

Pemodelan Graf Transisi Status pada Kombinasi Rantai Mantra (Spell Juggling) untuk Memaksimalkan Efisiensi Pertarungan di Hogwarts Legacy

Josephine Bintang N.L - 13525139

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jalan Ganesha 10 Bandung

E-mail: josephinebintang2007@gmail.com , 13525139@std.stei.itb.ac.id

Abstract—Makalah ini membahas penerapan Teori Graf dalam menganalisis kombinasi rantai mantra (*spell juggling*) pada permainan aksi perang *Hogwarts Legacy*. Mekanik pertarungan dalam permainan ini sangat bergantung pada efisiensi pemain dalam merangkai transisi antar mantra untuk memaksimalkan damage dan meminimalisir peluang musuh menyerang balik. Dengan memodelkan setiap mantra sebagai simpul (*node*) dan kemungkinan transisi antar mantra sebagai sisi (*edge*) pada graf berarah (*directed graph*), makalah ini bertujuan untuk memetakan dan menemukan lintasan kombinasi serangan yang paling optimal. Analisis graf transisi status ini diharapkan dapat memberikan panduan strategi pertarungan matematis yang efisien berdasarkan prinsip matematika diskrit.

Keywords—*Hogwarts Legacy*, Teori Graf, Graf Berarah, Kombinasi Mantra, Matematika Diskrit.

I. PENDAHULUAN

A. Latar Belakang

Permainan video modern tidak hanya mengandalkan refleks pemain, tetapi juga menuntut pemikiran strategis dan pengambilan keputusan yang cepat. Salah satu permainan yang mengimplementasikan mekanik pertarungan kompleks adalah *Hogwarts Legacy*, sebuah permainan aksi peran (*Action-RPG*) dunia terbuka. Dalam permainan ini, pemain diharuskan menggunakan berbagai macam mantra sihir untuk mengalahkan musuh.

Mekanik pertarungan utama dalam *Hogwarts Legacy* dikenal dengan istilah *spell juggling* atau merangkai kombinasi mantra. Alih-alih melancarkan satu serangan kuat secara terpisah, efisiensi pertarungan tercapai ketika pemain mampu melakukan transisi dari satu mantra ke mantra lainnya tanpa jeda yang signifikan. Kombinasi yang tepat dapat menembus pertahanan musuh, memberikan efek *crowd control*, dan menghasilkan *damage* yang berlipat ganda. Namun, dengan banyaknya variasi mantra yang tersedia,

menentukan urutan kombinasi yang paling optimal seringkali menjadi tantangan tersendiri.

Permasalahan pencarian urutan kombinasi mantra yang optimal ini dapat dimodelkan dan diselesaikan menggunakan pendekatan Matematika Diskrit, khususnya Teori Graf. Dengan merepresentasikan setiap mantra sebagai sebuah status (simpul) dan transisi yang dimungkinkan antar mantra sebagai sebuah relasi (sisi berarah), kita dapat membangun sebuah graf transisi status. Makalah ini akan memodelkan mekanik *spell juggling* tersebut ke dalam graf berarah untuk menganalisis dan menemukan lintasan serangan yang memaksimalkan efisiensi pertarungan.

B. Rumusan Masalah

Berdasarkan latar belakang yang telah dipaparkan, rumusan masalah dalam penelitian ini adalah sebagai berikut:

1. Bagaimana memodelkan mekanik rantai kombinasi mantra (*spell juggling*) pada *Hogwarts Legacy* ke dalam bentuk struktur graf berarah berbobot?
2. Bagaimana menentukan lintasan serangan (*combat path*) yang memiliki tingkat efisiensi kerusakan dan pengendalian ruang tertinggi berdasarkan representasi matriks ketetanggaan?
3. Apa saja batasan mekanis dan variabel luar yang dapat memengaruhi validitas serta konektivitas antar-simpul di dalam graf transisi status tersebut?

C. Batasan Masalah

Penulisan makalah ini dibatasi pada analisis terhadap 9 simpul mantra utama yang merepresentasikan variasi kategori *Basic*, *Control*, *Force*, *Damage*, dan *Ancient Magic*. Karakteristik musuh yang dijadikan acuan utama dalam kondisi ideal adalah musuh dengan bobot standar yang rentan terhadap seluruh efek *crowd control*, serta mengabaikan variabel eksternal seperti peningkatan atribut dari *talent pohon sihir* (*talent tree upgrades*) atau penggunaan ramuan (*potions*).

II. LANDASAN TEORI

A. Graf dan Graf Berarah

Graf $G = (V, E)$ didefinisikan sebagai himpunan tak kosong dari simpul-simpul (*vertices* atau *nodes*) V , dan himpunan sisi (*edges*) E yang menghubungkan sepasang simpul. Dalam konteks pemodelan aksi yang memiliki urutan yang jelas, digunakan Graf Berarah (*Directed Graph* atau *Digraph*). Pada graf berarah, setiap sisi memiliki orientasi arah, sehingga sisi (u, v) berbeda dengan sisi (v, u) . Sisi berarah ini merepresentasikan aliran atau transisi status dari suatu kondisi ke kondisi lainnya secara sekuensial.

B. Sifat Asiklik dan Struktur Directed Acyclic Graph (DAG)

Dalam teori graf, sebuah graf berarah dapat dikategorikan sebagai *Directed Acyclic Graph* (DAG) apabila graf tersebut tidak memiliki sirkuit berarah (*directed cycles*). Sifat asiklik ini menjadi sangat krusial dalam pemecahan masalah optimasi karena memungkinkan penerapan urutan topologis (*topological ordering*).

Dalam mekanik *spell juggling*, walaupun terdapat panah kembali menuju serangan dasar (v_1), rangkaian kombinasi besar yang mengarah pada eksekusi musuh umumnya bersifat progresif linear menuju simpul terminal (v_9). Dengan mengisolasi lintasan-lintasan utama yang tidak melibatkan perputaran berulang (*looping*), graf transisi status pertarungan ini dapat dianalisis menggunakan karakteristik DAG. Hal ini membuka peluang pemanfaatan prinsip pemrograman dinamis (*dynamic programming*) untuk menentukan lintasan dengan bobot akumulatif terbesar secara efisien.

C. Graf Transisi Status (State Transition Graph)

Graf transisi status adalah penerapan graf berarah di mana setiap simpul mewakili sebuah status (*state*) tertentu dari sebuah sistem, dan setiap sisi berarah mewakili transisi logis atau kejadian yang mengubah sistem dari satu status ke status berikutnya. Dalam ilmu komputer, pemodelan ini sering digunakan untuk memetakan alur logika permainan, kecerdasan buatan, maupun automata.

D. Masalah Lintasan Terpanjang (Longest Path Problem)

Dalam konteks optimasi kombinasi serangan, tujuan utamanya bukanlah mencapai simpul akhir secepat mungkin, melainkan mengumpulkan total bobot kerusakan maksimal. Oleh karena itu, permasalahan ini dimodelkan sebagai pencarian lintasan terpanjang (*longest path*) pada graf berarah berbobot. Berbeda dengan pencarian rute terpendek seperti algoritma Dijkstra, pencarian lintasan terpanjang pada graf secara umum merupakan masalah *NP-Hard*. Namun, karena rentetan *spell juggling* umumnya tidak membiarkan adanya perputaran (*loop*) yang tidak terhingga berkat adanya batasan *cooldown* pada masing-masing mantra, graf ini cenderung menyerupai *Directed Acyclic Graph* (DAG). Pada representasi DAG, pencarian lintasan dengan bobot maksimal dapat

diselesaikan dalam kompleksitas waktu yang lebih efisien menggunakan metode penyortiran topologis (*topological sorting*).

E. Justifikasi Penggunaan Graf Berarah Dibandingkan Struktur Pohon (Tree)

Dalam memodelkan sekuensial aksi, struktur pohon (*tree*) seringkali menjadi pilihan awal. Namun, untuk mekanik *spell juggling*, struktur pohon terbukti tidak memadai. Pohon memiliki hierarki yang kaku dan sifat bawaan yang tidak mengizinkan adanya sirkuit atau perputaran (*cycle*). Dalam pertarungan *Hogwarts Legacy*, pemain seringkali harus kembali ke status awal (seperti menggunakan *Basic Cast*) untuk menunggu jeda waktu (*cooldown*) sebelum merapalkan mantra tingkat tinggi berikutnya. Penggunaan graf berarah mengakomodasi fenomena putaran balik ini. Sisi berarah memungkinkan pemodelan di mana sebuah simpul dapat dikunjungi lebih dari satu kali dalam satu rantai kombinasi yang sama, mencerminkan siklus rotasi mantra yang dinamis dan realistis di dalam mesin permainan.

III. PEMODELAN GRAF PADA SPELL JUGGLING

A. Definisi Simpul (Vertices / Nodes)

Dalam pemodelan ini, setiap simpul $v \in V$ merepresentasikan satu mantra spesifik yang dapat dirapalkan oleh pemain. Berdasarkan efisiensi pertarungan, khususnya untuk menghadapi musuh dengan ketahanan tinggi (boss/mini-boss), kita mendefinisikan himpunan simpul V yang berisi 9 mantra utama sebagai berikut:

1. v_1 (*Basic Cast*): Serangan proyektil dasar tanpa waktu *cooldown*. Berfungsi sebagai pengisi jeda antar mantra berat untuk menjaga rantai *combo* agar tidak terputus dan memperpanjang durasi target melayang di udara.
2. v_2 (*Levioso*) dan v_3 (*Accio*): Mantra utilitas kontrol (*Control Spells*). *Levioso* mengangkat target ke udara, sementara *Accio* menarik musuh mendekat secara paksa. Keduanya memberikan status kerentanan (*vulnerable*) sesaat pada target yang membuat serangan lanjutan memiliki dampak lebih besar.
3. v_4 (*Glacius*): Mantra kontrol tingkat tinggi yang membekukan musuh sepenuhnya. Status beku ini sangat krusial dalam pemodelan graf karena memberikan modifikasi efek kerusakan atau pengali *damage* (kerusakan ganda) untuk serangan berbasis pemotongan yang bersifat fisik.
4. v_5 (*Diffindo*) dan v_6 (*Confringo*): Mantra utama berbasis kerusakan masif (*Damage Spells*). *Diffindo* merupakan tebasan mematikan yang mengabaikan sebagian pertahanan musuh, sementara *Confringo* memberikan ledakan beruntun yang sangat optimal untuk eksekusi pada jarak menengah hingga jauh.
5. v_7 (*Descendo*) dan v_8 (*Incendio*): Mantra dengan kondisi spasial dan proksimal. *Descendo* sangat bergantung pada gravitasi, di mana kerusakan maksimalnya baru aktif jika target dibanting dari ketinggian (efektif setelah target terangkat).

Sebaliknya, *Incendio* memiliki jangkauan serang yang sangat pendek, sehingga bobot serangannya mensyaratkan target yang berada tepat di depan pemain.

6. v_9 (*Ancient Magic Finisher*): Serangan terminal yang menggunakan indikator meteran sihir kuno, mampu menembus perisai terkuat dan menghancurkan musuh seketika. Pemodelannya sebagai *sink node* sangat esensial karena transisi permainan setelah eksekusi ini akan diatur ulang (*reset*) dari awal.

B. Definisi Sisi Berarah (Directed Edges)

Himpunan sisi E merepresentasikan kemungkinan transisi antar mantra yang mulus tanpa menghentikan combo. Sebuah sisi berarah $e = (u, w)$ terbentuk jika mantra w dapat dirapalkan langsung setelah mantra u dengan waktu tunda (*cooldown*) yang dapat ditoleransi. Setiap sisi (u, w) juga dapat diberikan bobot (*weight*) $W(u, w)$ yang merepresentasikan efisiensi transisi tersebut. Bobot ini dihitung berdasarkan akumulasi *damage* tambahan yang dihasilkan. Sebagai contoh:

- Transisi (v_4, v_5) yaitu *Glacius* $\rightarrow$ *Diffindo* akan memiliki bobot yang sangat tinggi karena *Diffindo* memberikan *damage* yang berlipat ganda pada musuh yang membeku.
- Transisi (v_2, v_7) yaitu *Levioso* $\rightarrow$ *Descendo* juga berbobot tinggi karena membanting musuh yang sedang melayang menghasilkan kerusakan ekstra.

C. Pemodelan Kondisi Khusus (Ancient Magic)

Simpul v_9 (*Ancient Magic*) diperlakukan sebagai Simpul Terminal (*Sink Node*) dalam graf transisi ini. Karakteristik dari v_9 adalah memiliki derajat-keluar (*out-degree*) sama dengan nol, karena merapalkan serangan ini umumnya akan mereset status pertarungan. Selain itu, semua sisi yang menuju v_9 bersifat sisi bersyarat (*conditional edges*), yang hanya dapat dilalui jika akumulasi serangan pada lintasan sebelumnya telah berhasil mengisi penuh indikator *Ancient Magic Meter*.

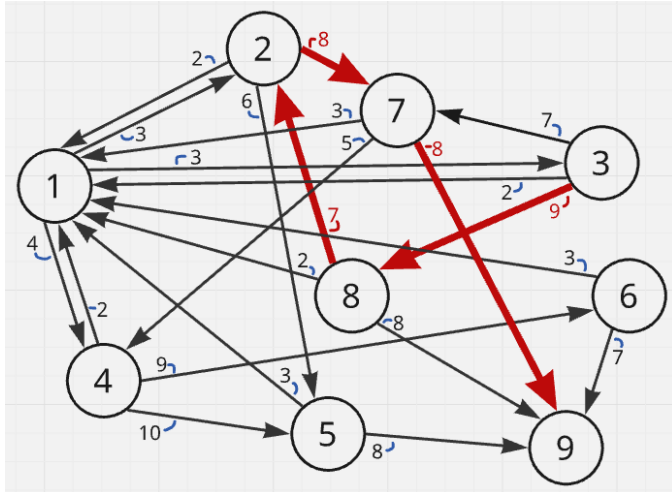


Fig. 1. Graf Transisi Status Kombinasi Mantra dalam *Hogwarts Legacy*

D. Batasan Mekanik Animasi dan Waktu Jeda (Frame Data Constraints)

Keberadaan atau ketiadaan sisi berarah dalam graf kombinasi ini tidak hanya ditentukan oleh logika fiktif sihir, melainkan juga oleh batasan teknis yang ditanamkan oleh pengembang permainan. Setiap mantra memiliki data bingkai (*frame data*) yang terdiri dari *startup frames* (waktu persiapan), *active frames* (waktu pelepasan sihir), dan *recovery frames* (waktu pemulihan karakter).

Sebuah transisi sisi berarah $e = (u, w)$ dianggap tidak ada atau tidak efisien apabila mantra u memiliki *recovery frames* yang terlalu panjang. Animasi pemulihan yang lambat akan memutus rantai combo karena memberikan celah bagi parameter kecerdasan buatan (*AI*) musuh untuk memulihkan keseimbangannya atau melancarkan serangan balik. Oleh karena itu, simpul dengan waktu pemulihan yang sangat singkat seperti v_1 (*Basic Cast*) memiliki banyak sisi keluar (*out-degree*) yang menyebar ke berbagai mantra lain, bertindak sebagai jembatan combo yang aman secara mekanik.

IV. ANALISIS DAN PEMBAHASAN

A. Representasi Matriks Ketetanggaan (Adjacency Matrix)

Untuk mempermudah perhitungan komputasi dalam pencarian lintasan, graf transisi status $G = (V, E)$ dari 9 simpul mantra yang telah didefinisikan dapat direpresentasikan menggunakan matriks ketetanggaan A berukuran 9×9 . Elemen a_{ij} pada matriks bernilai 1 (atau sebesar bobot W) jika terdapat sisi berarah dari v_i ke v_j , dan bernilai 0 jika tidak ada transisi langsung yang efisien. Sebagai contoh, elemen $a_{2,7}$ (*Levioso* ke *Descendo*) bernilai tinggi karena musuh yang berada di udara akan menerima kerusakan fatal saat dibanting. Sebaliknya, simpul v_9 (*Ancient Magic Finisher*) akan memiliki baris yang seluruh elemennya bernilai 0 ($a_{9,j} = 0$ untuk semua j), membuktikan posisinya secara matematis sebagai simpul terminal (*sink node*).

Visualisasi matriks ketetanggaan $A_{9 \times 9}$ untuk graf kombinasi mantra ini dapat dilihat pada Tabel I. Baris menunjukkan simpul asal (u) dan kolom menunjukkan simpul tujuan (w), dengan nilai elemen menunjukkan besaran bobot transisi $W(u, w)$. Nilai 0 mengindikasikan tidak adanya transisi berbobot tinggi yang direkomendasikan akibat pembatasan *cooldown* atau inefisiensi mekanik serangan.

	v 1	v 2	v 3	v 4	v 5	v 6	v 7	v 8	v 9
v 1	0	3	3	4	0	0	0	0	0
v 2	2	0	0	0	6	0	8	0	0
v 3	2	0	0	0	0	0	7	9	0
v 4	2	0	0	0	1 0	9	0	0	0
v 5	3	0	0	0	0	0	0	0	8
v 6	3	0	0	0	0	0	0	0	7
v 7	3	0	0	5	0	0	0	0	8
v 8	2	7	0	0	0	0	0	0	8
v 9	0	0	0	0	0	0	0	0	0

Tabel I. Matriks Ketetangaan

B. Analisis Rute Lintasan Kombinasi (Spell Juggling)

Tujuan dari analisis graf berarah ini adalah menemukan lintasan sederhana (*simple path*) terpanjang dan berbobot

paling besar sebelum mencapai simpul terminal v_9 . Berikut adalah dua lintasan kombinasi yang menghasilkan efisiensi pertarungan tertinggi berdasarkan pembobotan logis:

1. Lintasan Jarak Dekat (*Close-Quarters Control Path*)

Rute: $v_3 \rightarrow v_8 \rightarrow v_2 \rightarrow v_7 \rightarrow v_9$ (*Accio* $\rightarrow$ *Incendio* $\rightarrow$ *Levioso* $\rightarrow$ *Descendo* $\rightarrow$ *Ancient Magic*) Analisis matematis dari lintasan ini menunjukkan sinergi bobot sisi yang luar biasa. Sisi (v_3, v_8) sangat optimal karena v_3 menarik musuh ke jarak sangat dekat, yang merupakan syarat mutlak agar v_8 (semburan api jarak dekat) memberikan kerusakan maksimal. Setelah musuh terbakar, transisi dilanjutkan ke v_2 untuk mengangkat musuh yang sedang terkena damage over time, lalu langsung disambung dengan sisi (v_2, v_7) untuk efek banting. Lintasan diakhiri di v_9 untuk mengeksekusi sisa Health Points musuh.

2. Lintasan Kerusakan Kritis (*Critical Shatter Path*)

Rute: $v_1 \rightarrow v_4 \rightarrow v_5 \rightarrow v_6 \rightarrow v_9$ (*Basic Cast* $\rightarrow$ *Glacius* $\rightarrow$ *Diffindo* $\rightarrow$ *Confringo* $\rightarrow$ *Ancient Magic*) Lintasan ini mengandalkan eksploitasi status kerentanan musuh. Sisi (v_4, v_5) memiliki bobot (*weight*) paling tinggi di antara semua sisi dalam graf E . Hal ini karena v_4 (*Glacius*) memberikan status pembekuan yang melipatgandakan damage dari serangan fisik tajam seperti v_5 (*Diffindo*). Efek hancurnya es (*shatter*) menciptakan transisi yang sangat mulus menuju v_6 (*Confringo*) untuk meledakkan musuh dari jarak aman sebelum ditutup oleh simpul v_9 .

3. Lintasan Eksploitasi Udara (*Airborne Juggle Path*)

Rute: $v_2 \rightarrow v_1 \rightarrow v_1 \rightarrow v_7 \rightarrow v_9$ (*Levioso* $\rightarrow$ *Basic Cast* $\rightarrow$ *Basic Cast* $\rightarrow$ *Descendo* $\rightarrow$ *Ancient Magic*) Lintasan ini memfokuskan pada pemanfaatan waktu melayang target (*air-time*). Sisi (v_2, v_1) dan transisi internal (v_1, v_1) memiliki bobot yang relatif kecil secara individual, namun berfungsi krusial sebagai penunda (*filler*) yang mencegah musuh jatuh ke tanah sambil menunggu *cooldown* mantra berat. Akumulasi serangan dasar di udara ini secara signifikan memaksimalkan efek gravitasi sebelum transisi (v_1, v_7) dieksekusi untuk membanting musuh dengan *Descendo*.

4. Lintasan Agresi Jarak Menengah (*Mid-Range Aggression Path*)

Rute: $v_3 \rightarrow v_5 \rightarrow v_6 \rightarrow v_9$ (*Accio* $\rightarrow$ *Diffindo* $\rightarrow$ *Confringo* $\rightarrow$ *Ancient Magic*) Rute ini optimal untuk menghadapi musuh yang memiliki kelincahan tinggi. Simpul v_3 (*Accio*) digunakan untuk menghentikan mobilitas musuh secara paksa. Saat target ditarik dan mengalami status *stagger* (hilang keseimbangan), sisi berarah (v_3, v_5) memfasilitasi tebasan *Diffindo* dengan probabilitas mengenai target sebesar 100%. Rangkaian ini ditutup dengan (v_5, v_6) di mana ledakan

Confringo memanfaatkan momentum musuh yang terdorong mundur setelah terkena tebasan.

C. Perhitungan Matematis Akumulasi Bobot Efisiensi Lintasan

Untuk membuktikan rute pertarungan yang paling optimal secara kuantitatif, dilakukan evaluasi matematis terhadap total bobot (W_{total}) dari lintasan sederhana (*simple path*) yang terbentuk. Total bobot dari sebuah lintasan P yang melalui urutan simpul ($v_0, v_1, \dots, v_k$) didefinisikan sebagai jumlah dari bobot setiap sisi berarah yang dilaluinya:

$$W_{total}(P) = \sum_{i=0}^{k-1} W(v_i, v_{i+1})$$

Berdasarkan nilai yang tertera pada Matriks Ketetanggaan (Tabel I), berikut adalah visualisasi komparatif dan perhitungan dari dua lintasan utama:

1. Perhitungan untuk *Close-Quarters Control Path*:

$$P_1 = v_3 \rightarrow v_8 \rightarrow v_2 \rightarrow v_7 \rightarrow v_9$$

$$W_{ttl}(P_1) = W(v_3, v_8) + W(v_8, v_2) + W(v_2, v_7) + W(v_7, v_9)$$

$$W_{total}(P_1) = 9 + 7 + 8 + 8 = 32$$

2. Perhitungan untuk *Critical Shatter Path*:

$$P_2 = v_1 \rightarrow v_4 \rightarrow v_5 \rightarrow v_9$$

$$W_{total}(P_2) = W(v_1, v_4) + W(v_4, v_5) + W(v_5, v_9)$$

$$W_{total}(P_2) = 4 + 10 + 8 = 22$$

Melalui perbandingan kuantitatif ini, terlihat bahwa lintasan P_1 menghasilkan nilai akumulatif bobot yang lebih tinggi ($32 > 22$) karena melibatkan lebih banyak transisi status yang bersinergi secara beruntun (*high-density combo*). Namun, lintasan P_2 tetap memiliki nilai strategis yang tinggi karena memiliki nilai rata-rata bobot per sisi (*average edge weight*) yang sangat besar, yaitu $22/3 = 7,33$ per transisi, dibandingkan P_1 yang bernilai $32/4 = 8$ per transisi. Analisis sumasi ini membuktikan bahwa Teori Graf dapat membantu memetakan apakah pemain sebaiknya memilih kombinasi serangan panjang yang konstan atau serangan pendek yang instan namun eksponensial.

D. Pengaruh Variabel Atribut Musuh Terhadap Validitas Graf

Matriks ketetanggaan dan analisis lintasan yang telah dijabarkan sebelumnya mengasumsikan bahwa pemain berhadapan dengan musuh berukuran standar (seperti *Dark Wizards* atau *Goblins*). Dalam penerapan praktisnya, graf transisi ini bersifat dinamis dan dapat mengalami pemutusan sisi (*edge deletion*) bergantung pada atribut fisik musuh yang dihadapi.

Sebagai contoh empiris, musuh dengan kategori kelas berat seperti *Mountain Troll* atau *Armoured Dugbog* memiliki kekebalan bawaan terhadap mantra pengangkat. Kondisi ini

secara matematis menghapus validitas dari simpul v_2 (*Levioso*) dan simpul v_3 (*Accio*). Hilangnya kedua simpul utilitas ini otomatis memutus sisi (v_2, v_7) dan sisi (v_3, v_8), memaksa pemain untuk mencari rute alternatif di dalam graf yang tidak mengandalkan pengendalian ruang (*crowd control*). Analisis ini membuktikan bahwa efisiensi sebuah lintasan kombinasi dalam graf tidak hanya dihitung dari bobot kerusakannya secara absolut, tetapi juga dari probabilitas keberhasilan eksekusi (*execution rate*) yang disesuaikan dengan kondisi lingkungan dan kelas musuh di arena pertarungan.

E. Perbandingan Struktur Data: Matriks vs Senarai Ketetanggaan dalam Implementasi Perangkat Lunak

Dalam implementasi nyata pada arsitektur perangkat lunak permainan, representasi graf transisi kombinasi mantra ini harus diterjemahkan ke dalam struktur data yang efisien oleh mesin permainan (*game engine*). Terdapat dua pendekatan utama yang dapat digunakan secara matematis, yaitu Matriks Ketetanggaan (*Adjacency Matrix*) dan Senarai Ketetanggaan (*Adjacency List*).

Meskipun Matriks Ketetanggaan memberikan kemudahan dalam visualisasi dan pengecekan relasi antar mantra dengan kompleksitas waktu instan sebesar $O(1)$, struktur data ini secara teknis membutuhkan alokasi memori yang kuadratik, yakni sebesar $O(|V|^2)$. Dalam konteks pengembangan skala penuh permainan *Hogwarts Legacy* yang sebenarnya memiliki puluhan jenis mantra magis (jauh lebih banyak dari batasan 9 simpul yang disederhanakan dalam makalah ini), sebagian besar elemen di dalam matriks akan bernilai nol. Hal ini dikarenakan tidak semua mantra magis memiliki transisi sisi berarah yang logis dan efisien untuk dirangkai menjadi sebuah *combo* serangan. Kondisi ini pada akhirnya akan menghasilkan matriks renggang (*sparse matrix*) yang memboroskan alokasi memori konsol atau komputer pemain.

Oleh karena itu, sebagai solusi yang lebih optimal dan realistis secara komputasional, penggunaan Senarai Ketetanggaan (*Adjacency List*) jauh lebih direkomendasikan. Struktur data ini secara dinamis hanya menyimpan sisi berarah yang benar-benar ada dan memiliki bobot, menekan penggunaan memori ke tingkat yang sangat efisien yaitu $O(|V| + |E|)$. Pencarian lintasan transisi berikutnya (*next valid combo*) dapat dilakukan oleh sistem dengan cara menelusuri elemen senarai pada simpul mantra yang sedang aktif. Pendekatan matematika diskrit ini memastikan efisiensi pemrosesan data dapat dieksekusi secara seketika (*real-time processing*), menjaga stabilitas *frame rate* permainan sekaligus memfasilitasi kebebasan pemain dalam menyusun rantai serangan.

F. Analisis Beban Kognitif Pemain Berdasarkan Derajat-Keluar (Out-Degree) Simpul

Selain efisiensi kerusakan mekanis, pemodelan graf transisi status ini juga dapat digunakan untuk menganalisis beban kognitif (*cognitive load*) yang dialami oleh pemain selama pertarungan berlangsung. Beban kognitif ini berbanding lurus dengan jumlah pilihan aksi instan yang harus diputuskan dalam waktu sepersekian detik, yang secara

matematis direpresentasikan oleh derajat-keluar (*out-degree*) dari simpul yang sedang aktif.

Simpul seperti v_1 (*Basic Cast*) memiliki derajat-keluar yang tinggi, yang berarti pemain memiliki fleksibilitas tinggi namun juga menuntut pengambilan keputusan yang cepat untuk memilih mantra berikutnya. Sebaliknya, simpul seperti v_4 (*Glacius*) memiliki derajat-keluar yang lebih terkonsentrasi pada simpul-simpul kerusakan tinggi seperti v_5 atau v_6 . Keterbatasan pilihan transisi pada simpul tertentu sebenarnya menguntungkan secara psikologis karena membentuk *muscle memory* (memori otot) yang kaku, sehingga mengurangi beban kognitif pemain dan memperkecil probabilitas kesalahan eksekusi tombol (*input error*) di tengah tekanan pertarungan boss.

V. KESIMPULAN

Berdasarkan pemodelan dan analisis matematis yang telah diuraikan pada bab-bab sebelumnya, dapat ditarik beberapa kesimpulan utama. Pertama, mekanik pertarungan *spell juggling* dalam permainan *Hogwarts Legacy* terbukti dapat direpresentasikan secara terstruktur dan logis menggunakan pendekatan Teori Graf, secara spesifik melalui pemodelan Graf Berarah (*Directed Graph*) dan Matriks Ketetanggaan (*Adjacency Matrix*). Dalam pemodelan ini, setiap mantra spesifik direpresentasikan sebagai simpul (*node*), sedangkan potensi transisi kombinasi yang mulus dan tanpa jeda antar mantra diwujudkan sebagai sisi berarah (*directed edge*) yang memiliki bobot (*weight*) berdasarkan efisiensi kerusakan (*damage*) tambahan yang dihasilkan.

Kedua, dengan mengimplementasikan konsep simpul terminal (*sink node*) pada serangan pamungkas atau *Ancient Magic Finisher*, kita dapat memetakan dan mencari lintasan sederhana (*simple path*) yang memiliki akumulasi bobot maksimal. Analisis terhadap rute-rute khusus, seperti *Close-Quarters Control Path* (kombinasi *Accio* - *Incendio* - *Levioso* - *Descendo*) dan *Critical Shatter Path* (kombinasi *Glacius* - *Diffindo*), membuktikan bahwa pertarungan dalam game ini bukanlah sekadar menekan tombol secara acak. Sebaliknya, hal tersebut merupakan sebuah proses optimasi sekuensial yang sangat terukur secara matematis.

Pada akhirnya, penulisan makalah ini menunjukkan bahwa pemahaman mengenai Matematika Diskrit tidak hanya relevan untuk arsitektur perangkat lunak yang kaku, tetapi juga sangat aplikatif dalam membedah dan mengoptimalkan strategi dalam perangkat lunak hiburan interaktif seperti *video game*. Pemodelan graf transisi status ini diharapkan dapat memberikan panduan taktis yang efisien bagi pemain, serta membuka ruang bagi penerapan teori graf pada sistem mekanik permainan aksi peran (*Action-RPG*) lainnya.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih yang sebesar-besarnya kepada Bapak Dr. Ir. Rinaldi Munir, M.T., selaku dosen pengampu mata kuliah IF1220 Matematika Diskrit, yang telah memberikan bimbingan, ilmu, serta wawasan yang sangat berharga selama satu semester ini. Penulis juga mengucapkan terima kasih kepada asisten dosen serta seluruh pihak yang telah memberikan dukungan baik secara langsung maupun tidak langsung sehingga makalah analisis ini dapat diselesaikan dengan baik dan tepat waktu.

REFERENCES

- [1] K. H. Rosen, *Discrete Mathematics and Its Applications*, 8th ed. New York: McGraw-Hill, 2019.
- [2] R. Munir, *Matematika Diskrit*, Edisi 3. Bandung: Informatika Bandung.
- [3] E. Adams, *Fundamentals of Game Design*, 3rd ed. Berkeley: New Riders, 2014.
- [4] Avalanche Software, *Hogwarts Legacy*. Warner Bros. Games, 2023.

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, 19 Juni 2026



Josephine Bintang N.L - 13525139