

Analisis Penerapan Konsep Matematika Diskrit pada Struktur Grid dan Logika Perilaku Musuh dalam Permainan "Occidental Heroes"

Aditya Rasyid - 13525039

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jalan Ganesha 10 Bandung 40132, Indonesia

E-mail: 13525039@mahasiswa.itb.ac.id

E-mail: adityarasyid797@gmail.com

Abstract—Penelitian ini menganalisis penerapan konsep Matematika Diskrit dalam mekanisme game RPG taktis Occidental Heroes. Studi ini akan membedah penggunaan Teori Graf pada pemodelan peta komputasi, Aljabar Boolean dan Pohon Keputusan untuk penalaran Kecerdasan Buatan (AI), serta Notasi Big-O untuk analisis kompleksitas algoritma navigasi. Hasilnya membuktikan bahwa prinsip matematika diskrit berperan krusial dalam membangun sistem permainan waktu-nyata (real-time) yang seimbang, logis, deterministik, dan efisien secara komputasi.

Keywords—Occidental Heroes; Teori Graf; Pohon Keputusan; Kompleksitas Algoritma; Notasi Big-O.

I. PENDAHULUAN

Industri pengembangan permainan digital (video games) telah berkembang pesat dari sekadar media hiburan menjadi sistem simulasi yang sangat kompleks. Salah satu genre yang menuntut akurasi logika tingkat tinggi adalah Tactical Role-Playing Game (RPG) berbasis giliran (turn-based). Di balik visualisasi grafis dan interaksi antarmuka yang disajikan kepada pemain, terdapat arsitektur kode program yang digerakkan oleh aturan-aturan komputasional yang kaku, deterministik, dan harus berjalan secara efisien. Fondasi utama yang menjembatani aturan dunia permainan ke dalam sistem komputasi ini adalah Matematika Diskrit.

Matematika Diskrit menyediakan perangkat teoretis untuk memodelkan objek-objek yang bersifat terpisah (diskrit) dan saling berhubungan, seperti yang ditemukan pada elemen-elemen logika video game. Salah satu game yang merepresentasikan penerapan konsep ini secara intensif adalah Occidental Heroes, sebuah game RPG taktis mobile karya Kloppe Media. Game ini memiliki karakteristik unik, seperti sistem pergerakan karakter di atas papan ubin selang-seling (grid-based tactical movement), kecerdasan buatan (AI) musuh yang deterministik, sistem

jangkauan serangan taktis berdasarkan kelas unit, hingga perhitungan alokasi memori yang efisien agar dapat dieksekusi secara lancar pada perangkat dengan spesifikasi terbatas.

Seluruh mekanik utama dalam Occidental Heroes dapat dienkapsulasi ke dalam struktur matematika dasar. Sebagai contoh, peta pertempuran dan batas navigasi pergerakan karakter dapat dimodelkan secara matematis menggunakan Teori Graf berbasis Odd-R Hexagonal Grid. Sementara itu, penalaran taktis dan evaluasi kondisi serang yang dilakukan oleh AI musuh diturunkan dari operasi Aljabar Boolean dan struktur data Pohon Keputusan (Decision Tree). Lebih jauh lagi, untuk menjamin performa permainan tetap optimal saat memproses seluruh aturan tersebut secara waktu-nyata (real-time), analisis efisiensi sistem perlu diukur menggunakan landasan kompleksitas algoritma melalui Notasi Big-O. Melalui sudut pandang ini, elemen permainan tidak lagi dipandang sebagai entitas visual acak, melainkan sebagai fungsi-fungsi matematika yang saling berinteraksi secara logis dan terstruktur.

II. LANDASAN TEORI

2.1 Teori Graf dan Aplikasinya

Graf digunakan untuk memodelkan hubungan diskrit antara objek-objek.

Graf G didefinisikan sebagai $G = (V, E)$, yang dalam hal ini:

V = himpunan tidak-kosong dari simpul-simpul (*vertices*)
 $= \{v_1, v_2, \dots, v_n\}$

→ Himpunan V tidak boleh kosong, artinya graf tidak boleh tidak mengandung simpul

E = himpunan sisi (*edges*) yang menghubungkan sepasang simpul
 $= \{e_1, e_2, \dots, e_n\}$

→ Himpunan E boleh kosong, artinya graf boleh tidak mengandung sisi satu buah pun.

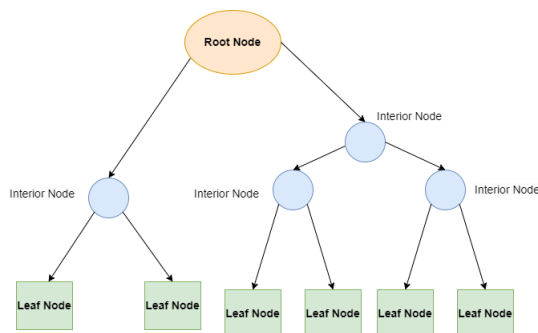
Gambar 2.1 Definisi Graf

(Sumber: <https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Teori-Graf-2020-Bagian1.pdf>)

Dalam ilmu komputer, graf sering digunakan untuk merepresentasikan jaringan atau peta. Pada struktur grid dua dimensi (seperti papan catur atau peta taktis), setiap kotak dapat direpresentasikan sebagai vertex, dan batas antar kotak yang bersebelahan direpresentasikan sebagai edge. Algoritma pencarian rute terpendek, seperti Dijkstra atau A* (A-Star), beroperasi di atas struktur graf ini untuk menemukan jalur dengan bobot atau biaya (cost) terkecil dari titik asal ke titik tujuan.

2.2 Pohon Keputusan (Decision Tree)

Pohon (Tree) adalah bentuk khusus dari graf terhubung yang tidak mengandung sirkuit atau siklus. Pohon Keputusan (Decision Tree) adalah model struktur pohon yang digunakan dalam kecerdasan buatan (AI) untuk memetakan langkah-langkah pengambilan keputusan berdasarkan kondisi tertentu.



Gambar 2.2 Struktur Pohon

(Sumber:

https://www.researchgate.net/figure/Structure-of-Decision-Tree-Nodes-Root-Interior-and-Leaf_fig2_382914996)

Struktur ini terdiri dari:

1. Simpul Akar (Root Node): Titik awal proses evaluasi kondisi.
2. Simpul Internal (Interior Node): Titik yang merepresentasikan pengujian terhadap suatu atribut atau variabel.
3. Simpul Daun (Leaf Node): Titik akhir yang merepresentasikan keputusan atau tindakan yang diambil.

Setiap cabang yang keluar dari simpul internal merepresentasikan hasil dari pengujian kondisi tersebut (misalnya: Ya/Tidak atau Benar/Salah).

2.3 Logika Matematika dan Aljabar Boolean

Aljabar Boolean adalah sistem aljabar yang beroperasi pada variabel-variabel yang hanya memiliki dua nilai kebenaran: True (1) atau False (0). Operasi dasar dalam Aljabar Boolean meliputi:

- Konjungsi (AND): Dinotasikan dengan " $\wedge$ ", menghasilkan nilai 1 hanya jika kedua operan bernilai 1.
- Disjungsi (OR): Dinotasikan dengan " $\vee$ ", menghasilkan nilai 1 jika salah satu atau kedua operan bernilai 1.

- Negasi (NOT): Dinotasikan dengan " $\neg$ " atau tanda aksen ($'$), membalikkan nilai kebenaran operan.

Dalam pemrograman game, logika ini menjadi fondasi dalam penyusunan instruksi kondisional (if-then-else) untuk memeriksa status permainan sebelum mengeksekusi fungsi tertentu.

2.4 Kompleksitas Algoritma dan Notasi Big-O

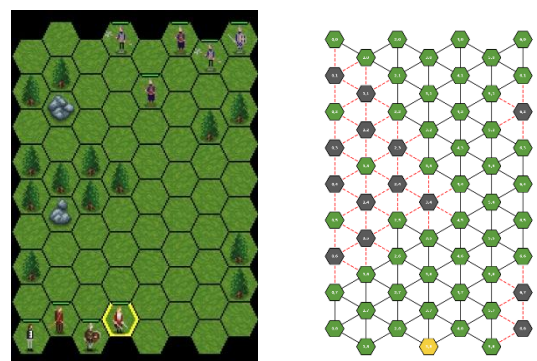
Dalam ilmu komputer dan matematika diskrit, kompleksitas algoritma digunakan untuk mengukur seberapa efisien suatu algoritma ketika ukuran data masukan (input size) berkembang menjadi semakin besar. Analisis ini dibagi menjadi dua aspek utama, yaitu:

- Kompleksitas Waktu (Time Complexity): Mengukur jumlah operasi komputasi dasar (seperti operasi aritmatika, perbandingan, atau penugasan variabel) yang dieksekusi oleh program sebagai fungsi dari ukuran masukan. Fokusnya bukan pada detik atau menit, melainkan pada pertumbuhan jumlah langkah eksekusi.
- Kompleksitas Ruang (Space Complexity): Mengukur efisiensi penggunaan memori RAM yang dialokasikan oleh algoritma selama runtime, termasuk ruang untuk variabel statis, alokasi dinamis, dan tumpukan fungsi (function call stack).

Untuk mengekspresikan kompleksitas tersebut secara standar tanpa bergantung pada spesifikasi perangkat keras komputer, digunakan Notasi Big-O. Notasi Big-O menggambarkan batas atas (upper bound) dari laju pertumbuhan waktu eksekusi sebuah algoritma pada skenario terburuk (worst-case scenario). Secara formal, notasi ini mengelompokkan algoritma ke dalam kelas efisiensi tertentu—seperti konstan $O(1)$ atau logaritmik $O(\log n)$ —sehingga performa sistem dapat diprediksi secara akademis.

III. ANALISIS DAN PEMBAHASAN

3.1 Pemodelan Medan Tempur dan Navigasi Karakter Menggunakan Teori Graf Berbasis Even-R Hexagonal Grid



Gambar 3.1 Medan Tempur dan Representasi Graf-nya
(Sumber: Arsip Penulis)

Berdasarkan analisis visual terhadap karakteristik permainan Occidental Heroes pada gambar peta dan model komputasinya, medan pertempuran dalam game ini mengadopsi sistem ubin segi enam dengan orientasi mendatar (Flat-Topped Hexagonal Grid). Dalam struktur matematika diskrit, seluruh papan permainan ini diabstraksikan menjadi model graf berarah dan berbobot $G = (V, E)$ dengan sistem koordinat bergeser horizontal (Even-R staggered row).

Model komputasi graf pada permainan ini dirancang dengan parameter berikut:

- Himpunan Titik (V): Setiap ubin segi enam di arena mewakili satu vertex V. Koordinat ubin diidentifikasi menggunakan pasangan matriks diskrit (kolom, baris) dengan ukuran tetap, yaitu lebar 4 kolom (0 hingga 3) dan panjang 16 baris vertikal ke bawah (0 hingga 15).
- Himpunan Sisi (E) dan Relasi Ketetanggaan: Tidak seperti grid persegi konvensional yang hanya memiliki 4 arah pergerakan, ubin berbentuk heksagonal ini dikelilingi langsung oleh 6 ubin tetangga. Hal ini berarti derajat (degree) maksimal dari setiap titik internal yang bebas adalah 6 ($\deg(v) = 6$).

Karena polanya berselang-seling horizontal, rumus matematika untuk menentukan hubungan sisi (edge) tetangga dibedakan berdasarkan nomor barisnya:

- Untuk Baris Genap (0, 2, 4, dst):
 $Tetangga(c, r) = \{(c+1, r), (c-1, r), (c, r+1), (c, r-1), (c-1, r+1), (c-1, r-1)\}$
- Untuk Baris Ganjil (1, 3, 5, dst):
 $Tetangga(c, r) = \{(c+1, r), (c-1, r), (c, r+1), (c, r-1), (c+1, r+1), (c+1, r-1)\}$

Fungsi Hambatan dan Pemutusan Sisi: Kehadiran objek rintangan lingkungan seperti batu besar dan pohon di dalam peta berfungsi sebagai pemutus interkoneksi graf. Akibatnya, semua garis sisi yang terhubung langsung menuju titik rintangan tersebut otomatis diputus (diubah status logikanya menjadi False) dan digambarkan sebagai garis merah putus-putus pada visualisasi graf komputer.



Archer



Man-At-Arms



Lancer

**Gambar 3.2 Tipe Serangan
Tiap Peran**
(Sumber: Arsip Penulis)

Kecerdasan Buatan (AI) yang menggerakkan pasukan musuh ataupun pemain bekerja secara deterministik dengan mengevaluasi kondisi pertempuran menggunakan Aljabar Boolean, lalu mengeksekusinya melalui struktur Pohon Keputusan (Decision Tree) tanpa siklus.

Setiap kali giliran (turn) dimulai, mesin permainan tidak langsung menggerakkan karakter secara acak, melainkan melakukan pengecekan status logika. Di dalam game ini, aturan logika dibedakan berdasarkan tipe serangan unit, yaitu unit jarak jauh (ranged) dan unit jarak dekat (melee).

3.2 Implementasi Aljabar Boolean dan Pohon Keputusan pada AI Musuh

Pasukan musuh maupun pemain memiliki 3 role (peran) utama, yaitu Archer, MAA(Man-At-Arms), dan Lancer. Berikut Adalah dokumentasi gameplay serangan dari tiap peran:

A. Aturan Unit Jarak, Archer

Pemanah memiliki jarak serangan sebesar 3 grid tetapi tidak dapat menyerang lawan yang bersebelahan (berjarak 1 grid) darinya. Untuk menentukan apakah unit pemanah musuh (Archer) dapat melancarkan serangan panah, program mengeksekusi ekspresi logika Boolean berikut:

BisaMenembak = (Jarak $\leq$ 3) AND (Jarak $>$ 1) AND NOT(TerhalangRintangan)

B. Aturan Unit Jarak Dekat, Man-at-Arms (MAA)

Unit Man-at-Arms merupakan tipe petarung infanteri jarak dekat murni. Berdasarkan aturan Teori Graf, MAA hanya bisa mengeksekusi perintah serang jika posisi target berada tepat di grid yang bertetangga langsung (berjarak 1 grid). Ekspresi Boolean untuk MAA didefinisikan sebagai:

BisaMenyerang_MAA = (Jarak == 1)

Jika nilai kebenaran dari variabel Jarak == 1 bernilai 0 (False) karena pemain berada 2 ubin atau lebih darinya, maka total nilai logika BisaMenyerang_MAA menjadi 0 (False), dan AI akan memilih untuk bergerak mendekat terlebih dahulu.

C. Aturan Unit Jarak Dekat, Lancer

Unit Lancer memiliki karakteristik unik berupa kemampuan menusuk (lunge). Secara mekanik matematika diskrit, Lancer mampu menyerang target yang berada sejauh 1 atau 2 blok ubin, dengan syarat arah ubin tersebut harus membentuk satu garis lurus (searah dengan orientasi ubin heksagonal) tanpa ada rintangan di antaranya. Ekspresi Boolean untuk Lancer didefinisikan sebagai:

BisaMenyerang_Lancer = ((Jarak == 1) OR (Jarak == 2 AND JalurLurus)) NOT(TerhalangRintangan)

D. Struktur Pohon Keputusan AI Musuh

Saat giliran musuh tiba, algoritma mengevaluasi langkah taktis melalui simpul-simpul keputusan:



Gambar 3.3 Pohon Keputusan AI
(Sumber: Arsip Penulis)

1. Simpul Akar (Root Node): Memeriksa kondisi kesehatan unit. Apakah HP dirinya < 25%?
 - Cabang True: AI mengambil tindakan defensif/mundur menjauh (Simpul Daun).
 - Cabang False: Sistem melanjutkan evaluasi ke Simpul Internal berikutnya (2).
2. Simpul Internal: Memeriksa jangkauan serangan taktis berdasarkan graf lingkungan. Apakah posisi karakter pemain berada dalam jangkauan serang?
 - Cabang True: Eksekusi perintah menyerang target (Simpul Daun).
 - Cabang False: Eksekusi perintah bergerak mendekati koordinat pemain (Simpul Daun).

3.3 Analisis Kompleksitas Algoritma Navigasi dan Pengambilan Keputusan AI

3.3.1 Kompleksitas Algoritma Pencarian Jalur (Pathfinding)

Sistem navigasi agen cerdas yang bertugas mencari rute terpendek di atas ubin heksagonal berpola Odd-R Hexagonal Grid diimplementasikan menggunakan algoritma pencarian jalur seperti Dijkstra atau A-Star (A*). Secara teoritis, efisiensi eksekusi dari algoritma

ini sangat bergantung pada representasi struktur data graf yang digunakan, yang didefinisikan oleh jumlah total ubin sebagai simpul (Vertex, V) dan jumlah garis penghubung antar-ubin sebagai sisi (Edge, E).

- Kompleksitas Waktu: Dengan menggunakan struktur data Priority Queue (Min-Heap) untuk menyimpan urutan ubin yang akan diperiksa, kompleksitas waktu pencarian rute terpendek adalah $O((V + E) \log V)$. Pada graf ukuran tetap 7x9 seperti Gambar 2.1, nilai $V = 63$, sehingga komputasi dapat diselesaikan dalam hitungan milidetik.
- Kompleksitas Ruang: Memori yang dibutuhkan algoritma adalah $O(V + E)$ untuk menyimpan struktur ketetanggaan ubin di dalam RAM, yang memastikan game tetap ringan dan tidak memicu kebocoran memori (memory leak).

3.3.2 Kompleksitas Pohon Keputusan AI Musuh

Berbeda dengan algoritma pencarian jalur yang melibatkan pencarian meluas (*breadth-first exploration*) di dalam ruang graf, penentuan aksi taktis musuh (seperti keputusan untuk menembak, mendekati target, atau memosisikan diri mundur) mengandalkan struktur Pohon Keputusan (*Decision Tree*). Struktur ini bekerja secara linier, deterministik, dan statis tanpa melibatkan proses pencarian heuristik yang dinamis.

- Kompleksitas Waktu: Karena AI mengevaluasi kondisi dari simpul akar (root node) langsung menuju simpul daun (leaf node) tanpa adanya perulangan (looping atau acyclic), kompleksitas waktunya adalah $O(d)$, di mana d adalah kedalaman maksimum (depth) dari pohon keputusan tersebut. Pada model AI di Bab 3.2, kedalaman pohon hanya berkisar antara 2 hingga 3 tingkat, sehingga kompleksitasnya bernilai konstan atau $O(1)$ (waktu konstan).
- Kompleksitas Ruang: Kompleksitas ruang untuk menyimpan status logika Boolean juga bernilai konstan $O(1)$ karena hanya membutuhkan beberapa variabel register dasar untuk menampung status kebenaran (True/False).

3.4 Implementasi dan Simulasi Program

```

1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main() {
5     int kelasMusuh, xMusuh, yMusuh, xPemainBaru, yPemainBaru, jarak;
6
7     printf("(1-Archer, 2-Lancer, 3-MAA): ");
8     scanf("%d", &kelasMusuh);
9
10    printf("Posisi Musuh (X Y): ");
11    scanf("%d %d", &xMusuh, &yMusuh);
12
13    printf("Posisi Tujuan Anda (X Y): ");
14    scanf("%d %d", &xPemainBaru, &yPemainBaru);
15
16    jarak = abs(xMusuh - xPemainBaru) + abs(yMusuh - yPemainBaru);
17    printf("Jaraknya : %d\n", jarak);
18
19    if (kelasMusuh == 1) {
20        if (jarak == 1) {
21            if (yMusuh < yPemainBaru) yMusuh--;
22            else if (yMusuh > yPemainBaru) yMusuh++;
23            else if (xMusuh < xPemainBaru) xMusuh--;
24            else if (xMusuh > xPemainBaru) xMusuh++;
25        }
26    }
27 }
  
```

Gambar 3.4 Penggalan Kode Program

Untuk membuktikan akurasi model teoritis Pohon Keputusan (Decision Tree) dan Aljabar Boolean yang

telah dibahas pada subbab sebelumnya, dibuat sebuah program simulasi taktis berbasis bahasa pemrograman C. Program ini dirancang untuk memprediksi respons pergerakan atau tindakan serangan AI musuh berdasarkan koordinat posisi dua dimensi (X, Y) setelah pemain menentukan calon langkah selanjutnya.

Perhitungan jarak antara ubin musuh $M(x_1, y_1)$ dan ubin tujuan pemain $P(x_2, y_2)$ di dalam kode ini dihitung menggunakan konsep Jarak Manhattan (Manhattan Distance) yang didefinisikan dengan rumus:

$$\text{Jarak} = |x_1 - x_2| + |y_1 - y_2|$$

Pemilihan metrik Jarak Manhattan didasarkan pada sifat pergerakan diskrit pada grid permainan turn-based, di mana agen cerdas hanya dapat melangkah ke ubin yang bertetangga secara ortogonal tanpa melibatkan interpolasi linier kontinu (Jarak Euclidean).

Informasi jarak mutlak yang dihasilkan kemudian ditransformasikan menjadi fungsi parameter masukan (input parameter) utama bagi fungsi logika bersyarat di dalam program. Melalui pencabangan ini, komputer dapat mengevaluasi secara instan ubin mana yang akan ditempati oleh entitas musuh pada turn berikutnya, sekaligus menentukan transisi status (state transition) musuh, baik dari status siaga (idle) menjadi menyerang (attacking), maupun berpindah posisi (moving).

```
PS C:\Users\Aditya\Downloads\Occidental\output> & .\Program OH.exe'
(1=Archer, 2=Lancer, 3=MAA): 3
Posisi Musuh (X Y): 4 4
Posisi Tujuan Anda (X Y): 5 4
Jaraknya : 1
MAA menyerang dari (4, 4)
PS C:\Users\Aditya\Downloads\Occidental\output> 
PS C:\Users\Aditya\Downloads\Occidental\output> & .\Program OH.exe'
(1=Archer, 2=Lancer, 3=MAA): 2
Posisi Musuh (X Y): 5 8
Posisi Tujuan Anda (X Y): 5 2
Jaraknya : 6
Lancer maju ke (5, 2)
```

Gambar 3.5 Output Program

Berikut adalah link video demo dan penjelasan dari program:

<https://drive.google.com/drive/folders/1syjsW270e3aoZ83EzzHGyHCExbHK219d?usp=sharing>

IV. KESIMPULAN

Secara keseluruhan, penelitian ini membuktikan bahwa prinsip Matematika Diskrit berperan krusial sebagai fondasi utama dalam menjembatani aturan permainan ke dalam sistem komputasi game Occidental Heroes agar berjalan secara logis, deterministik, dan efisien.

Peta ubin heksagonal berukuran 7x9 merupakan implementasi nyata dari Teori Graf, di mana rintangan lingkungan berfungsi memutus hubungan antar-simpul secara dinamis sehingga algoritma pathfinding dapat menghitung rute memutar yang optimal demi menjaga navigasi karakter tetap responsif. Selain itu, perilaku taktis AI musuh berhasil dikendalikan secara terstruktur melalui kombinasi Aljabar Boolean dan Pohon Keputusan (Decision Tree). Pendekatan ini sukses mengeksekusi aturan serang spesifik tiap kelas unit—seperti batasan jarak Archer, serangan dekat Man-at-

Arms, hingga tusukan lurus Lancer—secara instan tanpa membebani performa memori perangkat karena alur keputusannya yang dirancang searah.

Melalui implementasi simulasi berbasis bahasa C, penggunaan Jarak Manhattan terbukti akurat dalam memproyeksikan pergerakan ubin demi ubin pada ruang diskrit secara konsisten dengan kompleksitas waktu konstan $O(1)$. Integrasi struktur data graf dan logika Boolean ini menegaskan bahwa pendekatan matematika diskrit mampu menghasilkan performa permainan berbasis turn-based yang ringan, stabil, dan bebas dari risiko kebocoran memori (memory leak).

V. LAMPIRAN

- Source code yang digunakan dalam program prediksi (Program OH.c):

<https://github.com/zenyx7/Occidental-Heroes-Enemy-Predictions/tree/main>

- Link Video Demo + Penjelasan Makalah:

<https://drive.google.com/drive/folders/1syjsW270e3aoZ83EzzHGyHCExbHK219d?usp=sharing>

VI. REFERENSI

- [1] Munir, Rinaldi. 2020. "Teori Graf (Bagian 1)". <https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Teori-Graf-2020-Bagian1.pdf>, diakses 18 Juni 2026.
- [2] Munir, Rinaldi. 2020. "Aljabar Boolean (Bagian 1)". [https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Aljabar-Boolean-\(2020\)-bagian1.pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Aljabar-Boolean-(2020)-bagian1.pdf), diakses 18 Juni 2026.
- [3] Dice Cast Games. 2017. *Occidental Heroes Official Website*. <https://occidentalheroes.com/>, diakses 17 Juni 2026.
- [4] Munir, Rinaldi. 2020. "Pohon (Bagian 1)". <https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Pohon-2020-Bagian1.pdf>, diakses 18 Juni 2026.
- [5] Patel, Amit. 2014. "Heuristics for Pathfinding on a Grid". Red Blob Games. <https://www.redblobgames.com/pathfinding/a-star/introduction.html>, diakses 19 Juni 2026.

VII. UCAPAN TERIMA KASIH

Alhamdulillah, Puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa karena berkat rahmat dan karunia-Nya, penulis dapat menyelesaikan makalah yang berjudul "Analisis Penerapan Konsep Matematika Diskrit pada Struktur Grid dan Logika Perilaku Musuh dalam Permainan Occidental Heroes" ini dengan baik dan tepat pada waktunya.

Lalu, penulis menyampaikan terima kasih yang sebesar-besarnya kepada dosen pengampu mata kuliah Matematika Diskrit, Bapak Dr. Ir. Rinaldi Munir, M.T., atas bimbingan dan ilmu yang telah diberikan selama semester ini. Tak lupa, ucapan terima kasih penulis tujukan kepada orang tua,

keluarga, serta rekan-rekan yang senantiasa memberikan dukungandan motivasi selama proses penulisan.

Penulis menyadari bahwa makalah ini masih jauh dari kesempurnaan. Akhir kata, penulis berharap semoga karya ini dapat bermanfaat dan menjadi referensi akademik yang berguna bagi pembaca maupun penulis di masa depan.

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



Aditya Rasyid
13525039