

# Aplikasi Teori Graf Pada Visualisasi Hubungan Antar File Dalam Aplikasi Obsidian

Muhammad Dicky Isra - 13523075<sup>1</sup>

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia

[mdikiisra98@gmail.com](mailto:mdikiisra98@gmail.com), [13523075@std.stei.itb.ac.id](mailto:13523075@std.stei.itb.ac.id)

**Abstrak**—Teori graf memiliki banyak peran penting dalam aplikasi berbagai bidang, salah satu kegunaan dari teori graf adalah untuk pengelolaan dan visualisasi data. Contoh dari penggunaan graf pengelolaan dan visualisasi data adalah fitur *graph view* dalam aplikasi pencatat Obsidian. Fitur ini memungkinkan pengguna untuk melihat hubungan file atau catatan dengan file lainnya. Makalah ini bertujuan untuk mengkaji fitur ini dengan melihatnya dari sisi matematis mengenai bagaimana menginterpretasikannya ke dalam teori graf.

**Keywords**—Obsidian, Teori Graf, Visualisasi File.

## I. PENDAHULUAN

Sejak zaman dahulu, manusia menggunakan berbagai cara untuk menyimpan informasi dengan harapan untuk mengabadikan informasi. Aktivitas menyimpan informasi ini biasanya disebut dengan mencatat. Sejak saat yang lama, proses mencatat hanya memerlukan kertas dan pena ataupun media serupa. Namun, seiring berjalannya waktu dengan era digitalisasi proses pencatatan dimungkinkan dilakukan dengan media digital. Banyak sekali aplikasi yang memudahkan kita untuk melakukan pencatatan secara digital, seperti Obsidian, Notion, Microsoft OneNote, dll.

Obsidian sebagai salah satu aplikasi pencatat digital merupakan sebuah opsi yang dapat dipilih untuk melakukan pencatatan. Aplikasi ini memungkinkan pengguna untuk menulis catatan dengan file berbasis Markdown dan disimpan secara lokal. Karena catatan disimpan dalam file berbasis Markdown, maka catatan dapat ditulis dengan sintaks Markdown. Dengan demikian, dimungkinkan menggunakan fitur Markdown seperti link. Obsidian memiliki fitur khusus untuk menghubungkan catatan dengan catatan lain dengan sintaks tertentu.

File-file yang terhubung pada aplikasi Obsidian dapat divisualisasikan melalui fitur Graph View. Fitur Graph View memungkinkan pengguna untuk melihat hubungan antar file atau catatan dengan representasi graf. Hal ini dapat mempermudah untuk melihat bagaimana suatu catatan memiliki keterhubungan dengan catatan lainnya. Fitur Graph View merupakan fitur yang didasarkan oleh teori graf dalam matematika diskrit.

Makalah ini akan membahas salah satu cara untuk menjelaskan bagaimana fitur Graph View ini dijelaskan dalam konteks matematika diskrit.

## II. LANDASAN TEORI

### A. Graf

Berdasarkan [1], graf merupakan suatu sistem yang digunakan untuk merepresentasikan objek-objek diskrit serta hubungan dari objek-objek tersebut. Graf terdiri dari himpunan tidak kosong  $V = \{v_1, v_2, v_3, \dots, v_n\}$  yang merupakan himpunan *vertex* atau simpul dan juga terdiri dari himpunan  $E = \{e_1, e_2, e_3, \dots, e_n\}$  yang merupakan himpunan *edge* atau sisi [1]. Graf memiliki notasi yang umum digunakan dalam bentuk

$$G = (V, E)$$

Dengan keterangan:

$V$  = himpunan tidak-kosong dari simpul-simpul (*vertices*)

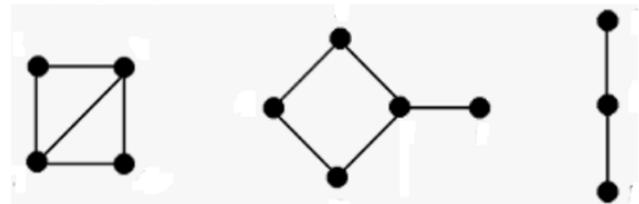
$E$  = himpunan sisi (*edges*) yang menghubungkan sepasang simpul

### B. Jenis-Jenis Graf Berdasarkan Sisi Ganda dan Simpul

Menurut [1], jenis-jenis graf digolongkan menjadi beberapa kategori. Salah satu kategori penggolongan adalah penggolongan graf berdasarkan ada atau tidaknya sisi ganda dan juga gelang. Berdasarkan penggolongan ini, graf dibagi menjadi 2 macam, yaitu:

#### 1. Graf Sederhana (*simple graph*)

Graf ini merupakan graf yang tidak memiliki sisi ganda maupun gelang.

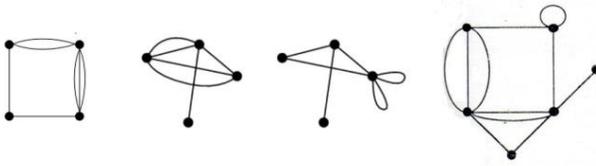


Gambar 1. Contoh Graf Sederhana

Sumber: <http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/20-Graf-Bagian1-2024.pdf>

#### 2. Graf tak-sederhana (*unsimple graph*)

Graf ini merupakan graf yang dapat memiliki sisi ganda ataupun gelang.

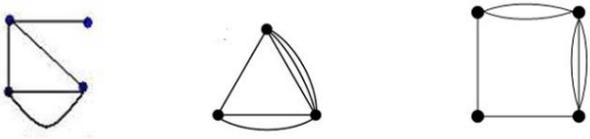


Gambar 2. Contoh Graf Tak-Sederhana

Sumber: <http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis2024-2025/20-Graf-Bagian1-2024.pdf>

Graf tak-sederhana ini dibagi kembali menjadi dua jenis, yaitu:

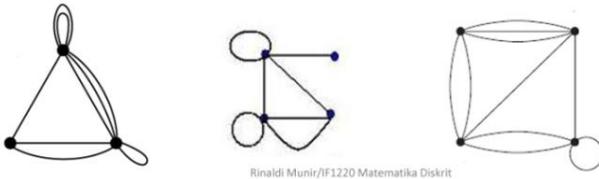
- a. Graf Ganda (*multigraph*)  
Graf ini merupakan bagian dari graf tak-sederhana yang memiliki ciri khas dengan memiliki sisi ganda.



Gambar 3. Contoh Graf Ganda

Sumber: <http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis2024-2025/20-Graf-Bagian1-2024.pdf>

- b. Graf Semu (*pseudograph*)  
Graf ini merupakan bagian dari graf tak-sederhana yang memiliki ciri khas dengan memiliki gelang.



Gambar 4. Contoh Graf Semu

Sumber: <http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis2024-2025/20-Graf-Bagian1-2024.pdf>

### C. Jenis-Jenis Graf Berdasarkan Arah

Selain dapat dikategorikan berdasarkan ada tidaknya sisi ganda dan simpul. Graf juga dapat dikategorikan berdasarkan ada atau tidaknya arah [1].

1. Graf tak-berarah (*undirected graph*)  
Graf ini merupakan graf yang tidak memiliki arah atau panah pada sisinya.

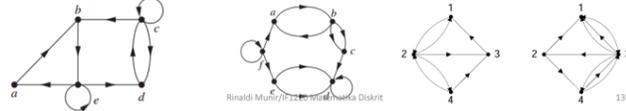


Gambar 5. Contoh Graf Tak-Berarah

Sumber: <http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis2024-2025/20-Graf-Bagian1-2024.pdf>

2. Graf berarah (*directed graph*)

Graf ini merupakan graf yang memiliki arah atau panah pada sisinya.



Gambar 6. Contoh Graf Berarah

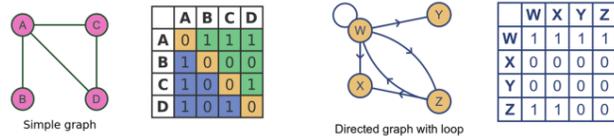
Sumber: <http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis2024-2025/20-Graf-Bagian1-2024.pdf>

### D. Representasi Graf

Graf memerlukan representasi struktur data agar dapat menyimpan data dengan efisien. Cara-cara penyimpanan graf, antara lain adalah sebagai berikut [2]:

1. Matriks Ketetanggaan (*adjacency matrix*)

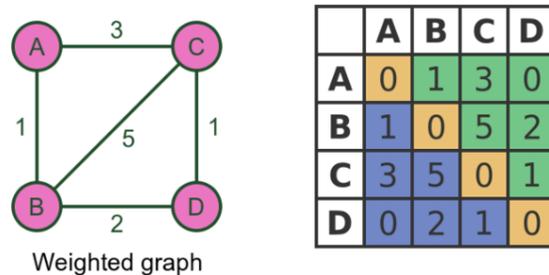
Matriks merupakan salah satu struktur data dua dimensi yang dapat digunakan untuk merepresentasikan. Pada Matriks ketetanggaan data disimpan sebagai baris dan kolom yang menyatakan benar atau tidaknya dua simpul bertetangga.



Gambar 7. Contoh Representasi Graf dengan Matriks Ketetanggaan

Sumber: <http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis2024-2025/21-Graf-Bagian2-2024.pdf>

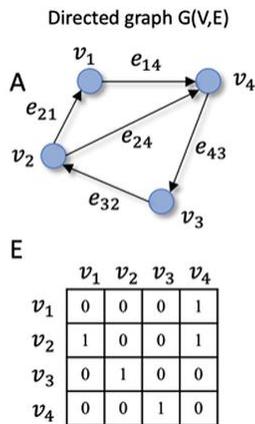
Matriks A dengan baris dan kolom yang akan diberi nama berdasarkan nama dari simpul.  $A[i,j]$  akan diberi nilai berdasarkan ada tidaknya sisi di antara baris dan kolom yang bersangkutan. Pada graf tidak berbobot, nilai ini hanya berisi 1 atau 0. Namun, pada graf berbobot nilai ini adalah nilai dari bobot sisi.



Gambar 8. Contoh Matriks Ketetanggaan Pada Graf Berbobot

Sumber: <http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis2024-2025/21-Graf-Bagian2-2024.pdf>

Kemudian pada graf berarah nilai  $A[i,j]$  perlu diperhatikan arahnya. Nilai 1 hanya akan diberikan kepada baris yang mengeluarkan arah.

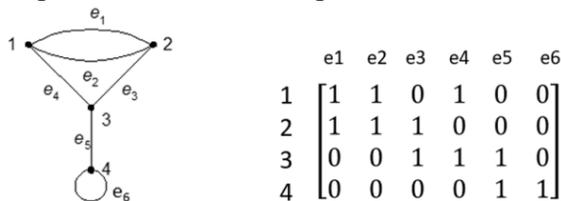


Gambar 9. Contoh Matriks Ketetanggaan Pada Graf Berarah

Sumber: <http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/21-Graf-Bagian2-2024.pdf>

## 2. Matriks Bersisian (*incidency matrix*)

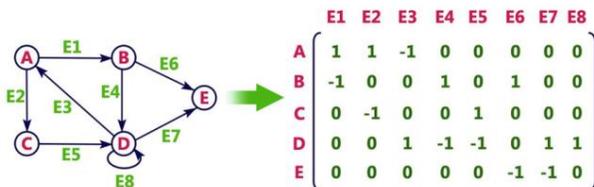
Sama seperti representasi dengan matriks ketetanggaan. Matriks bersisian juga menggunakan struktur data matriks. Perbedaannya adalah pada Matriks bersisian, digunakan simpul sebagai nama baris dan sisi sebagai nama kolom.



Gambar 10. Contoh Matriks Bersisian

Sumber: <http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/21-Graf-Bagian2-2024.pdf>

Pada graf berarah, baris dengan simpul yang mengeluarkan arah akan diberi nilai 1. Namun, jika sebaliknya diberi nilai -1 dan akan diberikan nilai 0 jika tidak bersisian [2].



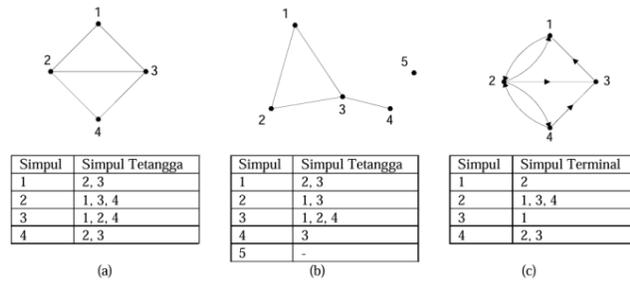
Gambar 11. Contoh Matriks Bersisian Pada Graf Berarah

Sumber: <http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/21-Graf-Bagian2-2024.pdf>

## 3. Senarai Ketetanggaan (*adjacency list*)

Pada representasi senarai ketetanggaan atau *adjacency list*. Akan digunakan senarai atau list sebagai dasar struktur

datanya. Data yang disimpan adalah data simpul yang saling bertetangga.



Gambar 12. Contoh Senarai Ketetanggaan

Sumber: <http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/21-Graf-Bagian2-2024.pdf>

Perhatikan bahwa pada Gambar 12 bagian a, jika  $x$  yang memiliki tetangga  $y$  maka  $y$  juga akan memiliki tetangga  $x$  sebab ini merupakan graf tak-berarah. Namun, pada Gambar 12 bagian c, jika  $x$  yang memiliki tetangga  $y$  maka tidak selalu memiliki tetangga  $x$ . Hal ini disebabkan oleh ketetanggaan pada graf berarah hanya dianggap satu arah saja.

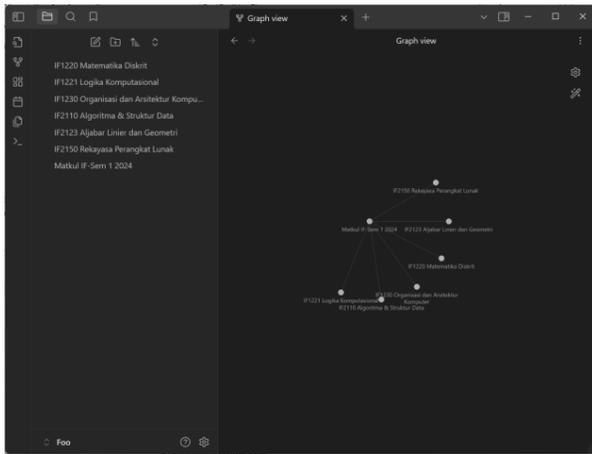
## III. PEMBAHASAN

### A. Aplikasi Obsidian dan Fitur Graph View

Aplikasi Obsidian merupakan aplikasi *cross-platform* yang memungkinkan penggunaannya untuk melakukan pencatatan berbasis file Markdown. Pencatatan dengan menggunakan file Markdown merupakan salah satu keunggulan Obsidian dibanding aplikasi pencatat lainnya. Karena berbasis Markdown, maka semua sintaks Markdown juga merupakan sintaks Obsidian yang valid. Selain itu, Obsidian juga menambahkan sintaks nya sendiri untuk memperluas Markdown itu sendiri.

Selain keunggulan dengan menggunakan file berbasis Markdown. File ini juga disimpan pada *local* folder yang disebut sebagai *vault*. Hal ini berarti file-file disimpan di dalam komputer/ponsel sendiri. Hal ini membuat aplikasi Obsidian meningkatkan privasi pengguna. Karena data tidak disimpan pada *server* yang rentan menjadi serangan maupun penyalahgunaan oleh pihak yang tidak bertanggung jawab.

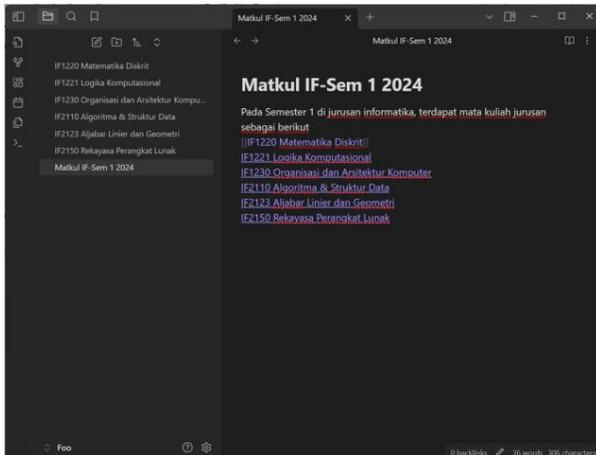
Aplikasi Obsidian, memiliki banyak fitur bawaan maupun fitur tambahan yang bisa ditambah melalui *plugins*. Salah satu fitur bawaan aplikasi Obsidian yang menggunakan matematika diskrit sebagai dasar aplikasinya adalah *Graph View*.



Gambar 13. Contoh Graph View Pada Obsidian

Sumber: Dokumentasi Pribadi

Graph View merupakan salah satu fitur pada aplikasi Obsidian yang memungkinkan penggunanya untuk melihat keterhubungan antara catatan. Hal ini dapat membantu penggunanya untuk mengetahui koneksi catatan, membantu untuk mengeksplorasi catatan, serta dapat membantu mengidentifikasi pola yang sulit terlihat jika hanya mengandalkan folder biasa.



Gambar 14. Tampilan Obsidian Dengan Menggunakan Internal Links

Sumber: Dokumentasi Pribadi

Obsidian menghubungkan antar catatan melalui salah satu fitur Markdown, yaitu *internal links* [3]. Obsidian menggunakan 2 sintaks untuk membuat *internal links*.

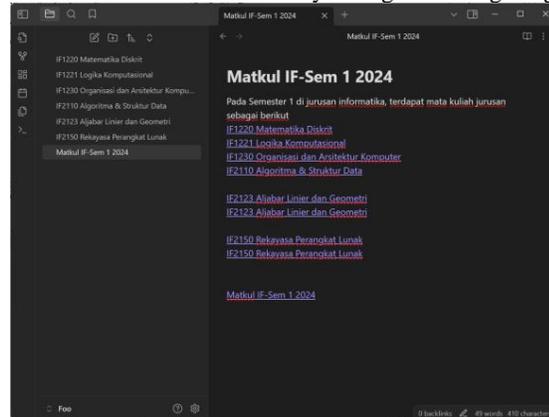
1. Format Wikilink: [[nama catatan lain]]
2. Format Markdown: [nama catatan lain](nama%20catatan%20lain.md)

### B. Pemodelan Graph View

Berdasarkan visualisasi yang diberikan oleh aplikasi. Kita dapat memodelkan bahwa graf yang dibentuk memiliki simpul yang merepresentasikan file/catatan dan juga merepresentasikan hubungan catatan sebagai sisi.

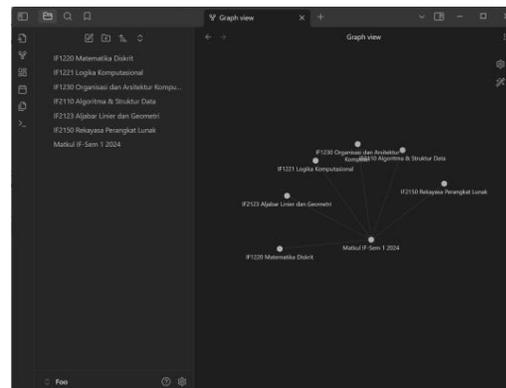
Selain itu, kita juga dapat mengategorikan graf ini ke dalam jenis-jenis graf yang ada.

#### 1. Berdasarkan ada atau tidaknya sisi ganda dan gelang



Gambar 15. Percobaan Sisi Ganda dan Gelang Dengan Internal Links

Sumber: Dokumentasi Pribadi



Gambar 16. Graph View Percobaan Sisi Ganda dan Gelang

Sumber: Dokumentasi Pribadi

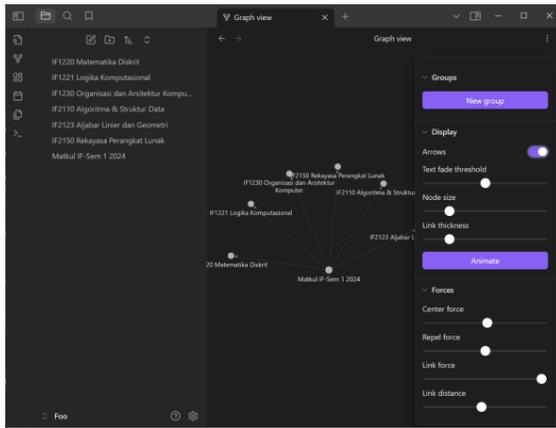
Dapat disimpulkan bahwa Obsidian tidak mengenal adanya sisi ganda maupun gelang. Obsidian hanya akan menambahkan satu sisi jika terdapat satu ataupun lebih *internal links* ke file yang sama.

Begitu juga dengan gelang, Obsidian tidak akan menambah sisi kepada simpulnya sendiri, walaupun terdapat *internal links* yang merujuk pada file nya sendiri.

Dengan demikian berdasarkan ada atau tidak sisi ganda dan simpul. *Graph View* tidak mengenal ini, sehingga bisa kita simpulkan bahwa graf yang terbentuk pada *Graph View* adalah graf sederhana.

#### 2. Berdasarkan ada atau tidaknya arah pada graf

Jika kita melihat pada Gambar 13 dan Gambar 16 terlihat bahwa tidak terdapat panah ataupun arah.



Gambar 17. Graf View Dengan Panah/Arah

Sumber: Dokumentasi Pribadi

Akan tetapi pada pengaturan *Graph View* ternyata arah ataupun panah dapat diatur untuk diperlihatkan atau tidak. Sehingga dengan ini berdasarkan adanya arah atau tidak *Graph View* dapat dikategorikan sebagai graf berarah.

Selanjutnya, kita akan memodelkan *Graph view* dalam representasi Graf yang memungkinkan. Hingga Saat ini graf yang terbentuk adalah graf pada Gambar 13 atau Gambar 16 atau Gambar 17. Jika kita daftarkan semua simpul tersebut ke dalam tabel, maka akan terbentuk tabel seperti berikut.

Tabel 1. Simpul Graf

Sumber: Dokumentasi Pribadi

Nomor	Simpul
1	IF1220 Matematika Diskrit
2	IF1221 Logika Komputasional
3	IF1230 Organisasi dan Arsitektur Komputer
4	IF2110 Algoritma & Struktur Data
5	IF2123 Aljabar Linier dan Geometri
6	IF2150 Rekayasa Perangkat Lunak
7	Matkul IF-Sem 1 2024

Sekarang kita akan merujuk simpul berdasarkan nomor yang sesuai dalam proses memodelkan representasi graf yang sesuai.

### 1. Matriks Ketetanggaan

Tabel 2. Matriks Ketetanggaan Graph View Pada Graf Berarah

Sumber: Dokumentasi Pribadi

	1	2	3	4	5	6	7
1	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0

6	0	0	0	0	0	0	0
7	1	1	1	1	1	1	0

Terbentuk matriks ketetanggaan untuk graf berarah berukuran 7x7. Yang menjadi perhatian adalah jika jumlah file membesar maka ukuran matriks juga akan membesar dan akan matriks bisa jadi akan menjadi matriks sparse karena tidak semua file akan berhubungan satu sama lain.

- Matriks Bersisian  
Representasi dalam matriks bersisian tidak akan dicoba, karena permasalahan yang sama dengan matriks ketetanggaan, yaitu memakan ruang terlalu besar.
- Senarai Ketetanggaan

Tabel 3. Senarai Ketetanggaan Graph View Pada Graf Berarah

Sumber: Dokumentasi Pribadi

Simpul	Simpul Terminal
1	-
2	-
3	-
4	-
5	-
6	-
7	1,2,3,4,5,6

Representasi ini tidak memakan banyak ruang seperti matriks ketetanggaan maupun matriks bersisian. Jika terdapat hubungan baru juga akan lebih mudah untuk memasukkan maupun mengeluarkan dari senarai

## IV. KESIMPULAN

Obsidian sebagai media untuk mencatat memberikan banyak fitur yang menarik seperti *Graph View*. Fitur ini akan dapat berguna jika dimanfaatkan dengan baik. Fitur ini dapat memudahkan pengguna untuk melihat hubungan antara file ataupun catatan. Selain itu juga dapat mempermudah eksplorasi ataupun navigasi antar catatan.

*Graph view* memanfaatkan teori graph sehingga dapat menghasilkan visualisasi yang mudah dipahami. Salah satu cara untuk menginterpretasikan *Graph view* adalah dengan menginterpretasikan graf yang dihasilkan adalah graf berarah. Selain itu, representasi senarai ketetanggaan juga memberikan representasi yang efisien dari segi ruang. Hal ini disebabkan senarai tidak memakan ruang sebanyak matriks.

## V. UCAPAN TERIMA KASIH

Puji dan Syukur atas kehadiran Allah SWT yang telah banyak membantu penulis untuk dapat menyelesaikan makalah berjudul "Aplikasi Teori Graf Pada Visualisasi Hubungan Antar File Dalam Aplikasi Obsidian". Tak lupa penulis ucapkan terima kasih dan rasa syukur kepada :

- Orang tua penulis yang selalu memberikan dukungan

- serta doa kepada penulis dalam kondisi apapun
2. Keluarga penulis yang senantiasa memberikan dukungan maupun doa.
  3. Bapak Dr. Ir. Rinaldi Munir, M.T. selaku dosen mata kuliah Matematika Diskrit yang senantiasa membantu dan membimbing mahasiswa Matematika Diskrit Kelas K-01, dan juga telah membuat *website* yang sangat membantu dalam pengerjaan makalah ini.

#### REFERENSI

- [1] R. Munir, "Graf (Bag.1)." Diakses: 4 Januari 2025. [Daring]. Tersedia pada:  
<http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/20-Graf-Bagian1-2024.pdf>
- [2] R. Munir, "Graf (Bag. 2)." Diakses: 4 Januari 2025. [Daring]. Tersedia pada:  
<http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/21-Graf-Bagian2-2024.pdf>
- [3] "Internal links - Obsidian Help." Diakses: 8 Januari 2025. [Daring]. Tersedia pada:  
<https://help.obsidian.md/Linking+notes+and+files/Internal+links>

#### 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, 8 Januari 2025



Muhammad Dicky Isra dan 13523075