# IMPLEMENTASI ALGORITMA *MULTIPLY WITH CARRY* PADA GAME PANAHAN

## *ALGORITHM IMPLEMENTATION OF MULTIPLY WITH CARRY ON ARCHERY GAME*

**Yogi Pribadi Mulya[1], Roswan Latuconsina[2], Anton Siswo Raharjo Ansori[3]**
[1]Prodi S1 Teknik Komputer, Fakultas Teknik, Universitas Telkom
[2]Prodi S1 Teknik Komputer, Fakultas Teknik, Universitas Telkom
[3]Prodi S1 Teknik Komputer, Fakultas Teknik, Universitas Telkom
[1]**yogipribadim@student.telkomuniversity.ac.id**, [2]**@telkomuniversity.co.id**,
[3]**@raharjo@telkomuniversity.ac.id**

**Abstrak**

**Ketika seseorang sedang dilanda kejenuhan akibat akitivitas pekerjaan yang cukup padat tentu akan menimbulkan stress dan akan menghambat aktivitas kedepannya. Maka dari itu, orang-orang melampiaskannya dengan memainkan sebuah permainan yang ada pada perangkat keras seperti komputer maupun gawai. Banyak hal positif yang dapat diambil dari bermain game atau permainan, contohnya seperti brainstorming, ketelitian, memperbaiki suasana hati, dan meningkatkan konsentrasi. Dari generasi ke generasi, permainan semakin canggih dan tampilan interface–nya realistis. Pengembang permainan sekarang, dituntut untuk semakin kreatif dalam membuat suatu permainan dengan tujuan menarik perhatian khalayak umum terutama gamer.**

**Dalam penelitian kali ini, penulis akan mengembangkan permainan panahan *(Archery Game)* dengan menggunakan *Procedural Movement Generator* sebagai prosedur dalam pemrograman dalam bentuk skrip yang ditulis dalam perangkat lunak game development yaitu *Unity3D*. Algoritma yang digunakan pada penilitian ini adalah Multiply with Carry yaitu implementasi dari metode *Pseudorandom Number Generator (PRNG)* yang menggabungkan dua atau lebih sebuah generator linear kongruensial. Gabungan *Multiply with Carry* juga memiliki sebuah algoritma khusus di dalamnya, di mana beberapa variabelnya mendeklarasikan modulus-modulus dari *Multiply with Carry* dan nilai acaknya.**

**Kata Kunci: Game, *Brainstorming, interface, Pseudorandom Number Generator(PRNG), Unity3D.***

**Abstract**

***When someone is hit by boredom due to work activities that are quite dense it will certainly cause stress and will hinder future activities. Therefore, people take it out by playing a game that is on hardware such as computers or devices. Many positive things can be taken from playing games or games, for example such as brainstorming, accuracy, improving mood, and increasing concentration. Over generations, the game has become more sophisticated and the interface looks realistic. Game developers now, are required to be more creative in making a game with the aim of attracting the attention of the general public - especially gamers.***

***In this research, the author will develop an archery game using the Procedural Movement Generator as a procedure in programming in the form of scripts written in game development software, Unity3D. The method used in this research is Multiply with Carry, which is the implementation of a Pseudorandom Number Generator (PRNG) which combines two or more congruential linear generators. Combined Multiply with Carry also has a special algorithm in it, in which some of the variables declare the modulus of Multiply with Carry and its random value***

**Keywords: Game, *Brainstorming, interface, Pseudorandom Number Generator(PRNG), Unity3D.***

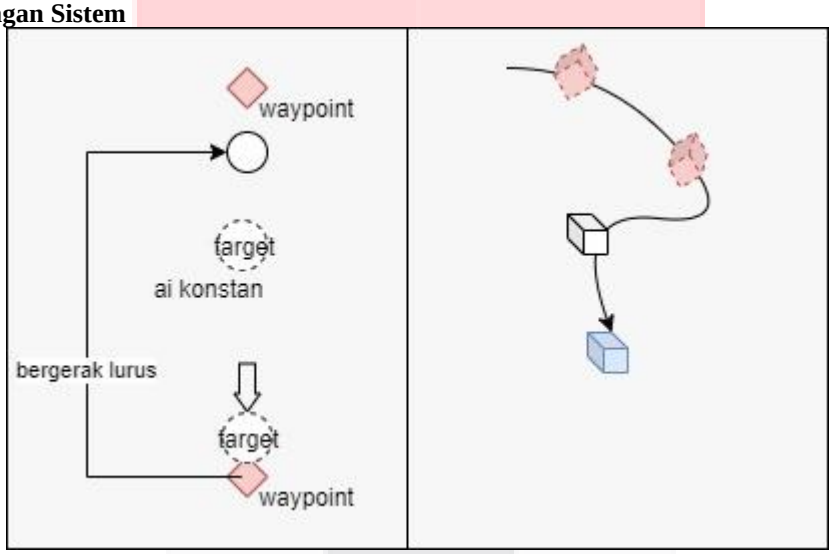
## 1. Pendahuluan

**Perkembangan aplikasi permainan pada zaman sekarang ada berbagai macam, mulai dari grafis yang makin ditingkatkan, berbagai macam genre dan ditujukan usia tertentu. Ini**

**menjadikan para developer aplikasi permainan yang diharuskan memiliki tantangan dan tingkat kesulitan yang menyesuaikan dengan kemampuan pemain. Pemain itu ada dua jenis yaitu *Hardcore Gamers* yang menyukai tantangan pada permainan dan ingin menyelesaikannya dan Casual Gamers yang cenderung mudah menyerah ketika permainannya terlalu sulit. Teknik/metode yang digunakan pada penilitian ini adalah dengan gabungan *Pseudorandom Number Generator (PRNG)* dan *Multiply-with Carry*. Tujuannya memberikan kesan atraktif seperti mengacak objek-objek yang dijelaskan sebelumnya, yaitu untuk membuat game lebih menarik atau tidak monoton dan memiliki tantangan bagi pemainnya.**

**Pada proses scripting, menggunakan metode *Pseudorandom Number Generator (PRNG)* melalui algoritma *Multiply-with Carry*. Metoda ini dipilih karena pada bagian target di dalam permainan menerapkan penggunaan scripting di mana semua bagian konten dapat dihasilkan dengan tulisan kode yang terstruktur dengan baik dan benar. Alasan yang memungkinkan dari penggunaan metode ini dilihat dari tingkat acak (*random*) dari konten yang dihasilkannya. Mengusahakan agar pemain nantinya merasakan permainan yang menantang. Dengan adanya *Multiply-with Carry*,pemain beradaptasi dengan target yang tidak monoton sebelumnya dengan berupa pergerakan target secara statis (kiri kanan) sehingga tingkat kesulitan semakin tinggi.**

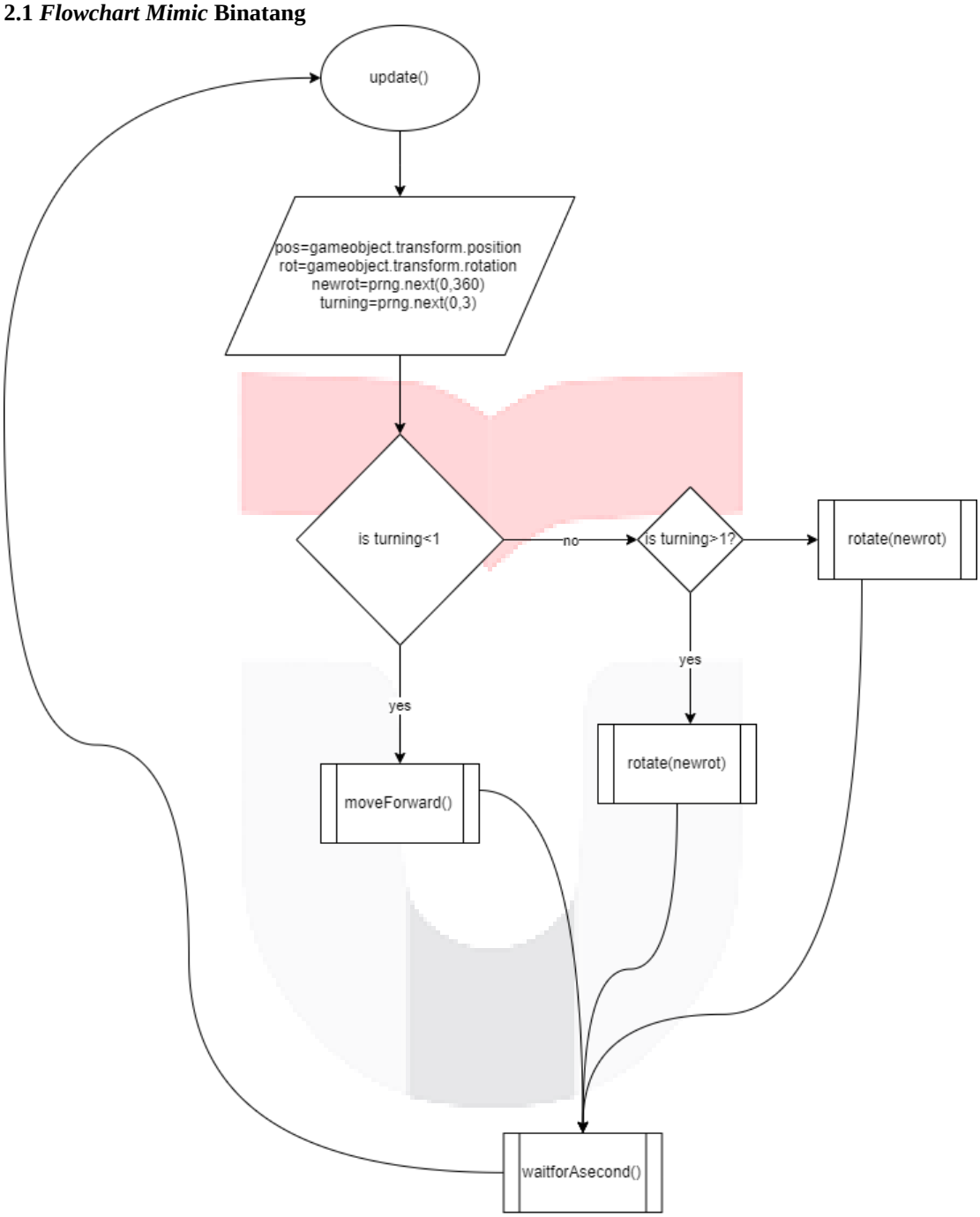
**2. Perancangan Sistem**




Gambar 1. gambaran umum sistem

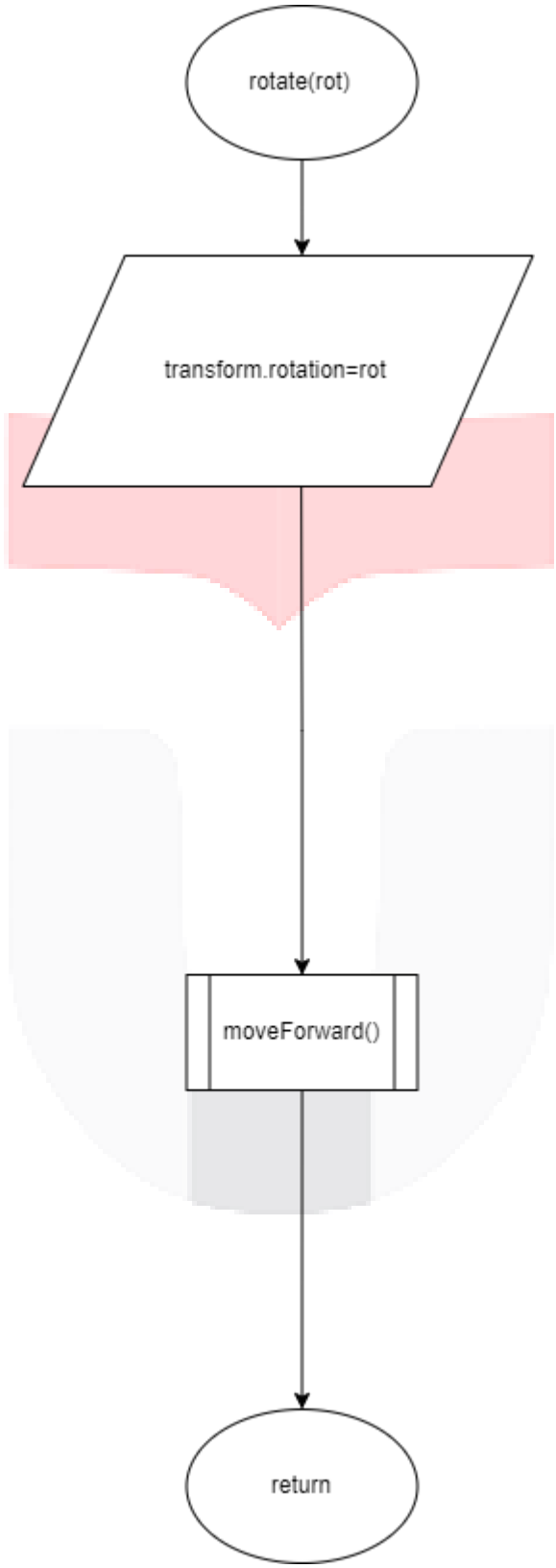
Dalam penelitian ini target akan bergerak pada *waypoint* yang ditentukan oleh *Pseucoderandom Number Generator* (PRNG) sebagai *spawn* suatu objek. PRNG yang digunakan adalah *Multiply with Carry* maka angka *random* yang digunakan akan selalu berbeda berdasarkan *seed*, akan tetapi jika menggunakan *seed* yang sama akan menghasilkan pola angka yang sama. Pada dasarnya sistem pergerakan *target* adalah bergerak lurus atau maju kedepan. Objek yang di *spawn* akan diacak peletakannya melalui property *transform* pada *Unity3d* dengan *range* yang sudah ditentukan, agar tidak melebihi dari luas arena.

**2.1 *Flowchart Mimic* Binatang**



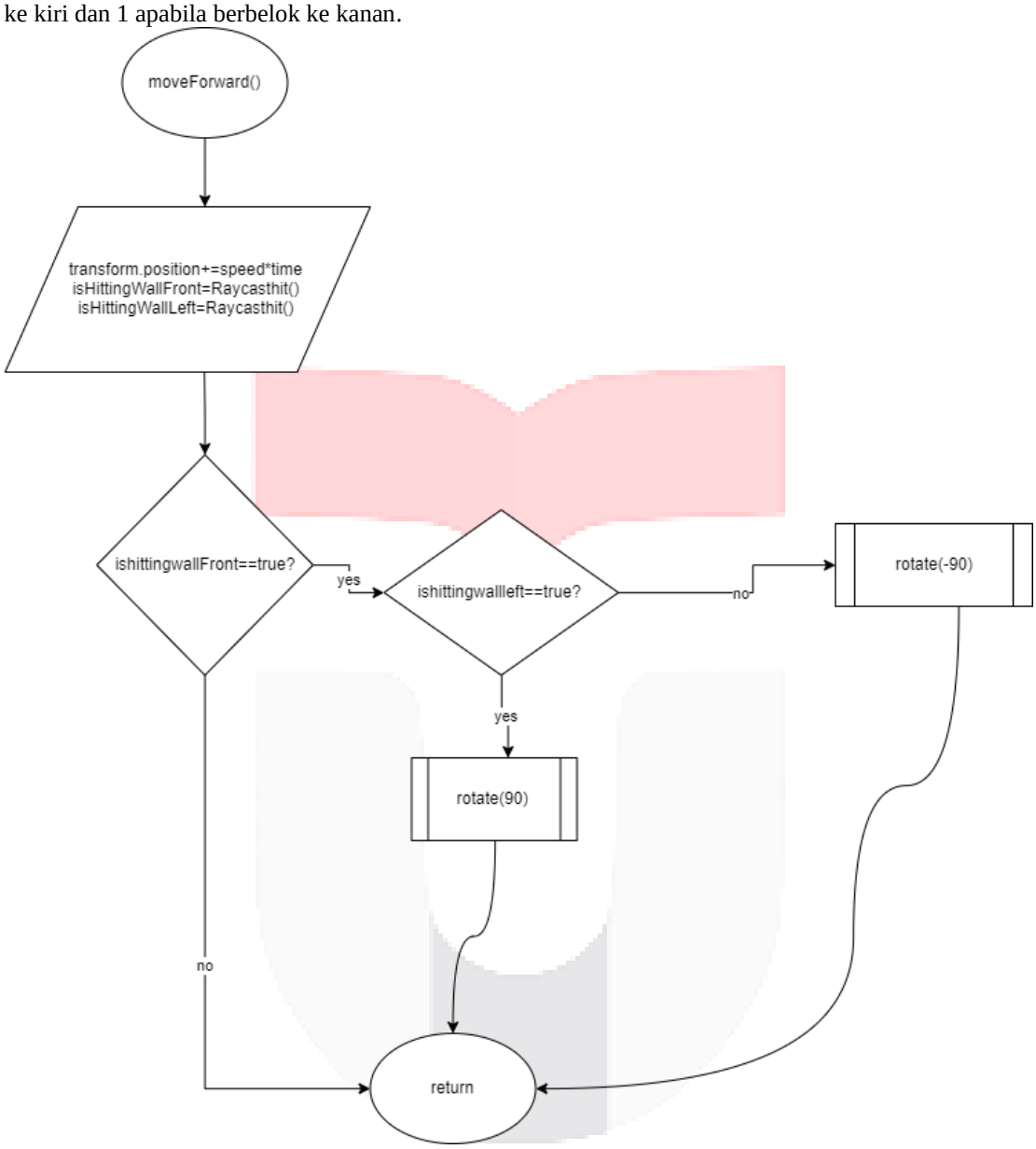

Gambar 2.1 *Mimic* Binatang

Setiap ai wanderer akan memiliki gambaran pada update kurang lebih seperti pada gambar di atas, nilai prng next 0,360 akan menghasilkan nilai dari 0 sampai 359 derajat. Akan di tentukan kemana akan berputar berdasarkan nilai turning. Nilai turning berada pada 0,1 dan 2 dimana 0 tidak berbelok 1 berbelok ke kanan dan 2 berbelok ke kiri.




Gambar 2.1.2 *Flowchart* Arah

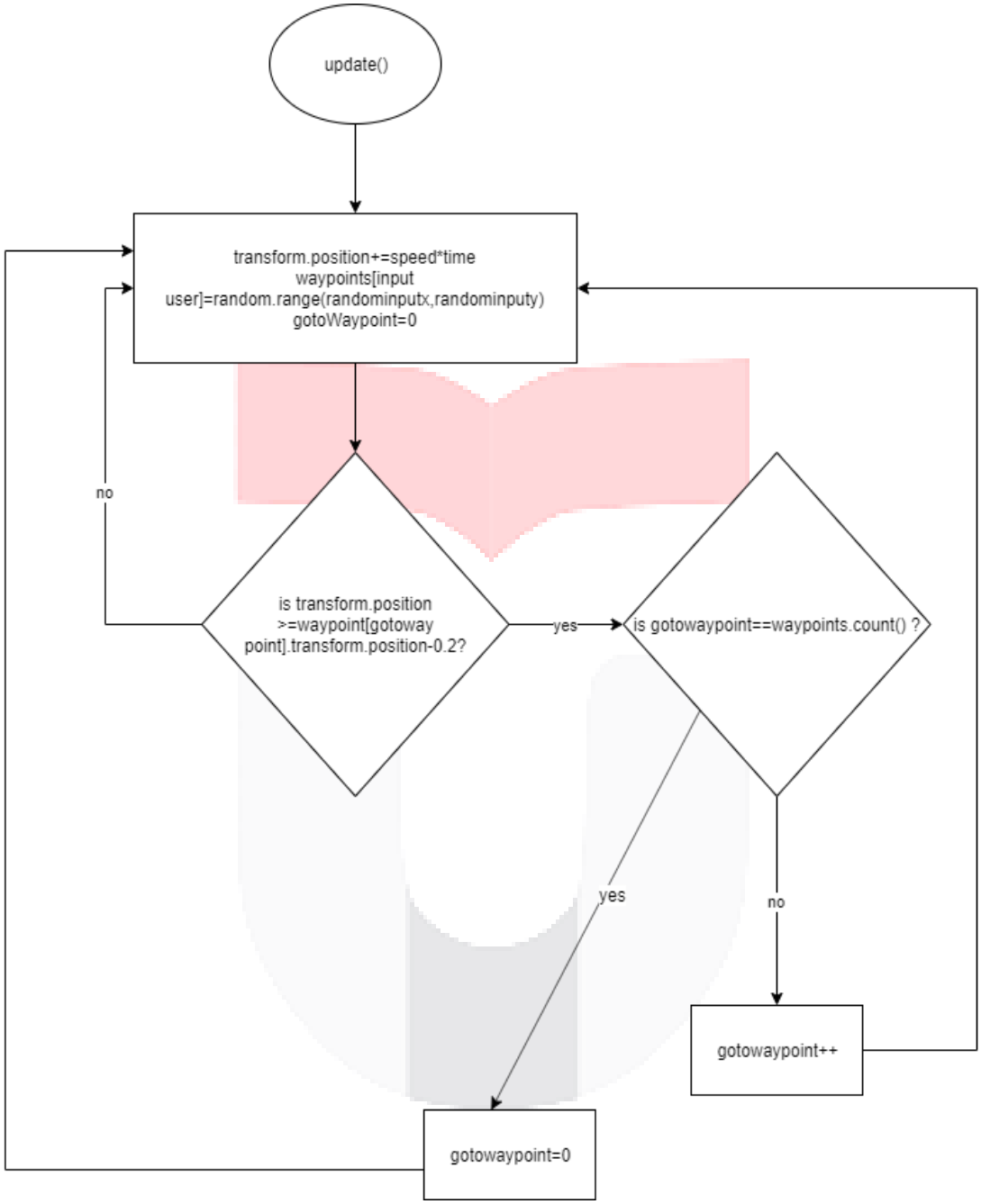
Nilai rotasi akan diterima melalui fungsi update. Nilai rotasi akan dikalikan dengan -1 apa bila berbelok ke kiri dan 1 apabila berbelok ke kanan.




Gambar 2.1.3 *Flowchart Rotate*

Untuk *move forward* atau jalan kedepan akan di jalankan setelah berbelok. Apabila di depan dan dikiri ada tembok dan ai akan *wanderer* akan bergerak ke kanan. Apabila ada tembok di sebelah kanan maka sebaliknya. Nilai hit akan didapatkan melalui sensor dari *unity3d* berupa *raycast*. Minimal terdapat dua *raycast* untuk deteksi tembok pada ai wanderer.

**2.2 *Flowchart* Statis**




Gambar 2.2 *Flowchart* Statis

Berikut merupakan gerakan untuk objek statis, untuk banyak *waypoint* akan di tentukan oleh *input user*, sedangkan untuk titik dimana *waypoint* akan berupa vektor dimensi tiga yang posisinya akan dirandom sesuai dengan batas maksimal di dimensi x dan z pada *unity3d*.

**3. Pengujian sistem**
***Position Bird***

Pada pengujian ini akan dilakukan 13 kali pengacakan, dengan seed mulai dari 2 sampai 1000 dengan vektor x, y, dan z dan didapatkan 74 data sehingga didaptkan nilai Mean, Standard error, Median, Standard deviation, Sample variance

Tabel 1. *Bird*

| **koordinat** | ***Mean*** | ***Standard error*** | ***Median*** | ***Standard deviation*** | ***Sample variance*** |
|---|---|---|---|---|---|
| X | 14.54935 | 0,913922 | 13,22719 | 68.09187381 | 4636.503279 |
| Y | 53.96991 | 0,628529 | 37,81376 | 46.82862849 | 2192.920446 |
| Z | 20.0843 | 1,732248 | 3,167259 | 129.0613194 | 16656.82416 |

***Position Kelinci***

Pada pengujian ini akan dilakukan 13 kali pengacakan, dengan *seed* mulai dari 2 sampai 1000 dengan vektor x dan y dan didapatkan 59.176 data sehingga didaptkan nilai *Mean, Standard error, Median, Standard deviation, Sample variance*

Tabel 2. Kelinci

| **koordinat** | ***Mean*** | ***Standard error*** | ***Median*** | ***Standard deviation*** | ***Sample variance*** |
|---|---|---|---|---|---|
| X | -26.754859 | 0,008130476 | -43,0199 | 225.4499636 | 50828.54504 |
| Z | -26.754859 | 0,006947926 | -235,018 | 192.6590329 | 37118.13022 |

***Rotation* Kelinci**

Pada pengujian ini akan dilakukan 13 kali pengacakan, dengan *seed* mulai dari 2 sampai 1000 dengan vektor x dan y dan didapatkan 72786 data sehingga didaptkan nilai *Mean, Standard error, Median, Standard deviation, Sample variance*

Tabel 3. Kelinci

| **koordinat** | ***Mean*** | ***Standard error*** | ***Median*** | ***Standard deviation*** | ***Sample variance*** |
|---|---|---|---|---|---|
| Y | 171,1077 | 0,003746 | 162,827 | 103,8569 | 103,8579 |

**Daftar Pustaka:**

[1] Wong .H, 2011, *A Study of The Video Game Industry In U.S Metropolitan Areas Using Occupational Analysis,*USA, University of Massachusetts Amherst.

[2] Larry Katz, James Parker, Hugh Tyreman, Gail Kopp, Richard Levy, Ernie Chang, 2006, *VIRTUAL REALITY in SPORT and WELLNESS: PROMISE and REALITY*, Canada, International Journal of Computer Science in Sport – Volume 4/Edition 1.

[3] Sasha Azad, Carl Saldanha, Cheng Hann Gan, and Mark O. Riedl, *Mixed Reality Meets Procedural Content Generation in Video Games*,USA, Experimental AI in *Game*s: Papers from the AIIDE Workshop AAAI Technical Report WS-16-22.

[4] Georgios N. Yannakakis, and Julian Togelius, 2011, *Experience-Driven Procedural Content Generation*, IEEE TRANSACTIONS ON AFFECTIVE COMPUTING, VOL. 2, NO. 3, JULY-SEPTEMBER 2011.

[5] Gilbert Nwankwo, Sabah Mohammed and Jinan Fiaidhi, 2017, *Procedural Content Generation for Dynamic Level Design and Difficulty in a 2D Game Using UNITY*, Canada, *International Journal of Multimedia and Ubiquitous Engineering Vol.12, No.9 (2017), pp.41-52.*

[6] C. Sommariva, E. Nardon, P. Beyer, M. Hoelzl, G. T. A. Huijsmans, D. van Vugt and JET Contributors, 2016, *Test particles dynamics in the JOREK 3D non-linear MHD code and application to electron transport in a disruption simulation*,France, CEA.

[7] Michael Stauffer, Thomas Hanne, Rolf Dornberger, 2016, *Uniform and Non-Uniform Pseudorandom number generators in a Genetic Algorithm Applied to an Order Picking Problem*, Switzerland, 2016 IEEE Congress on Evolutionary Computation (CEC).

[8] Melissa E. O'Neill, 2014, *PCG: A Family of Simple Fast Space-Efficient Statistikally Good Algorithms for Random Number Generation*, USA, ACM journals of mathematical software (submitted).

[9] Julian Togelius, Emil Kastbjerg, David Schedl and Georgios N, 2011 Yannakakis, *What is Procedural Content Generation? Mario on the borderline*, Denmark, DOI: 10.1145/2000919.2000922.

[10] Julian Togelius, Noor Shaker, and Mark J. Nelson, *Procedural Content Generation in Games*, Ch-1,[Online Books]

[11] Gillian Smith, Elaine Gan, Alexei Othenin-Girard and Jim Whitehead, 2011, *PCG-Based Game Design: Enabling New Play Experiences through Procedural Content Generation*, USA, DOI: 10.1145/2000919.2000926.

[12] E. Barker, *Recommendation for Key Management, Part 1: General*, NIST Special Publication 800-57 Part 1, Revision 4.

[13] Pierre L'Ecuyer, 2017, *History of uniform random number generation*, USA, WSC 2017 - Winter Simulation Conference.

[14] Jonas Freiknecht and Wolfgang Effelsberg, 2017, *A Survey on the Procedural Generation of Virtual Worlds*, Germany, Multimodal Technologies and Interact. 2017, 1, 27; doi:10.3390/mti1040027.

[15] Vincent Huber, Devon Miller, Aaron Nieto, Joseph Strawn, *Analyzing the Efficiency and Security of Permuted Congruential Number Generators*, [online].

[16] Jonathan Lockhart, Carla Purdy and Philip A. Wilsey, 2016, *Error Analysis and Reliability Metrics for Software in Safety Critical Systems*,USA, 2016 IEEE 59th International Midwest Symposium on Circuits and Systems (MWSCAS), 16-19 October 2016, Abu Dhabi, UAE.

[17] G.A.Sathishkumar, Srinivas Ramachandran, 2012, *Image Encryption Using Random Pixel Permutation by Chaotic Mapping*, India, 2012 IEEE Symposium on Computers & Informatics.

[18] Juergen Musil, Angelika Schweda, Dietmar Winkler, and Stefan Biffl, *Improving Video Game Development: Facilitating Heterogeneous Team Collaboration Through Flexible Software Processes*, Austria, Conference Paper in Communications in Computer and Information Science · September 2010 DOI: 10.1007/978-3-642-15666-3_8.

[19] I. Sazonov, D. Grebennikov, M. Kelbert1, G. Bocharov, 2017, *Modelling Stochastic and Deterministic Behaviours in Virus Infection Dynamics*, UK, Math. Model. Nat. Phenom. Vol. 12, No. 5, 2017, pp. 63–77.

[20] Harald Niederreiter and Igor E. Shparlinski, 2000, *ON THE DISTRIBUTION OF INVERSIVE CONGRUENTIAL PSEUDORANDOM NUMBERS IN PARTS OF THE PERIOD*, USA, MATHEMATICS OF COMPUTATION Volume 70, Number 236, Pages 1569–1574**.**

[21] Adam Noela, Karen C. Cheungb, Robert Schoberc, Dimitrios Makrakisa, Abdelhakim Hafidd, 2017, Simulating with AcCoRD: Actor-Based Communication via Reaction-Diffusion, Canada, arXiv:1612.00485v2 [physics.chem-ph] 2 Feb 2017.

[22] Tomasz Mazuryk and Michael Gervautz, 1999, *History, Applications, Technology and Future*, Austria, Institute of Computer Graphics Vienna University of Technology.

[23] H. Fuchs, G. Bishop, 1992, *Research Directions in Virtual Environments*, USA, NFS Invitational Workshop, Univ. North Carolina (1992).

[22] M. Gigante, 1993, *Virtual Reality: Definitions, History and Applications. "Virtual Reality Systems"*, Academic-Press, ISBN 0-12-22-77-48-1, pp. 3-14.

[24] Jargon, 1995, *Jargon Dictionary*. http://www.fwi.uva.nl/~mes/jargon/

[25] C. Cruz-Neira, 1993, *Virtual Reality Overview*, SIGGRAPH'93 Course, No. 23, pp. 1.1-1.18

[26] L. and E. von Schweber, 1995, *Virtually Here*, PC Magazine - March 14, 1995, pp. 168-198.

[27] Donald E.Knut, 1997, *The art of Computer Programming*, USA, American Mathematical Society

[28] Guy L. Steele Jr, Doug Lea, Christine H. Flood, *Fast Splittable Pseudorandom number generators*, USA, Proceedings of the 2014 ACM International Conference on *Object* Oriented Programming Systems Languages & Applications Pages 453-472.

[29] Raúl Lara-Cabrera, Mariela Nogueira-Collazo, Carlos Cotta and Antonio J. Fernández-Leiva, *Procedural Content Generation for Real-Time Strategy Games*, Spain, Universitas Málaga

[30] Leonardo T.Pereira, Claudio Toledo, Lucas N. Ferreira and Levi H.S Lelis, 2016,*Learning to Speed Up Evolutionary Content Generation in Physics-Based Puzzle Games*, USA, 2016 IEEE 28th International Conference on Tools with Artificial Intelligence.

[31] Yi-Si Xing, Xiaoping P. Liu, Shao-Ping Xu, *Efficient Collision Detection Based on AABB Trees and Sort Algorithm*, 2010 8th IEEE International Conference on Control and Automation.

[32] Rouse, Richard, *Game Design: Theory And Practice, Second Edition.* Los Rios Boulevard Plano, Texas: Wordware Publishing, 2004,pp 449-474.