



---

# **PENGANTAR TEORI GRAF**

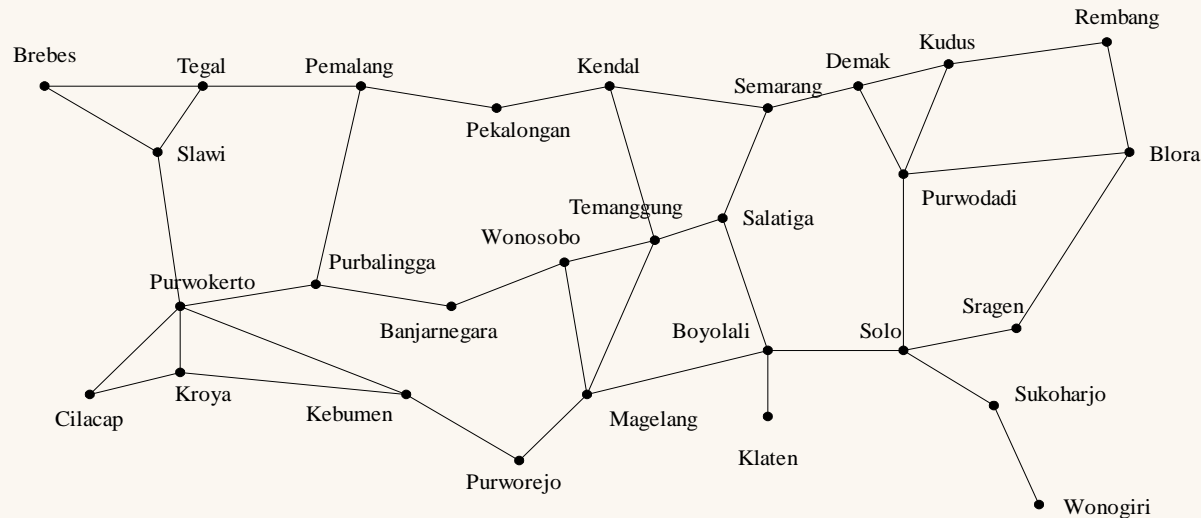
---

Egi Safitri

**Institut Informatika dan Bisnis  
Darmajaya**

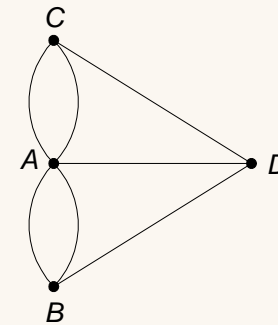
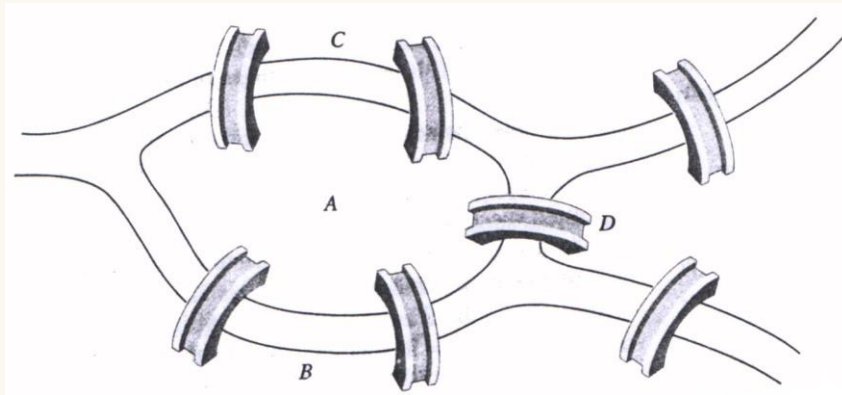
# Pendahuluan

- Graf digunakan untuk merepresentasikan objek-objek diskrit dan hubungan antara objek-objek tersebut.
- Gambar di bawah ini sebuah graf yang menyatakan peta jaringan jalan raya yang menghubungkan sejumlah kota di Provinsi Jawa Tengah.



# Sejarah Graf

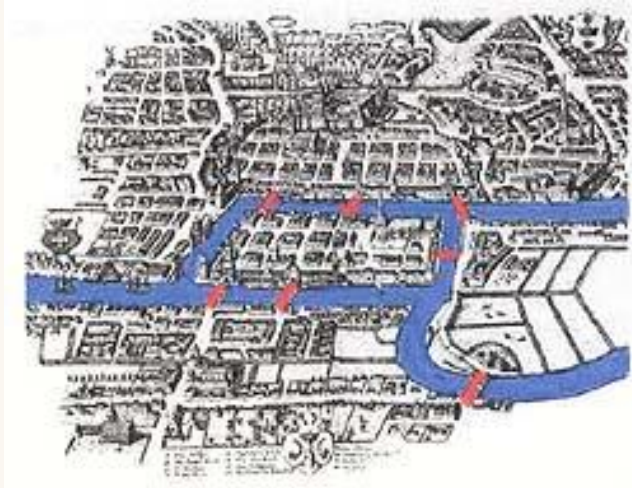
- Sejarah Graf: masalah jembatan Königsberg (tahun 1736)



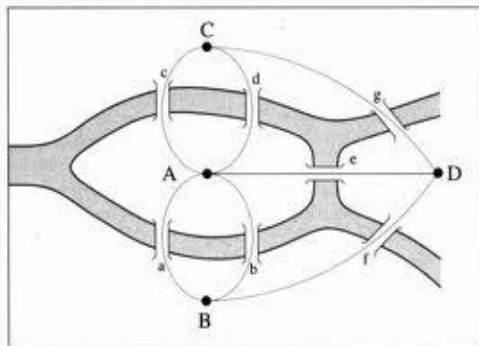
**Gambar 1.** Masalah Jembatan Königsberg

- Graf yang merepresentasikan jembatan Königsberg:
  - Simpul (*vertex*) → menyatakan daratan
  - Sisi (*edge*) → menyatakan jembatan
- Bisakah melalui setiap jembatan tepat sekali dan kembali lagi ke tempat semula?

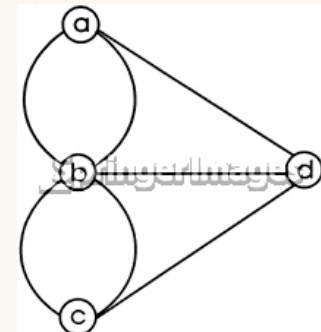
# Sejarah Graf



Königsberg  
Bridge Problem



Leonhard Euler  
15 April 1707 – 18  
September 1783



# Pendahuluan

---

- **Graf** adalah **diagram** yang digunakan untuk menggambarkan berbagai struktur yang ada.
- **Contoh :**  
Struktur Organisasi, Peta, Diagram Rangkaian Listrik.
- **Tujuan :**  
Sebagai visualisasi objek-objeknya agar mudah dimengerti.

# Dasar-Dasar Graf (1)

---

- Suatu Graf terdiri dari 2 himpunan yang berhingga, yaitu **himpunan titik-titik** tak kosong (simbol  **$V(G)$** ) dan **himpunan garis-garis** (simbol  **$E(G)$** ).
- Setiap garis berhubungan dengan satu atau dua titik. Titik-titik tersebut disebut **Titik Ujung**.
- Garis yang berhubungan dengan satu titik disebut **Loop**.

## Dasar-Dasar Graf (2)

---

- Dua garis yang menghubungkan titik yang sama disebut **Garis Paralel**.
- Dua titik dikatakan **berhubungan** bila ada garis yang menghubungkan keduanya.
- Titik yang tidak punya garis yang berhubungan dengannya disebut **Titik Terasing**.

## Dasar-Dasar Graf (3)

---

- **Graf Kosong** adalah graf yang tidak punya titik dan garis.
- **Graf Berarah** adalah graf yang semua garisnya memiliki arah (*Directed Graph / Digraph*).
- **Graf Tak Berarah** adalah graf yang semua garisnya tidak memiliki arah.

# Definisi Graf

---

Graf  $G = (V, E)$ , yang dalam hal ini:

- $V$  = himpunan tidak-kosong dari simpul-simpul (*vertices*)
- $= \{ v_1, v_2, \dots, v_n \}$
- $E$  = himpunan sisi (*edges*) yang menghubungkan sepasang simpul
- $= \{ e_1, e_2, \dots, e_n \}$

## Contoh 1.

---

- Ada 7 kota (A,...,G) yang diantaranya dihubungkan langsung dengan jalan darat. Hubungan antar kota didefinisikan sebagai berikut :

A terhubung dengan B dan D

B terhubung dengan D

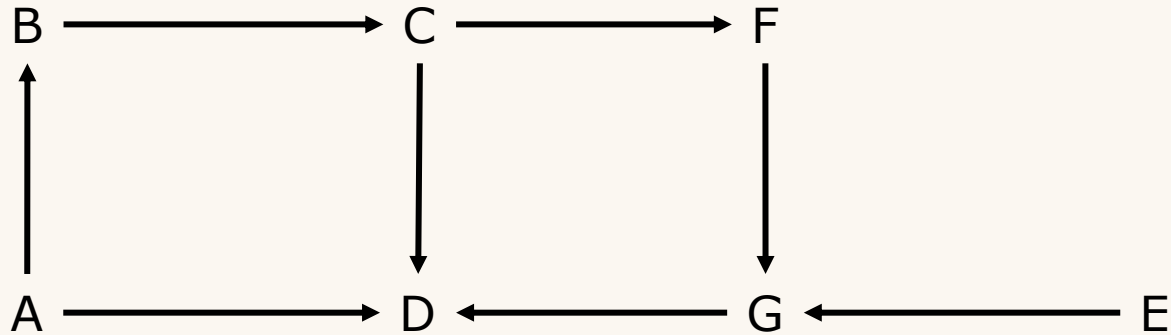
C terhubung dengan B

E terhubung dengan F

Buatlah graf yang menunjukkan keadaan transportasi di 7 kota tersebut !

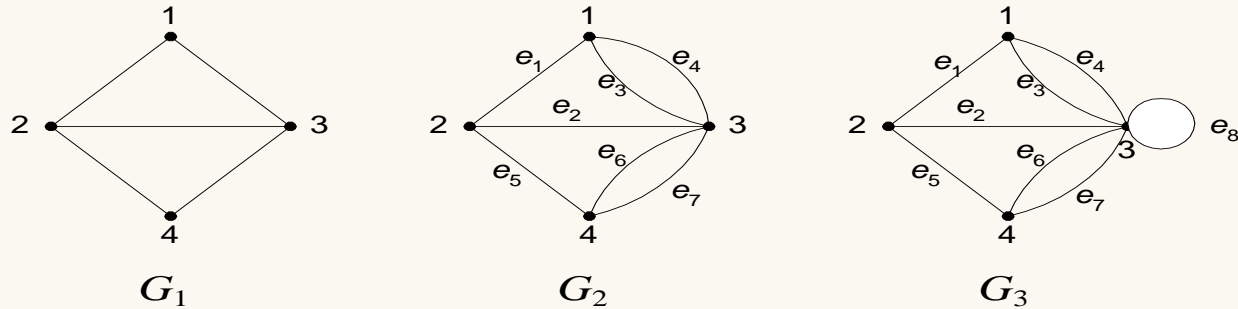
# Contoh 1.

---



Di sini, panah menunjukkan arah hubungan antar kota. Sebagai contoh, A terhubung dengan B dan D, B terhubung dengan D, dan seterusnya. Jadi, graf di atas mencerminkan hubungan antar kota sesuai dengan informasi yang diberikan.

# Contoh 2.



**Gambar 2.** (a) graf sederhana, (b) graf ganda, dan (c) graf semu

**Contoh 1.** Pada Gambar 2,  $G_1$  adalah graf dengan

$$V = \{ 1, 2, 3, 4 \} \quad E = \{ (1, 2), (1, 3), (2, 3), (2, 4), (3, 4) \}$$

$G_2$  adalah graf dengan

$$V = \{ 1, 2, 3, 4 \}$$

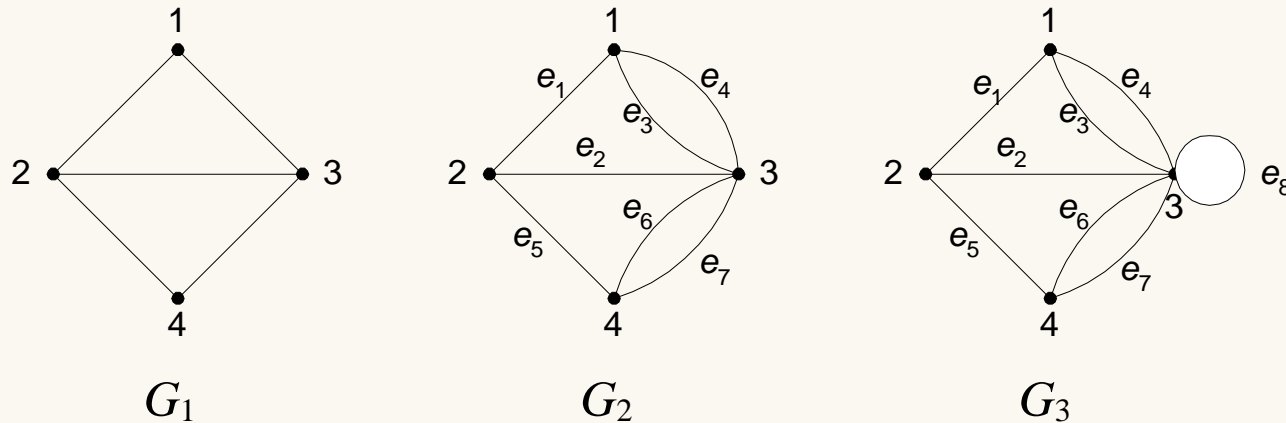
$$E = \{ (1, 2), (2, 3), (1, 3), (1, 3), (2, 4), (3, 4), (3, 4) \} \\ = \{ e_1, e_2, e_3, e_4, e_5, e_6, e_7 \}$$

$G_3$  adalah graf dengan

$$V = \{ 1, 2, 3, 4 \}$$

$$E = \{ (1, 2), (2, 3), (1, 3), (1, 3), (2, 4), (3, 4), (3, 4), (3, 3) \} \\ = \{ e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8 \}$$

## Contoh 2.



**Gambar 2.** (a) graf sederhana, (b) graf ganda, dan (c) graf semu

- Pada  $G_2$ , sisi  $e_3 = (1, 3)$  dan sisi  $e_4 = (1, 3)$  dinamakan **sisi-ganda** (*multiple edges* atau *parallel edges*) karena kedua sisi ini menghubungkan dua buah simpul yang sama, yaitu simpul 1 dan simpul 3.
- Pada  $G_3$ , sisi  $e_8 = (3, 3)$  dinamakan **gelang** atau **kalang** (*loop*) karena ia berawal dan berakhir pada simpul yang sama.

## Contoh 2.

---

- Gambarlah graf dengan titik-titik dan garis berikut :

$$V(G) = \{ v1, v2, v3, v4 \}$$

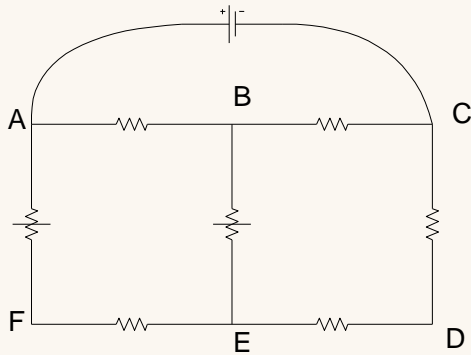
$$E(G) = \{ e1, e2, e3, e4, e5 \}$$

Titik-titik ujung garis adalah :

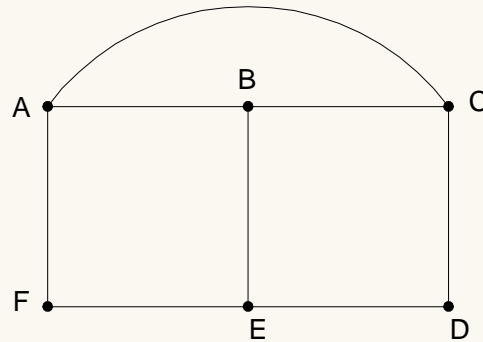
Garis	Titik Ujung
e1	{v1, v3}
e2	{v2, v4}
e3	{v1}
e4	{v2, v4}
e5	{v3}

# Contoh Terapan Graf

## 1. Rangkaian listrik.



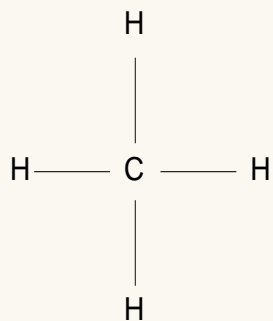
(a)



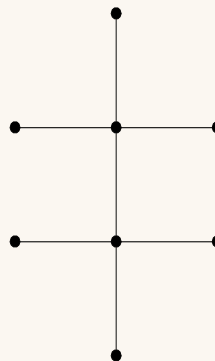
(b)

## 2. *Isomer senyawa kimia karbon*

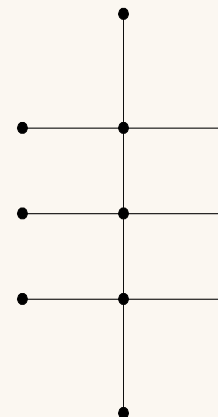
metana (CH<sub>4</sub>)



etana (C<sub>2</sub>H<sub>6</sub>)



propana (C<sub>3</sub>H<sub>8</sub>)



# Graf Tak Berarah

---

- **Graf Sederhana** adalah graf yang tidak memiliki Loop ataupun Garis Paralel.

## Contoh 3.

- Gambarkan semua graf sederhana yang dapat dibentuk dari 4 titik  $\{a,b,c,d\}$  dan 2 garis !

# Graf Tak Berarah

---

- **Graf Lengkap** dengan **n** titik (simbol  **$K_n$** ) adalah graf sederhana dengan **n** titik di mana **setiap** 2 titik yang berbeda **selalu** dihubungkan dengan suatu garis.
- **Banyaknya garis** dalam suatu graf lengkap dengan **n** titik adalah

$$\frac{n(n-1)}{2} \quad \text{buah}$$

# Graf Tak Berarah

---

- **Graf Bipartite** adalah graf  $G$  yang himp. titiknya/ $V(G)$  dapat dibagi menjadi 2 himp yaitu  $V_a$  dan  $V_b$ .
- Setiap garis dalam  $G$  menghubungkan titik di  $V_a$  dengan titik di  $V_b$ .
- Semua titik dalam  $V_a$  atau  $V_b$  tidak saling berhubungan.
- Apabila ***setiap*** titik di  $V_a$  berhubungan dengan ***setiap*** titik di  $V_b$  maka disebut **Graf Bipartite Lengkap**.

# Komplemen Graf

---

- **Komplemen suatu graf  $G$**  (simbol  $\overline{G}$ ) dengan  $n$  titik adalah suatu graf dengan :
  1. Titik-titik  $\overline{G}$  sama dengan titik-titik  $G$ .
  2. Garis-garis  $\overline{G}$  adalah komplemen garis-garis  $G$  terhadap Graf Lengkapnya ( $K_n$ )
- Titik-titik yang dihubungkan dengan garis pada  $G$  menjadi tidak terhubung dalam  $\overline{G}$
- Sebaliknya, titik-titik yang tidak terhubung pada  $G$  menjadi terhubung dalam  $\overline{G}$

# Sub Graf

---

- Misalkan  $G$  adalah graf. Graf  $H$  dikatakan subgraf dari  $G$  bila dan hanya bila :
  1.  $V(H) \subseteq V(G)$
  2.  $E(H) \subseteq E(G)$
  3. **Setiap garis dalam  $H$  memiliki titik ujung yang sama dengan garis tersebut dalam  $G$**

# Derajat

---

- Misal titik  $v$  adalah suatu titik dalam graf  $G$ . **Derajat titik  $v$**  (simbol  $d(v)$ ) adalah *jumlah garis yang berhubungan dengan titik  $v$ .*
- Derajat titik yang berhubungan dengan sebuah loop adalah **2**.
- **Derajat total** suatu graf  $G$  adalah **jumlah derajat semua titik dalam  $G$ .**
- **Derajat total** suatu graf selalu **genap**.
- Dalam sembarang graf **jumlah titik yang berderajat ganjil selalu genap**.

# Path dan Sirkuit (1)

---

Misalkan  $G$  adalah suatu graf,  $v_0$  dan  $v_n$  adalah 2 titik di dalam  $G$ .

- **Walk** dari titik  $v_0$  ke titik  $v_n$  adalah barisan titik-titik berhubungan dan garis secara berselang-seling diawali dari titik  $v_0$  dan diakhiri pada titik  $v_n$ .
- **Path** dari titik  $v_0$  ke titik  $v_n$  adalah walk dari titik  $v_0$  ke titik  $v_n$  yang semua garisnya berbeda.
- **Panjang** walk atau path = jumlah garis yang dilalui

## Path dan Sirkuit (2)

---

- ***Path sederhana*** dari titik  $v_0$  ke titik  $v_n$  adalah path dari titik  $v_0$  ke titik  $v_n$  yang semua titiknya berbeda.
- ***Sirkuit*** adalah path yang dimulai dan diakhiri pada titik yang sama.
- ***Sirkuit sederhana*** adalah sirkuit semua titiknya berbeda kecuali untuk titik awal dan titik akhir.

# Sirkuit Euler (1)

---

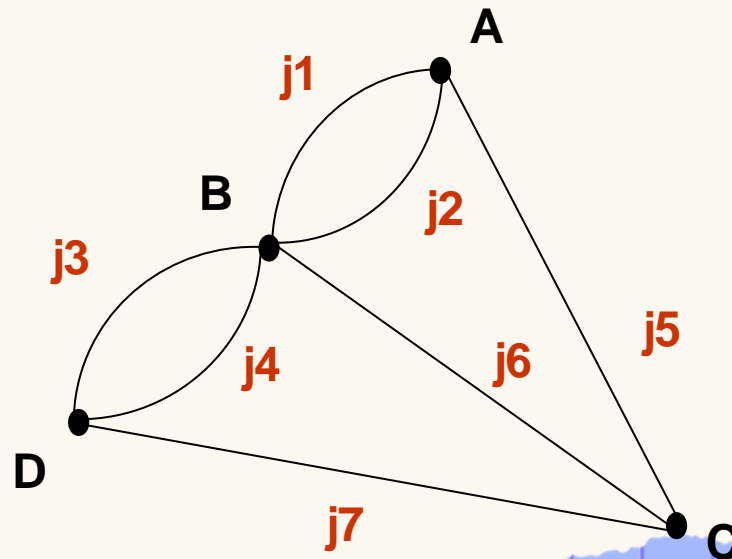
- ***Sirkuit Euler*** adalah sirkuit di mana **setiap titik** dalam graf  $G$  muncul **paling sedikit** satu kali dan **setiap garis** muncul **tepat** satu kali.

# Sirkuit Euler (2)

## ○ **Latar Belakang :**

Masalah 7 Jembatan yang menghubungkan 4 kota.

Apakah mungkin seseorang berjalan mengunjungi kota yang dimulai dan diakhiri pada tempat yang sama dengan melintasi 7 jembatan masing-masing tepat satu kali ?



# Teorema

---

- Graf  $G$  memiliki **Sirkuit Euler** bila dan hanya bila  $G$  adalah **graf yang terhubung** dan **semua titik dalam  $G$  mempunyai derajat genap**.

# Graf Terhubung dan Tidak Terhubung

---

Misalkan  $G$  adalah suatu graf

- 2 titik dalam  $G$ ,  $v_1$  dg  $v_2$  **terhubung** bila ada **walk** dari  $v_1$  ke  $v_2$ .
- Graf  $G$  dikatakan
  - ***Terhubung***  $\Leftrightarrow$  **setiap 2 titik** dalam  $G$  **terhubung**.
  - ***Tidak terhubung***  $\Leftrightarrow$  **ada 2 titik** dalam  $G$  yang **tidak terhubung**.

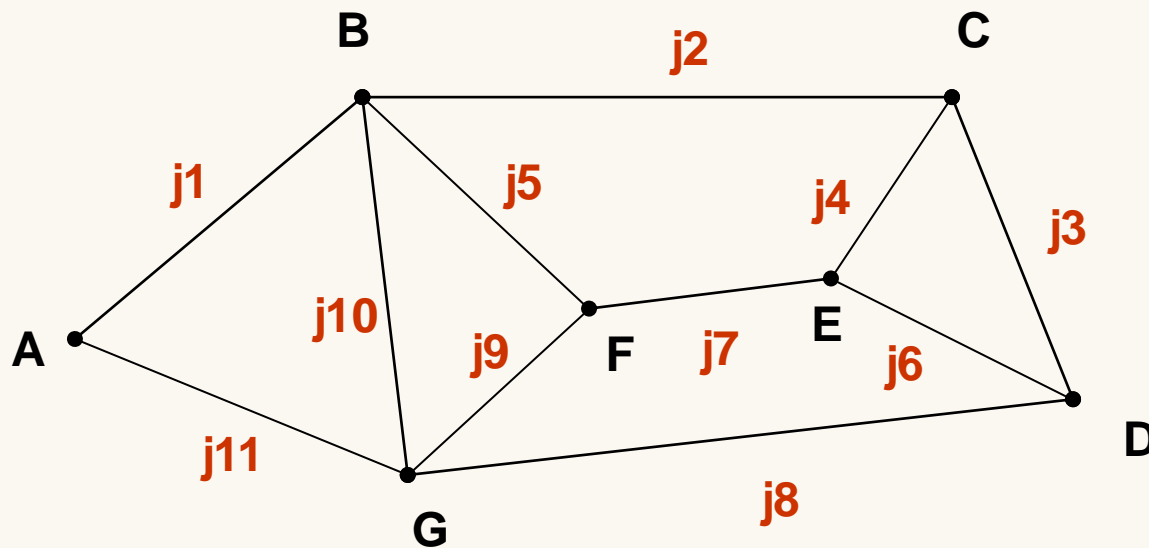
# Sirkuit Hamilton

---

- Suatu **graf terhubung  $G$**  memiliki **Sirkuit Hamilton** bila ada sirkuit yang mengunjungi setiap titiknya **tepat satu kali** (kecuali titik awal dan titik akhir).

# Contoh

- Gambar di bawah menyatakan peta kota A..G dan jalan-jalan yang menghubungkan kota-kota tsb. Seorang salesman akan mengunjungi tiap kota masing-masing 1 kali dari kota A kembali lagi ke kota A. Carilah rute perjalanan yang harus dilalui salesman tsb !



# Sirkuit Hamilton vs Euler

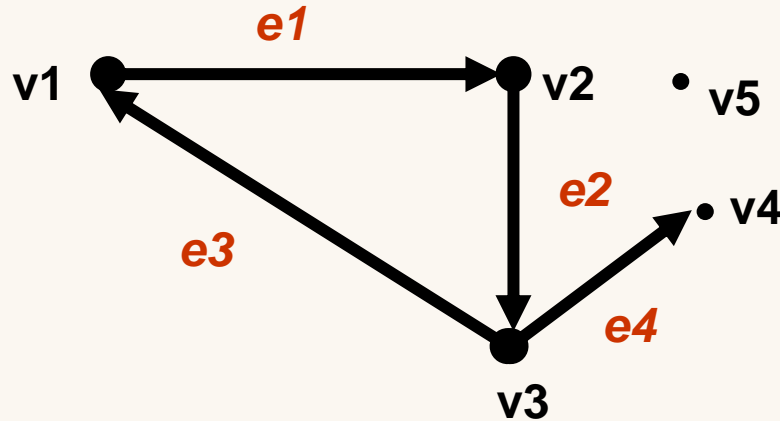
---

- Perbedaan Sirkuit Euler dengan Sirkuit Hamilton :
  - Dalam Sirkuit Euler **semua garis** harus dilalui **tepat satu kali**, sedangkan **semua titiknya** boleh dikunjungi **lebih dari sekali**.
  - Dalam Sirkuit Hamilton **semua titiknya** harus dikunjungi **tepat satu kali** dan **tidak harus melalui semua garis**.

# Graf Berarah (Digraph) - 1

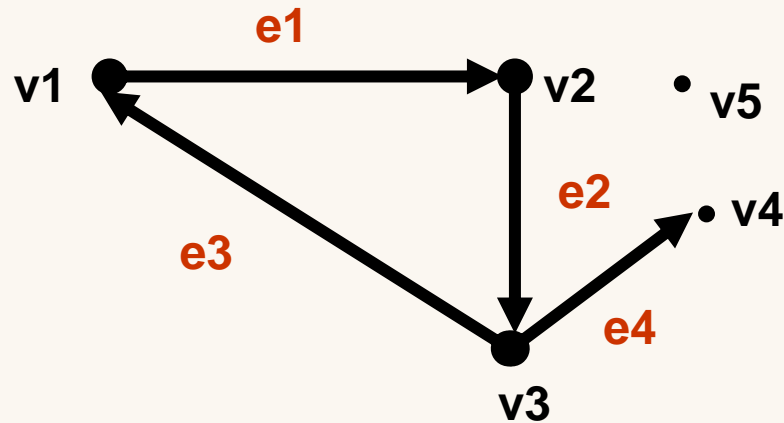
---

- Contoh graf G berikut :



- Titik  $v_1$  adalah **titik awal**  $e_1$ , titik  $v_2$  adalah **titik akhir**  $e_1$ . Arah garis dari  $v_1$  ke  $v_2$ .

# Graf Berarah (Digraph) - 2



- Jumlah garis yang keluar dari titik v1 disebut derajat keluar (**out degree**), simbol  $d^+(v_1)$
- Jumlah garis yang masuk ke titik v1 disebut derajat masuk (**in degree**), simbol  $d^-(v_1)$

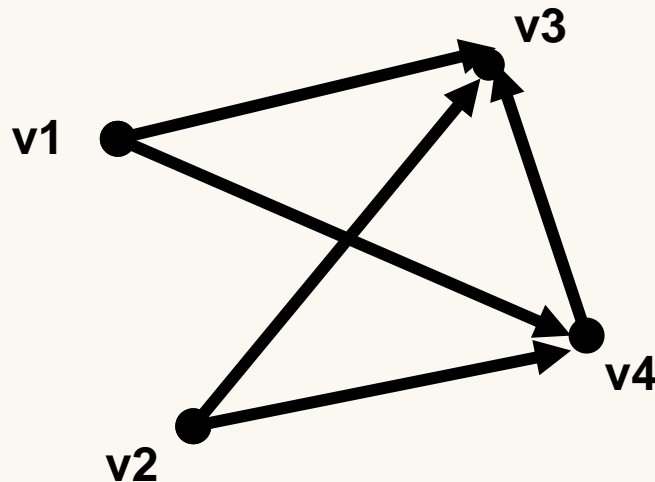
$$\sum_i d^-(v_i) = \sum_i d^+(v_i)$$

# Path Berarah dan Sirkuit Berarah

---

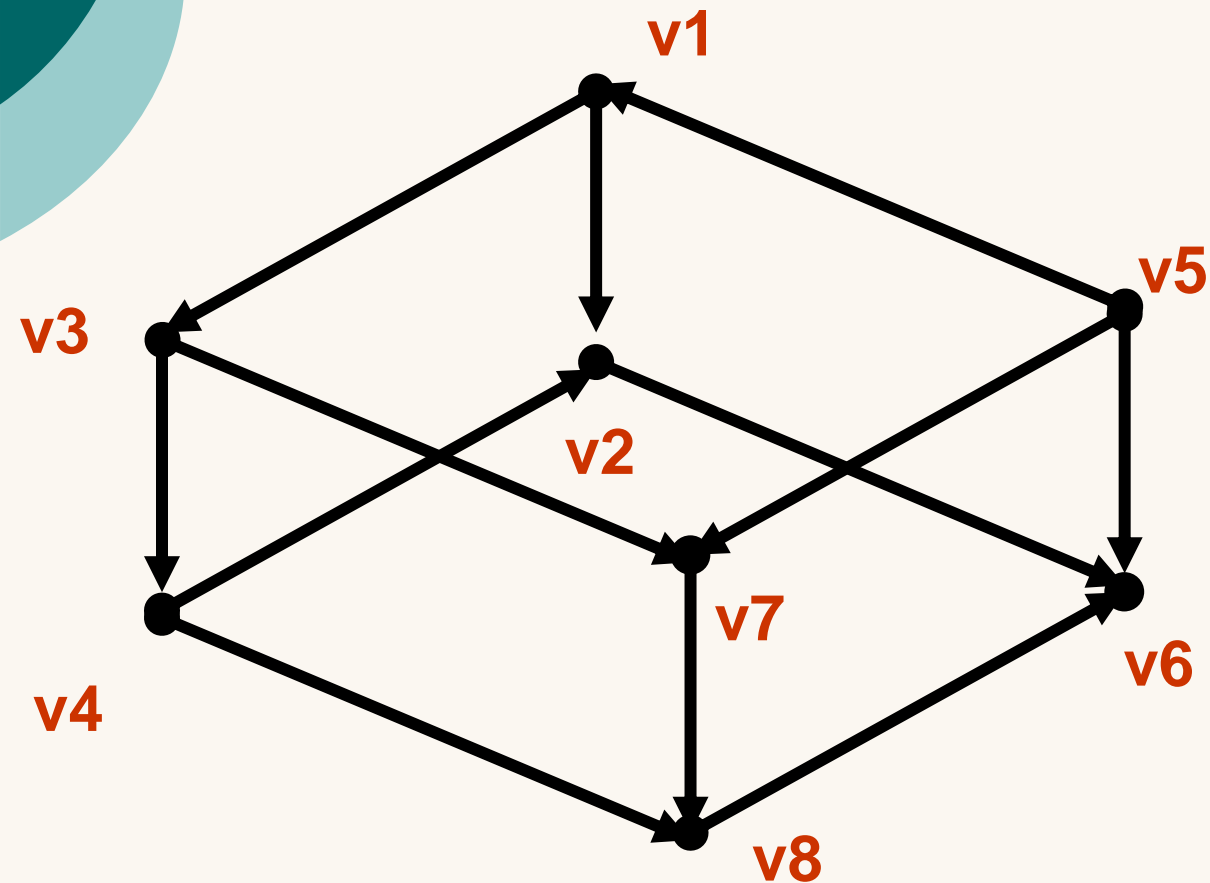
- Dalam graf berarah, perjalanan harus mengikuti arah garis.
- Suatu graf yang tidak memuat sirkuit berarah disebut **ASIKLIK**.

Contoh :



# Contoh

- Tentukan path berarah terpendek dari titik  $v_5$  ke titik  $v_2$  !



# Pohon (Tree)

---

- Struktur Pohon adalah salah satu kasus dalam graf.
- Penerapannya pada Teori Struktur Data.
- Graf  $G$  disebut **Pohon**  $\Leftrightarrow G$  merupakan graf sederhana yang tidak memuat sirkuit dan terhubung.

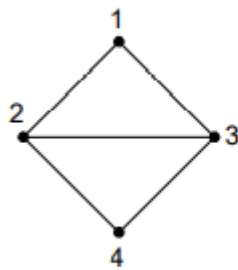
# Ketetanggaan

Dua buah simpul dikatakan bertetangga bila keduanya terhubung langsung.

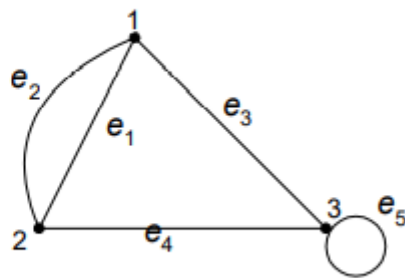
Tinjau graf  $G_1$  :

Simpul 1 bertetangga dengan simpul 2 dan 3

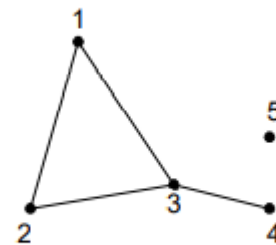
Simpul 1 tidak bertetangga dengan simpul 4



$G_1$



$G_2$



$G_3$

# Bersisian

Untuk sembarang sisi  $e = (v_j, v_k)$  dikatakan :

$e$  bersisian dengan simpul  $v_j$

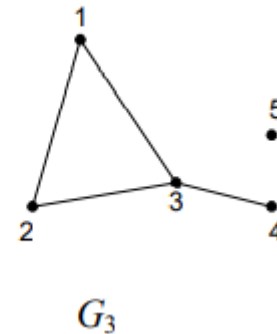
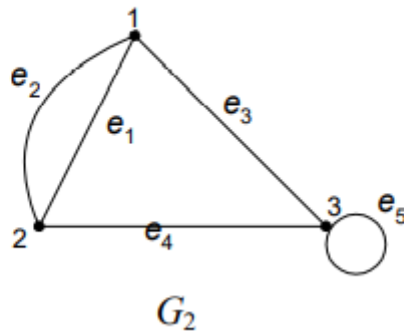
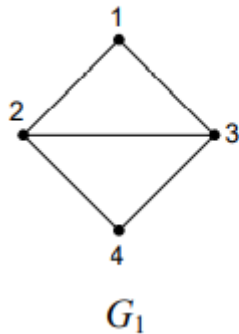
$e$  bersisian dengan simpul  $v_k$

Tinjau graf  $G_1$ :

Sisi  $(2, 3)$  bersisian dengan simpul 2 dan simpul 3

Sisi  $(2, 4)$  bersisian dengan simpul 2 dan simpul 4

Tetapi sisi  $(1, 2)$  tidak bersisian dengan simpul 4.

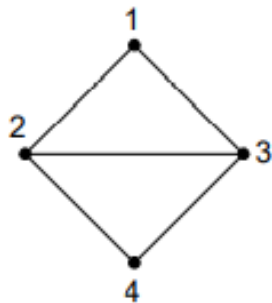


# Simpul Terpencil

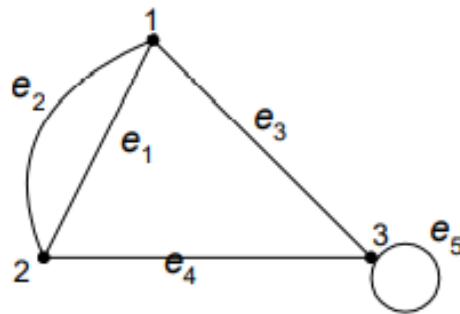
Simpul terpencil ialah simpul yang tidak mempunyai sisi yang bersisian dengannya.

Tinjau graf  $G_3$ :

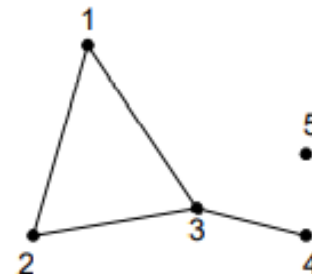
Simpul 5 adalah simpul terpencil.



$G_1$



$G_2$



$G_3$

## Pohon (2)

---

- **Daun** adalah titik di dalam Pohon yang berderajat 1.
- Titik dalam Pohon yang berderajat  $> 1$  disebut **Titik Cabang**.

### Teorema

*Suatu pohon dengan  $n$  titik memiliki  $(n-1)$  garis*

# Pohon Rentang

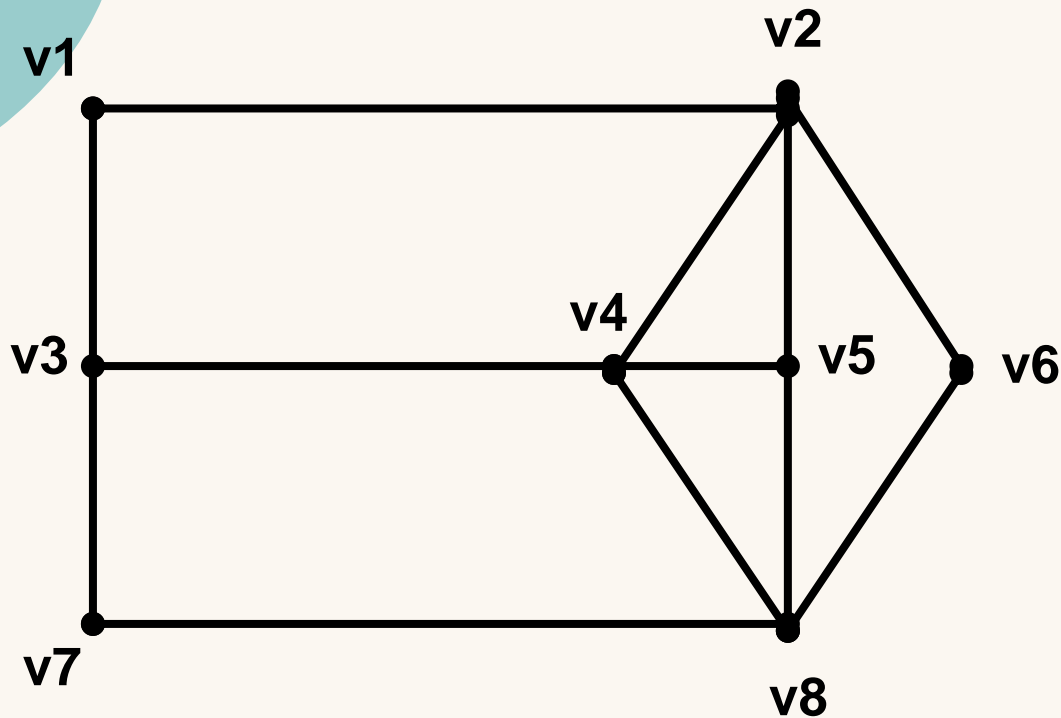
---

- **Pohon Rentang** dari graf terhubung  $G$  adalah subgraf  $G$  yang merupakan pohon dan memuat semua titik dalam  $G$ .

# Contoh

---

- Cari pohon rentang dari graf  $G$  !



# Graf Berlabel

---

- **Graf Berlabel** : graf tanpa garis paralel yang setiap garisnya berhubungan dengan bilangan riil positif yang menyatakan **bobot** garis tersebut.
- Simbol :  $w(e)$ .
- **Total Bobot** : jumlah bobot semua garis dalam graf.
- **Bobot** suatu garis dapat mewakili "jarak", "biaya", "panjang", "kapasitas", dll.

# Pohon Rentang Minimum

---

- Masalah : mencari pohon rentang dengan total bobot seminimal mungkin.
- Metode : ***Algoritma Kruskal***

# Algoritma Kruskal (1)

---

- Mula-mula urutkan semua garis dalam graf dari yang bobotnya terkecil sampai terbesar.
- **G** : graf mula-mula dg **n** titik,
- **T** : Pohon Rentang Minimum,
- **E** : himpunan semua garis dlm G

# Algoritma Kruskal (2)

---

## Algoritma :

1. Isi  $T$  dengan semua titik dalam  $G$  tanpa garis.
2.  $m = 0$
3. Selama  $m < (n-1)$  lakukan :
  - a. Pilih garis  $e$  dalam  $E$  dg bobot terkecil. Jika ada beberapa garis, pilih salah satu.
  - b. Hapus garis  $e$  dari  $E$ .
  - c. Jika garis  $e$  ditambahkan ke  $T$  tidak menghasilkan sirkuit, maka
    - I. Tambahkan  $e$  ke  $T$ .
    - II.  $m = m+1$  (Nilai  $m$  dinaikkan satu).

## Lintasan Terpendek

---

- Mencari path dengan total bobot paling minimal dari sebuah graf berlabel.
- Metode : Algoritma Djikstra

# Algoritma Dijkstra

---

- $V$  =  $\{v_1, v_2, \dots, v_n\}$   $\rightarrow$  titik awal :  $v_1$ , titik akhir :  $v_n$   
 $L(j)$  = jumlah bobot lintasan terpendek dari  $v_1$  ke  $v_j$   
 $w(i,j)$  = bobot garis dari titik  $v_1$  ke titik  $v_j$   
 $T$  = himp. titik yg sudah terpilih dlm alur lintasan terpendek

## ALGORITMA

1.  $T = \{ \}$   
 $L(v_1) = 0$   
 $L(v_2) = L(v_3) = \dots = L(v_n) = \sim$

# Algoritma Dijkstra

---

2. Selama  $v_n \notin T$  lakukan :
  - a. Pilih titik  $v_k \in V - T$  dengan  $L(v_k)$  terkecil  
 $T = T \cup \{ v_k \}$
  - b. Untuk setiap  $v_j \in V - T$  hitung :  
 $L(v_j) = \min[ L(v_j) , L(v_k) + w(v_k, v_j) ]$
3. Telusuri alur path minimum mulai dari titik akhir ( $v_n$ ) sampai titik awal ( $v_1$ )