Institut Teknologi
Telkom
Purwokerto

Teknik Informatika - Fakultas Informatika

Pertemuan 14 – Graph

Author: **Wahyu Andi Saputra [WAA]**

Institut Teknologi
Telkom
Purwokerto

Outline

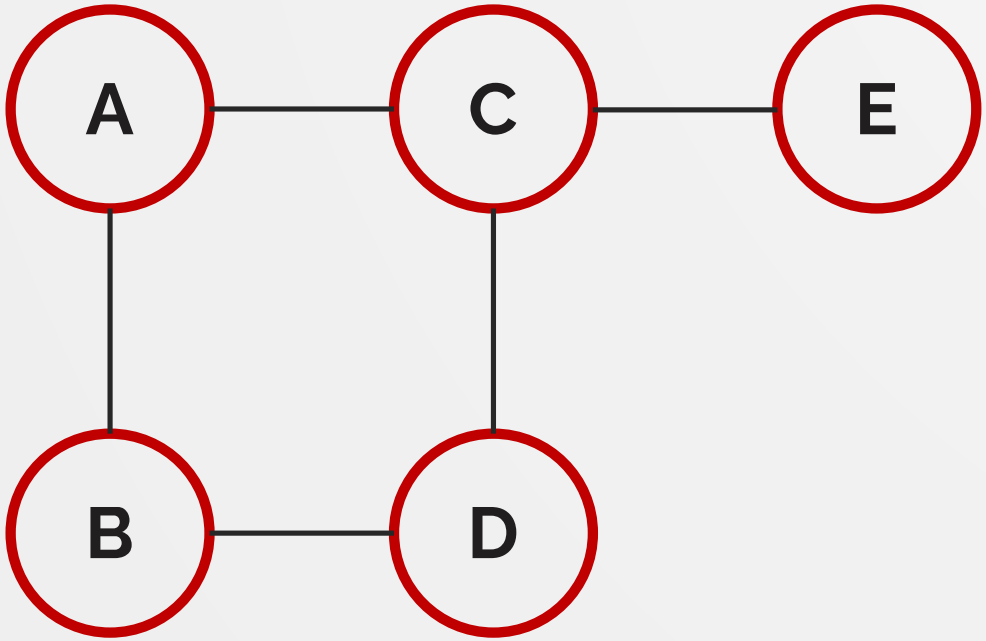
- Konsep Graph
- Jenis Graph
- Representasi Graph
- Implementasi Graph

Institut Teknologi
Telkom
Purwokerto

Konsep Graph

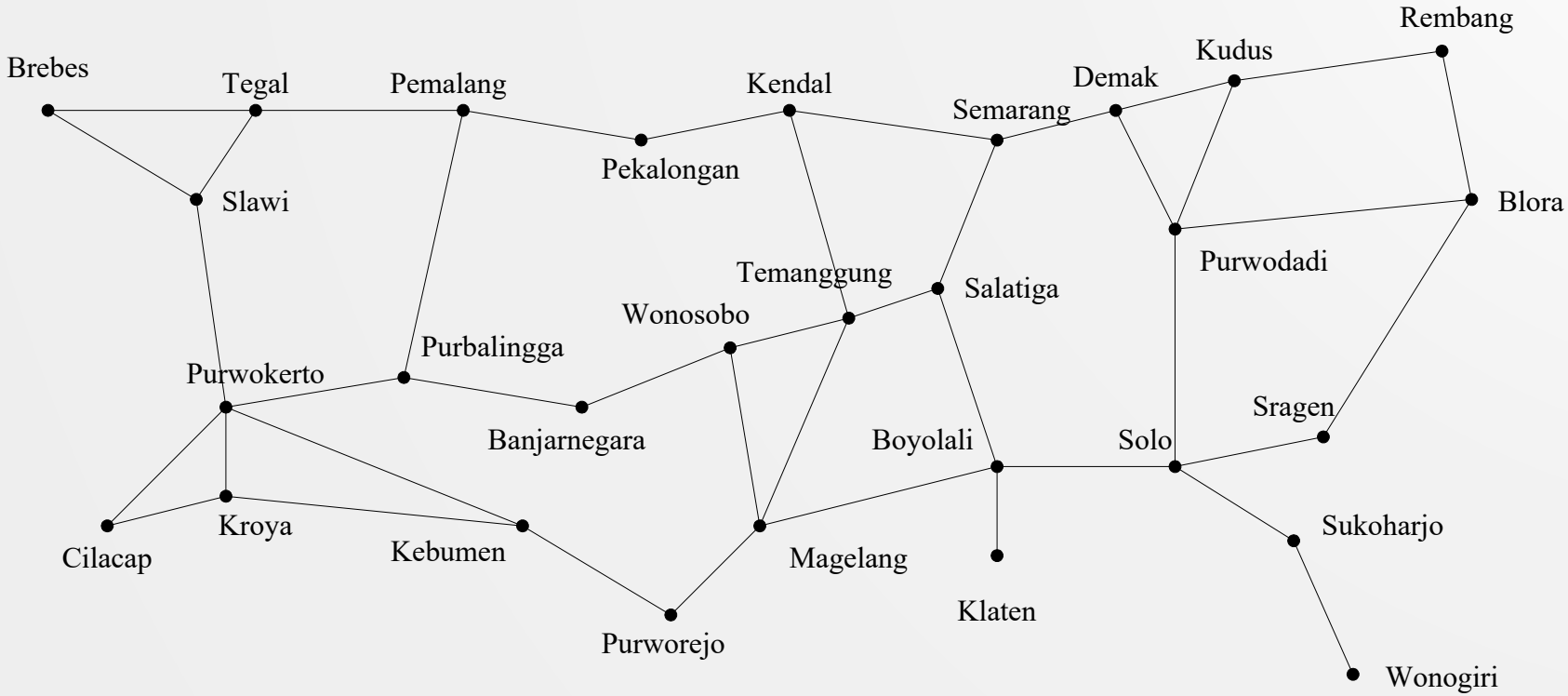
Graph

- Merepresentasikan objek-objek dan hubungan antar objek



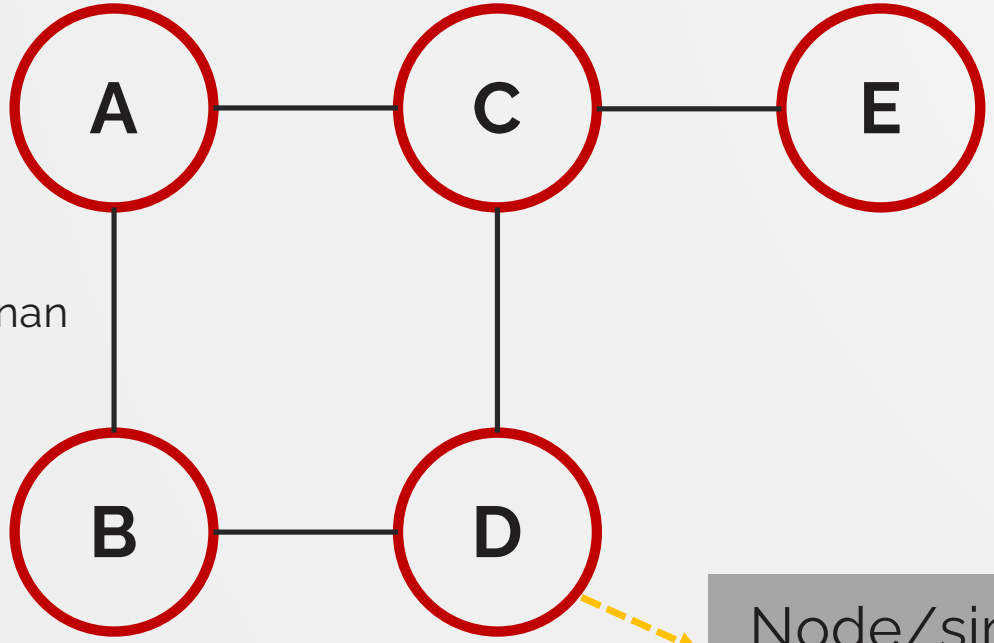
Graph

- Merepresentasikan objek-objek dan hubungan antar objek



Graph

Institut Teknologi Telkom Purwokerto

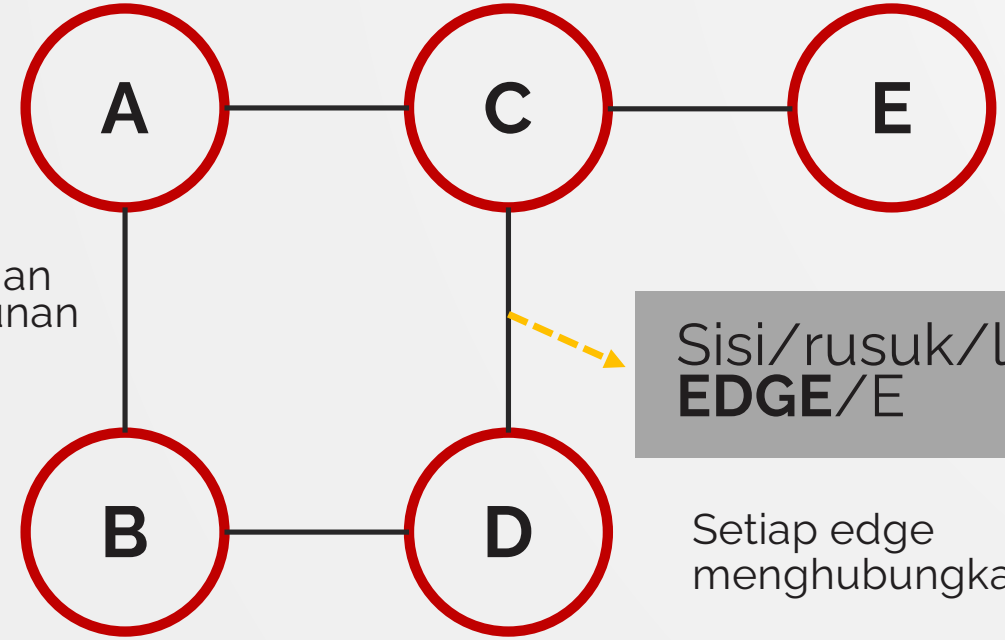


G(V,E) adalah pasangan himpunan V dan himpunan E pada suatu Graf G

Node/simpul/titik/ point/**VERTEX**/V

Graph

Institut Teknologi Telkom Purwokerto



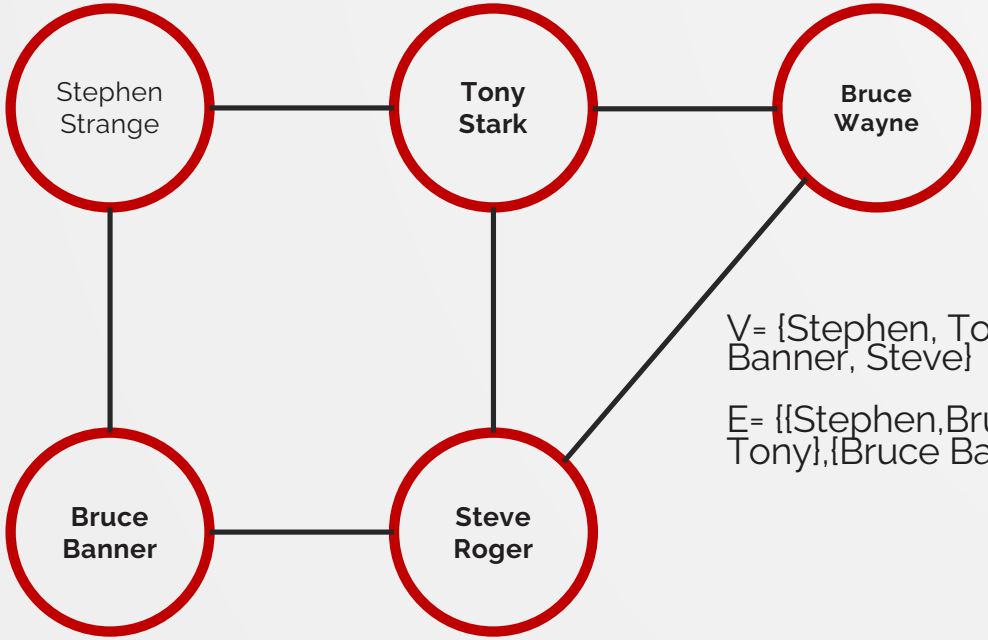
G = (V,E) adalah pasangan himpunan V dan himpunan E pada suatu Graf G

Sisi/rusuk/line/
EDGE/E

Setiap edge menghubungkan dua vertex

Contoh Penerapan Graph

Institut Teknologi Telkom Purwokerto



V= {Stephen, Tony, Bruce Wayne, Bruce Banner, Steve}

E= {{Stephen,Bruce Banner},{Stephen, Tony},{Bruce Banner, Steve Roger}, ... }

Contoh Penerapan Graph

Institut Teknologi Telkom Purwokerto

- PLN: mencari rute terpendek untuk menghubungkan jalur kelistrikan dari 2 desa
- Pertamina: mengetahui rute pipa paling optimal untuk mencapai suatu titik tujuan
- SI Mahasiswa: menentukan urutan pengambilan mata kuliah yang saling berkaitan

Perbedaan Konsep antar SD

Institut Teknologi Telkom Purwokerto

Challenge!

- Jelaskan perbedaan konsep struktur data linear vs struktur data tree vs struktur data graph!

Institut Teknologi
Telkom
Purwokerto

Jenis Graph

Jenis Graph

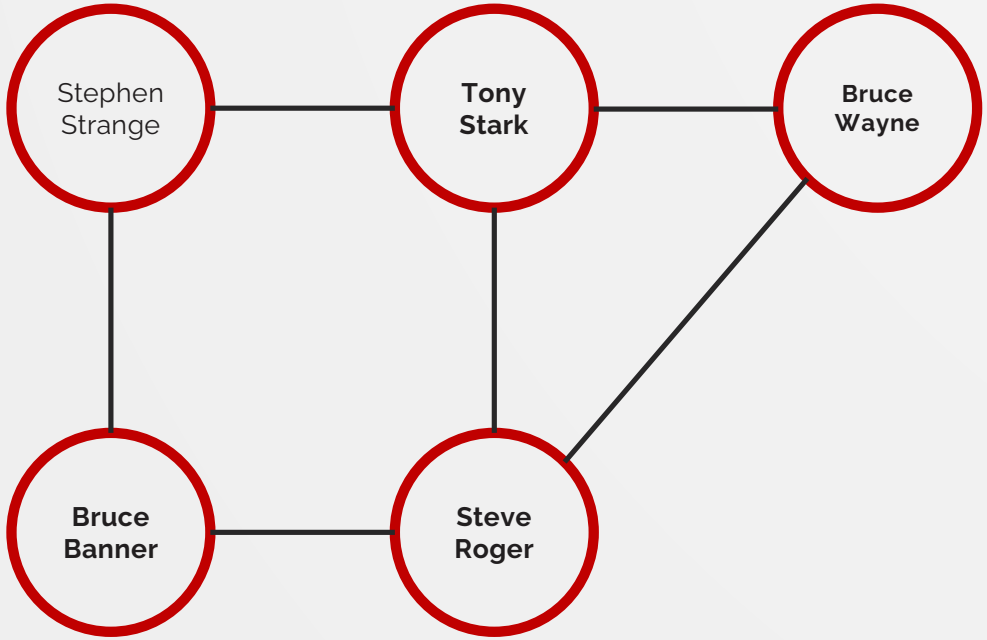
- Undirected Graph
- Directed Graph
- Weighted Graph

Institut Teknologi Telkom Purwokerto

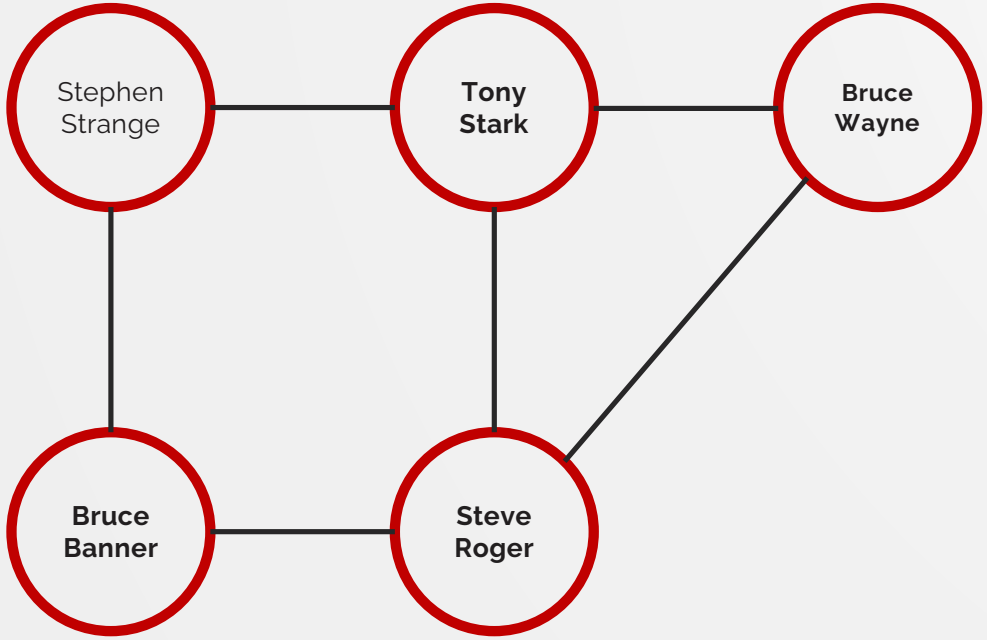
Jenis Graph: Undigraph

Institut Teknologi Telkom Purwokerto

- Graph yang tidak memiliki orientasi/arah, sehingga tidak terdapat arrow pada Edge



Jenis Graph: Undigraph

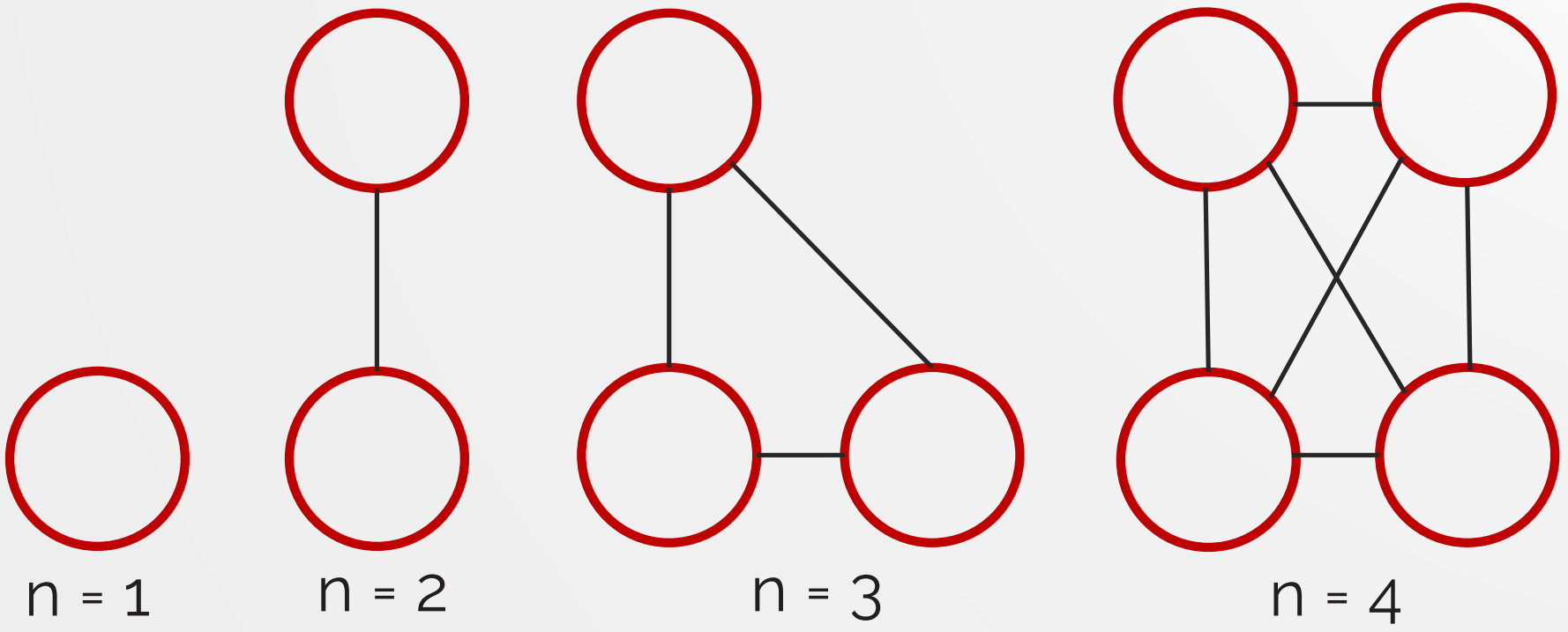


- G = {V,E}
- V = {SS, TS, BW, BB, SR}
- E = {{SS,TS}, {TS,BW}, {BW,SR}, {TS,SR}, {BB,SR}, {BB,SS}}

Jenis Graph: Undigraph

- Complete Undirected Graph: semua node saling terkait

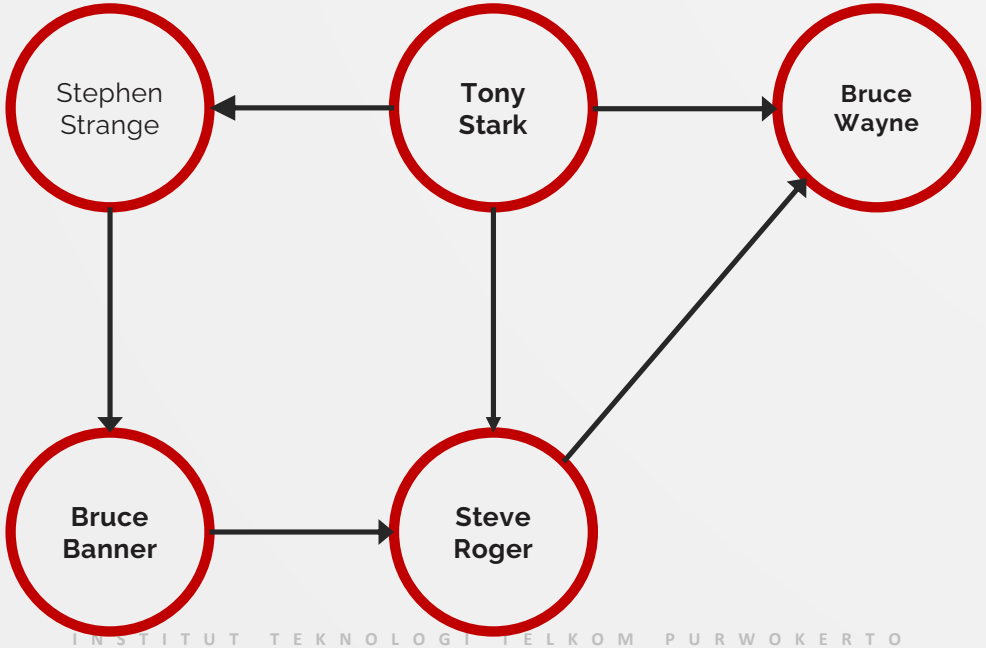
Institut Teknologi Telkom Purwokerto



Jenis Graph: Digraph

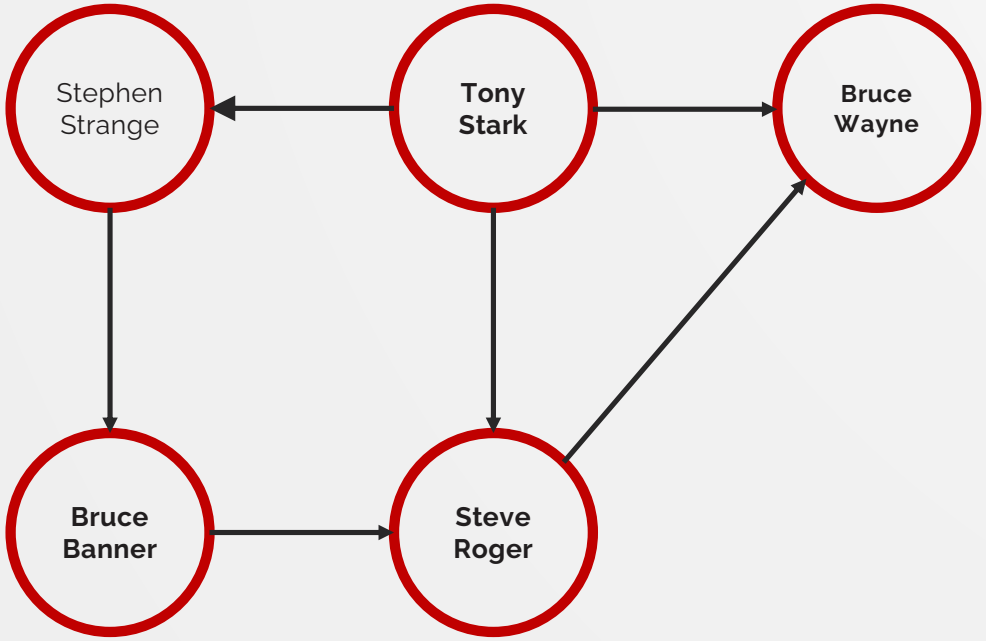
Institut Teknologi Telkom Purwokerto

- Graph yang memiliki orientasi/arah dan terdapat arrow pada Edge yang menunjukkan V asal dan V tujuan



Jenis Graph: Digraph

Institut Teknologi Telkom Purwokerto

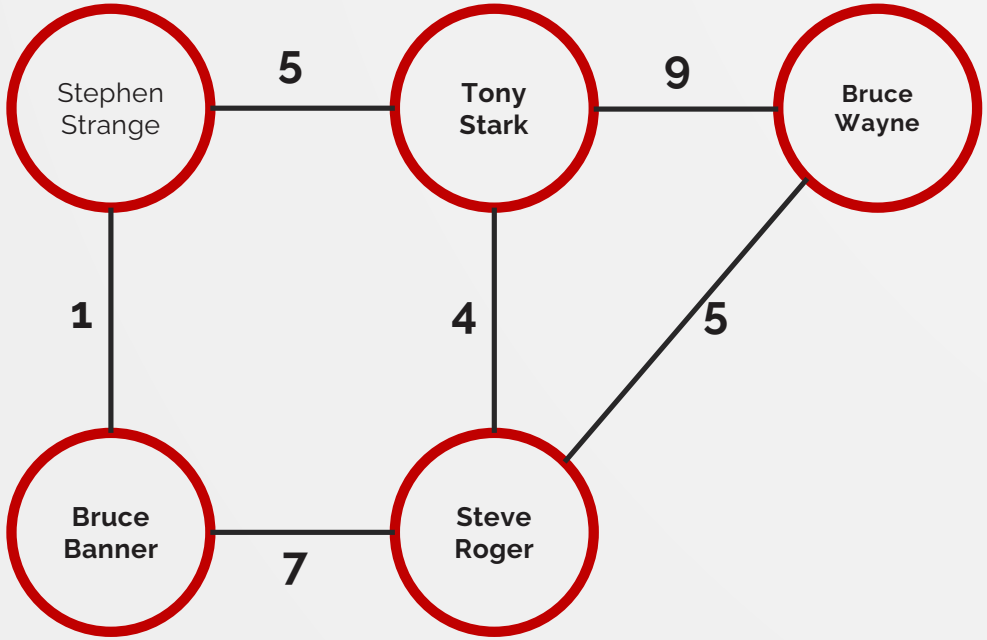


- G = {V,E}
- V = {SS, TS, BW, BB, SR}
- E = {{TS,SS}, {SS,BB}, {BB,SR}, {TS,SR}, {SR,BW}, {TS,BW}}

Jenis Graph: Weighted Graph

- Graph yang memiliki nilai/beban/bobot pada setiap Edge
- Bobot bisa merepresentasikan berbagai hal

Institut Teknologi Telkom Purwokerto

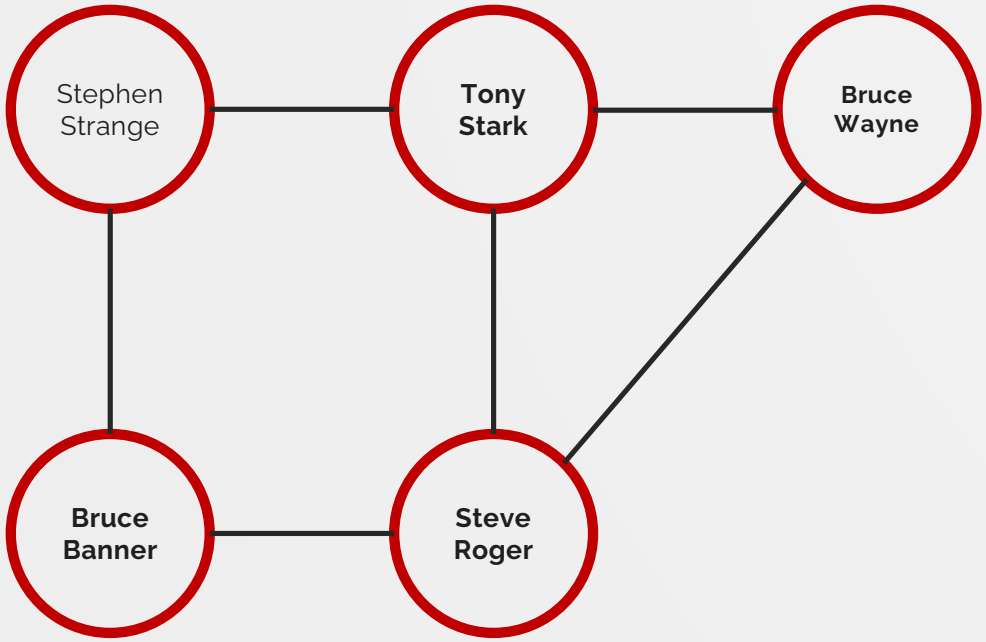


Institut Teknologi
Telkom
Purwokerto

Representasi Graph

Adjacency Matrix

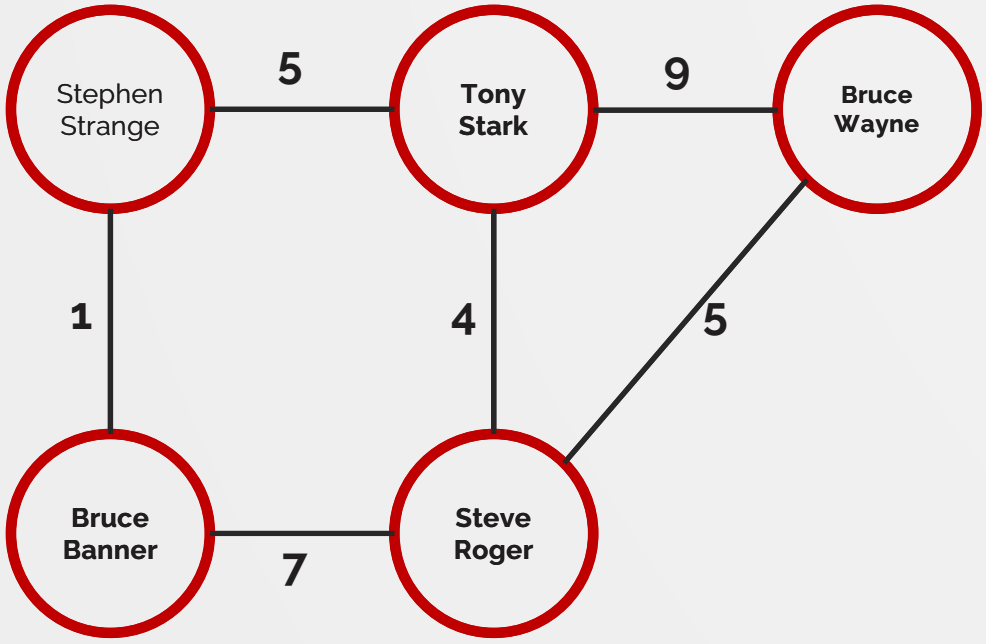
Institut Teknologi Telkom Purwokerto



	SS	TS	BW	BB	SR
SS	0	1	0	1	0
TS	1	0	1	0	1
BW	0	1	0	0	1
BB	1	0	0	0	1
SR	0	1	1	1	0

Adjacency Matrix

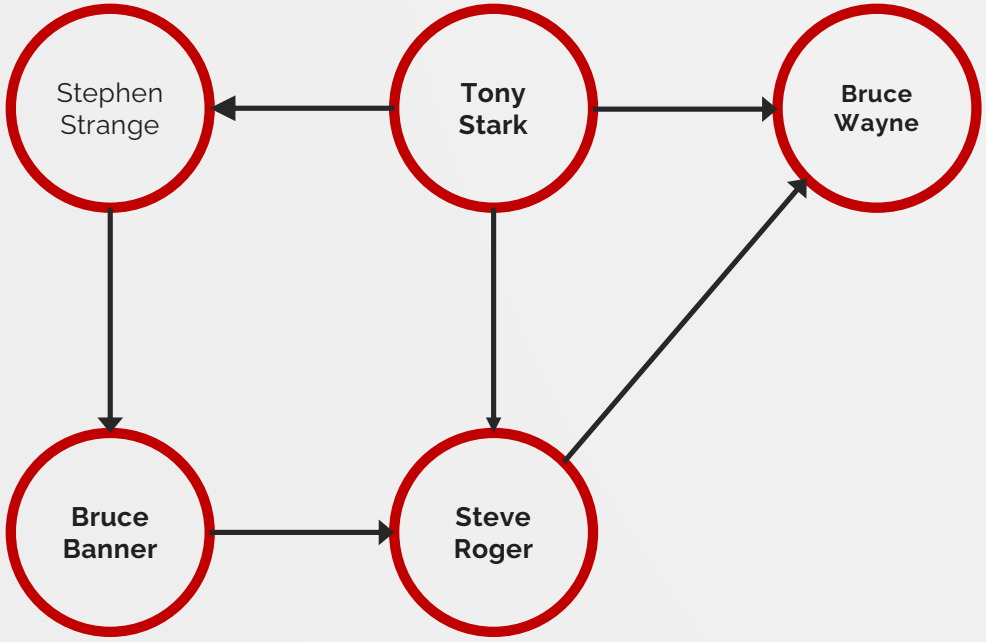
Institut Teknologi Telkom Purwokerto



	SS	TS	BW	BB	SR
SS	0	5	0	1	0
TS	5	0	9	0	4
BW	0	9	0	0	5
BB	1	0	0	0	7
SR	0	4	5	7	0

Adjacency Matrix

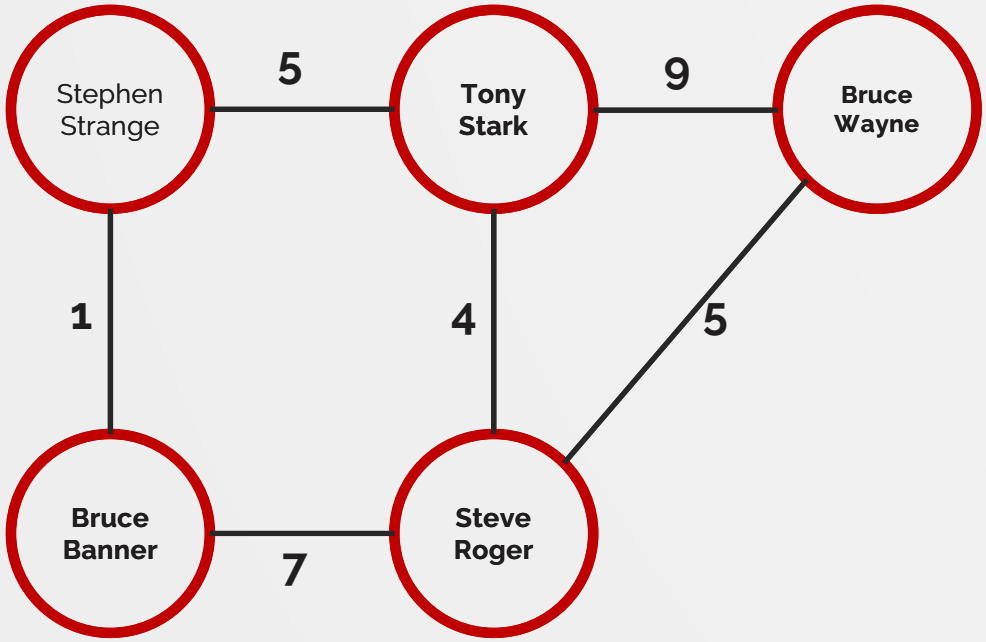
Institut Teknologi Telkom Purwokerto



dari \ ke	SS	TS	BW	BB	SR
SS	0	0	0	1	0
TS	1	0	1	0	1
BW	0	0	0	0	0
BB	0	0	0	0	1
SR	0	0	1	0	0

Adjacency Matrix

Institut Teknologi Telkom Purwokerto

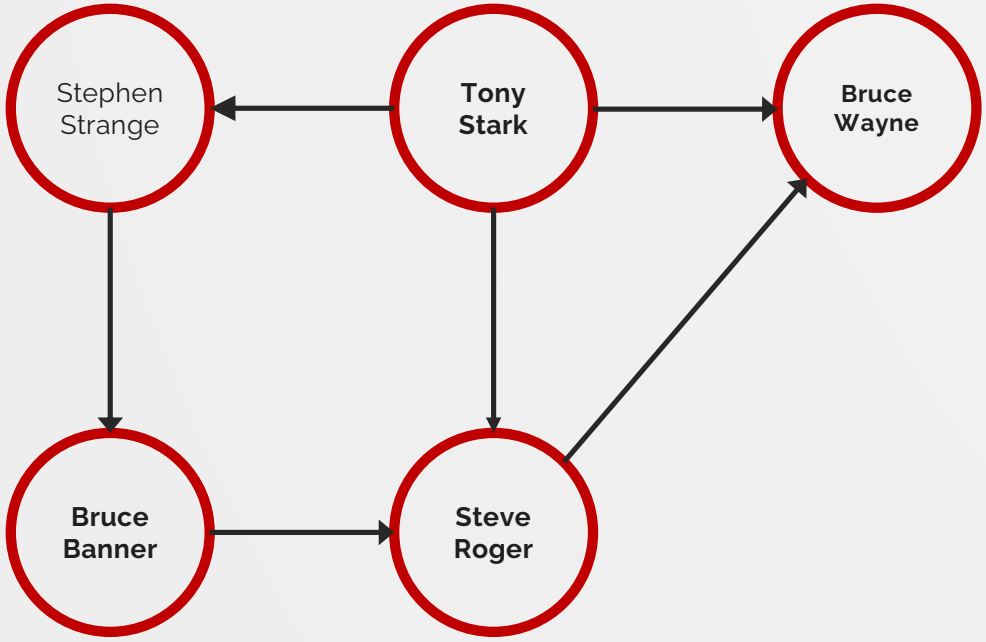


degree(SR) = 3

degree(BW = 2

Adjacency Matrix

Institut Teknologi Telkom Purwokerto



indegree(SR) = 2

indegree(BW) = 2

indegree(TS) = 0

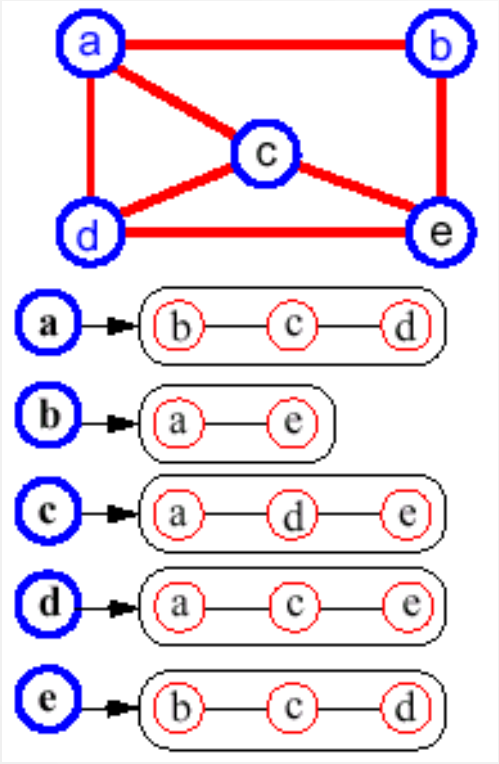
outdegree(SR) = 1

outdegree(BB) = 1

outdegree(SS) = 1

Adjacency List

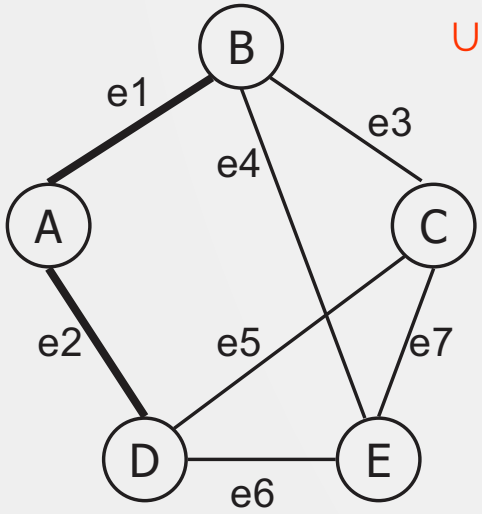
Bentuk adjacency list dari graph

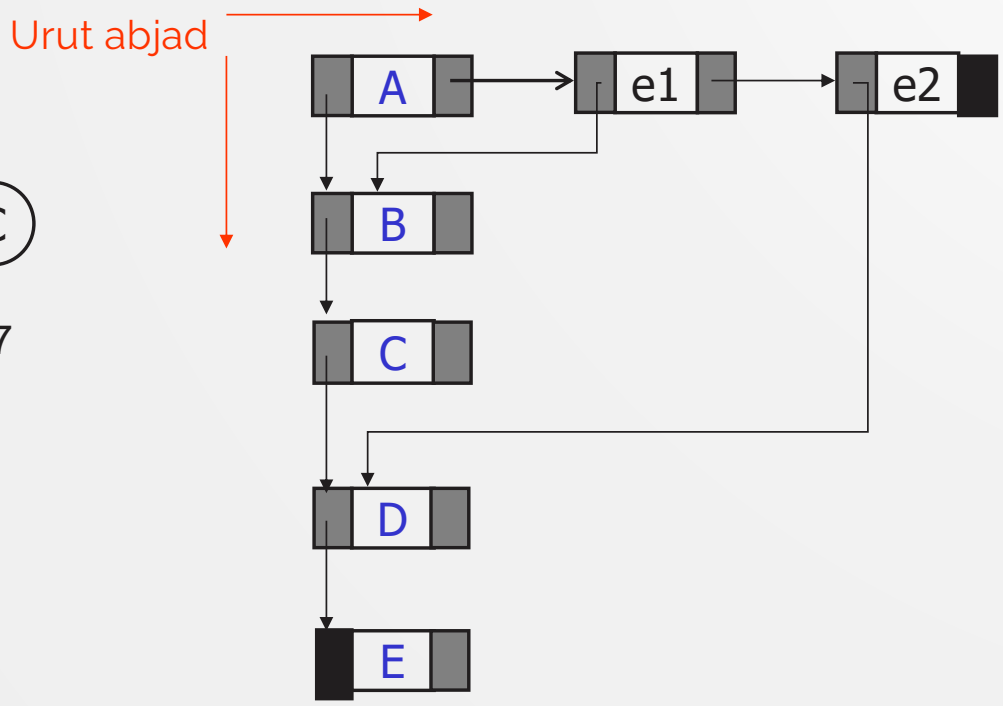


Institut Teknologi Telkom Purwokerto

Adjacency List

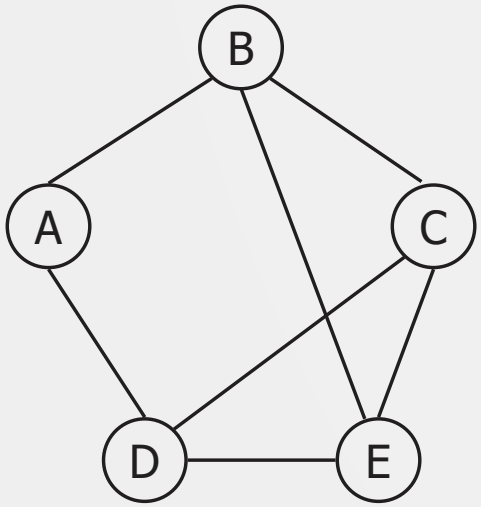
Institut Teknologi Telkom Purwokerto

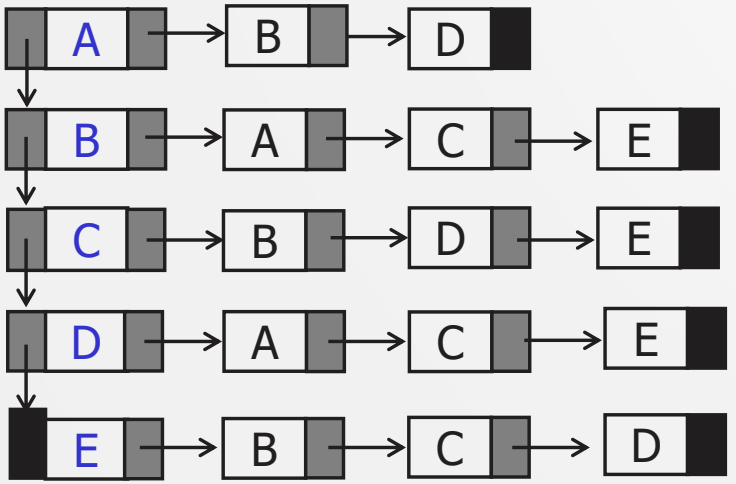




Adjacency List

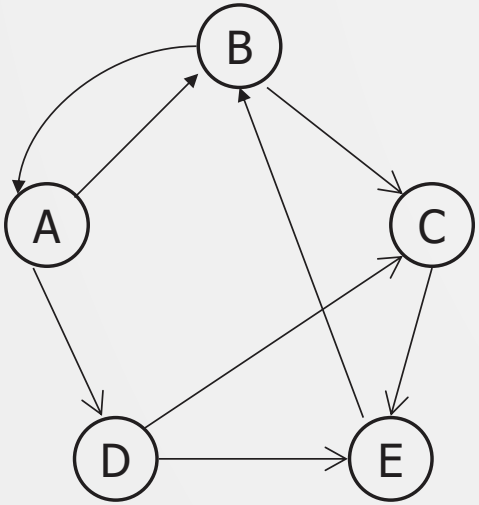
Institut Teknologi Telkom Purwokerto

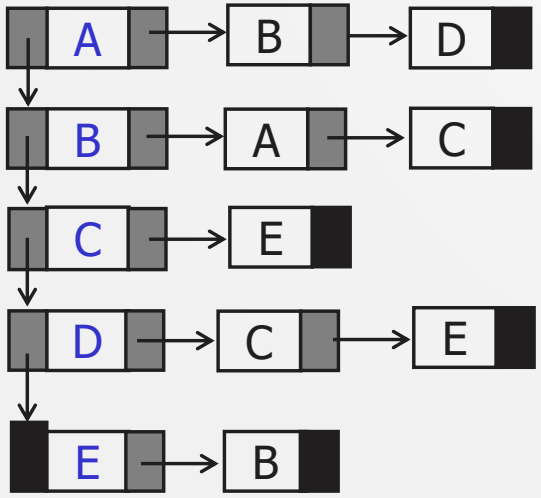




Adjacency List

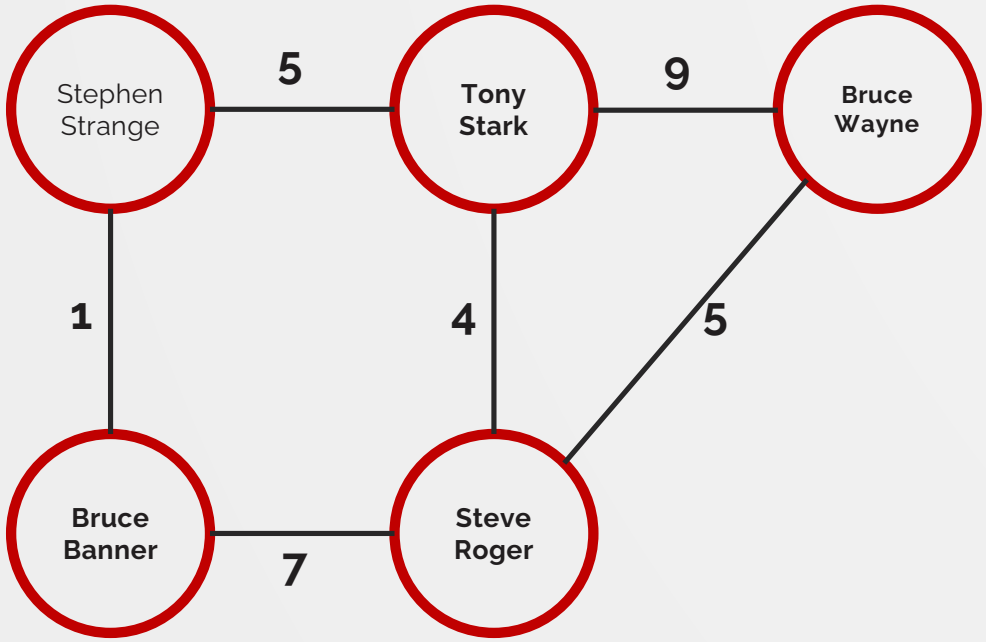
Institut Teknologi Telkom Purwokerto





Graph Path

Institut Teknologi Telkom Purwokerto



Graph Path dari BB ke BW:

1. BB-SR-BW
2. BB-SS-TS-BW
3. BB-SR-TS-BW
4. BB-SS-TS-SR-BW

Institut Teknologi
Telkom
Purwokerto

Implementasi Graph

Dijkstra's Algorithm

Institut Teknologi Telkom Purwokerto

- Algoritma Dijkstra menghitung jarak tiap simpul dari simpul awal hingga akhi diketahui jarak terpendek simpul akhir yang diinginkan.
- Algoritma mengingat simpul mana saja yang telah dihitung jarak terpendeknya dan dinyatakan dalam kelompok hijau (pada literatur dinyatakan sebagai awan putih/white cloud).
- Untuk simpul yang baru sebagian dihitung jaraknya dan belum bisa dipastikan apakah itu jarak terpendek, dinyatakan dengan kelompok abu-abu.
- Untuk simpul yang sama sekali belum dihitung, dinyatakan dalam kelompok hitam.

Dijkstra's Algorithm

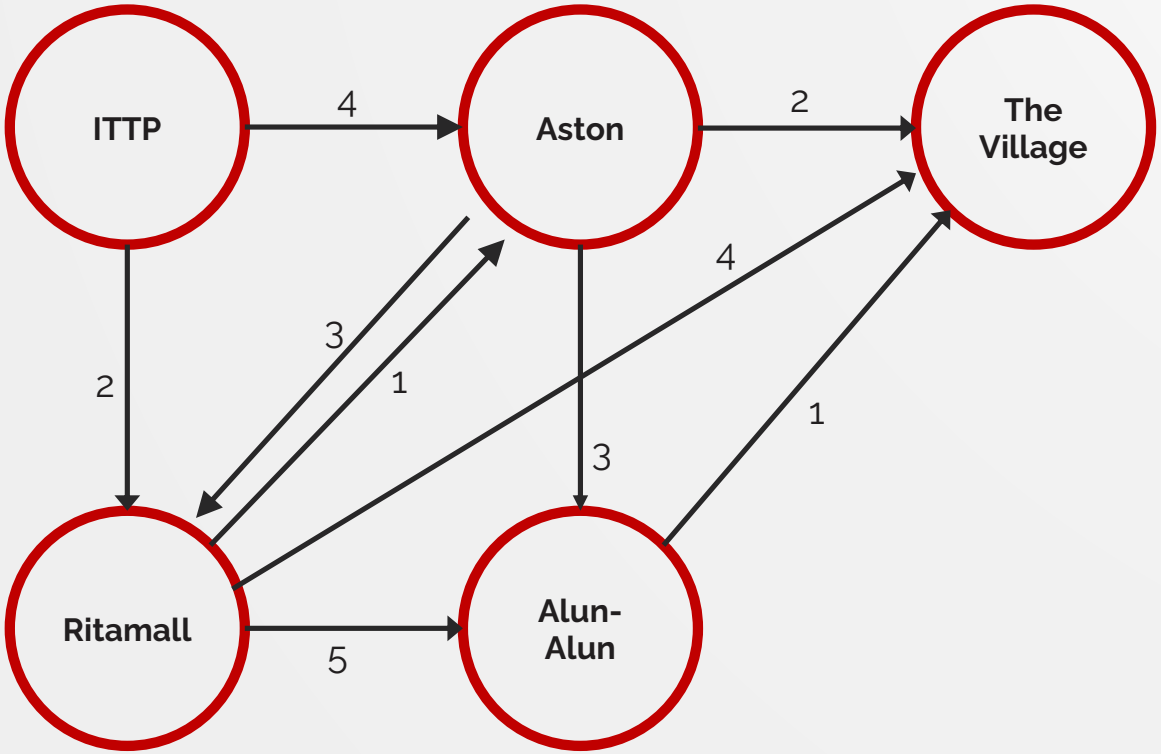
Institut Teknologi Telkom Purwokerto

- Algoritma menggunakan label D[v] untuk menyimpan perkiraan jarak terpendek antara s dan v.
- Ketika sebuah simpul v ditambahkan kedalam kelompok aba-abu nilai D[v] sama dengan bobot antara s dan v.
- Pada awalnya, nilai label D untuk setiap simpul adalah:
    - – D[s] = 0
    - – D[v] = ∞ untuk v ≠ s

Dijkstra's Algorithm

Institut Teknologi Telkom Purwokerto

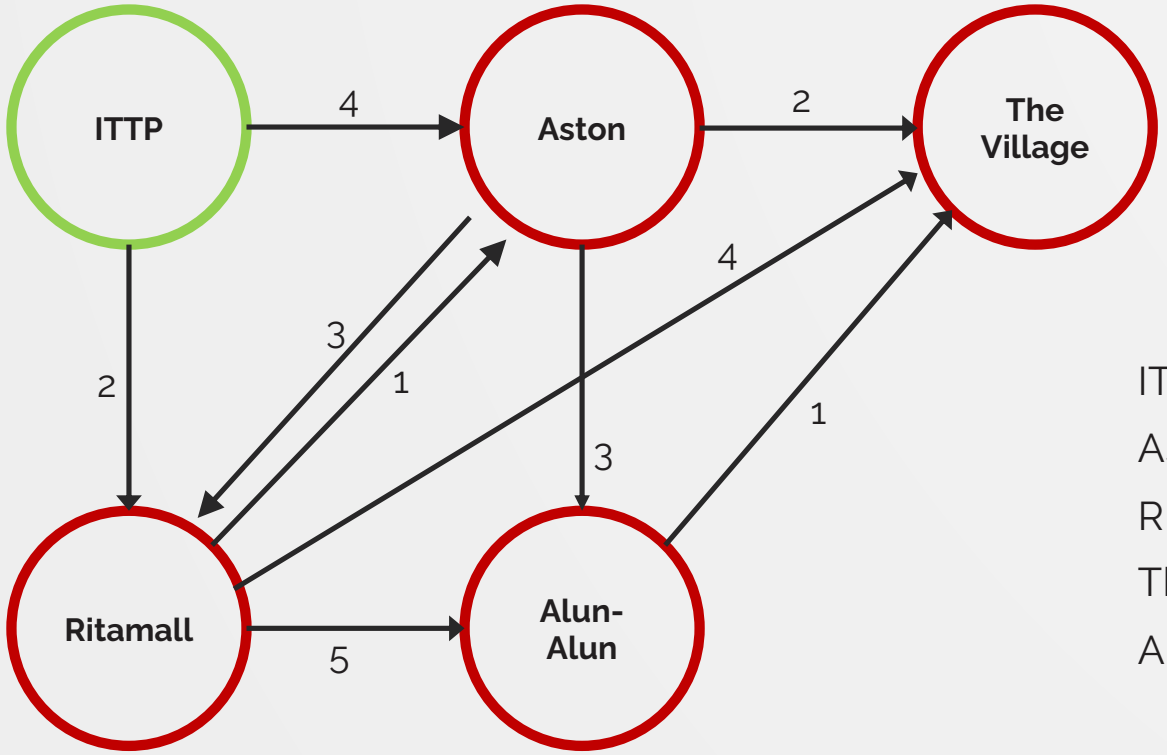
Mencari rute terpendek dari ITTP menuju node yang lain



Dijkstra's Algorithm

Mencari rute terpendek dari ITTP menuju node yang lain

- Step 1: memilih ITTP sebagai titik start



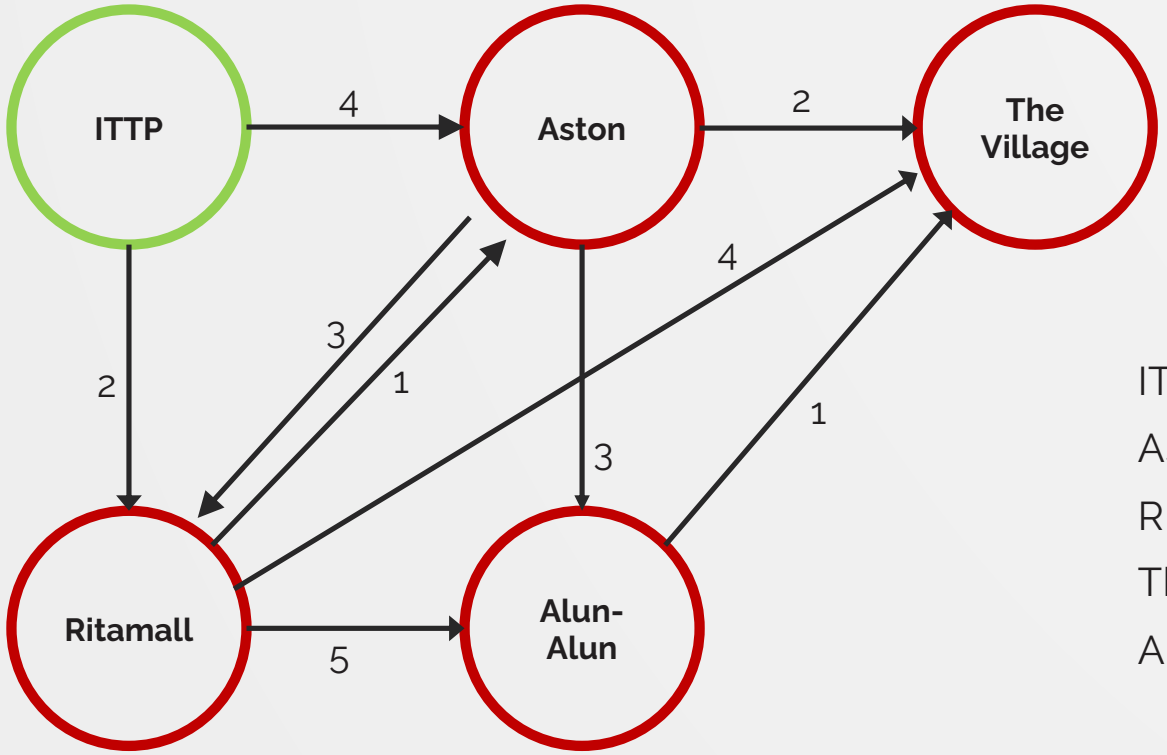
Unvisited node:
{ ~~ITTP~~, Aston, Ritamall, The Village, Alun-Alun }

ITTP	0
Aston	∞
Ritamall	∞
The Village	∞
Alun-Alun	∞

Dijkstra's Algorithm

Mencari rute terpendek dari ITTP menuju node yang lain

- Step 2: Visit node yang bisa dikunjungi



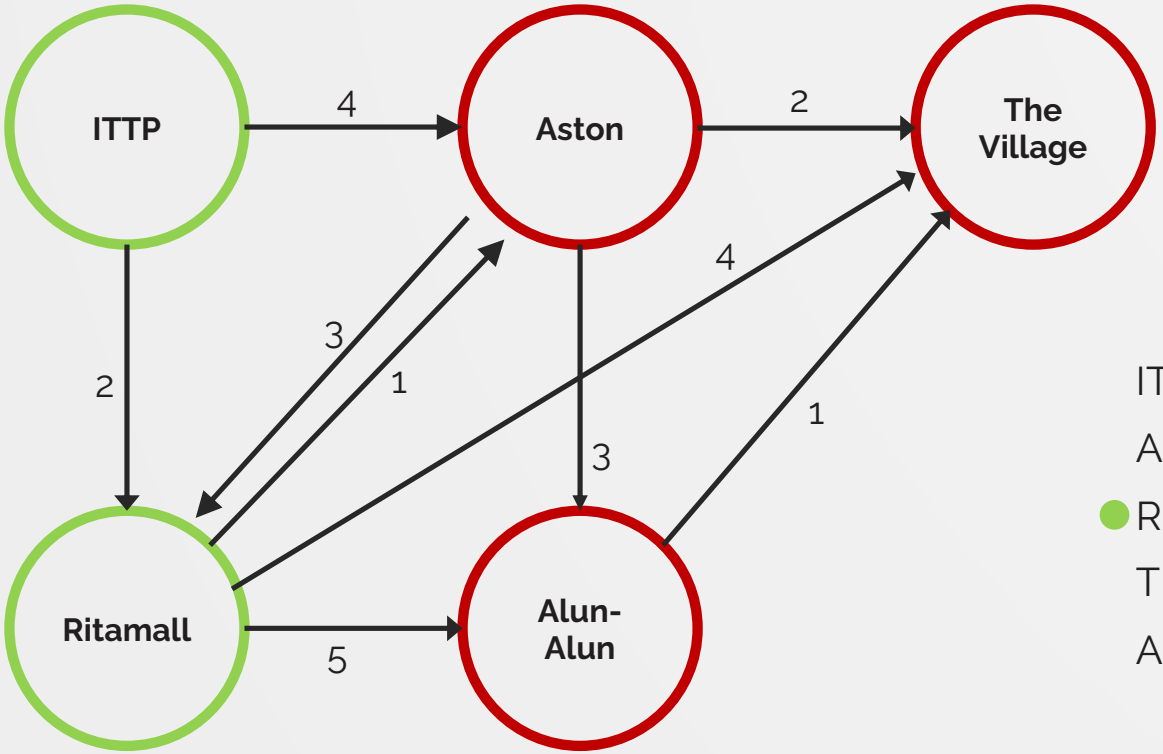
Unvisited node:
{ ~~ITTP~~, Aston, Ritamall, The Village, Alun-Alun }

ITTP	0
Aston	4
Ritamall	2
The Village	∞
Alun-Alun	∞

Dijkstra's Algorithm

Mencari rute terpendek dari ITTP menuju node yang lain

- Step 3: Lalu, pilih node yang terdekat, yaitu Ritamall



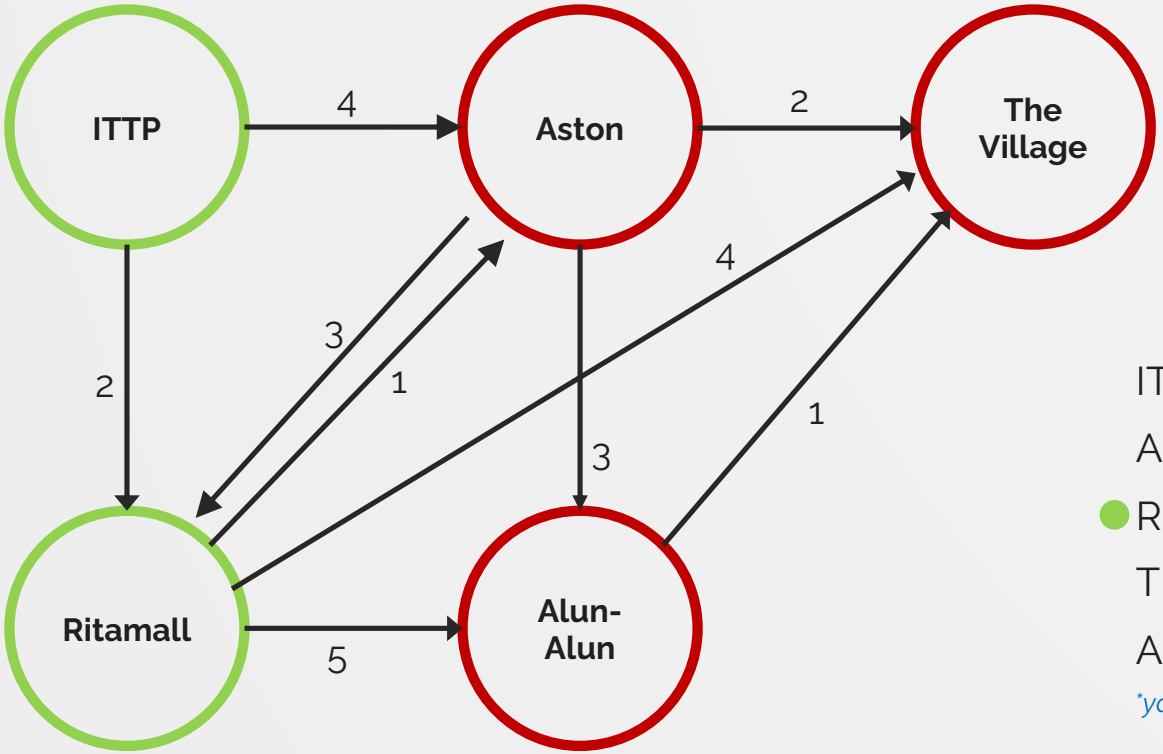
Unvisited node:
{ ~~ITTP~~, Aston, ~~Ritamall~~, The Village, Alun-Alun }

ITTP	0
Aston	4
Ritamall	2
The Village	∞
Alun-Alun	∞

Dijkstra's Algorithm

Mencari rute terpendek dari ITTP menuju node yang lain

- Step 4: Lakukan step 2 dan 3 secara berulang hingga seluruh node dicapai



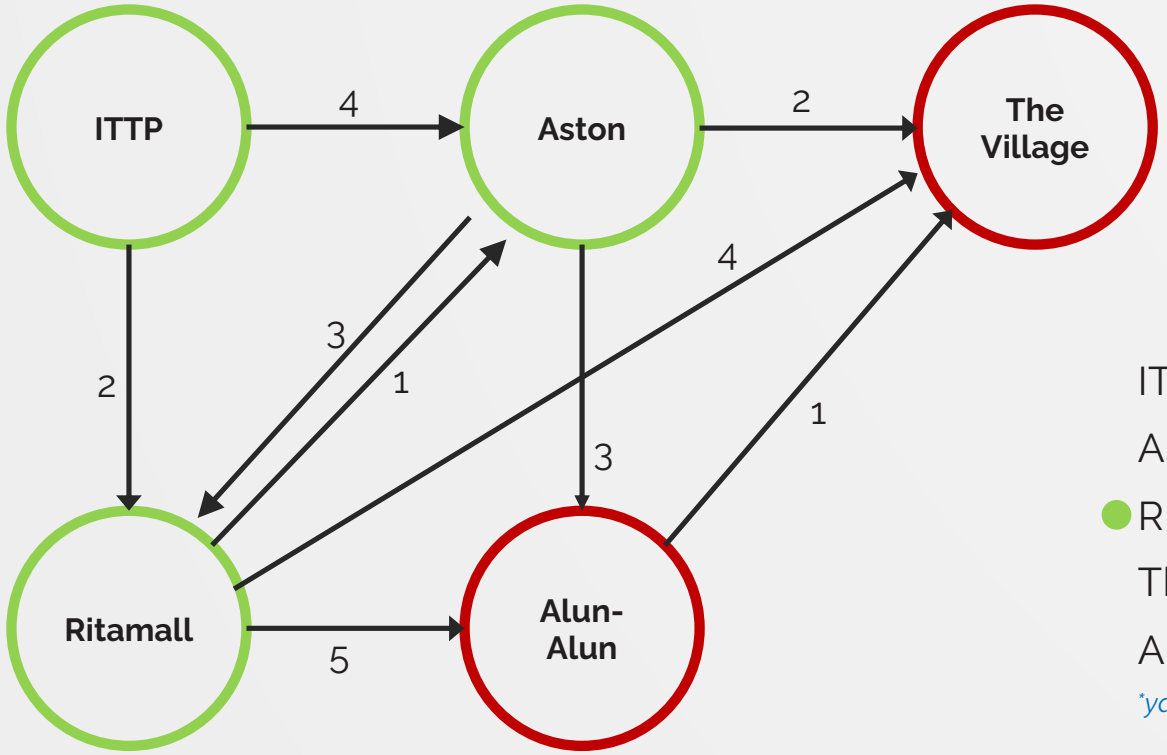
Unvisited node:
{ ~~ITTP~~, Aston, ~~Ritamall~~, The Village, Alun-Alun }

ITTP	0	ITTP	0
Aston	4	Aston	2+1
Ritamall	2	Ritamall	2
The Village	∞	The Village	2+4
Alun-Alun	∞	Alun-Alun	2+5
**yang lama*		**yang baru*	

Dijkstra's Algorithm

Mencari rute terpendek dari ITTP menuju node yang lain

- Step 4: Lakukan step 2 dan 3 secara berulang hingga seluruh node dicapai



Unvisited node:
{ ~~ITTP~~, ~~Aston~~, ~~Ritamall~~, The Village, Alun-Alun }

ITTP	0
Aston	4
Ritamall	2
The Village	∞
Alun-Alun	∞

*yang lama

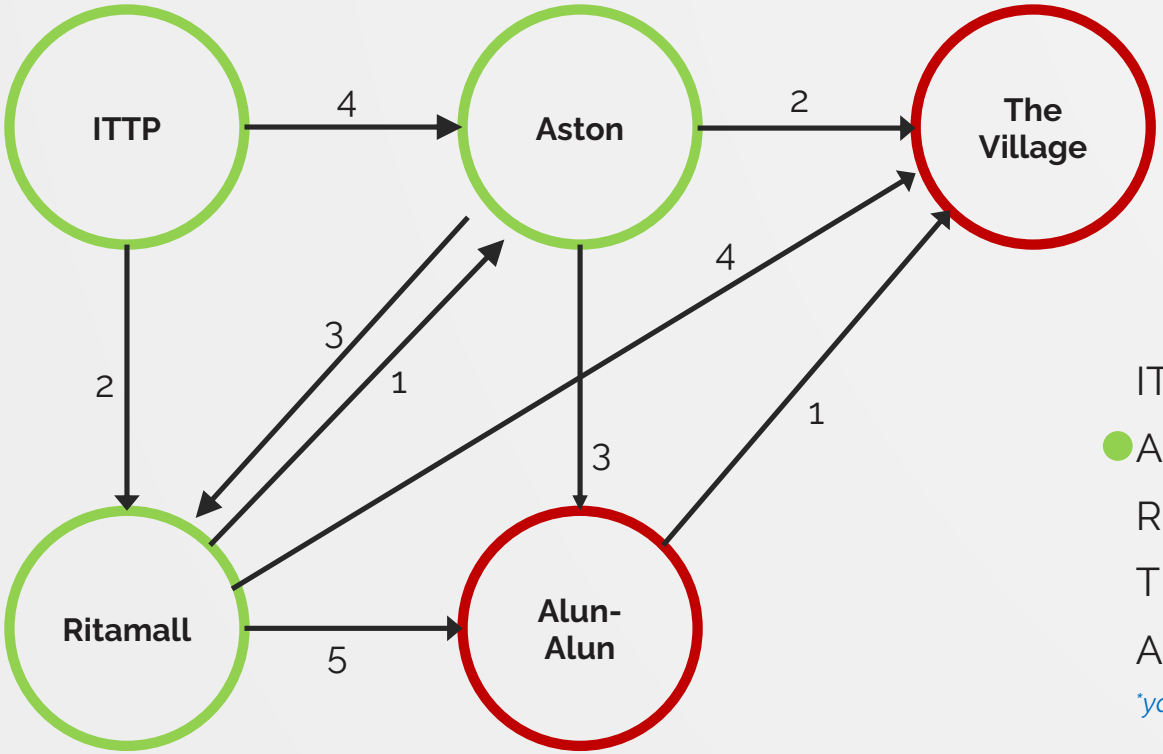
ITTP	0
Aston	3
Ritamall	2
The Village	6
Alun-Alun	7

*yang baru

Dijkstra's Algorithm

Mencari rute terpendek dari ITTP menuju node yang lain

- Step 4: Lakukan step 2 dan 3 secara berulang hingga seluruh node dicapai



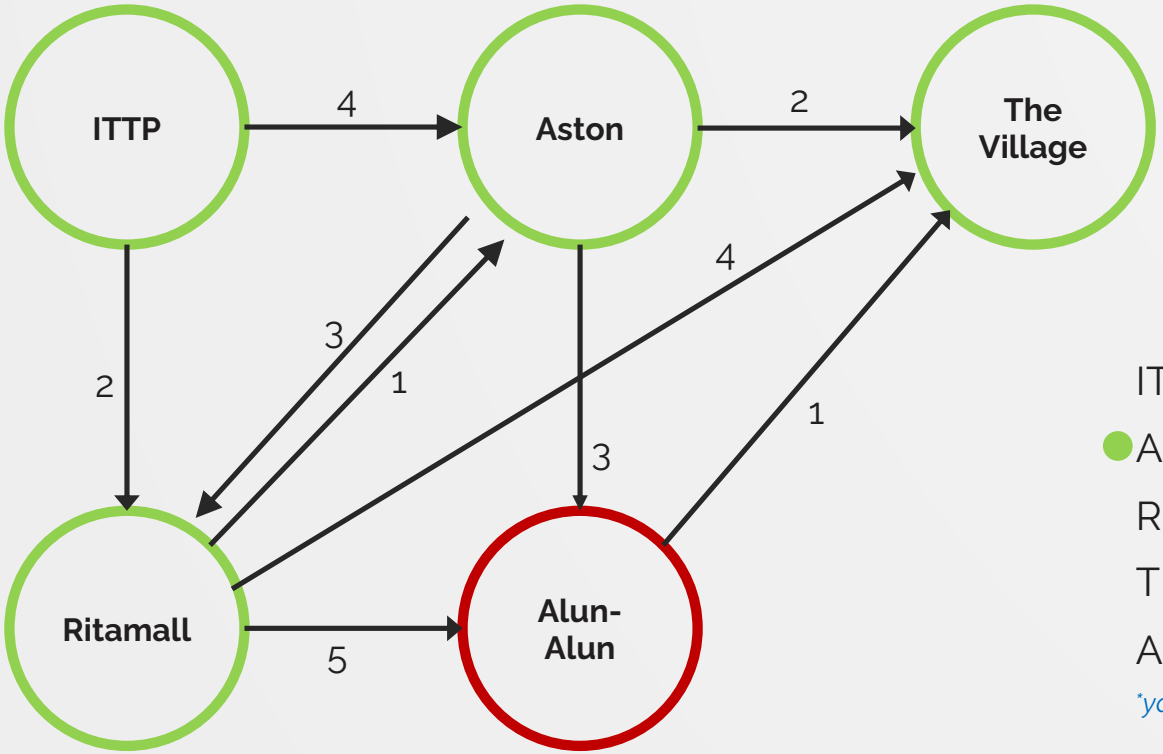
Unvisited node:
{ ~~ITTP~~, ~~Aston~~, ~~Ritamall~~, The Village, Alun-Alun }

ITTP	0	ITTP	0
Aston	3	Aston	3
Ritamall	2	Ritamall	2
The Village	6	The Village	3+2
Alun-Alun	7	Alun-Alun	3+3
*yang lama		*yang baru	

Dijkstra's Algorithm

Mencari rute terpendek dari ITTP menuju node yang lain

- Step 4: Lakukan step 2 dan 3 secara berulang hingga seluruh node dicapai



Unvisited node:
{ ~~ITTP~~, ~~Aston~~, ~~Ritamall~~, ~~The Village~~, Alun-Alun }

ITTP	0
Aston	3
Ritamall	2
The Village	6
Alun-Alun	7

*yang lama

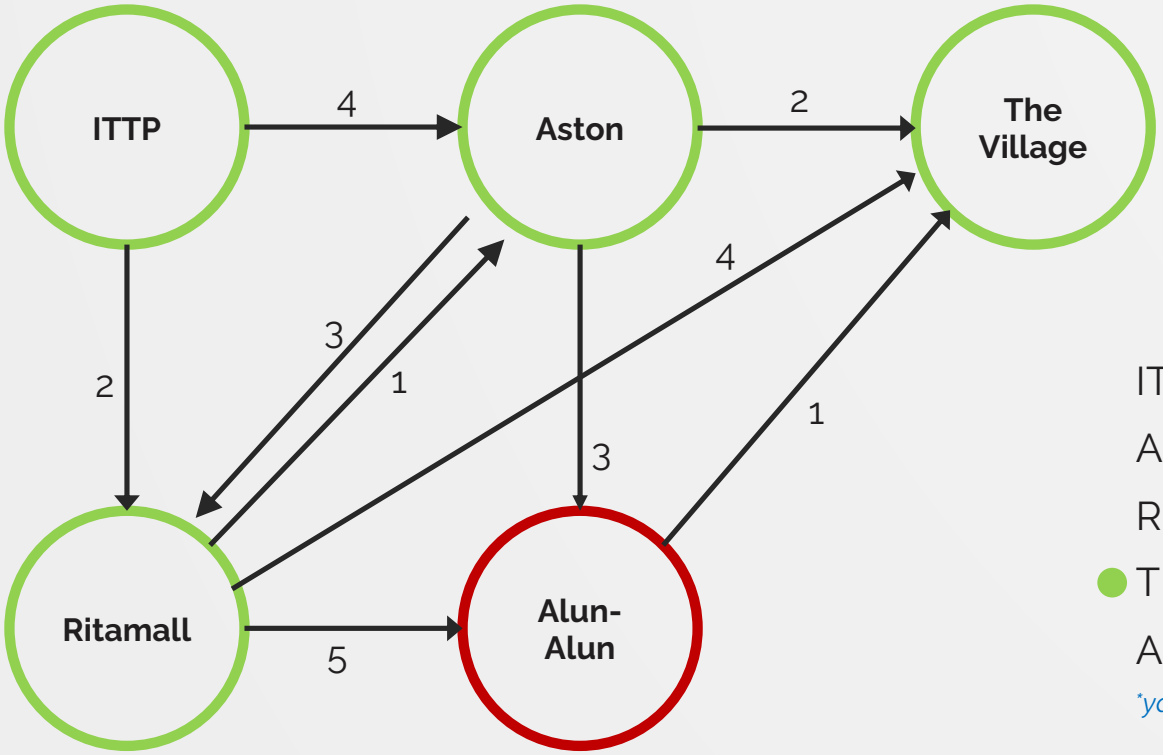
ITTP	0
Aston	3
Ritamall	2
The Village	5
Alun-Alun	6

*yang baru

Dijkstra's Algorithm

Mencari rute terpendek dari ITTP menuju node yang lain

- Step 4: Lakukan step 2 dan 3 secara berulang hingga seluruh node dicapai



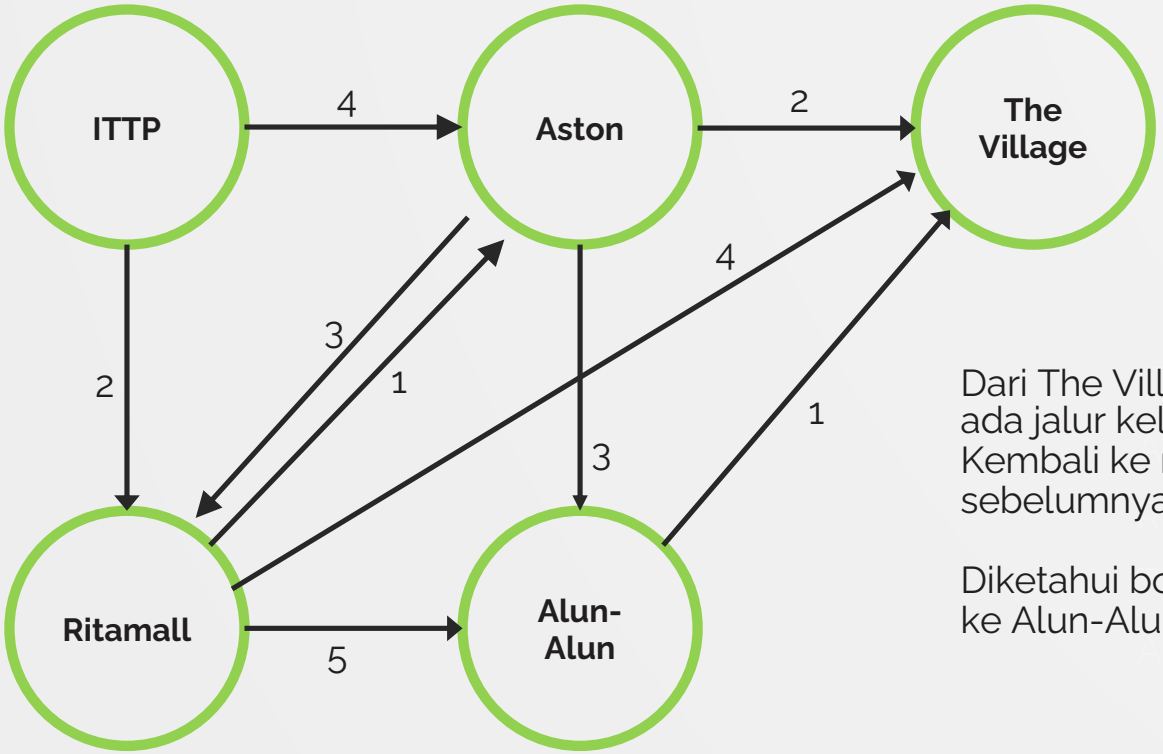
Unvisited node:
{ ~~ITTP~~, ~~Aston~~, ~~Ritamall~~, ~~The Village~~, Alun-Alun }

ITTP	0	ITTP	0
Aston	3	Aston	3
Ritamall	2	Ritamall	2
The Village	5	The Village	5
Alun-Alun	6	Alun-Alun	6
*yang lama		*yang baru	

Dijkstra's Algorithm

Mencari rute terpendek dari ITTP menuju node yang lain

- Step 4: Lakukan step 2 dan 3 secara berulang hingga seluruh node dicapai



Unvisited node:
{ ~~ITTP~~, ~~Aston~~, ~~Ritamall~~, ~~The Village~~, ~~Alun-Alun~~ }

Dari The Village karena tidak ada jalur keluar, maka Kembali ke node yang dipilih sebelumnya, yaitu Aston

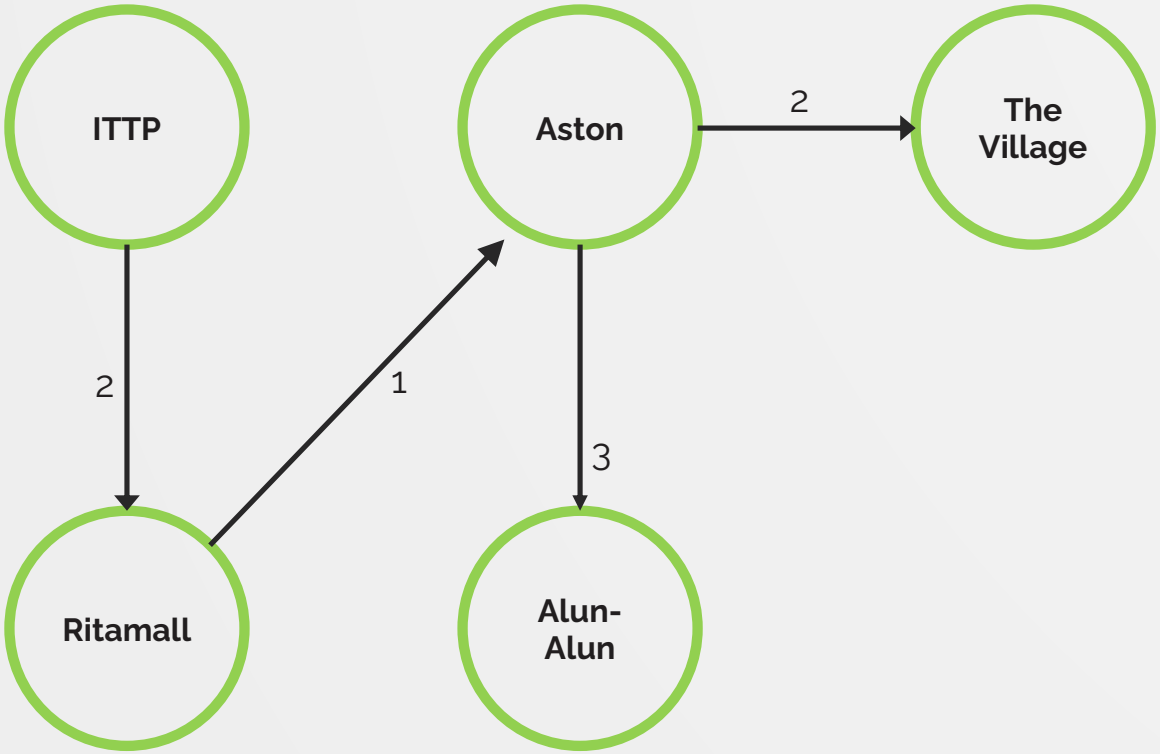
Diketahui bobot dari Aston ke Alun-Alun adalah 3+3

ITTP	0
Aston	3
Ritamall	2
The Village	5
Alun-Alun	6

Dijkstra's Algorithm

Mencari rute terpendek dari ITTP menuju node yang lain

- Step 5: Lakukan rekap jalur terpendek



Graph Path

Institut Teknologi Telkom Purwokerto

Challenge!

- Buatlah resume mengenai perbedaan *Dijkstra* vs *Prim's* vs *Kruskal* !

Institut Teknologi
Telkom
Purwokerto

TERIMA KASIH