

# Analisis Isomorfisme Graf untuk Mengecek Solvability Puzzle *Mutare Magnus* pada Game Honkai: Star Rail

Muhammad Daffa Arrizki Yanma - 13524133

Program Studi Teknik Informatika  
Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung  
[arrizkiyanma@gmail.com](mailto:arrizkiyanma@gmail.com) , [13524133@std.stei.itb.ac.id](mailto:13524133@std.stei.itb.ac.id)

**Abstrak**—Penelitian ini bertujuan untuk menganalisis solvabilitas puzzle *Mutare Magnus* pada permainan *Honkai: Star Rail* dengan pendekatan isomorfisme graf dari matematika diskrit. Setiap konfigurasi awal dan target pada puzzle dimodelkan sebagai graf tak berarah sederhana, di mana node cahaya direpresentasikan sebagai simpul, dan koneksi antar node sebagai sisi. Pemeriksaan solvabilitas dilakukan dengan memeriksa apakah kedua graf tersebut isomorfik menggunakan algoritma VF2, yang diimplementasikan melalui pustaka Python *NetworkX*. Uji coba dilakukan pada enam konfigurasi puzzle dari dua tingkat kesulitan (Easy dan Hard). Hasil pengujian menunjukkan bahwa seluruh konfigurasi awal dan target bersifat isomorfik, yang mengindikasikan bahwa seluruh puzzle tersebut dapat diselesaikan. Temuan ini menunjukkan bahwa pendekatan isomorfisme graf dapat digunakan untuk mengevaluasi solvabilitas puzzle.

**Kata Kunci**—*Isomorfisme graf, puzzle, algoritma VF2, graf tak berarah, Honkai: Star Rail.*

## I. PENDAHULUAN

Permainan berbasis giliran (turn-based) adalah jenis permainan yang menerapkan sistem di mana pemain dan lawan secara bergantian melakukan aksi. Mekanik ini memberi waktu kepada pemain untuk berpikir secara strategis sebelum mengambil keputusan, berbeda dengan permainan aksi yang menuntut respons cepat [1]. Model permainan ini umum digunakan dalam genre strategi dan role-playing game (RPG) karena memberikan kendali penuh atas tiap langkah permainan [1].

Turn-based RPG (TBRPG) memungkinkan pemain memilih aksi melalui antarmuka menu, yang kemudian dieksekusi secara bergiliran. Format ini sering dipadukan dengan unsur narasi, pertempuran strategis, dan puzzle sebagai tantangan berpikir [2].

Honkai: Star Rail, dikembangkan oleh HoYoverse, merupakan TBRPG modern yang menggabungkan sistem pertempuran berbasis giliran dengan eksplorasi dan elemen puzzle yang kompleks [3]. Salah satu puzzle penting dalam game ini adalah *Mutare Magnus*, yang mengharuskan pemain menyusun ulang koneksi antar simpul cahaya agar sesuai

dengan pola target. Puzzle ini terdiri dari simpul-simpul yang tersusun melingkar, serta satu simpul pusat, dengan tingkat kesulitan yang meningkat seiring mode (Easy ke Hard) [9].

Makalah ini mengusulkan pendekatan isomorfisme graf dari matematika diskrit sebagai metode untuk mengevaluasi solvability puzzle. Konfigurasi awal dan target puzzle dimodelkan sebagai graf tak berarah, di mana simpul merepresentasikan simpul, dan sisi menggambarkan koneksi antar simpul. Pemeriksaan isomorfisme dilakukan menggunakan algoritma VF2 melalui pustaka Python *NetworkX* untuk menentukan apakah konfigurasi awal dapat dipetakan ke target secara struktural.

## II. DASAR TEORI

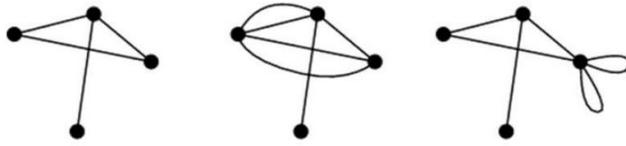
### A. Graf

Graf merupakan objek diskrit yang terdiri dari himpunan simpul (vertex atau simpul) dan himpunan sisi (Sisi) yang menghubungkan pasangan simpul. graf dinotasikan sebagai

$$G = (V, E)$$

di mana  $V$  adalah himpunan tidak-kosong dari simpul-simpul (vertices) dan  $E$  adalah himpunan sisi (Sisi) yang menghubungkan sepasang simpul [4].

Jenis graf yang digunakan dalam penelitian ini adalah graf tak berarah, yakni graf yang sisinya tidak mempunyai orientasi arah [4]. Graf jenis ini dipilih karena paling sesuai dalam merepresentasikan struktur topologis puzzle *Mutare Magnus* dalam gim *Honkai: Star Rail*, di mana tiap simpul atau pilar cahaya dihubungkan secara unik tanpa pengulangan koneksi maupun loop. Representasi struktur puzzle ke dalam graf ini memungkinkan analisis topologi secara matematis, yang membantu dalam menentukan solvability puzzle tersebut secara sistematis.



Gambar 2.1. Graf tak berarah

Sumber: <https://informatika.stei.itb.ac.id/~rinaldi.munir/>

### B. Graf Isomorfik

Dua graf dikatakan isomorfik apabila keduanya memiliki struktur yang identik, meskipun secara tampilan geometris atau penamaan simpulnya berbeda [5].

$$G1 = (V1, E1) \text{ dan } G2 = (V2, E2)$$

Dua buah graf,  $G1$  dan  $G2$  dikatakan isomorfik jika terdapat korespondensi satu-ke-satu antara simpul dan sisi kedua graf. sehingga hubungan kebersisian tetap dipertahankan [5].

Dengan kata lain, Dua buah graf yang isomorfik adalah graf yang sama, namun dengan penamaan simpul dan sisinya yang berbeda. Karena sebuah graf dapat digambarkan dalam banyak cara. [5].

Untuk implementasi, algoritma VF2 merupakan salah satu metode yang umum digunakan dalam pemeriksaan isomorfisme graf. Algoritma VF2, yang diperkenalkan oleh Cordella et al. [6], merupakan metode pencocokan graf berbasis pendekatan *depth-first search (DFS)* yang diperkuat dengan teknik *backtracking* dan *pruning* [6]. Dalam algoritma ini, pencocokan antar-simpul dua graf dilakukan secara sistematis dengan mempertimbangkan berbagai faktor topologi seperti urutan derajat simpul, relasi antar-simpul, dan struktur lokal lainnya.

Dalam penelitian ini, algoritma VF2 diimplementasikan melalui pustaka Python networkx, khususnya fungsi `is_isomorphic()`. Pemilihan algoritma ini didasarkan pada akurasi, efisiensi, serta kemudahan penggunaan dalam menentukan solvability puzzle melalui pendekatan isomorfisme graf.

### C. Honkai: Star Rail

*Honkai: Star Rail* merupakan permainan video RPG (*Role-Playing Game*) berbasis giliran (*turn-based*) yang dikembangkan oleh HoYoverse. Game ini diluncurkan sebagai game free-to-play di berbagai platform seperti PC, PlayStation, serta perangkat mobile Android dan iOS. [7].

Karakteristik unik dari *Honkai: Star Rail* adalah perpaduan mekanik pertempuran strategis dengan eksplorasi dunia yang mendalam serta berbagai jenis puzzle yang terintegrasi dalam *gameplay*. Puzzle dalam game ini tidak sekadar elemen tambahan, tetapi menjadi bagian esensial dalam eksplorasi, membuka area baru, dan memperoleh item spesial yang mendukung kemajuan pemain di dalam cerita utama [8].



Gambar 2.2. Honkai:Star Rail

Sumber: <https://www.playstation.com/en-id/games/honkai-star-rail/>

### D. Puzzle Mutare Magnus

Puzzle Mutare Magnus adalah teka-teki yang muncul dalam event-event tertentu di dalam game *Honkai: Star Rail*. Puzzle ini melibatkan pemain untuk menyusun ulang posisi simpul-simpul cahaya sehingga koneksi antar simpul tersebut sesuai dengan pola yang ditentukan. Puzzle ini umumnya tersusun secara melingkar dengan tambahan satu simpul pusat [9].

Puzzle ini memiliki dua tingkat kesulitan, yaitu *easy mode* yang relatif sederhana dengan sedikit simpul dan koneksi yang tidak terlalu kompleks, serta *hard mode* yang melibatkan lebih banyak simpul dan koneksi yang lebih kompleks, sehingga menuntut strategi berpikir yang lebih mendalam [9]. Dari sudut pandang matematika diskrit, puzzle ini dapat dimodelkan sebagai graf tak berarah, dengan simpul sebagai simpul dan koneksi antar simpul sebagai sisi. Penentuan apakah puzzle dapat diselesaikan atau tidak (*solvability*) dapat dianalisis dengan metode isomorfisme graf, di mana puzzle dianggap dapat diselesaikan (*solvable*) jika konfigurasi awal dapat diubah menjadi konfigurasi target yang diwakili oleh dua graf isomorfik.

## III. TERMINOLOGI PUZZLE MUTARE MAGNUS

Bagian ini menjelaskan istilah-istilah khusus yang digunakan dalam analisis puzzle *Mutare Magnus* dari permainan *Honkai: Star Rail*. Beberapa istilah merupakan adaptasi langsung dari mekanisme permainan, sedangkan istilah lain didefinisikan secara khusus untuk kebutuhan penelitian ini.

### 1. Simpul (Pilar Cahaya)

Simpul, atau dikenal juga sebagai pilar cahaya, merupakan titik-titik utama yang membentuk struktur puzzle *Mutare Magnus*. Setiap simpul dalam puzzle direpresentasikan sebagai sebuah simpul dalam graf. Simpul-simpul ini memiliki posisi tetap yang saling terhubung melalui koneksi garis. Dalam analisis graf, setiap simpul puzzle identik dengan konsep vertex atau simpul yang telah dibahas di bagian teori graf sebelumnya.

2. Sisi (Koneksi Garis)

Sisi, dalam konteks puzzle ini, adalah garis atau koneksi visual antar pilar cahaya (simpul). sisi menggambarkan hubungan langsung antar simpul dan secara formal direpresentasikan sebagai sisi dalam graf. Setiap sisi mengindikasikan adanya hubungan koneksi visual antar dua simpul dalam representasi graf puzzle. Hubungan ini menjadi dasar dalam pembentukan struktur graf yang digunakan untuk analisis solvability puzzle.

3. Konfigurasi Awal dan Konfigurasi Target

Dalam puzzle *Mutare Magnus*, pemain diberikan sebuah konfigurasi simpul dengan koneksi tertentu, yang disebut sebagai konfigurasi awal. Tujuan dari puzzle ini adalah mencapai sebuah konfigurasi akhir atau target yang telah ditentukan secara visual di dalam permainan. Konfigurasi awal dan target direpresentasikan sebagai dua buah graf berbeda, di mana analisis solvability dilakukan dengan memeriksa hubungan isomorfisme di antara keduanya.

4. Geser (Swap atau Rotasi)

Istilah Geser dalam puzzle ini merujuk pada tindakan pemain dalam mengubah konfigurasi puzzle. geser dapat berupa aksi swap (pertukaran posisi dua simpul secara langsung) ataupun rotasi (memutar posisi beberapa simpul mengikuti suatu pola tertentu). Setiap langkah (geser) bertujuan untuk secara bertahap mengubah konfigurasi awal menuju konfigurasi target yang diinginkan. Dalam konteks graf, setiap geser setara dengan tindakan relabeling atau permutasi posisi simpul.

5. Solvability (Keterselesaian Puzzle)

Solvability atau keterselesaian puzzle adalah istilah yang digunakan untuk menunjukkan apakah suatu konfigurasi awal dari puzzle bisa diubah secara sah menjadi konfigurasi target melalui rangkaian langkah yang diperbolehkan dalam permainan. Dalam penelitian ini, solvability puzzle *Mutare Magnus* ditentukan melalui pemeriksaan isomorfisme graf. Jika konfigurasi awal dapat direpresentasikan sebagai graf yang isomorfik dengan graf konfigurasi target, maka puzzle tersebut dikatakan dapat diselesaikan(solvable).

IV. RUANG LINGKUP ANALISIS

Bagian ini menjelaskan ruang lingkup dari analisis yang dilakukan terhadap puzzle *Mutare Magnus* dalam permainan *Honkai: Star Rail*. Dengan mendefinisikan ruang lingkup, pembaca dapat memahami konteks, batasan, serta jenis-jenis konfigurasi puzzle yang dianalisis dalam penelitian ini.

a) Penelitian ini difokuskan pada enam jenis konfigurasi puzzle yang terdiri dari tiga variasi mode *easy* dan tiga variasi mode *hard*. Masing-masing konfigurasi memiliki jumlah

simpul dan sisi yang berbeda, serta variasi dalam jumlah tempat yang dapat digeser. Rincian lengkapnya adalah sebagai berikut.

a. Easy Mode

Tabel 3.1 Konfigurasi Easy Mode

Level	Jumlah Simpul	Jumlah Sisi	Jumlah Tempat Geser
Easy 1	4	4	6
Easy 2	5	5	7 (6 simpul luar + 1 simpul pusat)
Easy 3	6	6	9 (8 simpul luar + 1 simpul pusat)

b. Hard Mode

Tabel 3.2 Konfigurasi Hard Mode

Level	Jumlah Simpul	Jumlah Sisi	Jumlah Tempat Geser
Hard 1	5	4	7 (6 simpul luar + 1 simpul pusat)
Hard 2	6	9	7 (6 simpul luar + 1 simpul pusat)
Hard 3	8	8	9 (8 simpul luar + 1 simpul pusat)

Konfigurasi ini diperoleh langsung dari pengamatan terhadap puzzle in-game dalam mode *Festive Revelry*. Masing-masing variasi dirancang oleh pengembang game untuk menguji kemampuan pemain dalam mengatur koneksi antar simpul cahaya dengan tingkat kesulitan yang berbeda.



Gambar 4.1. Contoh puzzle level easy(atas) dan puzzle level hard(bawah)

Sumber: <https://beebom.com/honkai-star-rail-mutare-magnus-event-puzzle-guide/>

Penelitian ini berfokus pada pengujian solvability masing-masing konfigurasi puzzle di atas menggunakan pendekatan isomorfisme graf. Proses analisis mencakup:

- Struktur koneksi atau sisi dari konfigurasi awal dan target.
- Pemeriksaan apakah kedua graf tersebut isomorfik.

Solvability didefinisikan sebagai kondisi ketika konfigurasi awal dapat diubah menjadi konfigurasi target melalui relabeling simpul, yang secara teoritis identik dengan uji isomorfisme antar graf.

## V. IMPLEMENTASI

### A. Penjelasan Program Python

Dalam penelitian ini, penulis mengembangkan sebuah program berbasis bahasa Python untuk memeriksa isomorfisme antara dua konfigurasi graf yang merepresentasikan puzzle Mutare Magnus. Tujuan dari program ini adalah untuk mengevaluasi apakah konfigurasi awal dapat diubah menjadi konfigurasi target melalui transformasi yang mempertahankan struktur graf. Pemeriksaan ini secara langsung berhubungan dengan solvability dari puzzle.

Program python diimplementasikan menggunakan pustaka networkx, khususnya memanfaatkan fungsi `is_isomorphic()` yang mengimplementasikan algoritma VF2 untuk pencocokan graf [5]. Program menerima input berupa jumlah simpul dan daftar sisi dari dua graf, kemudian mengevaluasi apakah keduanya isomorfik.

#### a) Fungsi `input_graph(n)`

Fungsi ini bertanggung jawab untuk membangun sebuah objek graf dari input pengguna. Input terdiri dari:

- Jumlah simpul (simpul)
- Jumlah sisi (Sisi)
- Daftar sisi yang masing-masing berupa pasangan angka  $u$  dan  $v$ , menunjukkan bahwa terdapat sisi antara simpul  $u$  dan  $v$

```
import networkx as nx

def input_graph(n):
    print(f'\nInput untuk Graf {n}')
    V = int(input("Jumlah simpul: "))
    E = int(input("Jumlah sisi: "))
    G = nx.Graph()
    print("Masukkan sisi sebagai pasangan u v (dipisah spasi):")
    for _ in range(E):
        u, v = input().split()
        G.add_Sisi(u, v)
    return G
```

Pustaka `networkx` digunakan untuk membentuk objek `Graph`, lalu sisi ditambahkan satu per satu berdasarkan input. Label simpul berupa angka (1, 2, 3, ...) dan tidak dibatasi urutan atau struktur, sehingga fleksibel digunakan untuk berbagai konfigurasi puzzle.

#### b) Fungsi `main()`

Fungsi `main()` merupakan titik masuk utama dari program. Di dalamnya, dua graf dibangun dengan memanggil `input_graph("G1")` dan `input_graph("G2")`. Setelah kedua graf terbentuk, fungsi `nx.is_isomorphic(G1, G2)` dijalankan untuk mengecek apakah kedua graf memiliki struktur konektivitas yang identik.

```
def main():
    print("== Cek Isomorfisme Dua Graf ==")
    G1 = input_graph("G1")
    G2 = input_graph("G2")

    if nx.is_isomorphic(G1, G2):
        print("\n ✓ Graf G1 dan G2 adalah isomorfik")
```

```

else:
    print("\n✗ Graf G1 dan G2 tidak isomorfik")

if __name__ == "__main__":
    main()

```

Output dari fungsi ini berupa informasi apakah kedua graf tersebut isomorfik atau tidak, yang secara langsung menunjukkan apakah konfigurasi awal dan target dalam puzzle memiliki bentuk topologi yang identik dan dengan demikian, dapat diselesaikan(solvable) atau tidak.

**B. Uji Kasus Puzzle Easy Level 1 – 4 Simpul, 4 Sisi**

Konfigurasi awal pada level ini terdiri dari 4 simpul yang terhubung oleh 4 sisi, membentuk siklus sederhana. Konfigurasi target juga memiliki jumlah simpul dan sisi yang sama.



Gambar 5.1 Konfigurasi awal (atas) dan Target(bawah) Puzzle Easy Level 1

Sumber: <https://beebom.com/honkai-star-rail-mutare-magnus-event-puzzle-guide/>

Tabel 5.1 Daftar sisi puzzle easy 1

Konfigurasi	Sisi
Awal	(1,2), (2,4), (4,5), (5,1)
Target	(1,2), (1,3), (2,4), (3,4)

Konfigurasi awal dan target kemudian dimasukkan ke dalam program sebagai daftar sisi. Program membentuk graf dari input tersebut dan memeriksa isomorfisme menggunakan fungsi `is_isomorphic()`. Hasil output ditunjukkan pada gambar

```

Input untuk Graf G1
Jumlah simpul: 4
Jumlah sisi: 4
Masukkan sisi sebagai pasangan u v (dipisah spasi):
1 2
2 4
4 5
5 1

Input untuk Graf G2
Jumlah simpul: 4
Jumlah sisi: 4
Masukkan sisi sebagai pasangan u v (dipisah spasi):
1 2
1 3
2 4
3 4

✓Graf G1 dan G2 adalah isomorfik

```

Gambar 5.2 Output Program Puzzle Easy 1

Kedua graf memiliki struktur yang identik, hanya berbeda urutan simpul. Karena itu, konfigurasi awal dapat diubah menjadi target: **dapat diselesaikan(solvable)**.

**C. Uji Kasus Penjelasan Puzzle Easy Level 2 -5 Simpul, 5 Sisi**

Konfigurasi awal pada level ini terdiri dari 5 simpul yang terhubung oleh 5 sisi, membentuk siklus sederhana. Konfigurasi target juga memiliki jumlah simpul dan sisi yang sama.



Gambar 5.3 Konfigurasi awal (atas) dan Target(bawah) Easy Level 2

Sumber: <https://beebom.com/honkai-star-rail-mutare-magnus-event-puzzle-guide/>

Tabel 5.2 Daftar sisi puzzle easy 2

Konfigurasi	Sisi
Awal	(1,2), (2,3), (3,7), (7,6), (6,1)
Target	(1,2), (2,3), (3,4), (4,5), (5,1)

Konfigurasi awal dan target kemudian dimasukkan ke dalam program sebagai daftar sisi. Program membentuk graf dari input tersebut dan memeriksa isomorfisme menggunakan fungsi `is_isomorphic()`. Hasil output ditunjukkan pada gambar

```

Input untuk Graf G1
Jumlah simpul: 5
Jumlah sisi: 5
Masukkan sisi sebagai pasangan u v (dipisah spasi):
1 2
2 3
3 7
7 6
6 1

Input untuk Graf G2
Jumlah simpul: 5
Jumlah sisi: 5
Masukkan sisi sebagai pasangan u v (dipisah spasi):
1 2
2 3
3 4
4 5
5 1

✓Graf G1 dan G2 adalah isomorfik

```

Gambar 5.4 Output Program Puzzle Easy 2

Kedua graf memiliki struktur yang identik, hanya berbeda urutan simpul. Karena itu, konfigurasi awal dapat diubah menjadi target: **dapat diselesaikan(solvable)**.

D. Uji Kasus Penjelasan Puzzle Easy Level 3 – 6 Simpul, 6 Sisi

Konfigurasi awal pada level ini terdiri dari 6 simpul yang terhubung oleh 6 sisi, membentuk siklus sederhana. Konfigurasi target juga memiliki jumlah simpul dan sisi yang sama.



Gambar 5.5 Konfigurasi awal (atas) dan Target(bawah) Easy Level 3

Sumber: <https://beebom.com/honkai-star-rail-mutare-magnus-event-puzzle-guide/>

Tabel 5.3 Daftar sisi puzzle easy 3

Konfigurasi	Sisi
Awal	(1,2), (2,3), (3,4), (4,7), (7,8), (8,1)
Target	(1,3), (1,4), (2,5), (2,6), (3,5), (4,6)

Konfigurasi awal dan target kemudian dimasukkan ke dalam program sebagai daftar sisi. Program membentuk graf dari input tersebut dan memeriksa isomorfisme menggunakan fungsi `is_isomorphic()`. Hasil output ditunjukkan pada gambar

```

Input untuk Graf G1
Jumlah simpul: 6
Jumlah sisi: 6
Masukkan sisi sebagai pasangan u v (dipisah spasi):
1 2
2 3
3 4
4 7
7 8
8 1

Input untuk Graf G2
Jumlah simpul: 6
Jumlah sisi: 6
Masukkan sisi sebagai pasangan u v (dipisah spasi):
1 3
1 4
2 5
2 6
3 5
4 6

✓Graf G1 dan G2 adalah isomorfik

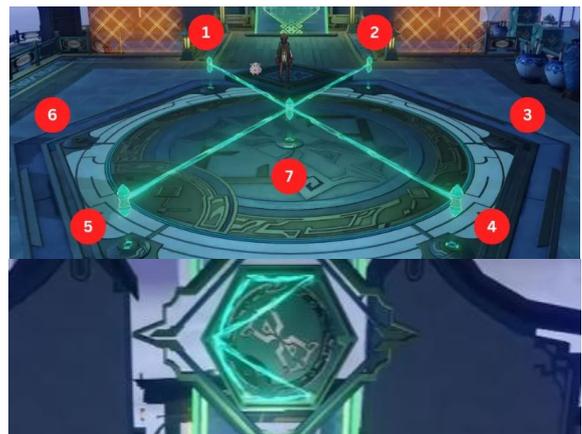
```

Gambar 5.6 Output Program Puzzle Easy 3

Kedua graf memiliki struktur yang identik, hanya berbeda urutan simpul. Karena itu, konfigurasi awal dapat diubah menjadi target: **dapat diselesaikan(solvable)**.

E. Uji Kasus Penjelasan Puzzle Hard Level 1 – 5 Simpul, 6 Sisi

Konfigurasi awal pada level ini terdiri dari 5 simpul yang terhubung oleh 6 sisi, membentuk silang yang. Konfigurasi target juga memiliki jumlah simpul dan sisi yang sama, namun dengan bentuk yang berbeda.



Gambar 5.7 Konfigurasi awal (atas) dan Target(bawah) Hard Level 1

Sumber: <https://beebom.com/honkai-star-rail-mutare-magnus-event-puzzle-guide/>

Tabel 5.4 Daftar sisi puzzle hard 1

Konfigurasi	Sisi
Awal	(1,4), (1,7), (2,5), (2,7), (4,7),

	(5,7)
Target	(1,2), (1,5), (2,5), (3,5), (3,4), (4,5)

Tabel 5.5 Daftar sisi puzzle hard 2

Konfigurasi	Sisi
Awal	(6,4), (6,3), (1,5), (1,3), (3,5), (3,4), (5,2), (4,5), (2,4)
Target	(1,2), (1,3), (1,5), (2,3), (3,5), (3,4), (4,5), (5,6), (1,6)

Konfigurasi awal dan target kemudian dimasukkan ke dalam program sebagai daftar sisi. Program membentuk graf dari input tersebut dan memeriksa isomorfisme menggunakan fungsi `is_isomorphic()`. Hasil output ditunjukkan pada gambar

```

Input untuk Graf G1
Jumlah simpul: 5
Jumlah sisi: 6
Masukkan sisi sebagai pasangan u v (dipisah spasi):
1 4
1 7
2 5
2 7
4 7
5 7

Input untuk Graf G2
Jumlah simpul: 5
Jumlah sisi: 6
Masukkan sisi sebagai pasangan u v (dipisah spasi):
1 2
1 5
2 5
3 5
3 4
4 5

✓Graf G1 dan G2 adalah isomorfik

```

Gambar 5.8 Output Program Puzzle Hard 1

Kedua graf memiliki struktur yang identik, hanya berbeda urutan simpul. Karena itu, konfigurasi awal dapat diubah menjadi target: **dapat diselesaikan(solvable)**.

*F. Uji Kasus Penjelasan Puzzle Hard Level 2 – 6 Simpul, 9 Sisi*

Konfigurasi awal pada level ini terdiri dari 6 simpul yang terhubung oleh 9 sisi. Konfigurasi target juga memiliki jumlah simpul dan sisi yang sama, namun dengan bentuk yang berbeda.



Gambar 5.9 Konfigurasi awal (atas) dan Target(bawah) Hard Level 2

Sumber: <https://beebom.com/honkai-star-rail-mutare-magnus-event-puzzle-guide/>

Konfigurasi awal dan target kemudian dimasukkan ke dalam program sebagai daftar sisi. Program membentuk graf dari input tersebut dan memeriksa isomorfisme menggunakan fungsi `is_isomorphic()`. Hasil output ditunjukkan pada gambar

```

Masukkan sisi sebagai pasangan u v (dipisah spasi):
6 4
6 3
1 5
1 3
3 5
3 4
5 2
4 5
2 4

Input untuk Graf G2
Jumlah simpul: 6
Jumlah sisi: 9
Masukkan sisi sebagai pasangan u v (dipisah spasi):
1 2
1 3
1 5
2 3
3 5
3 4
4 5
5 6
1 6

✓Graf G1 dan G2 adalah isomorfik

```

Gambar 5.10 Output Program Puzzle Hard 2

Kedua graf memiliki struktur yang identik, hanya berbeda urutan simpul. Karena itu, konfigurasi awal dapat diubah menjadi target: **dapat diselesaikan(solvable)**.

*G. Uji Kasus Penjelasan Puzzle Hard Level 3 – 8 Simpul, 8 Sisi*

Konfigurasi awal pada level ini terdiri dari 8 simpul yang terhubung oleh 8 sisi, Konfigurasi target juga memiliki jumlah simpul dan sisi yang sama, namun dengan bentuk yang berbeda.



Gambar 5.11 Konfigurasi awal (atas) dan Target(bawah) Hard Level 3

Sumber: <https://beebom.com/honkai-star-rail-mutare-magnus-event-puzzle-guide/>

Easy 3	6	6	Ya	Ya
Hard 1	5	4	Ya	Ya
Hard 2	6	9	Ya	Ya
Hard 3	8	8	Ya	Ya

Tabel 5.6 Daftar sisi puzzle hard 3

Konfigurasi	Sisi
Awal	(1,2), (2,3), (3,4), (4,5), (5,6), (6,7), (7,8), (8,1)
Target	(1,2), (1,4), (2,7), (3,7), (3,6), (4,8), (5,6), (5,8)

Konfigurasi awal dan target kemudian dimasukkan ke dalam program sebagai daftar sisi. Program membentuk graf dari input tersebut dan memeriksa isomorfisme menggunakan fungsi `is_isomorphic()`. Hasil output ditunjukkan pada gambar

```

Input untuk Graf G1
Jumlah simpul: 8
Jumlah sisi: 8
Masukkan sisi sebagai pasangan u v (dipisah spasi):
1 2
2 3
3 4
4 5
5 6
6 7
7 8
8 1

Input untuk Graf G2
Jumlah simpul: 8
Jumlah sisi: 8
Masukkan sisi sebagai pasangan u v (dipisah spasi):
1 2
1 4
2 7
3 7
3 6
4 8
5 6
5 8

✓Graf G1 dan G2 adalah isomorfik
    
```

Gambar 5.12 Output Program Puzzle Hard 3

Kedua graf memiliki struktur yang identik, hanya berbeda urutan simpul. Karena itu, konfigurasi awal dapat diubah menjadi target: **dapat diselesaikan(solvable)**.

#### H. Analisis Hasil

Pengujian dilakukan terhadap enam konfigurasi puzzle Mutare Magnus yang terdiri dari tiga level Easy dan tiga level Hard. Masing-masing konfigurasi terdiri dari graf awal dan graf target yang direpresentasikan menggunakan daftar sisi (*Sisi*). Hasil pemeriksaan menggunakan program Python menunjukkan bahwa seluruh pasangan graf adalah isomorfik.

Tabel berikut merangkum hasil pengujian:

Tabel 5.7 Hasil pengujian

Level	Jumlah simpul	Jumlah sisi	Isomorfik	Solvability
Easy 1	4	4	Ya	Ya
Easy 2	5	5	Ya	Ya

Berdasarkan hasil tersebut, dapat disimpulkan bahwa seluruh konfigurasi puzzle yang diuji memiliki struktur graf yang setara secara topologi. Meskipun jumlah simpul dan sisi bervariasi antara level, program berhasil mengidentifikasi bahwa struktur koneksi antar simpul tetap identic.

Seluruh konfigurasi puzzle Mutare Magnus yang diuji, baik pada mode Easy maupun Hard, menghasilkan output isomorfik dari program, yang berarti bahwa konfigurasi awal dan target memiliki struktur graf yang identik. Hal ini sesuai dengan kenyataan bahwa semua puzzle memang dapat diselesaikan secara *gameplay*. Dengan demikian, pendekatan isomorfisme graf terbukti relevan untuk mengidentifikasi solvability puzzle berdasarkan kesetaraan struktur topologi.

## VI. KESIMPULAN

Penelitian ini bertujuan untuk menguji apakah solvability puzzle Mutare Magnus dalam permainan Honkai: Star Rail dapat ditentukan melalui pendekatan isomorfisme graf. Dengan memodelkan konfigurasi awal dan target puzzle sebagai graf tak berarah sederhana, pemeriksaan dilakukan terhadap enam konfigurasi resmi yang terdiri dari tiga level Easy dan tiga level Hard.

Berdasarkan hasil implementasi program Python menggunakan pustaka `networkx`, seluruh pasangan graf yang diuji menghasilkan keluaran isomorfik, dan hal ini konsisten dengan kenyataan bahwa semua konfigurasi tersebut memang dapat diselesaikan(solvable) secara *gameplay*. Dengan demikian, pendekatan isomorfisme graf terbukti mampu merepresentasikan kesetaraan struktur puzzle secara matematis dan menjadi metode yang relevan untuk mengecek solvability dalam kasus seperti ini.

## VII. LAMPIRAN

- Repository Github program python: <https://github.com/daypft/Matdis-graf>

## VIII. UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada:

1. Tuhan Yang Maha Esa
2. Pak Rinaldi Munir, selaku dosen pengampu mata kuliah Matematika Diskrit
3. Orang tua yang selalu mendukung saya yang telah membantu penulis dalam menyelesaikan makalah ini.

## REFERENSI

- [1] E. Adams, *Fundamentals of Game Design*, 3rd ed. Berkeley, CA: New Riders, 2013..
- [2] J. Schell, *The Art of Game Design: A Book of Lenses*, 3rd ed. Boca Raton, FL: CRC Press, 2019.
- [3] HoYoverse Official, *Honkai: Star Rail – Official Website*, [Online]. Available: <https://hsr.hoyoverse.com/>. [Diakses 19 Juni 2025].
- [4] R. Munir, *Matematika Diskrit*, Informatika Bandung, 2025, [Online]. Available: <https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/20-Graf-Bagian1-2024.pdf>. [Diakses 19 Juni 2025].
- [5] R. Munir, *Matematika Diskrit*, Informatika Bandung, 2025, [Online]. Available: <https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/21-Graf-Bagian2-2024.pdf>. [Diakses 19 Juni 2025].
- [6] L.P. Cordella et al., "A (sub)graph isomorphism algorithm for matching large graphs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 10, pp. 1367-1372, 2004.
- [7] Fandom Honkai: Star Rail:, [Online]. Available: [https://honkai-star-rail.fandom.com/wiki/Honkai:\\_Star\\_Rail\\_Wiki](https://honkai-star-rail.fandom.com/wiki/Honkai:_Star_Rail_Wiki). [Diakses 19 Juni 2025].
- [8] IGN, "Honkai: Star Rail Review," [Online]. Available: <https://www.ign.com/articles/honkai-star-rail-review>. [Diakses 19 Juni 2025].
- [9] Beebom, "Honkai Star Rail: Mutare Magnus Event Puzzle Guide (Easy & Hard Modes)," [Online]. Available: <https://beebom.com/honkai-star-rail-mutare-magnus-guide/>. [Diakses 19 Juni 2025].

## 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, 20 Juni 2025



Muhammad Daffa Arrizki Yanma  
13524133