BAB II
LANDASAN TEORI

A. Definisi Graf

Dalam bagian ini dibahas dasar-dasar teori graf, seperti definisi graf, *orde*, *size*, *adjacent*, *incident*, $u - v\ walk$, $u - v\ trail$, $u - v\ path$, dan subgrab.

Definisi 1. Graf G didefinisikan sebagai pasangan himpunan (V, E), ditulis dengan notasi $G = (V, E)$, dalam hal ini V adalah himpunan tidak kosong dari *vertex* dan E adalah kumpulan *edge* yang menghubungkan sepasang *vertex* [12].

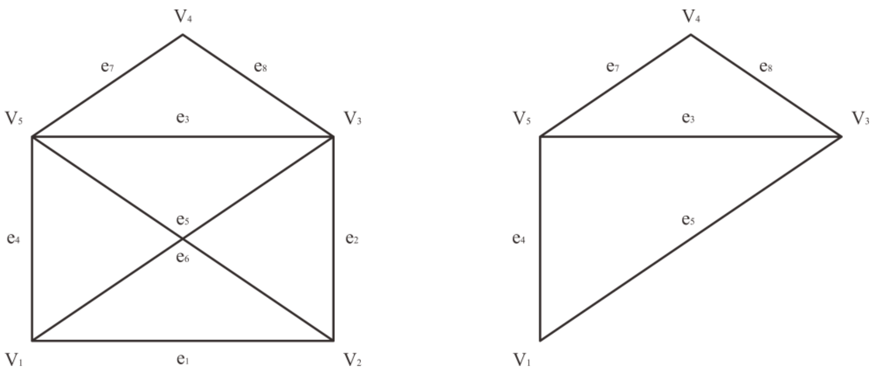
Definisi 2. Dalam graf $G, V(G)$ adalah himpunan semua *vertex* pada graf dan elemen $E(G)$ adalah himpunan semua *edge* pada graf G. Banyaknya *vertex* pada graf G disebut *orde* graf G, dinotasikan $|V(G)|$. Banyaknya *edge* pada graf G disebut *size* G, dinotasikan dengan $|E(G)|$.

Definisi 3. *Vertex u* dan *v* di katakan *adjacent* di graf *G* jika u, $v \in E(G)$. Lebih lanjut *vertex u incident edge e* dan juga *vertex v incident* dengan *edge e*, jika $e = u, v \in E(G)$, maka *u* dan *v* adalah *incident* dengan *e*.

Definisi 4. Suatu *u*-*v walk* pada suatu graf *G* adalah barisan bergantian dari *vertex* dan *edge G*, dimulai dari *vertex u* dan diakhiri dengan *vertex v*, dengan setiap *edge* menghubungkan *vertex* terhadap barisan tersebut. Suatu *u*-*v trail* pada suatu graf adalah lintasan *u*-*v walk* yang tidak mengulang *edge* manapun, sedangkan suatu *u*-*v path* adalah lintasan *u*-*v walk* yang tidak mengulang *vertex* manapun.

Definisi 5. Graf G dikatakan terhubung jika untuk sembarang dua *vertex* dalam graf *G* terdapat suatu *path* yang menghubungkan dua *vertex* tersebut. Apabila tidak terjadi hal tersebut, maka graf G tersebut tidak terhubung.

Definisi 6. Subgraf adalah graf yang himpunan *vertex* dan *edge*nya merupakan bagian dari graf yang lain. Misalkan $G = (V, E)$ adalah sebuah graf. $G_1 = (V_1, E_1)$ adalah subgraf dari G jika V_1 sub himpunan V $(V_1 \subseteq V)$ dan E_1 sub himpunan E $(E_1 \subseteq E)$.



G₁ ***G₂***

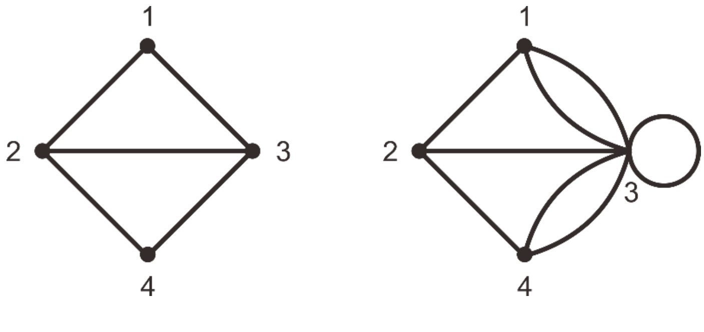
Gambar 2. Graf G_1 dan Graf G_2 Sumber [13]

Gambar 1. graf G_1 merupakan contoh graf terhubung dengan $|V(G\ )| = 5$ dan $|E(G\ )| = 8$. Dari graf G_1 didapatkan bahwasannya V_1 *adjacent* dengan V_2, V_3 dan V_5, e_1 *incident* dengan *vertex* V_1 dan V_2. Contoh dari *u – v walk* adalah V_1, e_1, V_2, e_2, V_3, e_3, V_4. Contoh dari *u – v trail* adalah V_1, e_1, V_2, e_2, V_3, e_8, V_4, e_7, V_5. Contoh dari *u – v path* adalah V_1, e_1, V_2, e_2, V_3, e_8, V_4. Contoh *cycle* adalah V_1, e_1, V_2, e_5, V_5, e_4, V_1 . Contoh dari *circuit* adalah V_1, e_1, V_2, e_2, V_3, e_3, V_5, e_4, V_1. Sedangkan pada graf G_2 merupakan contoh subgraf dari graf G_1.

B. Kelas-kelas Graf

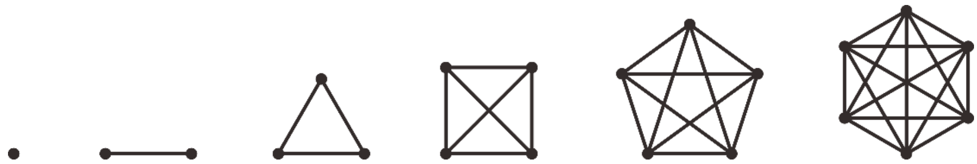
Kelas-kelas Graf yang akan dibahas adalah definisi dari graf lengkap, graf bipartit lengkap dan graf multipartit :

Definisi 8. Graf lengkap ialah graf sederhana yang setiap *vertex*nya mempunyai *edge* ke semua *vertex* lainnya. Graf dengan *n* buah *vertex* dilambangkan dengan K_n [15].



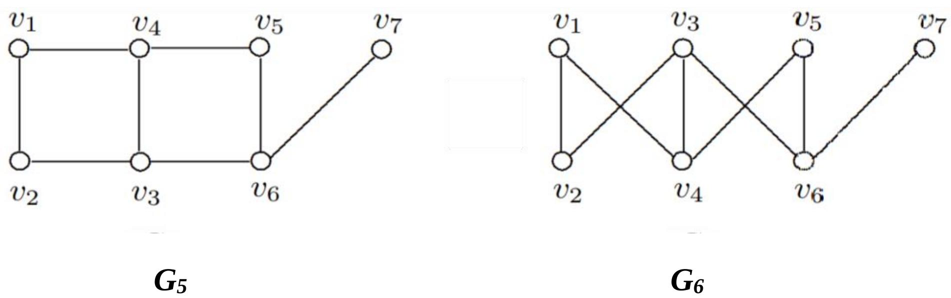
G₃ ***G₄***

Gambar 3. Contoh G_3 Graf Sederhana dan G_4 Graf tak-Sederhana Sumber [12]



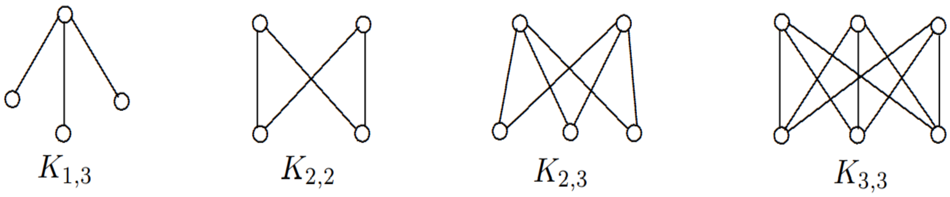
Gambar 4. Contoh Graf Lengkap Sumber [12]

Definisi 9. Suatu graf G disebut bipartit jika $V(G)$ dapat dipartisi menjadi dua himpunan U dan W, sedemikian sehingga setiap *edge* dari G menghubungkan suatu *vertex* di U dan suatu *vertex* di W [14].



Gambar 5. Contoh Graf Bipartit Sumber [14]

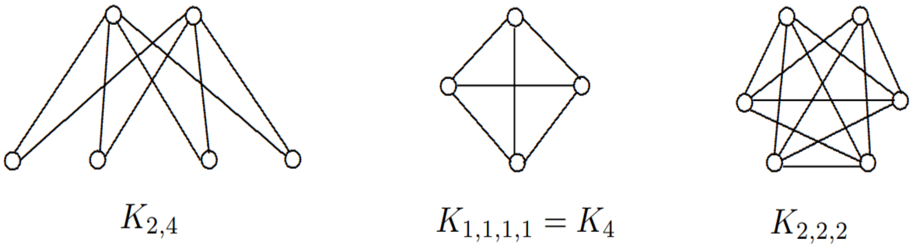
Definisi 10. Suatu graf G dikatakan graf bipartit lengkap jika $V(G)$ dapat dipartisi menjadi dua himpunan U dan W, sehingga uw adalah *edge* dari G jika dan hanya jika $u \in U$ dan $w \in W$ [14].



Gambar 6. Contoh Graf Bipartit Lengkap Sumber [14]

Definisi 11. Suatu graf G disebut graf k-partit ($k \geq 1$) jika $V(G)$ dapat dipartisi menjadi k himpunan bagian V_1, V_2, …., V_k, sehingga setiap *edge* dari G menghubungkan dengan *vertex-vertex* dalam dua himpunan partisi yang berbeda [14].

Definisi 12. Suatu graf k-partit ($k \geq 1$) lengkap G adalah suatu graf k-partit dengan sifat bahwa dua *vertex adjacent* di G jika dan hanya jika kedua *vertex* tersebut termuat di himpunan partisi yang berbeda [14].



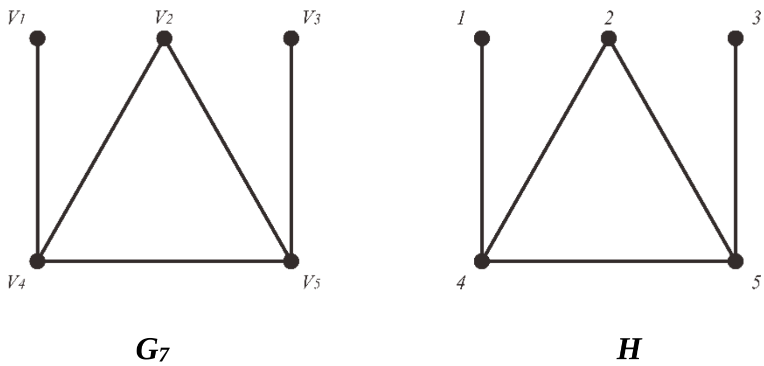
Gambar 7. Contoh Graf k-partit Lengkap Sumber [14]

C. Pelabelan Graf

Definisi 13. Pelabelan pada suatu graf adalah sebarang fungsi yang memasangkan unsur-unsur graf (*vertex* atau *edge*) dengan bilangan (biasanya bilangan bulat). Jika domain dari fungsi adalah himpunan *vertex* maka pelabelan disebut pelabelan *vertex* (*vertex labeling*). Jika domain dari fungsi adalah himpunan *edge*, maka pelabelan disebut pelabelan *edge* (*edge labeling*). Dan jika domain dari fungsi adalah himpunan *vertex* dan *edge*, maka pelabelan disebut pelabelan total (*total labeling*) [18].

Contoh diberikan graf G sebagai berikut :

Didefinisikan $f(v_1) = 1$, $f(v_2) = 2$, $f(v_3) = 3$, $f(v_4) = 4$, $f(v_5) = 5$



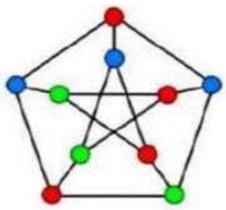
Gambar 8. Graf G_7 dan H merupakan Pelabelan Vertex Graf G_7
Sumber [15]

D. Pewarnaan Graf

Dalam teori graf, pewarnaan graf merupakan suatu bentuk pelabelan graf, yaitu dengan memberi warna pada elemen graf yang akan dijadikan subjek dalam memahami suatu permasalahan [16]. Ada tiga macam persoalan pewarnaan graf, yaitu pewarnaan *vertex*, *edge* dan wilayah.

1. Pewarnaan *Vertex*

   Pewaraan *vertex* adalah pemberian warna pada setiap *vertex* dimana warna yang tidak sama akan diberikan pada *vertex* yang saling bertetangga [17].

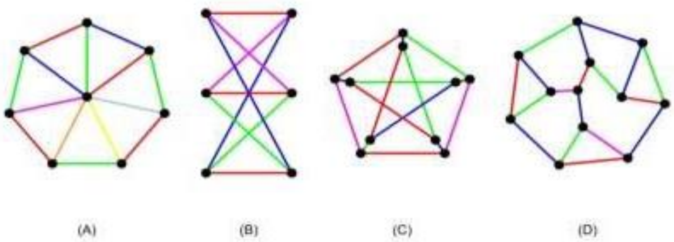


Gambar 9. Contoh Pewarnaan Vertex Sumber [17]

2. Pewarnaan *Edge*

   Pewarnaan *edge* adalah pemberian warna pada setiap *edge* pada graf sehingga *edge-edge* yang saling berhubungan tidak memiliki warna yang sama. Menurut [18] sebuah pewarnaan *edge* pada graf *G* adalah pewarnaan semua *edge G* sedemikian hingga setiap dua *edge* yang terkait pada titik yang sama mendapatkan warna yang berbeda. Jadi, graf yang berkaitan dengan pewarnaan *edge* hanya dibatasi dengan graf-graf yang sederhana saja [19].

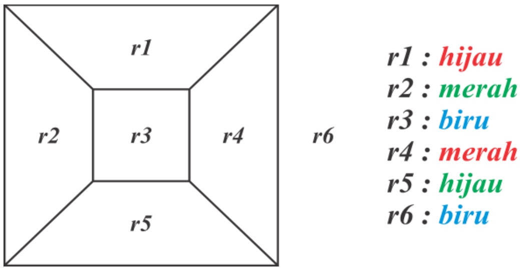
   Contoh :



Gambar 10. Contoh Pewarnaan Edge Sumber [20]

3. Pewarnaan Wilayah

   Pewarnaan wilayah adalah pemberian warna yang berbeda pada setiap wilayah atau daerah yang bertetangga, sehingga tidak ada dua wilayah yang bertetangga dengan warna yang sama [21]. Pewarnaan wilayah ini diterapakan pada pewarnaan peta. Pada pewarnaan peta, diberikan warna yang berbeda pada setiap provinsi yang saling bersebelahan. Dalam mengerjakan pewarnaan wilayah, kita dapat menggunakan prinsip pewarnaan *vertex* pada graf [19].



Gambar 11. Contoh Pewarnaan Wilayah

Dalam pewarnaan graf dikenal bilangan kromatik yaitu banyaknya warna minimum yang dapat digunakan untuk mewarnai *vertex* pada sebuah graf G dan dinotasikan dengan $\chi(G)$. Suatu graf G yang memiliki bilangan kromatik k dinyatakan dengan $\chi(G) = k$. Berdasarkan [12] bilangan kromatik dari beberapa graf dapat langsung ditentukan seperti:

a. Graf planar sederhana memiliki bilangan kromatik yang tidak lebih dari empat;
b. Graf kosong memiliki bilangan kromatik satu karena semua *vertex* tidak terhubung;
c. Graf lengkap dengan n buah *vertex* memiliki bilangan kromatik sebanyak juga banyaknya n karena semua *vertex* berhubungan satu sama lain;
d. Graf lingkaran dengan n ganjil memiliki $\chi(G) = 3$ sedangkan jika n genap memiliki $\chi(G) = 2$.

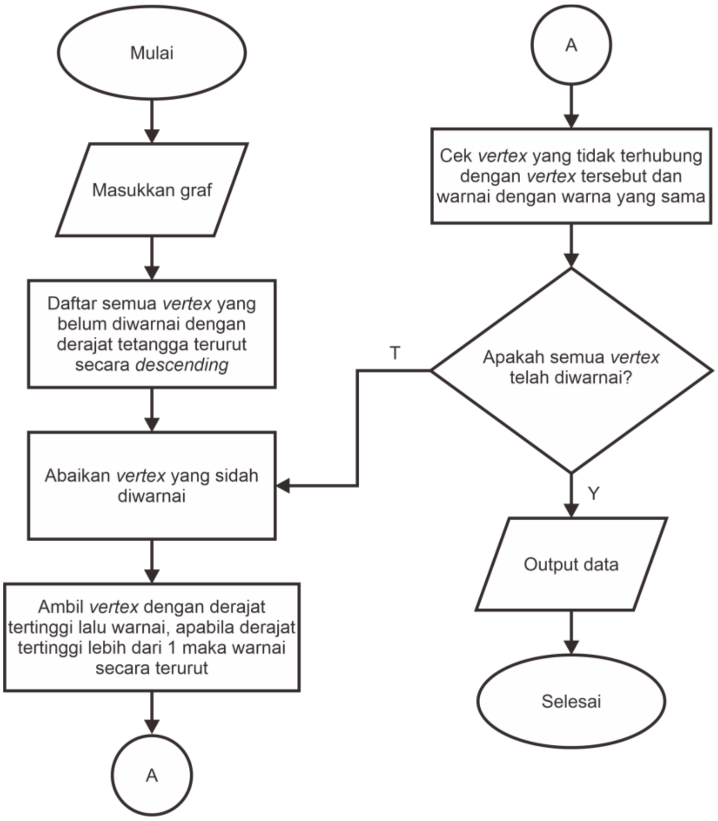
E. Algoritma *Recursive Largest First*

Algoritma *Recursive Largest First* adalah algoritma yang dapat digunakan untuk mengatasi masalah pada penjadwalan dengan mengimplementasikannya pada pewarnaan graf. Algoritma ini sesuai untuk graf yang memiliki order besar. Langkah-langkah yang terdapat dalam algoritma ini adalah sebagai berikut [22]:

1. Mendaftar semua semua derajat *vertex* dan mengurutkan secara menurun atau *descending*.
2. Dipilih *vertex* dengan derajat terbesar, kemudian diberikan sebuah warna.
3. Mencari *vertex* yang tidak bertetangga dengan simpul awal sehingga *vertex* – *vertex* tersebut menjadi calon *vertex* yang akan diwarnai sama dengan *vertex* awal.
4. Seleksi calon *vertex* sehingga diperoleh *vertex* yang dapat diwarnai sama dengan warna *vertex* awal.

5. Ulangi langkah-langkah tersebut hingga semua *vertex* terwarnai

Keterangan : Pada algoritma ini yang diprioritaskan hanya *vertex* awal saja.



Gambar 12. Flowchart Algoritma Recursive Largest First [22]

F. Algoritma *Greedy*

Algoritma Greedy merupakan salah satu algoritma yang sering digunakan dalam melakukan pewarnaan graf. Dalam algoritma ini *vertex-vertex* dari graf yang akan diwarnai diurutkan terlebih dahulu menurut aturan tertentu. Salah satu aturan pengurutan *vertex* yang lazim digunakan adalah pengurutan berdasarkan derajat dari masing-masing *vertex*. *Vertex* dengan derajat terbesar menempati urutan pertama, dan *vertex* dengan derajat terkecil menempati urutan terakhir. Setelah *vertex-vertex* diurutkan, algoritma Greedy dapat dimulai dengan urutan kerja sebagai berikut [23]:

1. Buatlah urutan pada warna-warna yang akan digunakan, sebagai contoh $\{1,2,3, \dots , k\}$.
2. Ambil *vertex* urutan pertama dan berikan warna 1.

3. Ambil *vertex* urutan berikutnya sesuai urutan *vertex*, beri warna dengan urutan terkecil yang belum diberikan kepada *vertex* lain yang *adjacent* terhadap *vertex* tersebut.
4. Ulangi langkah 3 hingga semua *vertex* terwarnai.

Keterangan : Algoritma ini tidak menggunakan derajat, jadi tidak cocok untuk data yang terdapat syarat khusus atau prioritas terkait mana dulu yang harus diberi warna (diselesaikan).

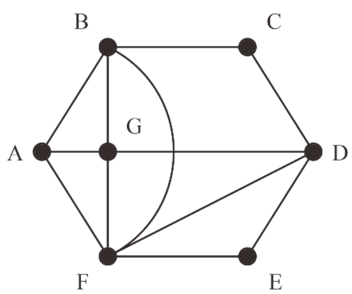
G. Algoritma Welch Powell

Algoritma Welch Powell merupakan salah satu algoritma pewarnaan graf yang melakukan pewarnaan berdasarkan derajat tertinggi dari *vertex-vertex*nya yang mengadopsi metode *Largest Degree Ordering* (LDO) [21]. Langkah-langkah Algoritma Welch Powell adalah sebagai berikut :

1. Menuliskan derajat setiap *vertex* dari graf *G*.
2. Urutkan *vertex* dari graf *G* dari derajat yang paling tinggi ke derajat yang paling rendah.
3. a) Pilih satu *vertex* dengan derajat tertinggi. Kemudian beri warna pada *vertex* tersebut.
   b) Pilih *vertex* yang tidak bertetangga dengan *vertex* yang telah terpilih pada a). Beri warna *vertex* tersebut dengan warna yang sama dengan *vertex* yang telah terpilih pada a).
   c) Jika ada *vertex* pada b) yang bertetangga maka hapus beberapa warna *vertex* sehingga tidak ada warna *vertex* yang bertetangga. Jika tidak ada *vertex* pada b) yang bertetangga maka lanjut ke nomor 4.
4. Ulangi langkah nomor 3 dengan memilih *vertex* yang belum mempunyai warna.
5. Pewarnaan *vertex* berakhir jika semua *vertex* telah terwarnai.

Contoh Algoritma Welch Powell [24]:

Diberikan graf *G* seperti di bawah ini :



Gambar 13. Contoh Graf G_8 Sumber [24]

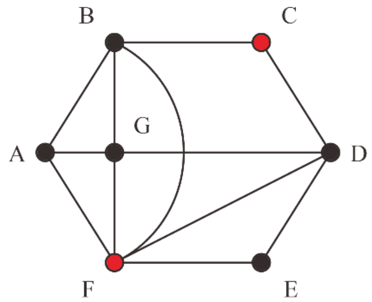
Berikut langkah-langkah pewarnaan *vertex* pada graf *G* menggunakan Algoritma Welch Powell :

1. Urutkan *vertex* dari *G*, seperti pada tabel

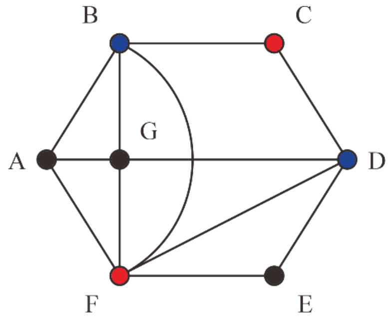
Tabel 1. Penentuan Warna *Vertex* Pada Graf *G*

Vertex	F	B	D	G	A	C	E
Derajat	5	4	4	4	3	2	2

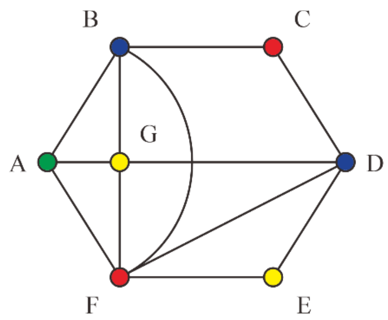
2. Berdasarkan Tabel 1. diperoleh beberapa titik dengan derajat terbesar yaitu F, dengan derajat 5. Ambil satu warna (misalnya merah) untuk titik F. Kemudian warnai titik yang tidak bertetangga dengan F dengan warna yang sama (merah). Diperoleh titik yang tidak bertetangga dengan F adalah C. Jadi titik C berwarna merah. Urutan titik-titik yang belum terwarnai menjadi B, D, G, A, E.



3. Ambil warna kedua (misalnya biru) untuk titik B. Titik yang tidak bertetangga dengan B adalah D dan E. Warnai titik D dengan warna biru. Karena titik D sudah berwarna biru dan E bertetangga dengan D, maka titik E tidak bisa diwarnai dengan biru. Sekarang urutan titik-titik yang belum terwarnai menjadi G, A, E.



4. Ambil warna ketiga (misalnya kuning) untuk titik G. Dengan cara yang sama, warnai pula titik E dengan kuning. Sehingga sisanya adalah titik A diwarnai dengan warna berbeda (misalnya hijau). Jadi graf G dapat diwarnai dengan 4 warna. Artinya $\chi(G) = 4$ atau G adalah 4 – kromatik. Hasilnya adalah sebagai berikut.



Gambar 14. Contoh Pewarnaan Vertex Graf G_8 Sumber [21]

Keterangan : Algoritma ini mempunyai prioritas yaitu derajat tertinggi hingga terendah, sehingga dapat digunakan dalam data yang memiliki syarat khusus dalam pewarnaannya untuk menyelesaikan permasalahan pewarnaan.

H. Penelitian yang Relavan

Dalam penelitian ini ada beberapa penelitian yang relevan sebagai referensi yang dapat ditunjukkan menggunakan tabel di bawah ini :

Tabel 2. Penelitian yang Relevan

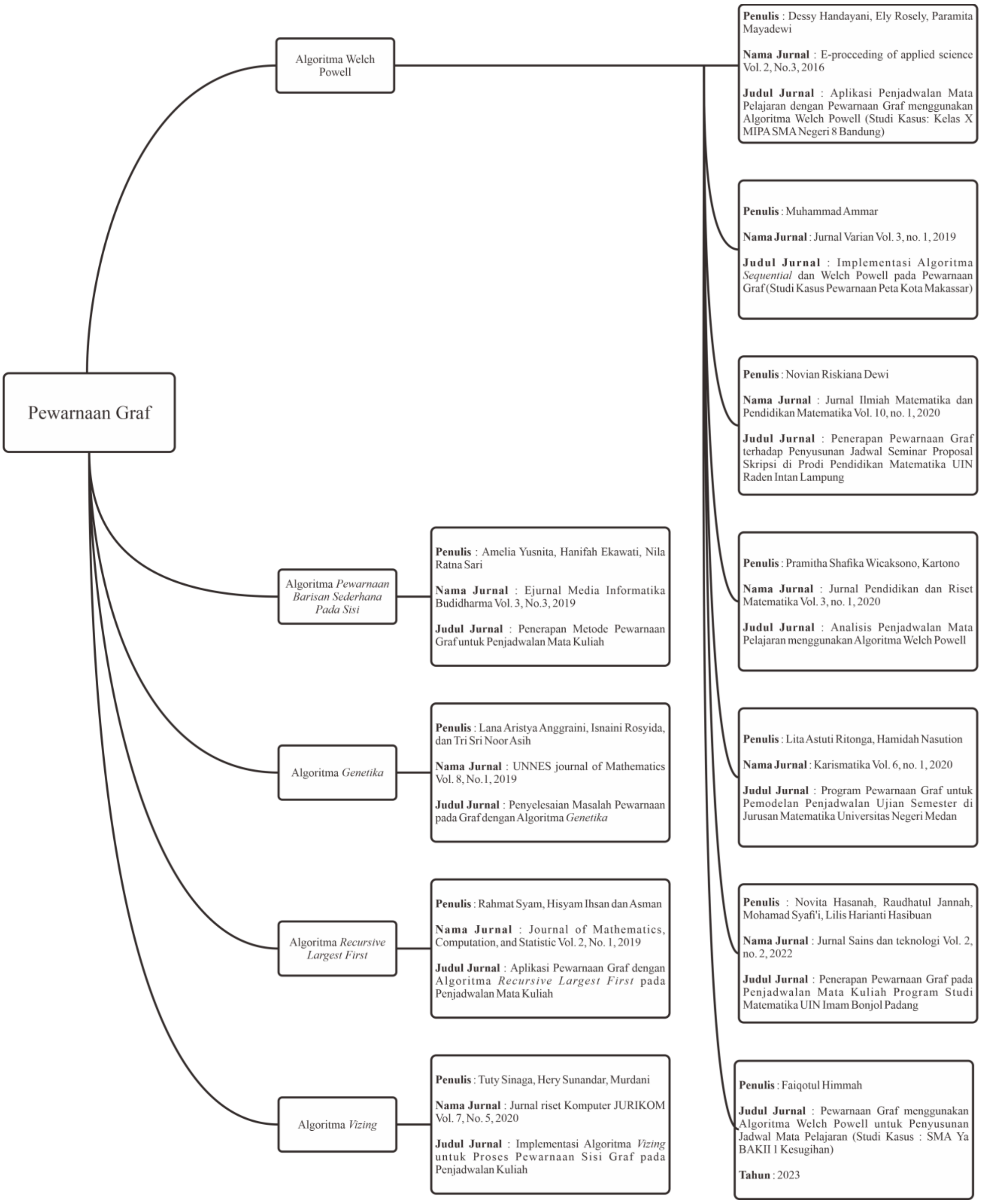
No	Peneliti	Keterangan
1	Dessy Handayani, Ely Rosely, Paramita Mayadewi (E-Proceeding Of Applied Science Vol. 2, No. 3, 2016)	**Judul** : Apilikasi Penjadwalan Mata Pelajaran dengan Pewarnaan Graf menggunakan Algoritma Welch Powell (Studi Kasus : Kelas XMIPA SMA Negeri 8 Bandung) **Metode** : Algoritma Welch Powell **Hasil** : Algoritma Welch Powell ini dapat menyusun jadwal mata pelajaran dan dapat mengampu mata pelajaran ke sebanyak bilangan kromatik yang dihasilkan.

No	Peneliti	Keterangan
2	Muhammad Ammar (Jurnal Varian Vol. 3, No. 1, 2019)	**Judul** : Implementasi Algoritma *Sequential* dan Welch Powell pada Pewarnaan Graf (Studi Kasus Pewarnaan Peta Kota Makassar). **Metode** : Algoritma *Sequential* dan Welch Powell **Hasil** : Pada Algoritma Sequential dan Algoritma Welch Powell sama-sama menghasilkan bilangan kromatik $\chi(G) = 4$. Kedua algoritma juga jauh lebih efektif dibandingkan pewarnaan peta sebelumnya.
3	Novian Riskiana Dewi (Jurnal Ilmiah Matematika dan Pendidikan Matematika, Vol. 10, No. 1, 2020)	**Judul** : Penerapan Pewarnaan Graf terhadap Penyusunan Jadwal Seminar Proposal Skripsi di Prodi Pendidikan Matematika UIN Raden Intan Lampung **Metode** : Algoritma Welch Powell **Hasil** : Pewarnaan graf pada titik dengan menggunakan Algoritma Welch Powell dapat diterapkan untuk menyusun jadwal seminar proposal skripsi di Prodi Pendidikan Matematika UIN Raden Intan Lampung. Sehingga jadwal yang disusun tidak bentrok dengan jadwal seminar yang lain dengan dosen pembimbing yang sama.
4	Pramitha Shafika Wicaksono, Kartono (Jurnal Pendidikan Dan Riset Matematika; Vol. 3, No. 1, 2020)	**Judul** : Analisis Penjadwalan Mata Pelajaran menggunakan Algoritma Welch Powell **Metode** : Algoritma Welch Powell **Hasil** : Proses penyusunan jadwal dapat dilakukan dengan menggunakan Algoritma Welch Powell yaitu teknik pada pewarnaan simpul pada graf dan diperoleh jadwal mata pelajaran yang tidak ada bentrok antar guru, mata pelajaran, dan jam mengajar.

No	Peneliti	Keterangan
5	Lita Astuti Ritonga, Hamidah Nasution, Lita Astuti Ritonga, Hamidah Nasution (Karismatika; Vol. 6, No. 1, 2020)	**Judul** : Program Pewarnaan Graf untuk Pemodelan Penjadwalan Ujian Semester di Jurusan Matematika Universitas Negeri Medan **Metode** : Algoritma Welch Powell **Hasil** : Pewarnaan pada graf digunakan untuk menyusun jadwal ujian berdasarkan pengelompokkan warna, sehingga jadwal ujian yang disusun berdasarkan hasil pewarnaan tidak akan terjadi bentrok jadwal.
6	Novita Hasanah, Raudhatul Jannah, Mohamad Syafi'i, Lilis Harianti Hasibuan (Jurnal Sains dan Teknologi, Vol. 2, No. 2, 2022)	**Judul** : Penerapan Pewarnaan Graf pada Penjadwalan Mata Kuliah Program Studi Matematika UIN Imam Bonjol Padang **Metode** : Algoritma Welch Powell **Hasil** : Penerapan pewarnaan graf dengan Algoritma Welch Powell bisa menjadi solusi untuk memudahkan dalam penyusunan jadwal mata kuliah, sehingga didapatkan jadwal mata kuliah lebih optimal dan tidak tumpang tindih.
7	Amelia Yusnita, Hanifah Ekawati, Nila Ratna Sari (Jurnal Media Informatika Budidarma Vol. 3, No. 3, 2019)	**Judul** : Penerapan Metode Pewarnaan Graf untuk Penjadwalan Mata Kuliah **Metode** : Algoritma Pewarnaan Barisan Sederhana Pada Sisi **Hasil** : Dengan menerapkan metode pewarnaan graf untuk penjadwalan mata kuliah dapat mempermudah program studi dalam menyusun jadwal perkuliahan tiap semesternya, sehingga menghindari terjadinya tumpeng tindih antara mata kuliah, dosen, ruangan dan waktu yang bersamaan.

No	Peneliti	Keterangan
8	Lana Aristya Anggraini, Isnaini Rosyida, dan Tri Sri Noor Asih (Unnes Journal of Mathematics Vol. 8, No. 1, 2019)	**Judul** : Penyelesaian Masalah Pewarnaan pada Graf dengan Algoritma Genetika **Metode** : Algoritma Genetika **Hasil** : Penyelesaian masalah pewarnaan graf dengan Algoritma Genetika bisa menjadi salah satu alternatif penyelesaian masalah pewarnaan graf.
9	Rahmat Syam, Hisyam Ihsan dan Asman (Journal Of Mathematics, Compotations, And Statistics Vol. 2, No. 1, 2019)	**Judul** : Aplikasi Pewarnaan Graf dengan Algoritma Recursive Largest First pada Penjadwalan Mata Kuliah **Metode** : Algoritma *Recursive Largest First* **Hasil** : Penjawalan mata kuliah menggunakan pewarnaan graf dengan Algoritma *Recursive Largest First* menjamin bahwa tidak ada jadwal dosen yang mengajar beberapa mata kuliah akan mendapatkan jadwal yang bersamaan untuk setiap mata kuliah yang diajarkannya.
10	Tuty Sinaga, Hery Sunandar, Murdani (Jurnal Riset Komputer (JURIKOM) Vol. 7, No. 5, 2020)	**Judul** : Implementasi Algoritma *Vizing* untuk Proses Pewarnaan Sisi Graf pada Penjadwalan Kuliah **Metode** : Algoritma Vizing **Hasil** : Proses pewarnaan graf pada penjadwalan kuliah ini mendapatkan jadwal yang minimal dalam jumlah hari, penggunaan ruang kuliah, bahkan waktu dalam perkuliahannya sangat minimal dalam perharinya.

Penelitian yang terkait dapat pula ditunjukkan melalui *mind map* seperti pada gambar di bawah ini :



Gambar 15. mind map penelitian yang terkait