≈ 58.8%), Rare: 6.9 item (≈ 29.3%), Epic: 3.5 item (≈ 11.9%). Distribusi ini menunjukkan bahwa algoritma *Weighted Random* Sampling yang diterapkan mampu menghasilkan sistem drop item yang cukup seimbang dan dapat dikendalikan sesuai bobot. Meskipun terdapat fluktuasi minor, proporsi kemunculan item common, rare, dan epic tetap dalam rentang yang mencerminkan kelangkaan relatif masing-masing jenis item. Setiap item memiliki nilai poin: Common = 5, Rare = 10, dan Epic = 25. Nilai ini diakumulasikan dan digunakan untuk meningkatkan status karakter dalam permainan. Rata-rata total poin dari seluruh pengujian adalah 156,5 poin. Sistem juga berhasil menampilkan item secara visual dengan layout grid yang rapi, border sesuai rarity, label instruksi, serta animasi tween yang mendukung pengalaman visual pemain. Tidak ditemukan gangguan fungsional maupun tampilan dalam pengujian. Pendekatan ini sejalan dengan tren desain game modern seperti Genshin Impact dan Borderlands, di mana pembagian drop item secara proporsional membuat pengalaman bermain lebih seimbang dan memotivasi pemain untuk melanjutkan progres (Hamari and Lehdonvirta 2010).

Simpulan

Berdasarkan hasil penelitian, implementasi algoritma *Weighted Random Sampling* dalam sistem drop rate item pada game action platformer berbasis Godot Engine terbukti efektif dalam menciptakan distribusi item yang seimbang dan adil sesuai tingkat kelangkaan yang ditetapkan. Sistem ini mampu mengatasi kelemahan metode acak seragam, menjaga ritme permainan tetap menantang, dan meningkatkan kepuasan pemain. Temuan ini menunjukkan bahwa penggunaan algoritma acak berbobot memiliki implikasi penting dalam meningkatkan kualitas pengalaman bermain, khususnya dalam sistem hadiah yang bersifat terukur namun tetap memberikan elemen kejutan. Penerapan pendekatan matematis seperti ini mendukung keterlibatan pemain dan dapat menjadi landasan bagi desain sistem reward yang lebih adaptif. Untuk penelitian selanjutnya, disarankan agar algoritma ini diterapkan dalam konteks yang lebih kompleks seperti sistem ekonomi game, jumlah item yang lebih beragam, atau dalam lingkungan multipemain. Selain itu, pengembangan lebih lanjut dapat mencakup penyesuaian bobot secara dinamis

berdasarkan performa atau preferensi pemain guna meningkatkan personalisasi dan retensi jangka panjang.

Daftar Pustaka

Arrazzaq, M. F., Sasmito, A. P., & Zahro', H. Z. (2023). Perancangan game 2D platformer 'Adventure Quest' dengan metode finite state machine berbasis Android. *JATI (Jurnal Mahasiswa Teknik Informatika)*, *7*(4), 2419–2427. https://doi.org/10.36040/jati.v7i4.7537

Bateman, S., Mandryk, R. L., Stach, T., & Gutwin, C. (2010). The effects of randomness on perceived fairness in games. In *Proceedings of the 28th ACM Conference on Human Factors in Computing Systems (CHI '10)* (pp. 1865–1874). ACM. https://doi.org/10.1145/1753326.1753595

Bradfield, C. (2018). *Godot engine game development projects: Build five cross-platform 2D and 3D games with Godot 3.0* (1st ed.). Packt Publishing.

Calin, A. (2020). *Making games with Godot*. Apress.

Cohen, E., & Kaplan, H. (2008). Weighted sampling without replacement from data streams. https://dl.acm.org/citation.cfm?id=3497495

Cook, M., & Colton, S. (2012). Multi-faceted evaluation for creativity support tools in game design. In *Proceedings of the International Conference on Computational Creativity (ICCC 2012)* (pp. 22–28).

Cox, B. (2024). Unity — How to create a weighted loot table. *Medium*. https://medium.com/@kshesho/unity-how-to-create-a-weighted-loot-table-3bcbf478eaf9

Dhule, M. (2022). *Beginning game development with Godot: Learn to create and publish your first 2D platform game*. Apress. https://doi.org/10.1007/978-1-4842-7455-2

Efraimidis, P. S. (2015). Weighted random sampling over data streams. *arXiv*. https://doi.org/10.48550/arXiv.1012.0256

Fullerton, T., & Zimmerman, E. (2019). *Game design workshop: A playcentric approach to creating innovative games* (4th ed.). CRC Press. https://doi.org/10.1201/b22309

Hamari, J., & Lehdonvirta, V. (2010). Game design as marketing: How game mechanics create demand for virtual goods. *International Journal of Business Science and Applied Management*, *5*(1), 14–29. https://doi.org/10.69864/ijbsam.5-1.48

Hunicke, R., LeBlanc, M., & Zubek, R. (2004). MDA: A formal approach to game design and game research. https://users.cs.northwestern.edu/~hunicke/MDA.pdf

Khandelwal, S., & Gupta, S. (2023). Designing fair reward systems in video games using probabilistic models. *Journal of Game Design and Development Education*, *8*, 45–58.

Lee, A. (2013). Visual feedback techniques in game design. *Game Developer Magazine*.

Li, Y. (2018). Efficient weighted random sampling for large-scale data processing. In *Proceedings of the International Conference on Very Large Data Bases* (pp. 1133–1142). ACM.

Lin, C., & Lin, J. (2021). *Godot engine game development projects* (2nd ed.). Packt Publishing.

Loot drop best practices. (2023). *GameDeveloper.com*. https://www.gamedeveloper.com/design/loot-drop-best-practices

Nielsen, R., & Grabarczyk, P. (2019). Are loot boxes gambling? Random reward mechanisms in video games. *Transactions of the Digital Games Research Association (ToDiGRA)*, *4*, 33–67.

Octodinata, S., Pragantha, J., & Haris, D. A. (2023). Pembuatan game 2D platformer 'Save the Foxy' pada website. *Jurnal Serina Sains, Teknik dan Kedokteran*, *1*(2), 431–442. https://doi.org/10.24912/jsstk.v1i2.31034

Schell, J. (2023). *The art of game design: A book of lenses* (4th ed.). CRC Press.

Shiffman, D. (2012). *The nature of code: Simulating natural systems with Processing*. https://natureofcode.com

Spicer, S. G., & Close, J. R. (2022). Loot boxes, problem gambling and problem video gaming: A systematic review. *New Media & Society*, *24*(1), 1–26. https://doi.org/10.1177/14614448211045645

Stokholm, A. (2019). How to: Weighted random selections. *LootLocker Blog*. https://lootlocker.com/blog/random-with-weights

Yee, N. (2006). Motivations for play in online games. *CyberPsychology & Behavior*, *9*(6), 772–775. https://doi.org/10.1089/cpb.2006.9.772