Aplikasi Konsep Graf Berarah Pada Animation System Flow Sebagai Fitur Esensial Game Engine Unity

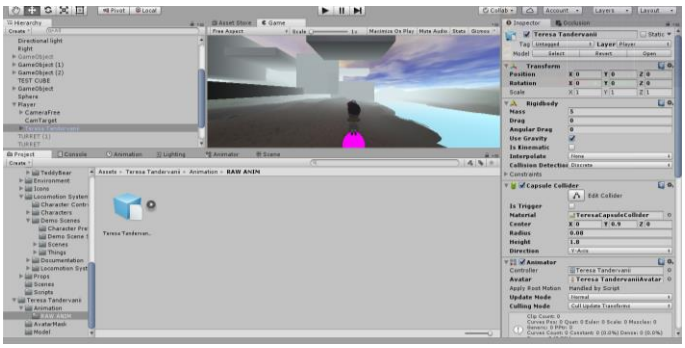
FARRELL ABIEZA ZIDAN / 13519182
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13519182@std.stei.itb.ac.id

***Abstraksi*— Animation Flow System merupakan sistem alur pada suatu clip animasi monoton menjadi animasi kompleks dengan beberapa clip berbeda sesuai dengan state yang dimilikinya saat itu. Animation Flow System diterapkan pada game engine Unity dalam konsep dan mekanisme animasi systemnya, karena metode system ini memudahkan pengguna pemula unity memahami interfacenya secara cepat, tanpa mengurangi fitur fitur penting dari metode sistem animasi lainnya. Penulis tertarik dengan mekanisme animation flow system yang dimiliki unity, ada banyak hal yang sebenarnya berkaitan dengan ilmu matematika diskrit, terutama pada konsep graf berarah pada sistem tersebut.**

***Keywords*— Graf berarah, System animation, Node, State, User Interface, Flow, Conditions**

I. INTRODUCTION

Unity merupakan game engine cross-platform yang bisa dibilang cukup dikenal banyak orang karena fiturnya dan hasil produksi gamenya yang berkualitas. Salah satu fitur esensial yang terdapat dalam unity adalah sistem animasinya yang sebenarnya menerapkan konsep graf berarah. Pengaplikasian graf berarah membuat unity mampu menciptakan objek dengan mekanisme alur animasi yang kompleks dengan algoritma pada script sederhana saja, atau bahkan tanpa menggunakan script. Animasi objek tanpa menggunakan script menandakan bahwa game engine unity menjadi engine user friendly yang mudah digunakan bagi setiap kalangan programer profesional maupun pemula.



Gambar 1.1 User Interface Game Engine Unity
(From My latest unity project game, "The Primitive")

Graf berarah berbobot menjadi konsep ekivalen pada Animator tab dalam Unity. Sebagai contoh jika dalam graf terdapat suatu istilah yang disebut sebagai titik atau vertex, maka dalam unity hal tersebut ekivalen dengan sebuah kontainer bernama "State", edge dalam graf merupakan flow pada Animator controller dalam unity yang fungsinya akan menghubungkan suatu State A ke state B, dengan catatan edge yang menghubungkan state A ke state B belum tentu menghubungkan state B ke state A (Karena itu disebut sebagai graf berarah). Dalam konsep graf pula, ada yang disebut sebagai weight atau bobot, Weight ini akan ekivalen dengan state flow yang terdapat dalam Edge flow animation pada unity yang akan dijelaskan pada segmen 3.I pada makalah ini.

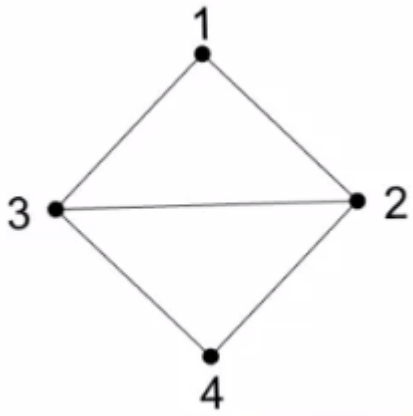
Istilah istilah ekivalen ini cukup unik sehingga bisa menjadi bahan pembicaraan yang panjang dan dapat dibuat sebagai topik makalah ini, karena itu penulis membuat makalah berjudul "Aplikasi Konsep Graf Berarah Pada Animation System Flow Sebagai Fitur Esensial Game Engine Unity".

II. TEORI DASAR

2.1 Graf

2.1.1 Definisi Graf

Suatu graf $G = (V,E)$ terdiri dari suatu himpunan V, yaitu himpunan tidak-kosong yang terdiri dari simpul-simpul yang ada di dalam G dan suatu himpunan E, yaitu suatu himpunan yang terdiri dari sisi-sisi yang menghubungkan sepasang simpul di dalam G.



Gambar 2.1 Contoh Graf
(https://lmsspada.kemdikbud.go.id/mod/resource/view.php?id=47640)

2.1.2 Jenis-Jenis Graf

Berdasarkan ada tidaknya gelang atau sisi ganda pada suatu graf, graf digolongkan menjadi dua jenis, yaitu:

1. Graf Sederhana (simple graph)

Graf sederhana adalah graf yang tidak mengandung gelang maupun sisi ganda.

2. Graf tak-sederhana (unsimple graph)

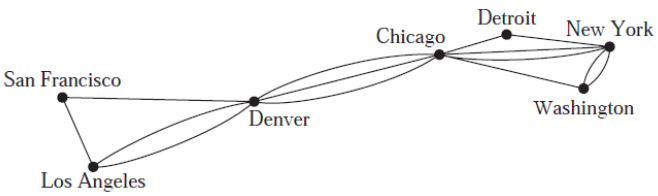
Graf tak-sederhana adalah graf yang mengandung sisi ganda atau gelang. Graf tak-sederhana kemudian dibagi lagi menjadi dua jenis, yaitu:

a. Graf Ganda (multigraph)

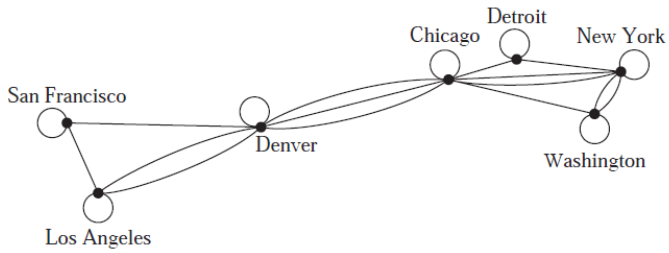
Graf ganda adalah graf yang mengandung sisi ganda.

b. Graf Semu (pseudograph)

Graf semu adalah graf yang mengandung sisi gelang.



Gambar 2.2 Graf Ganda
(https://www.slideshare.net/zachrisonmiruko/graf-matematika-diskrit)



Gambar 2.3 Graf Semu
(https://www.slideshare.net/zachrisonmiruko/graf-matematika-diskrit)

Sedangkan berdasarkan orientasi arah pada sisinya, graf digolongkan menjadi dua jenis, yaitu:

1. Graf tak-berarah (undirected graph)

Graf yang sisinya tidak mempunyai orientasi arah.

2. Graf berarah (directed graph)

Graf yang setiap sisinya diberikan orientasi arah.

2.1.3 Terminologi Graf

1. Ketetanggaan (Adjacent)

Dua buah simpul dikatakan bertetangga apabila terdapat suatu sisi yang menghubungkan kedua simpul tersebut secara langsung.

2. Bersisian (Incidency)

Suatu sisi e dikatakan bersisian dengan suatu simpul v apabila e menghubungkan v dengan simpul lain secara langsung.

3. Simpul Terpencil (Isolated Vertex)

Suatu simpul dikatakan terpencil apabila tidak ada sisi yang bersisian dengan simpul tersebut.

4. Graf Kosong (Null Graph)

Suatu graf dikatakan kosong apabila himpunan sisi dalam graf tersebut kosong.

5. Derajat (Degree)

Derajat suatu simpul adalah jumlah sisi yang bersisian dengan simpul tersebut. Pada graf berarah, derajat simpul kemudian dibagi lagi menjadi dua, yaitu derajat masuk dan derajat keluar.

6. Lintasan (Path)

Lintasan yang panjangnya n dari simpul awal $v0$ ke simpul tujuan vn di dalam graf G adalah barisan berselang-seling simpul-simpul dan sisi-sisi yang berbentuk $v0,e1,v1,\ldots,vn-1,en,vn$ sedemikian sehingga $e1=(v0,v1),e2=(v1,v2),\ldots,en=(vn-1,vn)$ adalah sisi-sisi dari graf G. Panjang lintasan adalah jumlah sisi di dalam lintasan tersebut.

7. Siklus (Cycle)

Siklus atau sirkuit adalah lintasan yang berawal dan berakhir pada simpul yang sama. Panjang sirkuit adalah jumlah sisi dalam sirkuit tersebut.

8. Keterhubungan (Connected)

Dua buah simpul dikatakan terhubung jika terdapat lintasan dari simpul satu ke simpul lainnya. Suatu graf G=(V,E) disebut graf terhubung jika untuk setiap pasang simpul dalam himpunan V terhubung. Jika ada satu atau lebih pasang simpul dalam himpunan V yang tidak terhubung, maka G disebut graf tak-terhubung.

Graf berarah G disebut terhubung apabila arah tiap-tiap sisinya dihilangkan akan menjadi graf tak-berarah terhubung. Dua simpul u dan v yang terdapat di dalam graf berarah G dikatakan terhubung kuat apabila terdapat lintasan berarah dari u ke v dan lintasan berarah dari v ke u. Apabila u dan v tidak terhubung kuat, tetapi jika arah tiap-tiap sisi pada graf dihilangkan akan menjadi terhubung, maka u dan v terhubung lemah.

9. Upagraf (Subgraph)

Upagraf dari suatu graf G=(V,E) adalah suatu graf yang himpunan simpulnya merupakan himpunan bagian dari V dan himpunan sisinya merupakan himpunan bagian dari E.

10. Upagraf Merentang (Spanning Subgraph)

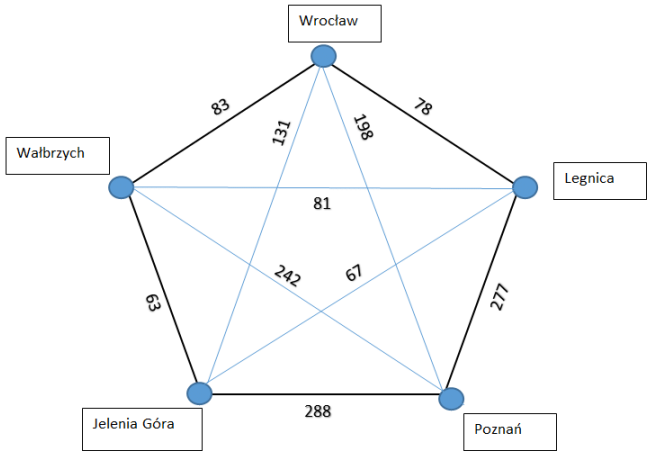
Upagraf merentang dari suatu graf G=(V,E) adalah suatu upagraf yang memiliki seluruh simpul yang dimiliki oleh G.

11. Cut-Set

Cut-set dari suatu graf terhubung G=(V,E) adalah suatu himpunan $E1$ yang merupakan himpunan bagian dari E dimana jika anggota E1 dibuang dari E akan mengakibatkan G menjadi graf tak-terhubung.

12. Graf Berbobot (Weighted Graph)

Graf berbobot adalah graf dimana tiap-tiap sisinya memiliki nilai tertentu.



Gambar 2.5 Weight Graf (https://www.researchgate.net/figure/Graf-weight-weight-distance-in-km-in-which-the-vertices-are-cities-By-carrying-out_fig1_327586118)

2.1.4 Graf Khusus

1. Graf Lengkap (Complete Graph)

Graf lengkap adalah graf yang setiap simpulnya bertetangga dengan seluruh simpul lain yang ada di graf. Jumlah sisi pada graf lengkap dengan n buah simpul adalah $n(n-1)2$.

2. Graf Lingkaran

Graf lingkaran adalah graf sederhana yang tiap simpulnya memiliki derajat dua. Jumlah sisi pada graf lingkaran dengan n simpul adalah n.

3. Graf Teratur (Regular Graph)

Graf teratur adalah graf yang tiap simpulnya memiliki derajat yang sama. Jumlah sisi pada graf teratur dengan n simpul yang tiap simpulnya berderajat r adalah $nr2$.

4. Graf Bipartit (Bipartite Graph)

Graf bipartite adalah graf yang himpunan simpulnya dapat dibagi menjadi dua himpunan sedemikian rupa sehingga seluruh sisi pada G menghubungkan sebuah simpul dari himpunan simpul satu ke salah salah satu simpul dari himpunan lainnya. Graf bipartite dinyatakan sebagai $G(V1,V2)$.

2.2 Animation Flow System

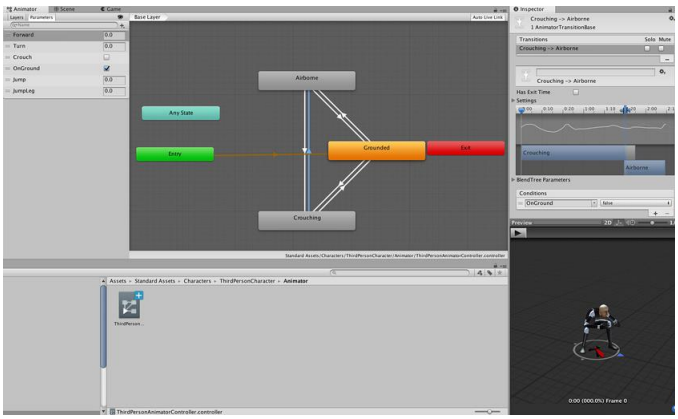
2.2.1 Mekanim

Animation system pada unity umumnya disebut sebagai mekanim, fiturnya adalah sebagai berikut:

- Workflownya dan setupnya mudah dipahami
- Support dengan clip animasi yang bahkan diimport diluar aplikasi unity
- Humanoid Animation Retargeting, yang

merupakan fitur khusus dalam pembuatan Inverse Kinematict pada anggota tubuh lengan dan kaki

- Manajemen interaksi setiap state pada mekanim yang kompleks
- Layering dan fitur Masking bone animasi



Gambar 2.6 Animation Flow Overview
(https://docs.unity3d.com/Manual/AnimationOverview.html)

2.3 Animation Overflow

Sistem animasi Unity didasarkan pada konsep "Animation Clip" Animasi, yang berisi informasi tentang bagaimana objek tertentu harus mengubah posisi, rotasi, atau properti lainnya dari waktu ke waktu. Setiap klip dapat dianggap sebagai rekaman linier tunggal. Klip animasi dari sumber eksternal dibuat oleh artist atau animator dengan third-party application seperti Autodesk, 3ds Max, atau Blender3D, atau berasal dari studio motion capture atau sumber lain.

Klip Animasi kemudian disusun menjadi sistem seperti diagram alur terstruktur yang disebut "Animator Controller"

Animator Controller sebagai "State Machine" yang pernah dipelajari pada mata kuliah lain, yaitu Teori Bahasa Dan Otomata.

Animator Controller yang sangat sederhana contohnya seperti membuat animasi powerup yang berputar dan memantul, atau animasi untuk menghidupkan pintu, membuka dan menutup pada waktu yang tepat. Animator Controller yang lebih canggih mungkin berisi lusinan animasi humanoid untuk semua aksi karakter utama, dan mungkin menyatu di antara beberapa klip pada saat yang sama untuk memberikan gerakan yang lancar saat pemain bergerak di sekitar adegan.

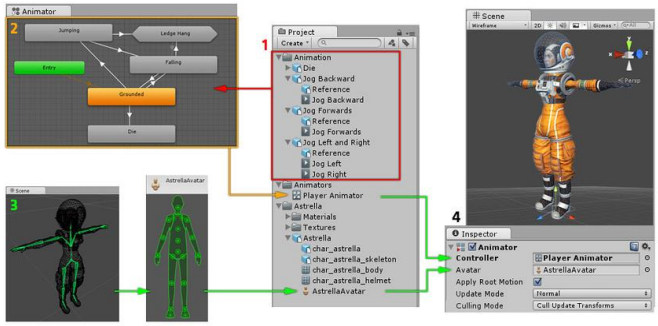
.

Sistem Animasi Unity juga memiliki banyak fitur khusus untuk menangani karakter humanoid yang kemampuan untuk menargetkan ulang animasi humanoid dari sumber mana pun (misalnya: penangkapan gerak; Asset Store; atau library / asset animasi third party application lainnya) ke model karakter, serta menyesuaikan definisi muscle dalam rig yang ada.

Fitur khusus ini dilakukan di dalam Unity's Avatar sistem, di mana karakter humanoid dipetakan ke format internal umum.

Masing-masing bagian ini - Klip Animasi, Animation Controler, dan Avatar, disatukan dalam GameObject melalui Komponen Animator

Komponen ini memiliki referensi ke Animator Controller, dan (jika diperlukan) Avatar untuk model ini. Untuk Animator Controller, berisi referensi ke Klip Animasi yang digunakannya.

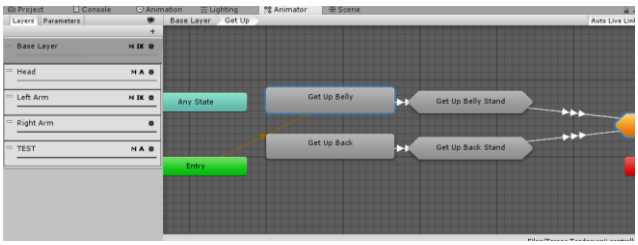


Gambar 2.7 Animation Overflow, Dan Humanoid Feature pada Unity
(https://docs.unity3d.com/Manual/AnimationOverview.html)

III. MEKANISME ANIMATION FLOW UNITY

3.1 Animator

Salah satu Interface dasar pada Unity disebut sebagai "Animator", merupakan gudang dari semua sistem kerja animasi yang ada dalam folder project unity tersebut. Interface ini umumnya merupakan sistem terapan dari graf berarah yang setiap edge-nya akan dilalui berdasarkan waktu yang sudah dilalui, atau menggunakan kondisional yang lebih spesifik



Gambar 3.1 Contoh Interface Animator
(From My latest unity project game, "The Primitive")

Dalam Unity, vertex atau titik diganti sebagai sebuah container yang disebut sebagai State. State ini bisa berbentuk dua hal, antara lain:

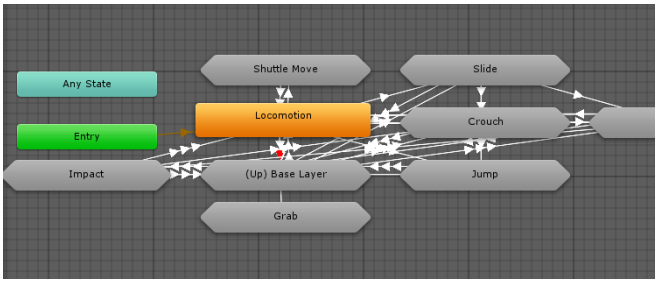
3.1.1 Single State

Single State merupakan state sederhana yang hanya memiliki satu clip animasi, state ini kemudian bisa

menuju beberapa state tergantung dengan kondisional apa yang mesti dipenuhinya. Jika state A misalnya akan dituju dengan state saat ini, maka edge yang menghubungkan state sekarang dengan state A hanya perlu satu saja kalaupun kondisionalnya berupa “Or conditional (|)”

3.1.2 Group State

Group state pada dasarnya merupakan state yang digabung untuk keperluan memudahkan animatornya dibaca, fiturnya kurang lebih sama, namun untuk memasuki group state biasanya pengguna unity hanya akan membuat sebuah state kosong yang nantinya menentukan kemana state selanjutnya memasuki state state yang hanya ada didalam group state tersebut.



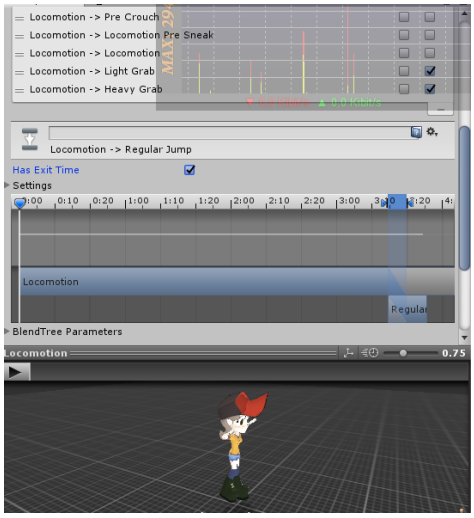
Gambar 3.2 Kumpulan State Kompleks, Untuk Single State Dinotasikan sebagai Container Kotak Lancip, Untuk Group State Dilambangkan Sebagai Container Yang Sedikit Tumpul, Untuk Container Berwarna Biasanya Merupakan Container Start Animasi
(From My latest unity project game, “The Primitive”)

3.2 Kondisional

Jika suatu animasi sedang berada pada suatu state, maka ada saat dimana state tersebut berpindah dari state ke state lainnya dengan kondisional tertentu atau dengan durasi tertentu sebagai pengganti kondisional jika tidak terdapat. Terdapat tiga kondisi yang mengakibatkan terjadinya perpindahan state ke state lain diantaranya:

3.2.1 Transisi Yang Memiliki Exit Time

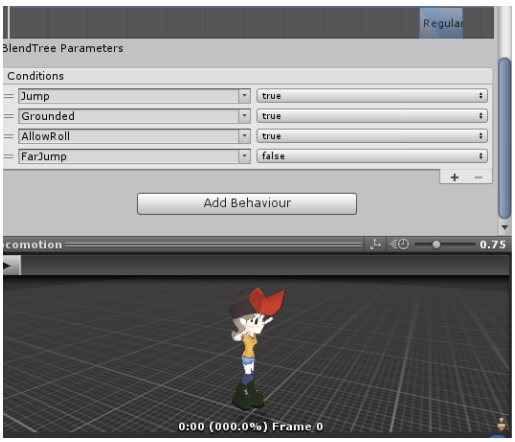
Transisi ini akan terjadi jika suatu Flow (Edge dalam Graf) menggunakan tipe Has Exit Time didalam state flow tersebut. Jika iya, maka state tersebut akan berpindah state jika dan hanya jika clip animasi sudah mencapai loop akhirnya, Atau dengan menggunakan konsep weight graph, maka transisi akan membuat clip saat ini dengan clip tujuan menjadi bercampur secara semu (Blending) dan faktor pencampurannya didasarkan pada konsep weight graph yang ada dalam ilmu graf.



Gambar 3.2 Weight Graf dianalogikan sebagai dua clip animasi yang digabung (Blending). Pada kondisi state flow ini, Memiliki “Has Exit Time” option yang membuat state berpindah jika state nya sudah selesai secara semu, sesuai pada clip berwarna biru pada gambar
(From My latest unity project game, “The Primitive”)

3.2.2 Transisi Dengan Kondisional Spesifik

Transisi dengan kondisional spesifik menggunakan variable tambahan yang dibuat oleh pengguna. Variable tambahan ini umumnya akan dimanipulasikan didalam script yang ada pada karakter, dan perubahannya akan di cross-check oleh system animation flow tersebut. Sebagai contoh, jika ada beberapa kondisional yang sudah terpenuhi, flow akan bernilai satu sehingga state dapat berpindah ke state tujuan yang ada, dalam konsep graf berarah, weight ketika kondisional belum terpenuhi akan ekivalen dengan nilai 0



Gambar 3.2 Contoh kondisional dalam suatu state untuk menuju state “Jumping” , diantaranya Suatu variable “Jump” harus true, variable “Grounded” harus bernilai true, “AllowRoll” harus bernilai true, dan “FarJump” harus bernilai False
(From My latest unity project game, “The Primitive”)

3.2.3 Transisi dengan Exit Time dan Kondisional Spesifik

Transisi ini merupakan transisi yang menganggap waktu sebagai kondisional yang harus dipenuhi juga selain kondisional lainnya yang ada. State akan berpindah kedalam state tujuan jika kondisional yang dimilikinya terpenuhi ketika clip animasi sudah mencapai loop akhir.

IV. KESIMPULAN

Aplikasi Graf Berarah Berbobot sangatlah banyak, bahkan untuk purpose pembuatan game salah satunya ditemukan dalam mekanisme sistem alur animasi pada game engine unity.

Konsep graf berarah yang diterapkan dalam fitur unity ini tentulah membuat game engine unity menjadi salah satu game engine ternama dikarenakan User Interface nya yang user friendly berkat konsep graf berarah, dan juga konsep tersebut tidak mengurangi fitur fiitur esensial lain yang memang seharusnya ada.

Sifat natural manusia yang selalu ingin tahu sangatlah berguna untuk aplikasi aplikasi kasus lainnya, dengan konsep Graf yang ada, bahkan secara tidak langsung menjadi peluang besar untuk kemajuan teknologi, yang diantaranya dari segi entertainmen dan game developing. Terlihat bahwa keberadaan game engine unity meningkatkan minat orang orang untuk menjadi seorang game developer secara pesat, dapat dibuktikan bahwa game game bergenre indie saat ini seringkali dijumpai di android, playstore, atau bahkan game pc yang rumit.

Hal kecil yang memudahkan game developer untuk memahami game engine yang digunakannya mungkin bukanlah hal yang besar, namun jika hal tersebut selalu diaplikasikan secara kontinu, maka seharusnya inovasi tersebut membuktikan bahwa dengan pemahaman konsep yang contohnya berada dalam ilmu matematika diskrit mampu membuka peluang seorang game developer untuk mengeksplor pasionnya lebih jauh.

VII. UCAPAN TERIMA KASIH

Segala puji dan syukur penulis panjatkan kepada kehadirat Allah SWT, karena berkat rahmat, hidayah dan karunia-Nya maka penulis dapat menyelesaikan makalah ini tepat pada waktunya.

Penulis juga mengucapkan terimakasih kepada Ibu Harlili sebagai Dosen pengampu mata kuliah IF2120 Matematika Diskrit – Sem. I Tahun 2020/2021 yang membimbing penulis hingga dapat menyelesaikan makalah ini tepat pada waktunya.

Terimakasih juga kepada seluruh teman kuliah yang selalu memberikan kritik mengenai makalah ini sebelum sesi upload makalah sehingga hasil akhir dari makalah yang dibuat menjadi lebih baik dari revisi yang sudah ada sebelumnya.

REFERENSI

1. http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/matdis20-21.htm
2. https://docs.unity3d.com/Manual/AnimationOverview.html
3. https://docs.unity3d.com/Manual/AnimationSection.html
4. https://en.wikipedia.org/wiki/Unity_(game_engine)
5. https://en.wikipedia.org/wiki/Graph_(discrete_mathematics)
6. https://www.geeksforgeeks.org/shortest-path-weighted-graph-weight-edge-1-2/
7. https://docs.unity3d.com/540/Documentation/Manual/Playables-GraphVisualizer.html

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 11 Desember 2020

Farrell Abieza Zidan / 13519182