

# Greedy algorithms

wijanarto

# Materi

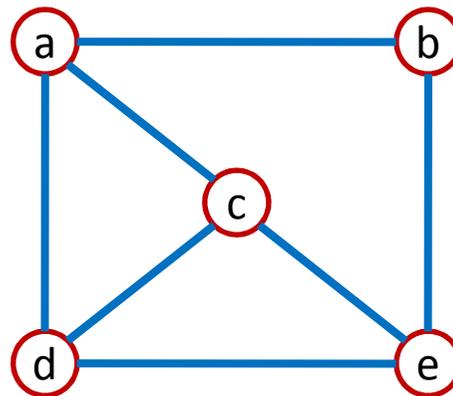
- Graph
- MST
- Kruskal
- Prim
- Dijkstra

# Graph

- Definisi dan terminologi
- Contoh dan aplikasinya

# Definisi dan Terminologi

- Apa itu Graph ?
- Direpresentasi dengan 2 tuple
  - Graph  $G=(V,E)$ , dimana
    - **V** adalah himpunan vertek/node
    - **E** adalah himpunan edge/garis yang menghubungkan vertek/node
  - Edge/garis adalah pasangan vertek dalam graph,  $e=(u,v)$ , yang tidak berurutan.

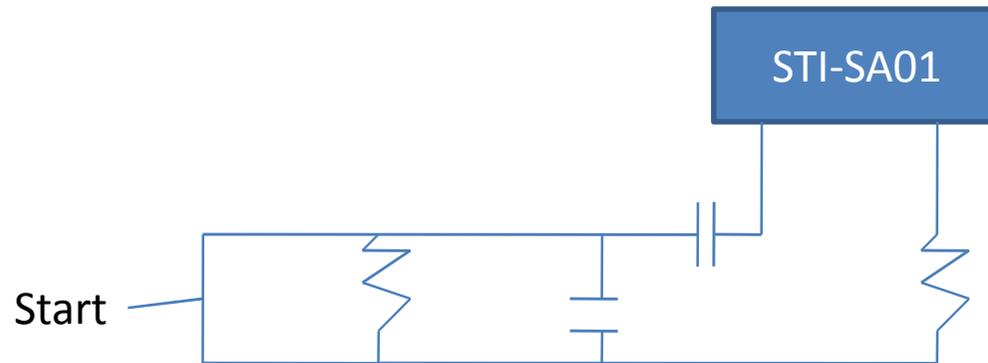


$V=\{a, b, c, d, e\}$

$E=\{(a,b), (a,c), (a,d), (b,e), (c,d), (c,e), (d,e)\}$

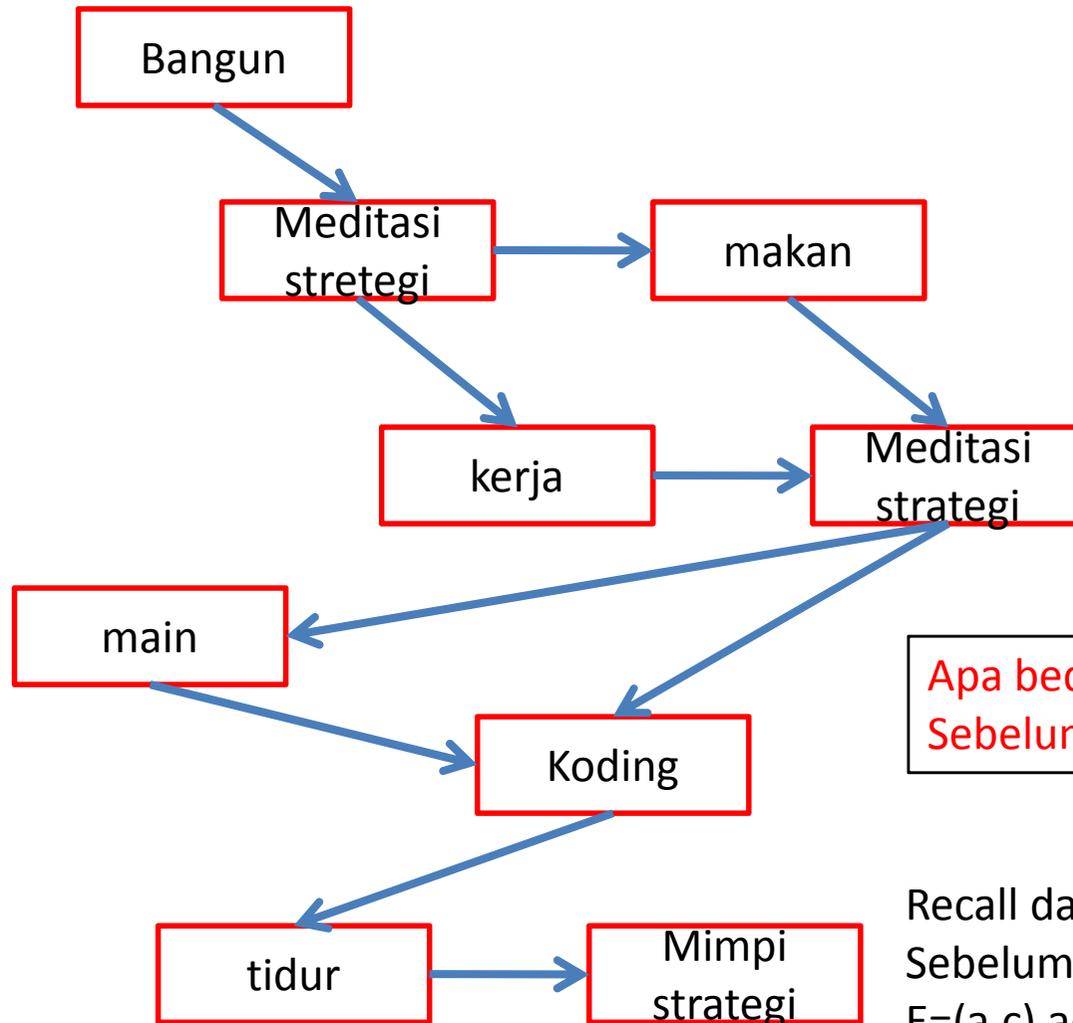
# Aplikasi Graph

- Circuit Elektronik



- Temukan jalur dg hambatan terkecil ke STI-SA01
- Network (peta jalan, penerbangan, komunikasi)

# Aktifitas sehari-hari

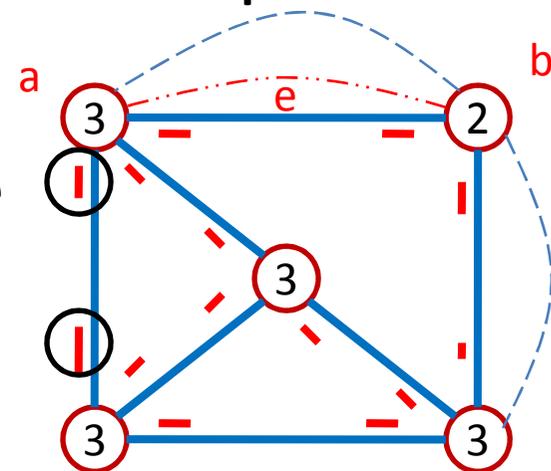


Apa beda dengan graph  
Sebelumnya ?

Recall dari undirected graph  
Sebelumnya, maka urutan dari  
 $E=(a,c)$  adalah penting, ini menunjukkan  
Arah dari a ke c

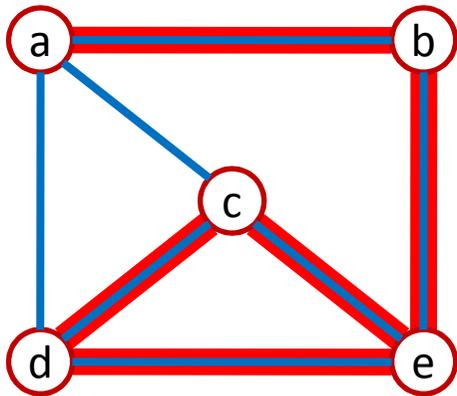
# Terminologi Graph

- Adjacent Vertek : vertek yang di hubungkan oleh edge
- Degree (vertex): jumlah adj vertex
  - Vertek 2 punya degree 2 (3 dan 3) → grs biru putus
- Incident edge :  $e=(a,b)$  adalah incident pada vertek dari a ke b
- # degree seluruh vertex =  $2 * \text{edge}$

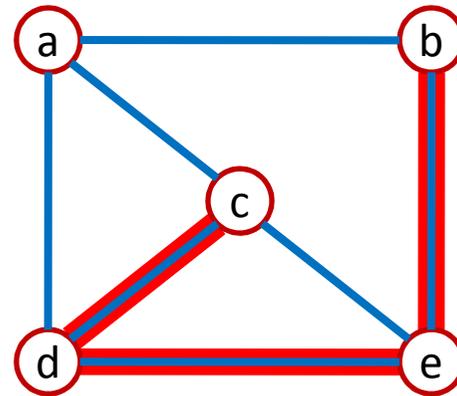


# Terminologi

- Path: urutan vertek  $v_1, v_2, v_3, \dots, v_k$  sedemikian sehingga menyebabkan vertek  $v_i$  dan  $v_{i+1}$  saling berhubungan.



abecde

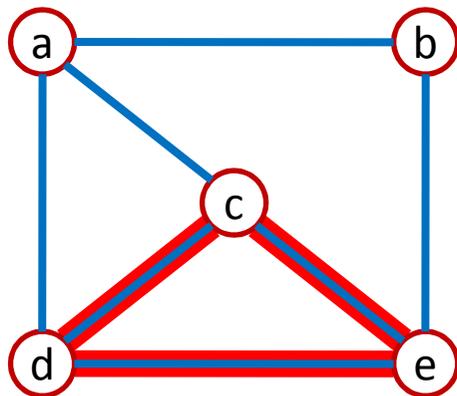


bedc

# Terminologi

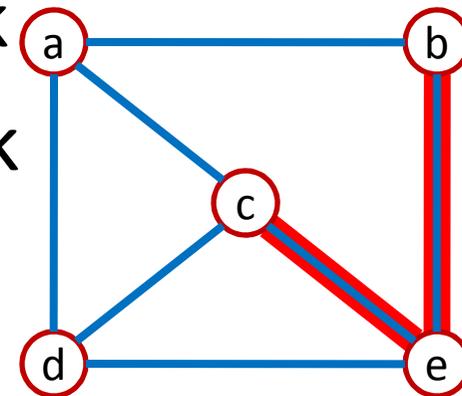
- **Simple path** : tidak ada path yang berulang
- **Cycle** : adalah SP dengan vertek Terakhir yang sama dengan vertek

awal



ecde

cycle

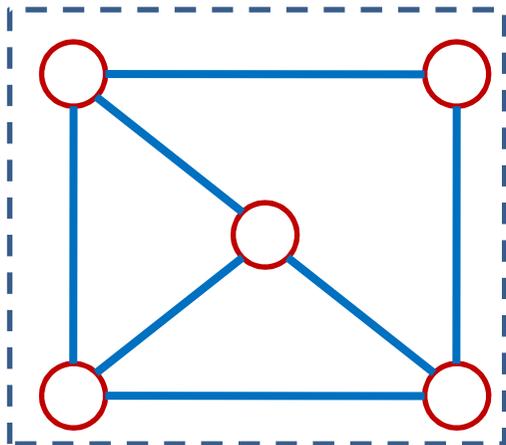


bec

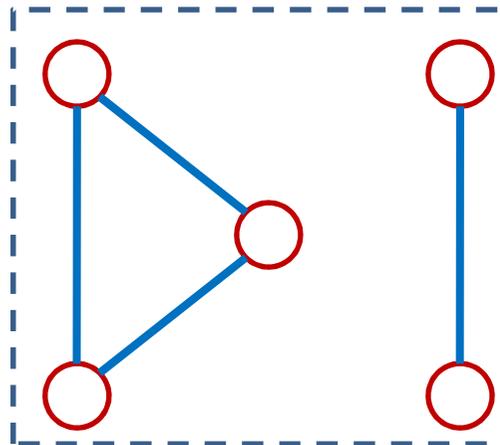
# Terminologi

- **Connected graph** : tiap 2 vertek di hubungkan oleh suatu **path**.

connected

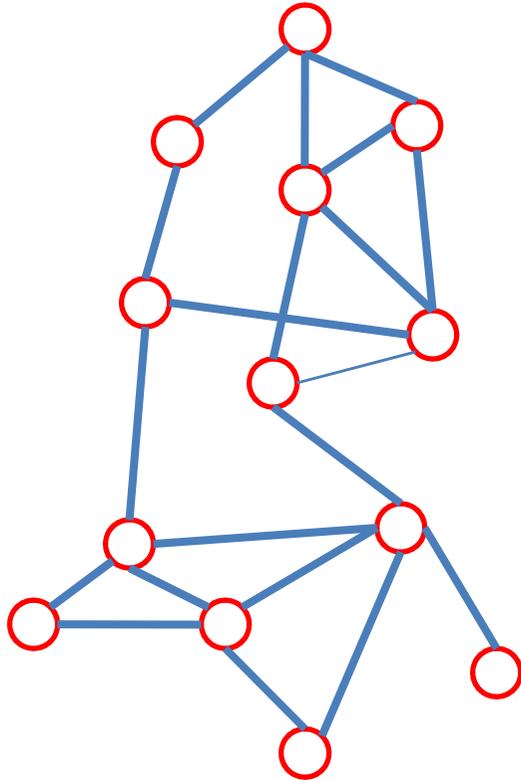


Not connected

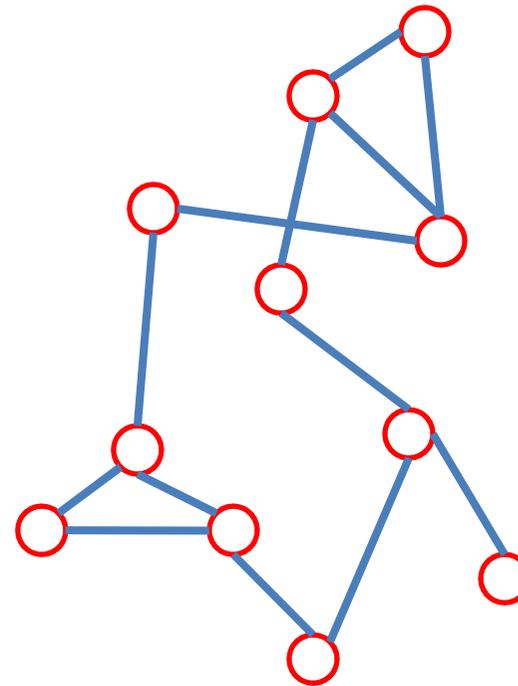


# Terminologi

- Sub Graph : subset vertek dan edge yang membentuk suatu graph



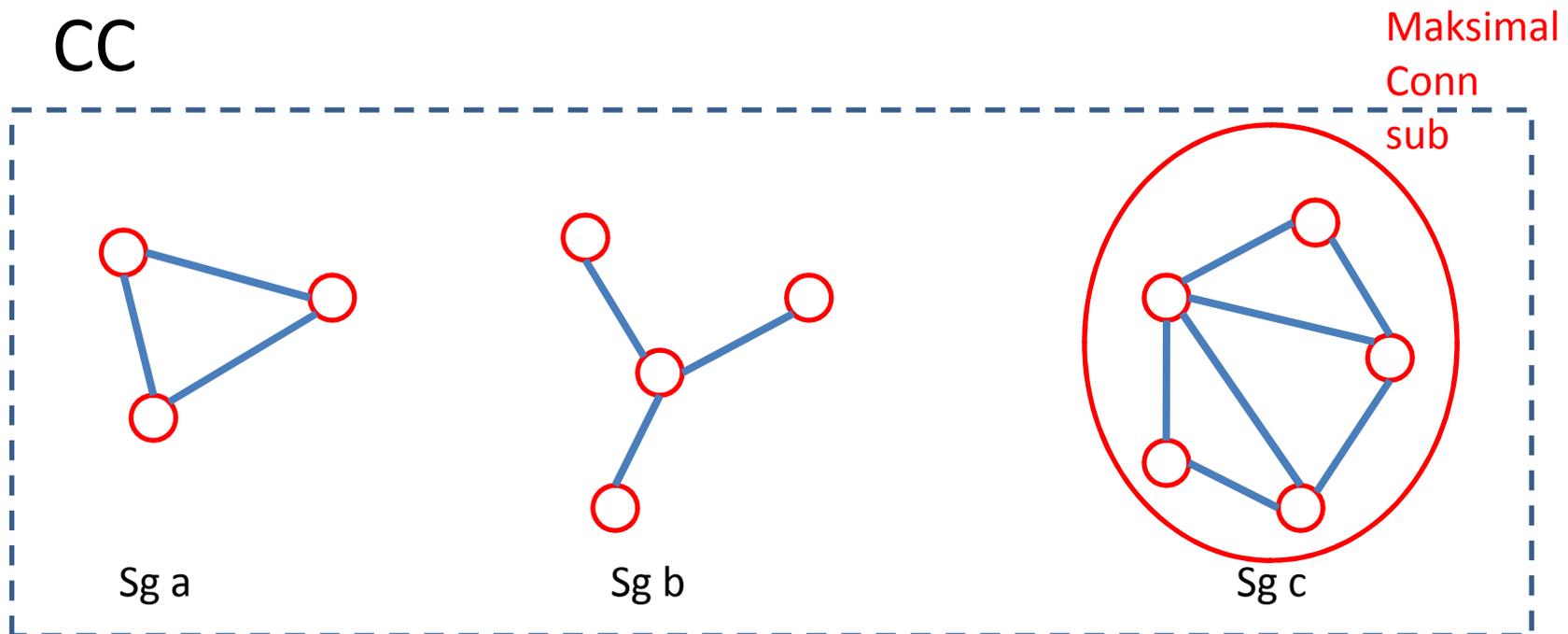
G



Subgraph G

# Terminologi

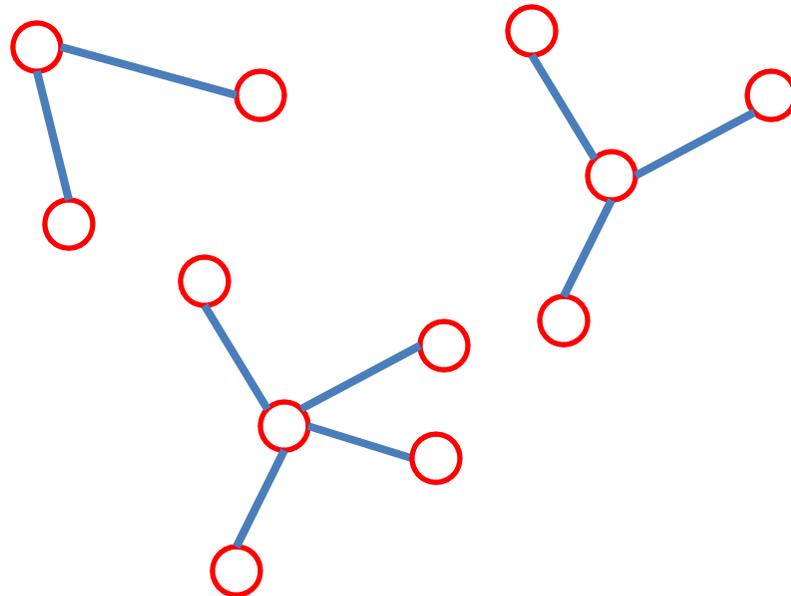
- Connected component : **subgraph terhubung maximal**, contoh di bawah ini Graph dengan 3 CC



STM: bila suatu sg tidak dapat lagi di tambah baik vertek maupun edge, sedemikian rupa Sg tersebut membentuk himpunan vertek dan edge yang berhubungan

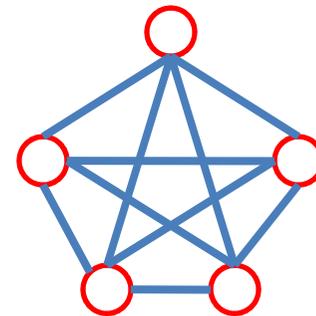
# Terminologi

- (free) tree : graph terhubung tanpa cycle
- **Hutan(forest)**:koleksi tree



# Konektifitas

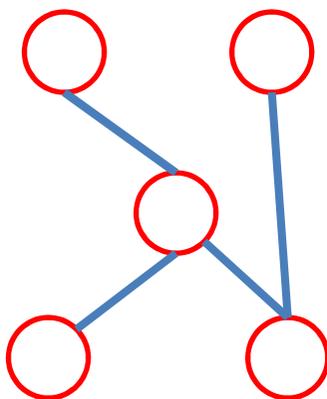
- Diberikan  $n = \# \text{vertex}$  dan  $m = \# \text{edge}$
- **Complete Graph** : jika semua pasangan vertek terhubung.
- Berapa edge dalam complete graph ?
  - Ada sejumlah  $n(n-1)/2$  pasangan vertek, sehingga  $m = n(n-1)/2$



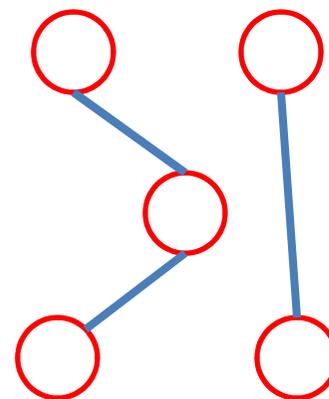
$$N=5$$
$$M=(5*4)/2=10$$

# Konektifitas

- $n = \#$  vertek
- $m = \#$  edge
- Untuk tree  $m = n - 1$ , Buktikan ????
- Jika  $m < n - 1$  maka  $G$  tidak terhubung



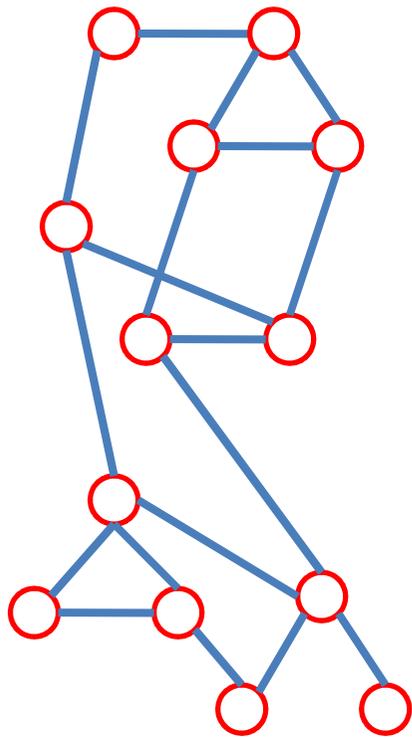
$n=5$   
 $m=4$



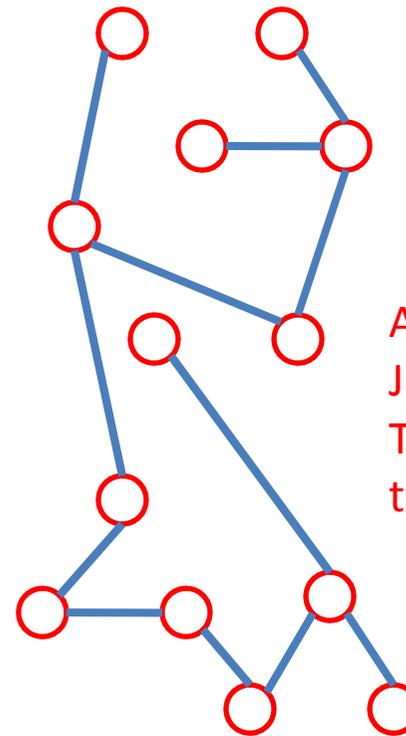
$n=5$   
 $m=3$

# Spanning Tree (SP)

- **SP** dari  $G$  adalah subgraph yang mrp tree dan mengandung seluruh vertek (spanning) dalam  $G$ .



**G**

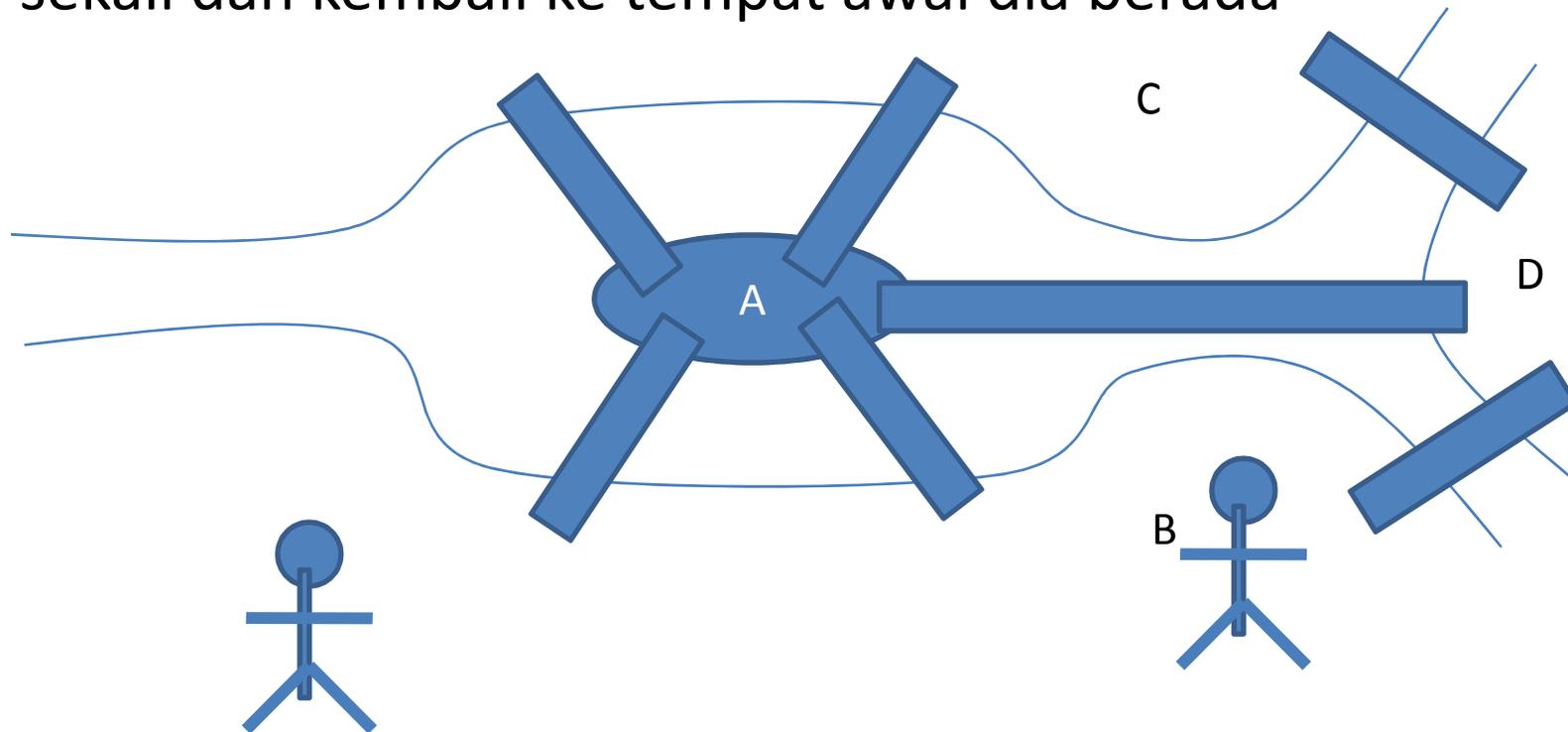


Akan tjd kegagalan  
Jika ada edge yg  
Terpotong (least fault  
tolerance)

Spanning tree dari G

# Jembatan Koenigsberg

- Dapatkah seseorang berjalan melewati jembatan tepat sekali dan kembali ke tempat awal dia berada



Pada tahun 1736 EULER membuktikan hal ini tidak mungkin di lakukan

# Model Graph (parallel edge)

- Multigraph:
  - Tdp byk edge pd pasangan vertek

## Euler Tour :

Path yg dapat mengunjungi  
Tiap edge tepat sekali dan  
Kembali ke titik asalnya

## Euler Teorema :

G berupa Euler Tour

IIF seluruh verteknya

Mempunyai degree genap

