

## 13. Pewarnaan dan Dekomposisi Edge

Oleh : Ade Nurhopipah

### Pokok Bahasan :

1. Pewarnaan Edge
2. Algoritma Pewarnaan Edge
3. Edge Dekomposisi

### Sumber :

Aldous, Joan M. ,Wilson, Robin J. 2004. *Graph and Applications*. Springer: UK.

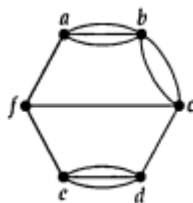
Pada bab ini, kita akan meninjau masalah yang berkaitan dengan pewarnaan edge pada sebuah graph. Kita akan mempelajari sebuah algoritma untuk pewarnaan edge dan meninjau masalah yang berkaitan dengan pembagian himpunan edge ke dalam sub himbunan *disjoint* dengan sifat tertentu. Pembahasan kita pada materi ini berkaitan dengan masalah yang berhubungan dengan papan sirkuit, *matching* dan penjadwalan.

### Pewarnaan Edge

#### Contoh 13.1 : Pewarnaan kabel

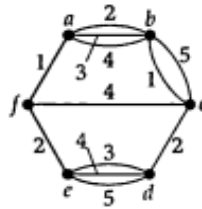
Masalah yang akan kita bahas ini, diperkenalkan oleh C.E Shannon pada tahun 1949 tentang jaringan elektronik. Seorang insinyur ingin merancang sebuah panel di mana komponen listik  $a, b, \dots$  akan dibuat terhubung. Koneksi kabel  $a$  dengan  $b$  dilakukan dengan cara menghubungkan kabel  $a$  yang keluar pada sebuah lubang dengan kabel  $b$  yang keluar dari lubang yang lain dan seterusnya. Untuk membedakan kabel yang keluar dari lubang yang sama, mereka mewarnai kabel dengan warna berbeda-beda. Berapa jumlah warna yang dibutuhkan untuk keseluruhan sistem?

Untuk mempelajari masalah ini, kita merepresentasikan titik dengan vertex dan kabel dengan edge. Sebagai contoh, graph pada Gambar 13.1 merepresentasikan sebuah panel dengan enam komponen  $a, \dots, f$ .



Gambar 13.1 Masalah pewarnaan kabel

Karena  $b$  memiliki lima edge yang insiden dengannya, dan karena edge tersebut harus memiliki warna yang berbeda, maka setidaknya dibutuhkan lima warna. Pada Gambar 13.2 ditunjukkan edge dengan warnanya masing-masing.


**Gambar 13.2 Solusi masalah pewarnaan kabel**
**Index Kromatik**

Penetapan warna kabel pada contoh 13.1 menggambarkan definisi berikut.

**Definisi 13.1**

Misalkan  $G$  adalah graph tanpa loop, sebuah pewarnaan  $k$ -edge adalah penetapan paling banyak  $k$  warna pada edge di  $G$  sedemikian rupa sehingga dua edge yang bertemu pada sebuah vertex ditandai dengan warna yang berbeda.

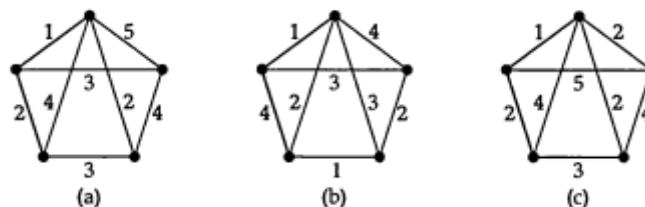
Jika  $G$  memiliki pewarnaan  $k$ -edge, maka  $G$  disebut  $k$ -edge colourable.

Index kromatik dari  $G$ , dinotasikan dengan  $\chi'(G)$  adalah bilangan terkecil  $k$  di mana  $G$  adalah  $k$ -edge colourable.

Pada contoh masalah kabel 13.1, graph tersebut memiliki index kromatik 5.

Catat bahwa definisi yang diberikan di atas hanya berlaku untuk graph tanpa loop. Loop tidak dimasukkan karena, pada setiap pewarnaan  $k$ -edge, edge yang bertemu pada sebuah vertex harus memiliki warna yang berbeda. Namun, kita membahas graph dengan multiple edge seperti pada masalah pewarnaan kabel sebelumnya. Adanya multiple edge mungkin akan mempengaruhi nilai indeks kromatik,

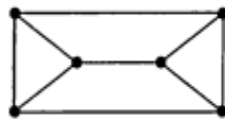
Kita menunjukkan pewarnaan  $k$ -edge dengan menulis angka  $1, 2, \dots, k$  di samping edge yang bersesuaian. Sebagai contoh, diagram (a) dan (b) pada Gambar 13.3 menunjukkan sebuah pewarnaan 5-edge dan pewarnaan 4-edge dari sebuah graph dengan delapan edge. Catat bahwa diagram (c) bukanlah pewarnaan 5-edge dari  $G$ , karena dua edge yang insiden pada sebuah vertex diwarnai yang sama.


**Gambar 13.3 Pewarnaan  $k$ -edge pada sebuah graph**

Karena  $G$  memiliki pewarnaan 4-edge, maka  $\chi'(G) \leq 4$ , sehingga 4 adalah batas atas dari  $\chi'(G)$ . Graph  $G$  juga memuat empat edge yang bertemu di sebuah vertex (sebuah vertex dengan derajat 4) yang harus diwarnai dengan warna berbeda, maka  $\chi'(G) \geq 4$ , maka 4 adalah batas bawah dari  $\chi'(G)$ . Dengan mengkombinasikan pertidaksamaan ini kita memperoleh  $\chi'(G) = 4$ .

**Latihan 13.1**

1. Tentukan  $\chi'(G)$  untuk setiap graph berikut.



(a)



(b)



(c)

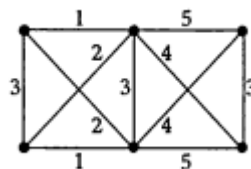
Petunjuk : Untuk setiap graph, beri pewarnaan yang mungkin dan jelaskan mengapa tidak ada pewarnaan edge dengan lebih sedikit warna.

2. Apa yang dapat anda katakan tentang graph  $G$  dengan
- $\chi'(G)=1$
  - $\chi'(G)=2$
3. Tuliskan indeks kromatik pada setiap graph berikut.
- Graph komplit  $K_4$
  - Graph bipartit komplit  $K_{2,3}$
  - Graph cycle  $C_6$
4. Tentukan apakah setiap pernyataan berikut tentang sebuah graph  $G$  adalah benar atau salah, dan berikan sebuah pembuktian atau contoh yang sesuai.
- Jika  $G$  memuat sebuah vertex dengan derajat  $r$ , maka  $\chi'(G) \geq r$
  - Jika  $\chi'(G) \geq r$ , maka  $G$  memuat sebuah vertex dengan derajat  $r$

Jika kita diberi sebuah graph  $G$ , bagaimanakah kita dapat menentukan index kromatiknya?

*Batas atas untuk  $\chi'(G)$  dapat diperoleh dengan membangun sebuah pewarnaan eksplisit untuk edge  $G$ . Sedangkan batas bawah untuk  $\chi'(G)$  dapat diperoleh dengan menemukan derajat vertex paling besar pada  $G$ .*

Jika kita menemukan sebuah batas atas dan batas bawah yang sama, maka nilai  $\chi'(G)$  sama dengan nilai keduanya. Sebagai contoh, edge pada Graph  $G$  pada Gambar 13.4 dapat diwarnai dengan lima warna maka  $\chi'(G) \leq 5$ . Namun  $G$  tidak dapat diwarnai kurang dari lima warna, karena  $G$  memuat sebuah vertex dengan derajat 5, maka  $\chi'(G) \geq 5$ . Dengan mengkombinasikan kedua pertidaksamaan, kita memperoleh  $\chi'(G)=5$ .



**Gambar 13.4 Graph untuk ilustrasi batas atas dan batas bawah  $\chi'(G)$**

Catat bahwa jika sebuah graph memiliki  $m$  edge, maka  $\chi'(G) \leq m$ . Namun batas atas ini biasanya kurang bagus. Pertidaksamaan ini menjadi sebuah persamaan  $\chi'(G)=m$ , hanya jika  $G$  adalah sebuah graph komplit bipartit dengan bentuk  $K_{1,m}$ .

Batas atas yang lebih baik diberikan oleh V.G Vinzing dan C.E Shannon. Untuk graph sederhana, Vinzing membuktikan hasil berikut pada tahun 1963.

**Teorema 13.1 : Teorema Vinzing**

Misalkan  $G$  adalah graph sederhana dengan derajat vertex maksimal  $d$ , maka berlaku:

$$d \leq \chi'(G) \leq d+1$$

Hasil ini menunjukkan kita bahwa jika  $G$  adalah graph sederhana dengan derajat vertex maksimal  $d$ , maka indeks kromatik  $G$  adalah  $d$  atau  $d+1$ . Ini membuat kita dapat mengklasifikasikan graph sederhana menjadi dua kelas, yaitu graph yang memiliki  $\chi'(G) = d$  dan graph yang memiliki  $\chi'(G) = d+1$ .

**Latihan 13.2**

Untuk setiap graph sederhana berikut, tuliskan :

Batas atas dan batas bawah untuk  $\chi'(G)$  yang diberikan oleh teorema Vinzing

Nilai sebenarnya dari  $\chi'(G)$  dan sebuah pewarnaan edge menggunakan  $\chi'(G)$  warna.

- Graph cycle  $C_7$
- Sebuah graph bipartit komplit  $K_{2,4}$
- Sebuah graph komplit  $K_6$ .

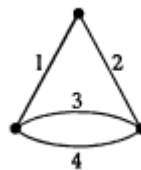
Terdapat pengembangan teorema Vinzing yang melibatkan jumlah edge maksimal yang menghubungkan pasangan vertex sebagai berikut.

**Teorema 13.2 : Teorema Vinzing (Versi Pengembangan)**

Misalkan  $G$  adalah graph yang memiliki derajat vertex maksimal  $d$ , dan misalkan  $h$  adalah jumlah edge maksimal yang menghubungkan pasangan vertex, maka berlaku :

$$d \leq \chi'(G) \leq d+h$$

Sebagai contoh, untuk graph  $G$  yang ditunjukkan di bawah, diperoleh  $d = 3$  dan  $h = 2$ , karena ada dua edge yang menghubungkan sebuah pasangan vertex. Maka batas bawah  $\chi'(G)$  adalah 3 dan batas atasnya adalah 5. Nilai  $\chi'(G)$  yang tepat untuk masalah ini adalah  $\chi'(G) = 4$ .



Gambar 13.5 Graph untuk ilustrasi teorema Vinzing dan Shannon

Batas atas lainnya untuk indeks kromatik dari graph didapatkan oleh Shannon untuk masalah pewarnaan kabel.

**Teorema 13.3 : Teorema Shannon**

Misalkan  $G$  adalah graph dengan derajat vertex maksimal  $d$ , maka berlaku :

$$d \leq \chi'(G) \leq 3d/2 \quad \text{jika } d \text{ genap}$$

$$d \leq \chi'(G) \leq (3d-1)/2 \quad \text{jika } d \text{ ganjil}$$

Sebagai contoh, untuk  $G$  di atas,  $d=3$ , dan  $(3d-1)/2=4$ . Maka batas bawah adalah 3 dan batas atas adalah 4. Nilai  $\chi'(G)$  sebenarnya adalah 4.

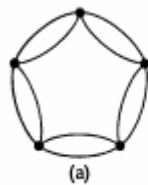
**Latihan 13.3**

Untuk setiap graph berikut, tulislah :

Batas atas dan batas bawah untuk  $\chi'(G)$  yang diberikan oleh versi pengembangan teorema Vinzing.

Batas atas dan batas bawah untuk  $\chi'(G)$  yang diberikan oleh teorema Shannon

Nilai sebenarnya dari  $\chi'(G)$  dan sebuah pewarnaan edge menggunakan  $\chi'(G)$  warna.



Kita dapat merangkum pencarian nilai batas atas dan batas bawah  $\chi'(G)$  sebagai berikut.

**Mencari indeks kromatik  $\chi'(G)$  graph  $G$  tanpa loop**

Temukan batas atas dan batas bawah yang sama, nilai  $\chi'(G)$  sama dengan nilai tersebut.

**Batas atas yang mungkin untuk  $\chi'(G)$**

- Jumlah warna pada pewarnaan  $G$  secara eksplisit
- Jumlah  $m$ , yaitu jumlah edge dari  $G$
- $d+1$ , dimana  $d$  adalah derajat vertex maksimal di  $G$ , untuk  $G$  yang tidak memiliki multiple edge (teorema Vinzing)
- $d+h$ , dimana  $d$  adalah derajat vertex maksimal dari  $G$  dan  $h$  adalah jumlah edge maksimal yang menghubungkan sepasang vertex (versi pengembangan teorema Vinzing)
- $3d/2$ , dimana  $d$  adalah derajat vertex maksimal dan  $d$  adalah genap (Teorema Shannon)
- $(3d-1)/2$ , dimana  $d$  adalah derajat vertex maksimal dan  $d$  adalah bilangan ganjil (teorema Shannon)

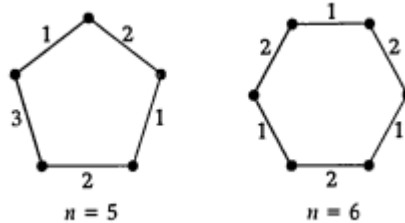
**Batas bawah yang mungkin**

$d$ , derajat vertex maksimal dari  $G$

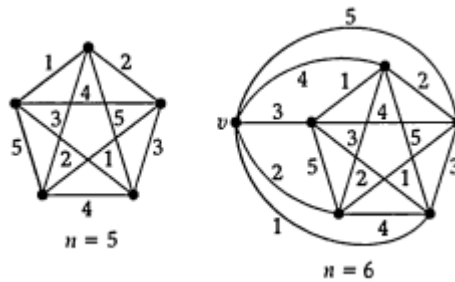
**Klasifikasi Beberapa Graph Sederhana**

Kita sekarang beralih ke masalah klasifikasi graph sederhana ke dalam dua kelas yaitu graph dengan  $\chi'(G) = d$  dan graph yang memiliki  $\chi'(G) = d + 1$ .

Untuk beberapa tipe graph, ini mudah ditunjukkan. Sebagai contoh, untuk graph cycle  $C_n$  ( $n \geq 3$ ),  $\chi'(C_n) = 2$  jika  $n$  genap, dan  $\chi'(C_n) = 3$  jika  $n$  ganjil. Sebagai contoh untuk  $n = 5$  dan  $n = 6$ , kita dapat memperoleh pewarnaan sebagai berikut.



Untuk graph Komplit  $K_5$  dan  $K_6$  kita mendapatkan pewarnaan sebagai berikut.



Secara umum, kita memperoleh teorema sebagai berikut.

**Teorema 13.4**

Untuk setiap graph Komplit  $K_n$ , berlaku :

- $\chi'(K_n) = n - 1$ , untuk  $n$  genap
- $\chi'(K_n) = n$ , untuk  $n$  ganjil

**Bukti :**

Karena setiap vertex memiliki derajat  $n - 1$ , maka menurut teorema vizing,  $\chi'(K_n)$  bernilai  $n - 1$  atau  $n$ . Jika  $n$  ganjil, maka jumlah edge maksimum yang dapat ditandai dengan warna yang sama adalah  $(n - 1) / 2$ , karena dua dari edge ini bertemu pada sebuah vertex yang sama. Namun  $K_n$  tepat memiliki  $n(n - 1) / 2$  edge, maka jumlah warna setidaknya  $n$ , sehingga  $\chi'(K_n) \geq n$ .

Kita bisa mendapatkan sebuah pewarnaan  $n$ -edge dari  $K_n$  dengan menggambar vertex dalam bentuk sebuah polygon teratur  $n$  dan mewarnai edge dari batasannya menggunakan warna yang berbeda untuk setiap edge. Setiap edge yang tersisa ditetapkan dengan warna yang sama dengan edge batasan yang paralel dengannya, oleh karena itu,  $\chi'(K_n) \leq n$ .

Mengkombinasikan pertidaksamaan, kita menyimpulkan bahwa  $\chi'(K_n) = n$ , jika  $n$  adalah ganjil.





Jika  $n$  bilangan genap, kita buktikan bahwa  $\chi'(K_n)=n-1$ , dengan secara eksplisit membangun sebuah pewarnaan  $(n-1)$ -edge dari  $K_n$ . Jika  $n=2$  maka ini mudah dibuktikan. Jika  $n>2$ , kita memilih vertex  $v$  dan menghapusnya, bersama-sama dengan edge yang insiden dengannya. Ini meninggalkan sebuah graph koplit  $K_{n-1}$  dengan jumlah vertex yang ganjil, dimana edge dapat diwarnai dengan  $n-1$  warna, dengan konstruksi sebelumnya. Pada setiap vertex ada tepat satu warna yang hilang, dan warna hilang ini semuanya berbeda. Edge  $K_n$  yang insiden dengan  $v$  dapat diwarnai menggunakan warna yang hilang tersebut. Maka  $\chi'(K_n)=n-1$ , jika  $n$  adalah genap.

**Latihan 13.4**

- Misalkan bahwa 31 tim berkompetisi di mana setiap tim harus bermain tepat satu kali melawan setiap 30 pemain lainnya. Jika tidak ada tim yang dapat bermain lebih dari satu pertandingan sehari, berapa banyak hari yang dibutuhkan?
- Apa jawaban yang sesuai untuk jumlah 32 tim, setiap tim harus bermain tepat satu kali melawan 31 tim lainnya.

Selanjutnya kita mempelajari teorema König. Dénes König adalah seorang matematikawan Hungaria yang menulis *Theorie der Endlichen und Unendlichen Graphen* (Theory of Finite and Infinite Graphs) pada tahun 1936. Teorema tersebut menunjukkan bahwa edge pada setiap graph bipartit graph (tidak perlu sederhana) dengan derajat vertex maksimal  $d$  dapat diwarnai dengan  $d$  warna.

**Teorema 13.5 : Teorema König**

Misalkan  $G$  adalah graph bipartit dengan derajat vertex maksimal  $d$ , maka berlaku  
$$\chi'(G)=d$$

**Bukti :**

Bukti dengan induksi matematika pada  $m$ , jumlah edge dari  $G$

**Langkah 1**

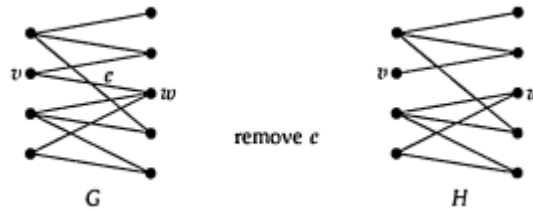
Pernyataan benar untuk  $m=1$ , untuk setiap graph bipartit  $G$  dengan satu edge  $\chi'(G)=1$  dan  $d=1$ .

**Langkah 2**

Kita asumsikan bahwa  $\chi'(G)=d$  untuk setiap graph bipartit dengan jumlah edge kurang dari  $m$ . Kita akan membuktikan bahwa  $\chi'(G)=d$  untuk semua graph bipartit dengan  $m$  edge.

Misalkan  $G$  adalah graph bipartit dengan  $m$  edge dan derajat vertex maksimal  $d$ . Misalkan  $H$  adalah graph yang diperoleh dari  $G$  dengan menghapus sebuah edge  $e$  yang insiden dengan vertex  $v$  dan  $w$ .

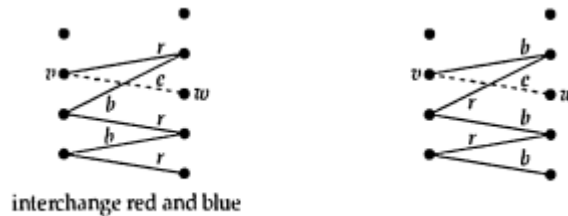




Karena  $H$  memiliki jumlah edge kurang dari  $m$  dan derajat vertex maksimal  $d$  atau kurang, menurut asumsi kita bahwa  $\chi'(H) \leq d$  maka  $H$  adalah  $d$ -edge colourable. Kita sekarang mewarnai edge pada  $H$  dengan  $d$  warna, dan menyimpan kembali edge  $e$ . Jika kita dapat mewarnai  $e$  dengan sebuah warna dari  $d$  warna, maka kita memperoleh sebuah pewarnaan  $d$ -edge seperti yang diinginkan.

Kita akan menunjukkan bahwa edge  $e$  selalu dapat diwarnai seperti ini. Karena  $H$  diperoleh dari  $G$  dengan menghapus edge  $e$ , ada setidaknya sebuah warna yang hilang di  $v$ , dan setidaknya satu warna yang hilang di  $w$ . Jika ada satu warna yang hilang di kedua  $v$  dan  $w$ , maka kita dapat mewarnai warna ini ke edge  $e$ , sehingga melengkapi pewarnaan  $d$ -edge.

Jika tidak ada warna yang hilang di kedua  $v$  dan  $w$ , misalkan warna yang hilang di  $v$  tersebut adalah biru, dan warna yang hilang di  $w$  adalah merah, tinjau path yang dimulai di  $v$  dan memuat keseluruhan edge merah dan biru. Edge pada path ini harus memilih warna, dan memilih antara vertex di kiri dan di kanan pada graph bipartit. Karena tidak ada edge berwarna biru di  $v$ , warna merah pasti muncul disini. Sehingga  $w$  tidak dapat dicapai dari  $v$  dengan path merah-biru ini, karena  $w$  harus dicapai oleh edge merah.



Kini kita mengganti warna pada path tersebut, edge biru menjadi merah dan edge merah menjadi biru. Warna yang muncul di  $w$  tidak berubah, dan warna merah sekarang hilang pada  $v$  maupun  $w$ . Kita dapat menandai edge  $e$  dengan merah, sehingga melengkapi pewarnaan  $d$ -edge.

Pernyataan ini benar untuk semua graph bipartit dengan  $m$  edge. Ini melengkapi langkah 2. Maka, dengan prinsip induksi matematika, pernyataan ini juga benar untuk semua graph bipartit dengan  $m$  edge, untuk setiap bilangan  $m$  integer positif.

**Latihan 13.5**

Gunakan teorema König untuk mencari indeks kromatik pada setiap graph berikut.

1. Graph komplit bipartit  $K_{r,s}$  ( $r$   $s$ )
2. Graph cube
3. Graph  $k$ -cube  $Q_k$ .



## Algoritma Pewarnaan Edge

Algoritma yang digunakan adalah algoritma greedy untuk pewarnaan edge sebagai berikut.

### Algoritma Greedy untuk Pewarnaan Vertex

Mulai dengan sebuah graph  $G$  dan sebuah daftar warna  $1, 2, 3, \dots$

Langkah 1 :

Labeli edge dengan  $a, b, c$  secara acak.

Langkah 2 :

Identifikasi edge yang belum diwarnai. Warnai edge tersebut dengan warna pertama pada daftar yang belum dipakai edge tetangga yang sudah diwarnai.

Ulangi langkah 2 sampai semua edge diwarnai dan berhentilah.

Sebuah pewarnaan edge dari  $G$  didapatkan. Jumlah warna yang diperoleh bergantung pada pelabelan yang dipilih untuk edge di langkah 1.

Kita mencoba dua contoh dengan sebuah graph yang sama dengan pelabelan yang berbeda.

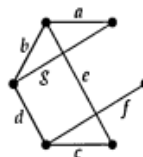
### Ilustrasi A

Temukan sebuah pewarnaan edge pada graph  $G$  berikut.



Langkah 1 :

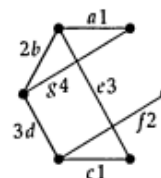
Labeli edge dengan  $a, \dots, g$  seperti sebagai berikut.



Langkah 2 :

Secara berturut-turut warnai :

- Edge  $a$  dengan warna 1
- Edge  $b$  dengan warna 2
- Edge  $c$  dengan warna 1
- Edge  $d$  dengan warna 3
- Edge  $e$  dengan warna 3
- Edge  $f$  dengan warna 2
- Edge  $g$  dengan warna 4

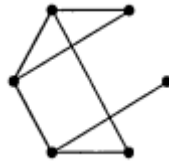


Semua edge sekarang sudah diwarnai, maka berhenti

Kita memperoleh pewarnaan 4-edge  $G$  yang ditunjukkan di atas.

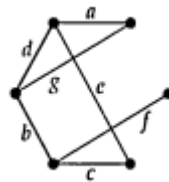
**Ilustrasi B**

Temukan sebuah pewarnaan edge dari graph  $G$  berikut.



Langkah 1 :

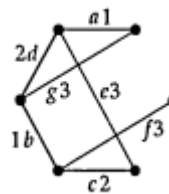
Labeli edge dengan  $a, \dots, g$  seperti sebagai berikut.



Langkah 2 :

Secara berturut-turut warnai :

- Edge  $a$  dengan warna 1
- Edge  $b$  dengan warna 1
- Edge  $c$  dengan warna 2
- Edge  $d$  dengan warna 2
- Edge  $e$  dengan warna 3
- Edge  $f$  dengan warna 3
- Edge  $g$  dengan warna 3

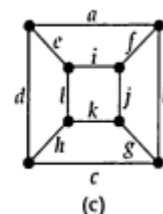
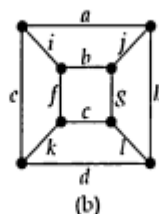
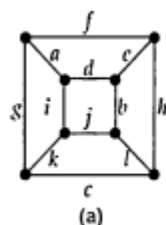


Semua edge sekarang sudah diwarnai, maka berhenti  
 Kita memperoleh pewarnaan 3-edge  $G$  yang ditunjukkan di atas.

Catat bahwa, pada contoh di atas  $\chi'(G)=3$  dan kita menemukan sebuah pewarnaan vertex  $G$  dengan tiga warna.

**Latihan 13.6**

Gunakan algoritma greedy untuk mewarnai edge pada graph  $G$  berikut, menggunakan masing-masing pelabelan berikut.



Berapa nilai  $\chi'(G)$  dari  $G$ ?

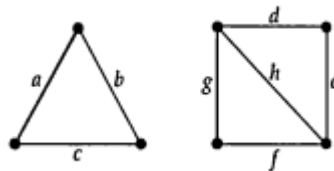
**Teorema 12.6**

Untuk setiap graph  $G$ , ada sebuah pelabelan edge untuk setiap algoritma greedy yang menghasilkan sebuah pewarnaan edge dengan  $\chi(G)$  warna.

**Dekomposisi Edge**

Beberapa permasalahan yang paling menarik dalam teori graph melibatkan dekomposisi graph  $G$  ke dalam subgraph dengan tipe tertentu. Pada beberapa masalah, kita membagi himpunan edge dari  $G$  ke dalam sub himpunan yang disjoint, ini disebut sebuah dekomposisi edge  $G$ .

Sebagai contoh, tinjau graph tidak terhubung berikut.



Sebuah dekomposisi vertex edge adalah pembagian himpunan vertex ke dalam sub himpunan edge yang disjoint yang berkoresponden dengan komponen  $G$  :

$$\{a, b, c\}, \{d, e, f, g, h\}.$$

Dekomposisi edge lainnya biasanya muncul pada graph Euler. Pada bagian sebelumnya, kita mempelajari bahwa setiap graph Euler dapat dibagi ke dalam cycle yang disjoint. Ini artinya kita dapat membagi edge di  $G$  ke dalam sub himpunan yang disjoint. Sebagai contoh, pada graph Euler  $G$  ditunjukkan sebagai berikut, ada lima dekomposisi edge dari  $G$  sebagai berikut.

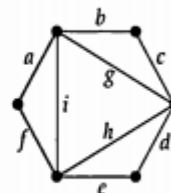
$$\{a, b, c, d, e, f\}, \{g, h, i\};$$

$$\{a, f, i\}, \{b, c, g\}, \{d, e, h\};$$

$$\{a, f, h, g\}, \{b, c, d, e, i\};$$

$$\{b, c, h, i\}, \{a, f, e, d, g\};$$

$$\{d, e, i, g\}, \{a, b, c, h, f\}.$$



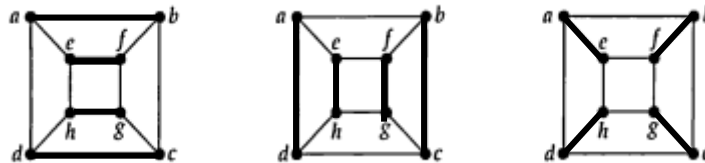
Pada bagian ini kita mengadopsi pendekatan serupa untuk beberapa masalah. Setiap masalah dapat dibangun dalam bentuk teoritis, dan melibatkan pembagian himpunan edge ke dalam sub himpunan yang disjoint dengan sifat tertentu. Dengan melakukan ini, kita mempelajari persamaan antara permasalahan yang berbeda dan dapat mengklasifikasikannya.

**Dekomposisi dalam Matching**

Gambar 13.6 menunjukkan graph cube dan tiga himpunan edge yang ditandai oleh garis yang tebal. Terdapat tiga himpunan dengan sifat bahwa setiap edge pada graph tersebut memuat edge yang tidak memiliki vertex yang sama, dan ini mengacu dekomposisi edge berikut:

$$\{ab,cd,ef,gh\}, \{ad,bc,eh,fg\}, \{ae,bf,cg,dh\}$$

Setiap himpunan yang memuat edge yang tidak memiliki vertex yang sama seperti ini di sebut *matching*.



**Gambar 13.6** Dekomposisi untuk matching

**Definisi 13.1**

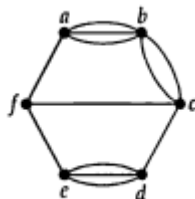
Sebuah *Matching* pada graph  $G$  adalah himpunan edge pada  $G$ , di mana tidak ada dua himpunan di antaranya yang memiliki vertex yang sama.

Setiap graph dapat di dekomposisi ke dalam matching, karena jika ada  $m$  edge, maka kita dapat membuat  $m$  matching, di mana setiap himpunan hanya memuat sebuah edge. Namun, masalah penentuan jumlah minimal matching dalam sebuah himpunan memerlukan dekomposisi yang lebih sulit dan secara umum belum terselesaikan.

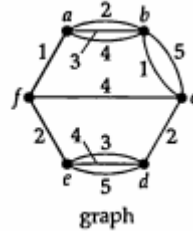
Masalah matching ini bukan hanya menjadi *interest* pada bidang akademik, namun juga muncul pada berbagai konteks. Dua diantaranya adalah masalah pewarnaan kabel dan penjadwalan. Catat bahwa masalah dekomposisi graph ke dalam jumlah minimal matching adalah sebuah masalah pewarnaan edge di mana edge pada matching ditandai dengan warna yang sama.

**Contoh 13.2 : Pewarnaan Kabel**

Misalkan sebuah panel dengan enam komponen listrik  $a..f$  dipasang dan saling berhubungan. Koneksi kabel dihubungkan dari  $a$  yang muncul melalui sebuah lubang dan dihubungkan dengan  $b$  yang muncul pada lubang yang lain. Untuk membedakan kabel yang keluar dari lubang yang sama, kabel diwarnai dengan warna berbeda.



Untuk menentukan jumlah minimum warna yang dibutuhkan untuk keseluruhan sistem, kita merepresentasikan komponen listrik dengan vertex dan kabel dengan edge. Kita menemukan bahwa lima warna dibutuhkan untuk mewarnai keseluruhan kabel pada sistem. Diagram berikut menunjukkan edge dengan warnanya masing-masing.



Dekomposisi edge yang berkoresponden dengan pewarnaan edge di atas memuat lima sub himpunan edge sebagai berikut.

$$\{af, bc\}, \{ab, cd, ef\}, \{ab, de\}, \{ab, cf, de\}, \{bc, de\}.$$

Pada masalah pewarnaan kabel, edge pada setiap warna membentuk sebuah matching, maka masalah menemukan jumlah terkecil warna kabel yang dibutuhkan sama dengan menentukan jumlah minimal matching yang dibutuhkan untuk dekomposisi graph. Dengan kata lain, ini adalah masalah dekomposisi edge pada graph di mana edge pada setiap sub himpunan membentuk matching.

Karena graph yang ditinjau pada masalah pewarnaan kabel biasanya memiliki multiple edge, kita dapat menyatakan bahwa jumlah matching dapat dibatasi oleh batasan indeks kromatik yang diberikan oleh versi pengembangan teorema Vinzing dan teorema Shannon.

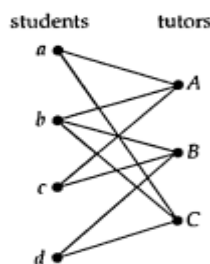
$$d \leq \chi'(G) \leq d+h \quad \text{dan} \quad d \leq \chi'(G) \leq 3d/2$$

Di mana  $d$  adalah derajat vertex minimal pada graph  $G$  dan  $h$  adalah jumlah maksimal edge yang menghubungkan pasangan vertex.

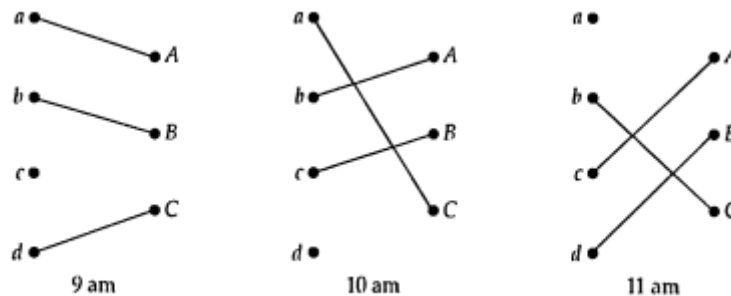
**Contoh 13.2 : Penjadwalan Ujian**

Pada akhir tahun ajaran, semua siswa akan melaksanakan ujian dengan masing-masing tutornya. Pada permasalahan ini, satu tutor hanya menguji satu siswa dalam satu waktu. Berapa banyak periode waktu yang diperlukan untuk ujian ini?

Untuk melihat pemecahannya, tinjau sebuah contoh sederhana dari empat siswa  $a, b, c, d$  dan tiga tutor  $A, B, C$ . Kita representasikan murid dan tutor sebagai vertex pada graph bipartit, dan terdapat garis antara murid dengan tutor jika terdapat ujian murid oleh tutor tersebut. Contoh graph ditunjukkan pada graph berikut.



Terdapat dua edge yang bertemu pada vertex yang sama, sehingga ujian tidak dapat dilakukan bersama. Masalah ini adalah masalah dekomposisi edge di mana kita harus membagi graph ke dalam sub himpunan di mana tidak ada edge yang bertemu. Masalah ini juga merupakan masalah mencari jumlah minimal matching. Jumlah minimal matching yang menjadi solusi dalam ini adalah 3, dan pasangan waktu yang sesuai ditunjukkan sebagai berikut.



Dekomposisi edge yang sesuai adalah sebagai berikut :

$$\{aA, bB, dC\}, \{aC, bA, cB\}, \{bC, cA, dB\}$$

Masalah ini juga dapat dinyatakan sebagai masalah pewarnaan. Jika kita mewarnai jam 9 am sebagai edge merah, jam 10 am sebagai edge kuning dan jam 11 sebagai edge biru, maka warna yang muncul pada setiap vertex (murid atau tutor) adalah berbeda. Semua edge yang memiliki warna yang sama membentuk matching.

Pada masalah tipe penjadwalan di atas, graph yang ditinjau adalah graph bipartit. Masalah ini merupakan masalah penentuan indeks kromatik pada graph bipartit dan masalah ini dijawab oleh teorema Koenig, yaitu jumlah terkecil yang dibutuhkan sama dengan derajat vertex terbesar pada graph bipartit.

### Latihan 13.7

Lima siswa  $a, \dots, e$ , akan diuji oleh lima tutor  $A, \dots, E$ .

Tutor  $A$  akan menguji siswa  $b$  dan  $d$ .

Tutor  $B$  akan menguji siswa  $a, b$  dan  $e$ .

Tutor  $C$  akan menguji siswa  $b, c$  dan  $e$ .

Tutor  $D$  akan menguji siswa  $a$  dan  $c$ .

Tutor  $E$  akan menguji siswa  $b, d$  dan  $e$ .

Setiap tutor hanya menguji satu siswa dalam satu waktu dan ujian membutuhkan waktu yang sama. Temukan jumlah periode minimal yang dibutuhkan, dan tentukan jadwal yang sesuai.



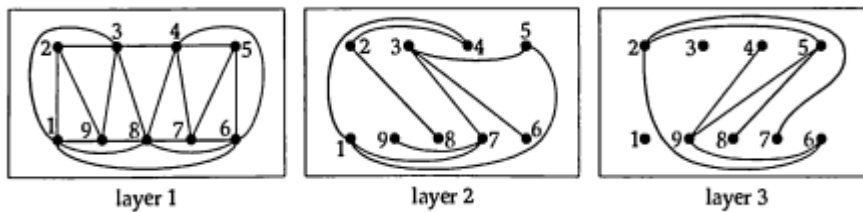
**Dekomposisi Pada Subgraph Planar**

**Masalah Papan Sirkuit**

Pada papan sirkuit, komponen elektronik dihubungkan dengan jalur yang dicetak pada papan. Penghubung ini tidak boleh saling memotong, karena dapat menimbulkan kontak listrik yang tidak diinginkan. Sirkuit di mana banyak perpotongan tidak dapat dihindarkan dapat dicetak pada beberapa papan. Setiap papan memuat sebuah sirkuit cetak tanpa perpotongan. Berapa jumlah minimal lapisan papan yang dibutuhkan untuk membuat sirkuit ini?

**Masalah 13.1 : Sirkuit Cetak**

Terdapat sebuah sirkuit cetak yang memiliki 36 koneksi dan direpresentasikan dengan graph komplit  $K_9$ . Kita tidak mungkin menyusun semua interkoneksi pada sebuah lapisan atau bahkan dua lapisan. Pada masalah ini, tiga lapisan dibutuhkan, dan solusinya diberikan pada Gambar 13.7. Catat bahwa edge pada  $K_9$  dimasukkan pada salah satu lapisan. Sebagai contoh, edge 28 muncul pada lapisan 2 dan edge 69 muncul pada lapisan 3.



**Gambar 13.7** Tiga lapisan sirkuit cetak untuk membentuk  $K_9$

Setiap tiga graph yang terbentuk dari setiap lapisan adalah graph planar. Maka masalah sirkuit cetak ini adalah masalah dekomposisi graph ke dalam graph yang lebih kecil, di mana setiap subgraphnya merupakan graph planar. Dengan kata lain, ini adalah masalah dekomposisi edge di mana edge pada setiap sub himpunan membentuk sebuah graph planar. Pada kasus  $K_9$ , kita memperoleh dekomposisi edge yang sesuai pada tiga lapisan sebagai berikut.

- {12, 13, 16, 18, 19, 23, 29, 34, 38, 39, 45, 46, 47, 48, 56, 57, 67, 68, 78, 89},
- {14, 15, 17, 24, 28, 35, 36, 37, 79},
- {25, 26, 27, 49, 58, 59, 69}.

Ide membagi graph ke dalam subgraph yang planar mengantarkan kita pada definisi *thickness* pada graph  $G$  yang dinotasikan dengan  $t(G)$ . *Thickness* adalah jumlah minimal graph planar yang dapat disatukan untuk membentuk graph  $G$ . Sebagai contoh, *thickness* pada setiap graph planar adalah 1, dan *thickness* pada graph komplit  $K_9$  adalah 3.

**Latihan 13.8**

Tentukan *thickness* pada setiap graph berikut.

- a. Graph komplit  $K_5$
- b. Graph komplit bipartit  $K_{3,3}$
- c. Graph Petersen

Secara umum, tidak diketahui rumus yang menentukan *thickness* sebuah graph  $G$ . Namun, kita dapat dengan mudah memperoleh sebuah batas bawah dari  $t(G)$  yang biasanya juga merupakan nilai yang sama dengan nilai sebenarnya. Kita membatasi perhatian kita pada graph sederhana, karena kita dapat mengganti multiple edge dengan sebuah edge dan menghapus loop, seperti yang kita lakukan sebelumnya.

Misalkan  $a$  adalah bilangan positif. Maka  $\lfloor a \rfloor$  adalah bilangan integer yang diperoleh dari pembulatan ke bawah  $a$ , dan  $\lceil a \rceil$  adalah bilangan integer yang diperoleh dari pembulatan ke atas  $a$ . sebagai contoh :

Suatu hubungan antara fungsi ini diberikan oleh,

$$\lceil a/b \rceil = \lfloor (a/b) + (b-1)/b \rfloor;$$

Sebagai contoh ,

$$\lceil 7/5 \rceil = \lfloor 7/5 + 4/5 \rfloor = \lfloor 11/5 \rfloor = 2$$

Catat bahwa jika  $a$  adalah sebuah integer maka  $\lfloor a \rfloor = \lceil a \rceil = a$ .

**Teorema 13.7**

Misalkan  $G$  adalah graph sederhana terhubung dengan  $n \geq 3$  vertex dan  $m$  edge, maka:

- $t(G) \geq \lfloor m/(3n-6) \rfloor$
- $t(G) \geq \lfloor m/(2n-4) \rfloor$ , jika  $G$  tidak memiliki segitiga.

**Bukti :**

- Dengan akibat 11.1, jumlah edge pada graph planar terhubung dengan  $n \geq 3$  vertex dan  $m$  edge paling banyak  $3n-6$ , maka jumlah edge pada setiap lapisan  $G$  paling banyak  $3n-6$ . Karena terdapat  $m$  edge, jumlah graph planar harus setidaknya  $m/(3n-6)$ . Maka jumlah graph planar adalah sebuah integer, maka  $t(G) \geq \lfloor m/(3n-6) \rfloor$
- Menurut akibat 11.2, jumlah edge pada graph sederhana terhubung dengan  $n \geq 3$  vertex dan  $m$  edge serta tidak terdapat segitiga, paling banyak  $3n-6$ . Karena terdapat  $m$  edge, jumlah graph planar harus setidaknya  $m/(2n-4)$ . Maka jumlah graph planar adalah sebuah integer, maka  $t(G) \geq \lfloor m/(2n-4) \rfloor$

Selanjutnya kita mempelajari batasan bawah untuk *thickness*  $K_n$  dan  $K_{r,s}$

**Teorema 13.8**

- $t(K_n) \geq \lfloor (n+7)/6 \rfloor$ ;
- $t(K_{r,s}) \geq \lfloor rs/(2r+2s-4) \rfloor$ ;

**Bukti**

- a. Jika  $G = K_n$ , maka  $m = \frac{1}{2}n(n-1)$ .

Menurut Teorema 13.7 bahwa  $t(K_n) \geq \lceil \frac{1}{2}n(n-1)/(3n-6) \rceil$ ;

Menggunakan  $\lceil a/b \rceil = \lfloor (a+b-1)/b \rfloor$ , kita dapat menulis persamaan sebagai berikut

$$\begin{aligned} & \lceil \frac{1}{2}n(n-1)/(3n-6) \rceil \\ &= \lfloor \frac{1}{2}n(n-1)/(3n-6) + (3n-7)/(3n-6) \rfloor \\ &= \lfloor (\frac{1}{2}(n^2-n) + (3n-7))/(3n-6) \rfloor \\ &= \lfloor ((n^2-n) + (6n-14))/2(3n-6) \rfloor \\ &= \lfloor (n^2+5n-14)/2(3n-6) \rfloor \\ &= \lfloor (n+7)(n-2)/6(n-2) \rfloor \\ &= \lfloor (n+7)/6 \rfloor \end{aligned}$$

maka,

$$t(K_n) \geq \lfloor (n+7)/6 \rfloor$$

- b. Jika  $G = K_{r,s}$ , maka  $m = rs$  dan  $G$  tidak mempunyai segitiga.

Menurut Teorema 13.7 bahwa  $t(K_{r,s}) \geq \lceil m/(2n-4) \rceil = \lceil rs/(2(r+s)-4) \rceil$ ;

Maka

$$t(K_{r,s}) \geq \lceil rs/(2r+2s-4) \rceil$$

Dapat dibuktikan bahwa  $t(K_n) \geq \lfloor (n+7)/6 \rfloor$  untuk semua  $n$ , kecuali untuk  $n=9$  dan  $n=10$ , karena  $t(K_n)=3$ .

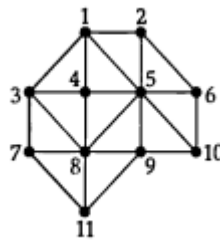
Tidak diketahui apakah pertidaksamaan pada bagian  $b$  juga berlaku untuk semua  $r$  dan  $s$ , namun untuk semua graph komplit bipartit untuk vertex kurang dari 48 ini berlaku. Jadi, walaupun kita tidak dapat memecahkan masalah papan sirkuit secara umum, kita bisa mendapatkan batas bawah dari solusi dan batasan ini sering menjadi nilai solusi sebenarnya.

**Dekomposisi Pada Spanning Subgraph****Masalah rute bus**

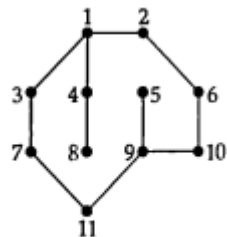
Pada beberapa tempat terdapat persaingan perusahaan bus. Setiap perusahaan ingin memberikan pelayanan yang memuat setiap kota, di mana penumpang dapat menggunakan perusahaan untuk bepergian dari satu kota ke kota lain. Namun, pemerintahan kota tidak akan mengizinkan perusahaan berbeda untuk beroperasi sepanjang jalan yang sama. Berapa banyak perusahaan yang dapat diakomodasi oleh pemerintah pada suatu area tertentu? Kita menyelesaikan masalah ini dengan menggambar sebuah graph di mana vertex mewakili kota dan edge mewakili jalan yang menghubungkan kota.

**Contoh 13.4 : Rute Bus**

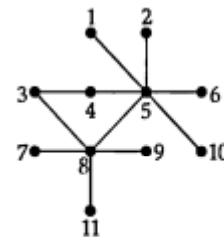
Graph berikut merepresentasikan sebuah daerah yang memiliki 11 kota dengan 22 jalan.



Setiap perusahaan bus memerlukan sebuah jaringan untuk menghubungkan 11 kota, maka setiap perusahaan harus ditandai untuk setidaknya 10 interkoneksi jalan. Karena hanya ada 22 jalan, maka jumlah maksimal perusahaan yang dapat diakomodasi adalah 2. Diagram berikut menunjukkan alokasi jalan pada dua perusahaan.



Red Devil bus company



Purple Peril bus company

Alokasi jalan pada perusahaan seperti ini menghasilkan sebuah dekomposisi edge dari graph asal. Setiap subgraph harus memuat edge yang insiden dengan semua vertex dan harus terhubung, sehingga penumpang dapat bepergian dari satu kota ke kota lain dengan bus-bus pada setiap perusahaan. Masalah ini merupakan masalah dekomposisi graph ke dalam jumlah maksimal subgraph terhubung, setiapnya subgraph memuat setiap vertex pada graph, subgraph seperti ini disebut *spanning subgraph*.

Kita menotasikan jumlah spanning subgraph pada  $G$  dengan  $s(G)$ . Sebuah teori untuk jumlah  $s(G)$  dikemukakan oleh W.T. Tutte yang membuktikannya pada tahun 1961.

**Teorema 13.9**

Misalkan  $G$  adalah graph terhubung dengan  $n$  vertex. Jumlah spanning subgraph  $s(G)$  adalah bilangan integer terbesar yang memenuhi pernyataan berikut.

Untuk setiap bilangan integer positif  $k \leq n$ , setidaknya  $(k-1) \times s(G)$  edge harus dihapus untuk membuat graph  $G$  tidak terhubung ke dalam  $k$  komponen.

Kita mengilustrasikan teori tersebut untuk  $s(G)=2$  pada graph  $G$  yang ditunjukkan pada contoh 13.4 berikut.

Untuk membuat  $G$  tidak terhubung ke dalam :

- 2 komponen, kita harus menghapus setidaknya 3 edge, maka :

$$s(G) \leq 3/(2-1) = 3$$

- 3 komponen, kita harus menghapus setidaknya 5 edge, maka :

$$s(G) \leq 5/(3-1) = 5/2$$

- 4 komponen, kita harus menghapus setidaknya 7 edge, maka :

$$s(G) \leq 7/(4-1) = 7/3$$

...

- 11 komponen, kita harus menghapus setidaknya 22 edge, maka :

$$s(G) \leq 22/(11-1) = 22/10$$

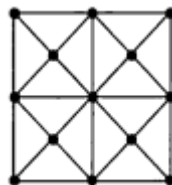
Bilangan integer  $s(G)$  terbesar yang memenuhi semua pertidaksamaan ini adalah 2.

Bukti formal pada hasil tersebut secara garis besar adalah dengan mengasumsikan graph  $G$  telah tidak terhubung ke dalam  $k$  komponen dengan menghapus  $r$  edge. Untuk membuatnya terhubung, setiap perusahaan bus harus memiliki setidaknya  $k-1$  edge yang terhubung antara komponen yang beragam. Maka jika ada  $s(G)$  perusahaan bus, maka

$$r \geq (k-1) \times s(G)$$

### Latihan 13.9

Temukan nilai dari  $s(G)$  untuk jaringan jalan  $G$  berikut.

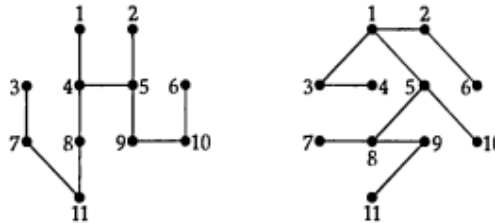


### Dekomposisi menjadi spanning tree

Variasi masalah rute bus di atas adalah, misalkan setiap perusahaan bus beroperasi dari sebuah depot di sebuah kota dan setiap rutenya merupakan sebuah path yang ke luar dari suatu vertex, dan kembali dengan jalan yang sama. Setiap subgraph terhubung yang terbentuk merupakan sebuah tree. Dengan kata lain, graph dapat di dekomposisi menjadi spanning tree. Dekomposisi ini hanya mungkin dilakukan jika hanya jika, jumlah edge pada graph adalah hasil perkalian dari jumlah edge pada spanning tree, jika graph memiliki  $n$  vertex, dan  $m$  edge, maka  $m$  harus merupakan hasil perkalian dari  $(n-1)$ .

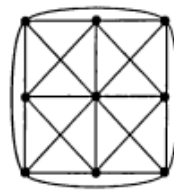
**Contoh 13.5 : Rute Bus + Variasi**

Pada contoh 13.4,  $m=22$  bukanlah hasil perkalian dari 10, ( $n-1=10$ ). Graph ini hanya dapat didekomposisi ke dalam spanning tree, jika hanya jika dua jalan tidak dipakai. Sebagai contoh, jika jalan 3-8 dan 5-6 dihapus dari graph, graph yang dihasilkan dapat di dekomposisi ke dalam spanning sebagai berikut.



**Latihan 13.10**

Dekomposisi graph berikut ke dalam spanning tree yang disjoint.



Teorema berikut memberikan syarat perlu dan syarat cukup untuk keberadaan sebuah dekomposisi ke dalam spanning tree.

**Teorema 13.10**

Misalkan  $G$  adalah graph terhubung dengan  $n$  vertex dan  $s(n-1)$  edge, maka  $G$  dapat di dekomposisi ke dalam  $s$  spanning tree jika hanya jika :

Untuk setiap bilangan integer positif  $k \leq n$ , setidaknya  $(k-1) \times s$  edge harus dihapus untuk membuat  $G$  tidak terhubung ke dalam  $k$  komponen.

**Bukti :**

Dengan teorema 13.9, menegaskan bahwa  $G$  dapat di dekomposisi ke dalam  $s$  spanning tree jika hanya jika  $s=s(G)$ . Namun jika  $G$  dapat di dekomposisi ke dalam  $s$  subgraph terhubung setiapnya memuat setiap vertex dalam graph, maka setiap subgraph harus memiliki  $n-1$  edge, dan ini menjadi sebuah spanning tree, karena tidak ada edge yang menciptakan sebuah cycle.

Kita mendapatkan solusi jumlah maksimal perusahaan bus yang dapat diakomodasi pada tipe permasalahan pertama, dan kita memiliki sebuah syarat perlu dan syarat cukup untuk tipe permasalahan kedua.