Aplikasi Graf dalam Pembuatan Game dengan n-Jumlah Stage

Muhammad Fauzan Azhim - 13522153[1]
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
[1]13522153@std.stei.itb.ac.id

***Abstract*—Game development sudah menjadi salah satu cabang pekerjaan bagi lulusan computer science. Namun dalam pengembangan game, kadang menjadi sebuah kesulitan untuk membuat game yang minim *Bug*. Game yang minim *Bug* dapat membuat pemainnya menikmati game tersebut. Oleh karena itu kita dapat mengimplementasikan graf untuk menjadi acuan pembuatan game berbasis *stage*.**

***Keywords*—Graf, Game development, Algoritma A*.**

I. PENDAHULUAN

Dalam pembuatan game yang berbasis stage, sangat diperlukan perencanaan *design* game yang sesuai. Game yang dibuat secara asal dan tidak memerhatikan keseimbangan dari mekanisme game itu sendiri akan tidak disukai.

Namun untuk membuat game yang seimbang itu sendiri dipengaruhi oleh kompleksitas game itu sendiri. Semakin banyak mekanik yang dibuat dan semakin banyak yang saling berkaitan akan semakin banyak yang perlu diperhatikan.

Salah satu permasalahan sederhana yang dapat di selesaikan menggunakan graf adalah game yang berbasis stage yang saling terhubung dan dapat pemain sendiri dapat memilih stage mana yang ingin di tuju. Penggunaan graf ini akan memudahkan game developer untuk mengembangkan gamenya dengan acuan-acuan tertentu.

II. TEORI DASAR

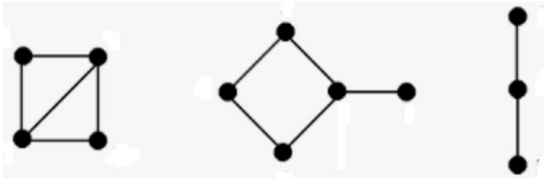
A. Graf

Graf adalah hubungan antara objek-objek diskrit yang direpesentasikan dengan simpul dan sisi. Graf dapat direpresentasikan dengan tuple:

$$G = (V, E)$$

Yang dalam hal ini, V adalah simpul dan E adalah sisinya. Graf juga memiliki banyak jenis:

1. Graf sederhana (simple graph)
   Graf yang tidak mengandung gelang maupun sisi ganda.

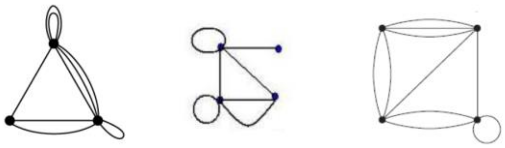


Gambar 2.1 Contoh Graf Sederhana
(sumber : Website Rinaldi Munir) diakses pada 11 Desember 2023

2. Graf tak-sederhana (unsimple-graph).
   Graf yang mengandung sisi ganda atau gelang dan dapat dibedakan lagi menjadi:
   a. Graf ganda (multi-graph)
      Graf mengandung sisi ganda

Gambar 2.2 Contoh Graf Ganda
(sumber : Website Rinaldi Munir) diakses pada 11 Desember 2023

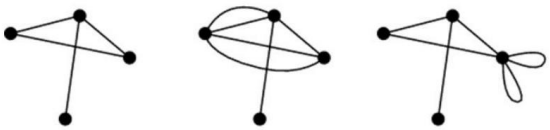
   b. Graf semu (pseudo-graph)
      Graf mengandung sisi gelang



Gambar 2.3 Contoh Graf Semu
(sumber : Website Rinaldi Munir) diakses pada 11 Desember 2023

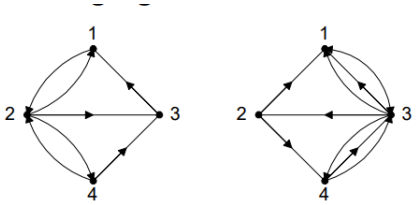
Selain itu graf juga dapat dibedakan berdasarkan orientasi arah pada sisi:

1. Graf tak-berarah (undirected graph)
   Graf yang sisinya tidak mempunyai orientasi arah disebut graf tak-berarah.



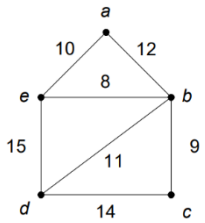
Gambar 2.4 Contoh Graf Tak Berarah
(sumber : Website Rinaldi Munir) diakses pada 11 Desember 2023

2. Graf berarah (directed graph atau digraph)
   Graf yang setiap sisinya diberikan orientasi arah disebut sebagai graf berarah.



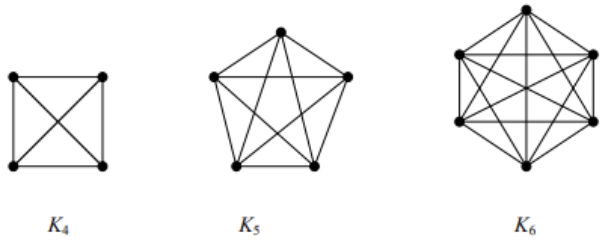
Gambar 2.5 Contoh Graf Berarah
(sumber : Website Rinaldi Munir) diakses pada 11 Desember 2023

Ada juga graf berbobot. Graf berbobot adalah graf yang sisinya memiliki sebuah nilai yang mempunyai arti.



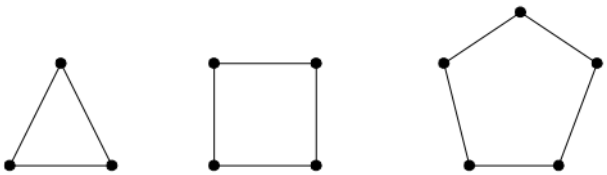
Gambar 2.6 Contoh Graf Berbobot
(sumber : Website Rinaldi Munir) diakses pada 11 Desember 2023

Graf Lengkap (Complete Graph), Graf lengkap ialah graf sederhana yang setiap simpulnya mempunyai sisi ke semua simpul lainnya.



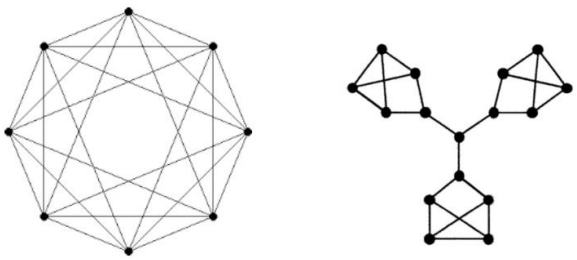
Gambar 2.7 Contoh Graf Lengkap
(sumber : Website Rinaldi Munir) diakses pada 11 Desember 2023

Graf lingkaran, Graf lingkaran adalah graf sederhana yang setiap simpulnya berderajat dua.



Gambar 2.8 Contoh Graf Lingkaran
(sumber : Website Rinaldi Munir) diakses pada 11 Desember 2023

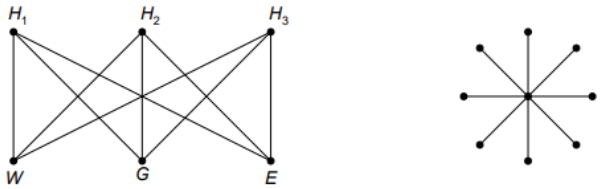
Graf Teratur (Regular Graphs), Graf yang setiap simpulnya mempunyai derajat yang sama disebut graf teratur.



Gambar 2.9 Contoh Graf Teratur

(sumber : Website Rinaldi Munir) diakses pada 11 Desember 2023

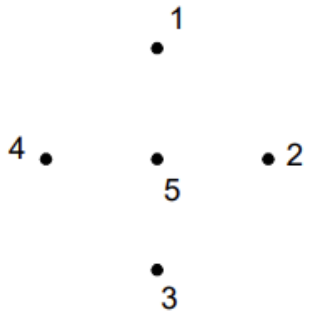
Dan Graf Bipartite (Bipartite Graph), Graf G yang himpunan simpulnya dapat dipisah menjadi dua himpunan bagian V1 dan V2, sedemikian sehingga setiap sisi pada G menghubungkan sebuah simpul di V1 ke sebuah simpul di V2.



Gambar 2.10 Contoh Graf Bipartit
(sumber : Website Rinaldi Munir) diakses pada 11 Desember 2023

Terminologi di dalam Graf:

1. Ketetanggaan (Adjacent)
   Dua buah simpul dikatakan bertetangga bila keduanya terhubung langsung.
2. Bersisian (Incidency)
   Suatu sisi akan bersisian dengan kedua simpul pada ujungnya.
3. Simpul Terpencil (Isolated Vertex)
   Simpul terpencil ialah simpul yang tidak mempunyai sisi yang bersisian dengannya.
4. Graf Kosong (null graph atau empty graph)
   Graf yang himpunan sisinya merupakan himpunan kosong (Nn).



Gambar 2.11 Contoh Graf kosong
(sumber : Website Rinaldi Munir) diakses pada 11 Desember 2023

5. Derajat (Degree)
   Derajat suatu simpul adalah jumlah sisi yang bersisian dengan simpul tersebut.
6. Lintasan (Path)
   Lintasan yang terdiri dari sisi dan simpul. Sisi–sisi saling menyambung panjang dan terdapat simpul di antara sisi-sisi tersebut.
7. Siklus (Cycle) atau Sirkuit (Circuit)
   Lintasan yang berawal dan berakhir pada simpul yang sama disebut sirkuit atau siklus.
8. Kerterhubungan (Connected)
   Dua buah simpul v1 dan simpul v2 disebut terhubung jika terdapat lintasan dari v1 ke v2.
9. Upagraf
   Jika sebuah graf merupakan subset dari graf lain berarti graf itu adalah upagraf dari graf lain.
10. Cut-Set

Cut-set dari graf terhubung G adalah himpunan sisi yang bila dibuang dari G menyebabkan G tidak terhubung.

B. Game Berbasis *Stage*

Game berbasis stage, atau yang sering disebut juga sebagai stage-based games atau level-based games, adalah jenis permainan di mana gameplay atau narasi dibagi menjadi bagian-bagian yang lebih kecil, yang dikenal sebagai "stages" atau "levels".

Setiap stage ini biasanya memiliki tujuan, tantangan, dan karakteristiknya sendiri. Hal ini membuat alur game tertata rapih saat dimainkan. Genre ini banyak ditemukan di game-game saat ini dimana terdapat sistem progresi yang bertahap.

Gambar 2.12 Contoh Game Berbasis Stage
(sumber : cs.stackexchange.com) diakses pada 11 Desember 2023

Pada gambar diatas, kita dapat melihat sebuah peta yang saya ambil dari game Eternal Return Black Survial. Pada game tersebut terdapat banyak tempat yang bisa di kunjungi. Namun pada setiap interval 2 menit, akan ada beberapa tempat yang di larang untuk ditempati. Hal tersebut termasuk progresi yang bertahap. Dimana mereka menyempitkan ruang gerak para pemain agar dapat terjeda pertemuan antara tim dari pemain yang berbeda.

III. SOLUSI PERMASALAHAN

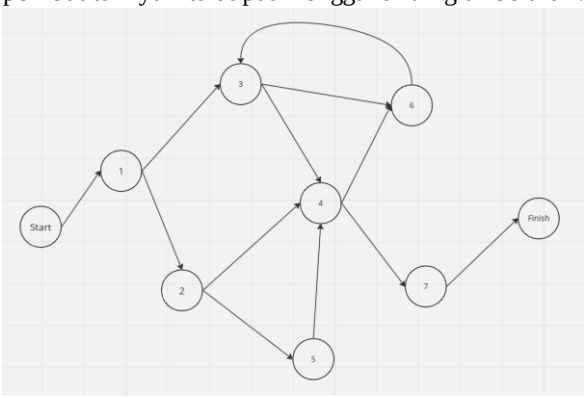
Dalam video game dengan basis stage dimana dalam setiap stage kita juga dapat memilih pilihan tersendiri, kita harus memikirkan kompleksitas dari game tersebut agar tetap dapat menarik untuk dimainkan. Salah satu cara membuat konsep game yang tetap dapat menarik saat dimainkan adalah dengan membuat skema permainannya menggunakan graf.

Dengan menyimbolkan stage dengan nodes dan hubungan antara stage dengan simpul berarah yang menandakan alur dari permainan, kita dapat menggambarkan bagaimana alur dari permainan dari mulai sampai selesai. Kita juga dapat menggambarkan alur dari permainan yang dipengaruhi oleh pilihan pemain menggunakan graf. Dan kita juga bisa membuat permainan yang didasarkan oleh graf.

Sebagai contoh, misalkan:

A. Penggambaran desain level menggunakan Graf

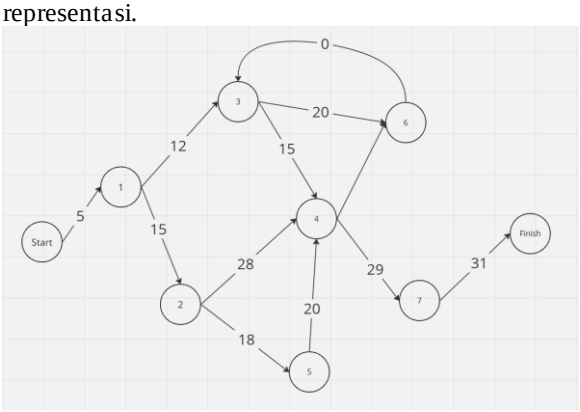
Untuk dapat menggambarkan bagaimana stage saling terhubung kita dapat menggunakan graf. Seperti saat kita bermain game labirin yang membuat kita memilih arah permainan kita. Dalam pembuatannya kita dapat menggunakan graf berarah.



Gambar 3.1 Contoh Alur Permainan dalam Graf
(sumber : Pribadi) dibuat pada 11 Desember 2023

Dalam graf tersebut angka pada simpul menandakan nomor level dan sisinya menandakan kemana player dapat pergi melalui level tersebut.

Kita juga dapat mengembangkan graf ini dengan menjadikannya graf berbobot. Dengan banyak representasi.



Gambar 3.2 Contoh Alur Permainan dalam Graf
(sumber : Pribadi) dibuat pada 11 Desember 2023

Pada contoh diatas, saya membuat graf berbobot tersebut menandakan rata-rata poin kerusakan yang berkurang dari pemain. Misalkan pemain mulai dari poin kerusakan 100 dan akan kalah jika turun sampai nol. Kita dapat memperhitungkan bagian mana yang tidak adil, seperti apakah dengan design level seperti ini player akan mungkin menyelesaikan permainan tersebut?

Penyelesaiannya dapat dilakukan dengan menerapkan algoritma *A**. Algoritma *A** dapat dengan cepat mencari jalur tercepat sampai tujuan. Hal ini sesuai dengan yang kita inginkan dimana kita ingin tau apakah kita bisa sampai dengan tujuan yaitu finish stage.

Namun perlu ada beberapa modifikasi dari algoritma tersebut dimana kita harus menambahkan

batasan bahwa *cost* tidak boleh melewati seratus. Batasan tersebut dibuat agar sesuai dengan yang ktia inginkan bahwa poin kerusakan dari *player* tidak boleh melebihi seratus. Dan juga kita harus menambahkan *handle edge case* dimana terdapat sisi yang tidak mengarah ke *finish line*.

Jika kita implementasikan ini dalam *code python* maka bentuknya akan seperti ini.

```python
import heapq

def heuristic(node, goal):
    return abs(goal - node)

def get_neighbors(node, grid):
    directions = [(0, 1), (0, -1), (1, 0), (-1, 0)]  # Up, Down, Right, Left
    result = []
    for dir in directions:
        neighbor = (node[0] + dir[0], node[1] + dir[1])
        if 0 <= neighbor[0] < len(grid) and 0 <= neighbor[1] < len(grid[0])
and grid[neighbor[0]][neighbor[1]] == 0:
            result.append(neighbor)
    return result

def astar_modified(start, goal, graph):
    """Modified A* algorithm for directed weighted graph."""
    num_nodes = len(graph)
    open_set = [(0, start)]
    came_from = {}
    g_score = [float("inf")] * num_nodes
    g_score[start] = 0

    while open_set:
        current = heapq.heappop(open_set)[1]

        if current == goal:
            path = []
            while current in came_from:
                path.append(current)
                current = came_from[current]
            return path[::-1]

        for neighbor in range(num_nodes):
            if graph[current][neighbor] != 0:  # Check if an edge exists
                tentative_g_score = g_score[current] + graph[current][neighbo
r]
                if tentative_g_score < g_score[neighbor] and
 tentative_g_score <= 100:
                    came_from[neighbor] = current
                    g_score[neighbor] = tentative_g_score
                    f_score = tentative_g_score + heuristic(neighbor, goal)
                    if neighbor not in [i[1] for i in open_set]:
                        heapq.heappush(open_set, (f_score, neighbor))

    return None
```

Gambar 3.3 Contoh Program dalam python
(sumber : Pribadi) dibuat pada 11 Desember 2023

Kode ini akan menerima nomor awal graf dan tujuan graf, dan akan mengeluarkan none jika tidak ditemukan jalur yang tepat dan mengeluarkan jalur yang ditempuh jika ditemukan jalurnya. Dengan batasan dia akan mencari jalur lain bila tentative_g_score-nya sudah melebihi seratus.

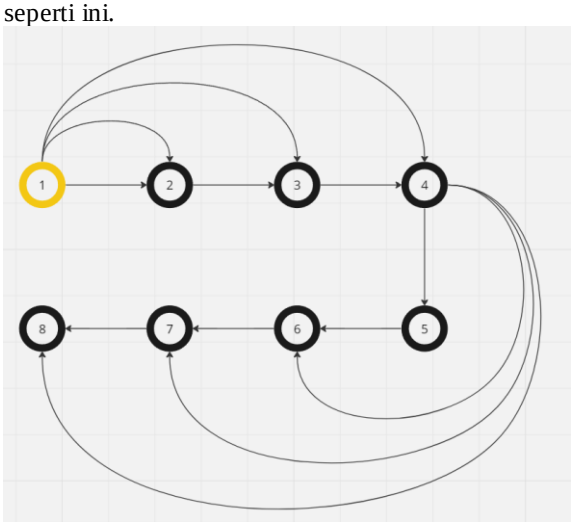
B. Penggambaran alur permainan menggunakan graf

Pada permainan Original Super Mario Bros, terdapat area tersembunyi dimana player dapat memilih stage selanjutnya.

Gambar 3.4 Contoh Alur Permainan dalam Graf
(sumber :wikihow.com) diakses pada 11 Desember 2023

Pada contoh gambar diatas, dapat kita lihat bahwa pemain berada di stage 4. Pemain diberikan pilihan untuk melanjut ke stage delapan, tujuh, enam. Pada game ini diberikan kebebasan pada pemain yang berhasil menemukan area tersembunyi tersebut dengan melompati beberapa stage.

Jika kita buat progresi nya dalam graf akan terlihat seperti ini.

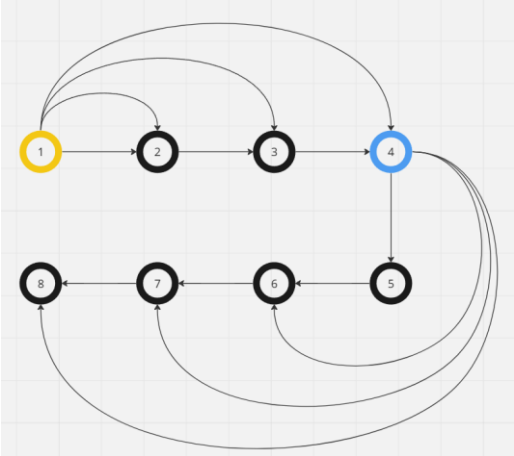


Gambar 3.5 Alur Permainan pada permainan Super Mario Bros
(sumber : Pribadi) dibuat pada 11 Desember 2023

Progresi pada permainan tersebut seharusnya memulai dari stage satu sampai delapan dengan pertambahan satu untuk setiap levelnya. Namun terdapat lompatan di stage satu dan 4 yang memungkinkan pemain untuk bisa sampai lebih cepat pada akhir dari game tersebut.

Pada game tersebut itu dinamakan *easter egg*, dimana player diberikan hadiah karena menemukan ruang tersembunyi. Pada game Super Mario Bros hal ini tidak terlalu memberikan efek yang besar pada game, karena progresi dari mario bros tidak mempunyai poin penting yang wajib di miliki pemain selama progresi.

Dengan menerapkan pembuatan alur mekanik ini kita bisa dengan mudah membuat apapun yang player pilih dia akan tetap melewatu suatu rangkaian tertentu yang kita wajibkan agar progresi dari game tersebut sesuai dengan yang kita inginkan.



Gambar 3.6 Alur Permainan pada permainan Super Mario Bros yang dimodifikasi
(sumber : Pribadi) dibuat pada 11 Desember 2023

Sebagai contoh, saya memakai alur dari permainan Super Mario Bros sebelumnya. Pada permainan Super Mario Bros, terdapat dua buah stage yang akan selalu dilewati pemain saat mereka memulai permainan. Yaitu stage satu dan stage empat. Karena kita dapat dengan mudah melihat bahwa pemain pasti akan selalu menuju stage empat dan akan selalu mulai dengan stage satu, kita dapat menaruh progresi penting disini yang tidak mungkin akan dilewati oleh pemain.

Selain itu kita juga dapat membuat bentuk graf ini untuk memudahkan kita menemukan stage yang tidak akan dilewati sama sekali atau tidak mempunyai lanjutan.

C. Konsep game yang menggunakan graf sebagai generatornya

Salah satu contoh game yang dapat dibuat ulang menggunakan konsep graf adalah *Candy Crush Saga*. Game ini dikenal oleh banyak kalangan mulai dari yang muda sampai yang tua. Game ini bersifat *infinite gameplay*, yaitu game yang tidak ada akhirnya.

Gambar 3.7 Game Candy Crush Saga
(sumber apkpac.com) diakses pada 11 Desember 2023

Pada game ini kita diharuskan menukar salah satu kotak sebanyak satu kotak secara horizontal dan vertikal, jika setelah digeser terdapat beberapa kotak dengan isi yang sama lebih dari 3 secara vertikal maupun horizontal maka kotak-kotak tersebut akan hilang. Konsep ini mirip dengan konsep graf, dimana kita dapat mengecek ketetanggan kita dan mencari secara vertikal maupun horizontal pada node yang kita ubah untuk mendapatkan akhir yang diinginkan.

IV. Hasil Program A* Modified

Graf dibuat secara acak dengan parameter parameter yang sudah ditentukan, seperti:

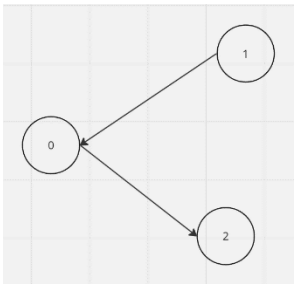
a. Jumlah Nodes
b. Probabilitas suatu sisi memiliki simpul
c. Maksimal bobot
d. Titik Awal
e. Titik Akhir

Batasan yang diberikan adalah jika bobot setelah melewati node sudah melebihi sama dengan 100.

1. Percobaan pertama

   Jumlah nodes : 3
   Probabilitas sisi : 0.5
   Titik awal : 0
   Titik akhir : 2
   Maksimal bobot : 1

   Hasil generasi graf
   Graph (Adjacency Matrix):
   [0 0 1]
   [1 0 0]
   [0 0 0]



Gambar 4.1 Hasil Generasi Graf
(sumber : Pribadi) dibuat pada 11 Desember 2023

   Hasil Jalur :
   [2]

2. Percobaan kedua

   Jumlah nodes : 10
   Probabilitas sisi : 0.5
   Titik awal : 0
   Titik akhir : 9
   Maksimal bobot : 10

   Hasil generasi graf
   Graph (Adjacency Matrix):
   [0 3 9 10 0 6 2 0 0 0]
   [0 0 4 0 5 4 6 10 3 0]
   [10 5 0 0 0 3 0 4 10 0]
   [0 0 0 0 8 3 0 0 4 3]
   [0 1 10 0 0 6 0 0 3 0]
   [0 5 4 4 2 0 0 9 0 6]
   [2 0 0 0 5 0 0 4 4 0]
   [0 0 2 9 0 0 0 0 0 0]
   [7 10 2 0 0 0 0 0 0 2]
   [0 0 9 0 0 2 8 0 0 0]

Hasil Jalur :
[6, 8, 9]

3. Percobaan ketiga
   Jumlah nodes : 3
   Probabilitas sisi : 0.5
   Titik awal : 0
   Titik akhir : 2
   Maksimal bobot : 10

   Hasil generasi graf
   Graph (Adjacency Matrix):
   [0 20 0 12 0 9 0 0 0 0 0 0 0 0 0]
   [0 0 0 0 0 19 0 0 29 0 30 18 0 0 0]
   [1 0 0 0 0 0 27 20 0 0 0 0 0 5 0]
   [0 0 26 0 0 26 22 0 0 25 0 0 0 14 0]
   [0 0 0 0 0 27 0 0 0 0 0 0 0 0 0]
   [13 0 22 0 0 0 0 13 0 0 11 0 0 0 28]
   [11 0 10 0 0 0 0 0 3 0 14 0 6 0 0]
   [0 25 0 0 0 4 6 0 0 2 0 21 2 0 0]
   [0 0 0 26 0 24 0 0 0 21 30 16 0 28 0]
   [0 0 0 0 26 0 0 0 0 0 0 13 0 4 0]
   [0 0 15 0 0 0 14 20 0 0 0 0 0 0 20 4]
   [0 0 0 0 0 0 0 0 0 0 26 0 0 0 0 7]
   [2 0 0 9 0 18 11 0 28 0 0 0 0 23 0]
   [6 0 0 27 25 0 14 30 0 0 0 0 0 0 0]
   [8 0 0 18 0 1 0 0 0 0 0 0 0 0 0]

   Hasil Jalur :
   [5, 10, 14]

4. Kasus tidak ada jalan
   Jumlah nodes : 10
   Probabilitas sisi : 0.1
   Titik awal : 0
   Titik akhir : 9
   Maksimal bobot : 15

   Hasil generasi graf
   Graph (Adjacency Matrix):
   [0 0 0 0 0 0 0 12 0 0]
   [0 0 0 0 0 0 0 0 0 0]
   [0 0 0 0 0 0 0 0 0 0]
   [0 0 0 0 0 0 0 0 0 1]
   [0 0 0 0 0 0 0 6 0 0]
   [0 0 0 5 0 0 0 0 0 12]
   [0 0 0 0 0 0 0 5 0 0]
   [0 0 0 0 15 0 0 0 0 0]
   [0 10 0 0 0 0 0 0 0 0]
   [0 0 0 0 0 0 0 12 0 0]

   Hasil Jalur :
   No path found.

5. Kasus nilai melebihi seratus
   Jumlah nodes : 10
   Probabilitas sisi : 0.1
   Titik awal : 0
   Titik akhir : 9
   Maksimal bobot : 300

   [0 0 0 0 0 0 0 0 0 151]
   [0 0 291 0 0 0 55 0 0 0]
   [0 0 0 0 0 247 0 0 0 0]
   [0 0 0 0 0 0 0 0 0 0]
   [0 0 0 0 0 0 0 121 0 0]
   [0 0 228 0 0 0 0 0 0 0]
   [0 0 0 0 0 0 0 0 0 0]
   [0 151 0 0 124 0 0 0 0 0]
   [2 288 0 0 0 0 0 0 0 0]
   [0 0 45 0 0 0 0 269 0 0]

   Hasil Jalur :
   No path found.

V. KESIMPULAN

Dapat disimpulkan bahwa kita dapat mempermudah pembuatan konsep game dalam pengembangan game menggunakan teori graf. Teori-teori tersebut dapat menjadi acuan yang tidak akan membuat konsep dari game tersebut tetap sejalan dengan yang diinginkan pembuat game.

Penerapan ilmu graf ini juga memudahkan kita membuat program *python* karena kita mempunyai gambaran seperti apa struktur data yang akan kita cek.

Dengan menggunakan ilmu-ilmu matematika diskrit, dapat memudahkan kita dalam pembuatan game. Masih banyak cabang matematika diskrit yang dapat membantu kita dalam pengembangan game. Seperti Relasi rekurens dapat digunakan untuk membuat kesulitan dari game bertambah seiring waktu, aljabar boolean untuk membuat logika komputer saat menerima input dari pemain, dll. Dan tentunya masih banyak juga yang dapat di implementasikan dari graf pada pengembangan game.

REFERENCES

[1] https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2023-2024/19-Graf-Bagian1-2023.pdf diakses pada 11 Desember 2022

[2] https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2023-2024/20-Graf-Bagian2-2023.pdf diakses pada 11 Desember 2023

[3] https://cs.stackexchange.com/questions/144342/traversal-algorithm-for-an-optimal-item-collecting-route-in-the-game-eternal-re diakkses pada 11 Desember 2023

[4] https://www.wikihow.com/Beat-Super-Mario-Bros.-on-the-NES-Quickly diakses pada 11 Desember 2023

[5] https://www.apkpac.com/app/com.king.candycrushsaga?details=dl diakses pada 11 Desember 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, 11 Desember 2023

Muhammad Fauzan Azhim - 13522153