Pohon

STIMKK106 Matematika Diskrit

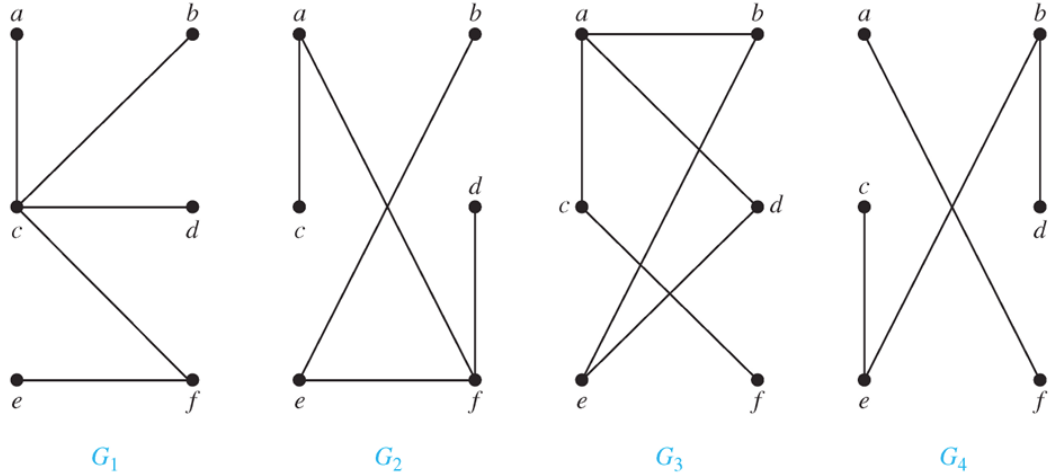
Made **Prastha** Nugraha, S.Kom., M.Kom.

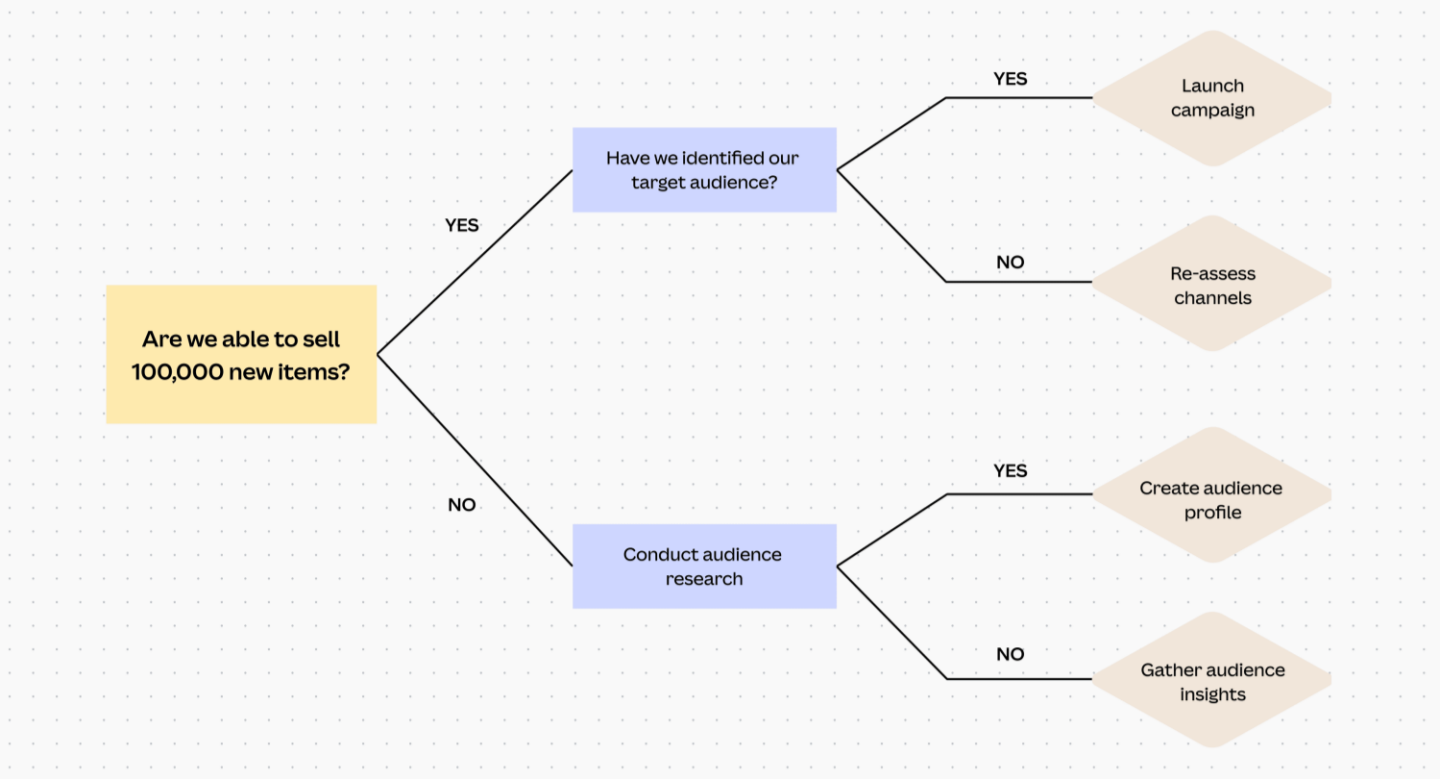
prasthanugraha@idbbali.ac.id

Pohon dan Hutan

Pohon (Tree)

- Pohon merupakan graf tidak berarah terhubung yang tidak mengandung sirkuit.
- Hutan (Forest) merupakan kumpulan pohon yang saling lepas atau graf tidak terhubung yang tidak mengandung sirkuit

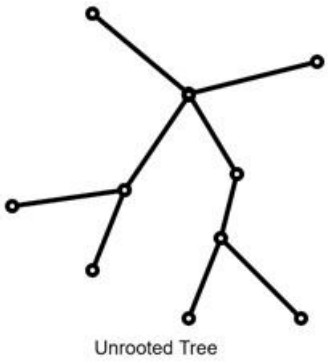


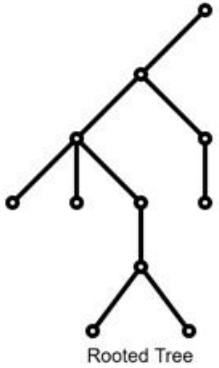

Are we able to sell 100,000 new items?
YES
NO
Have we identified our target audience?
Conduct audience research
YES
NO
YES
NO
Launch campaign
Re-assess channels
Create audience profile
Gather audience insights

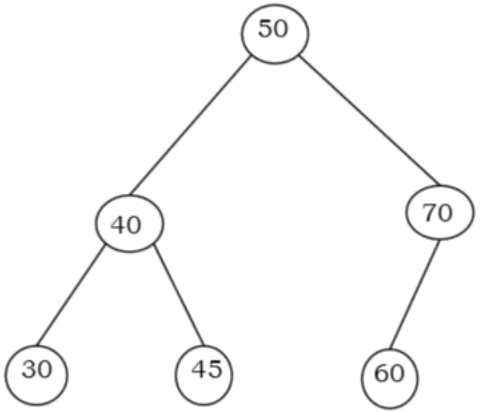
Jenis Pohon

Jenis Pohon

- Pohon dapat dibedakan berdasarkan jumlah cabang dan tingkatannya, yaitu:
  - Pohon Berakar
  - Pohon Biner
  - Pohon Rentang





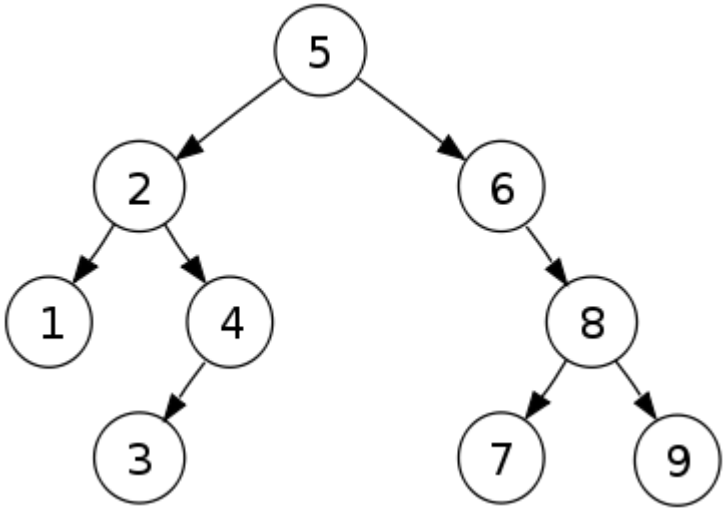


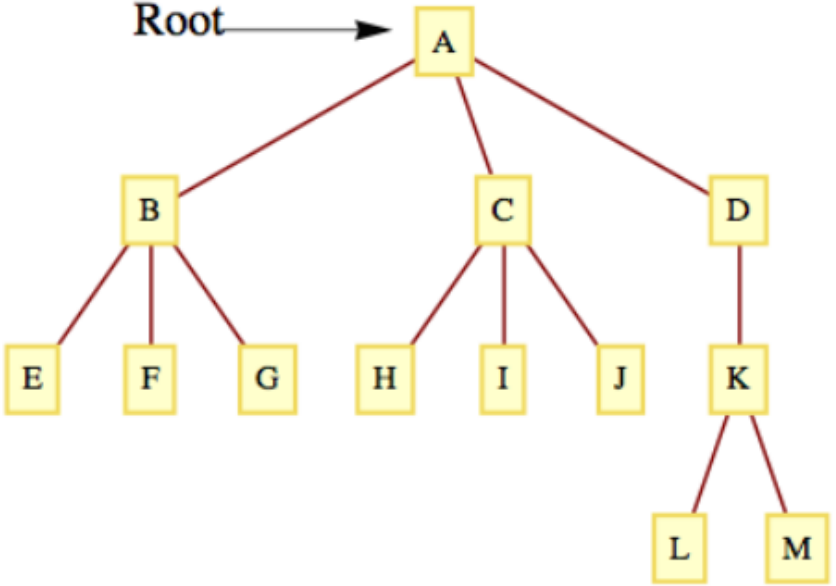
IDB BALI
INSTITUT DESAIN & BISNIS BALI

Pohon Berakar (Rooted Tree)

- Pohon berakar (Rooted Tree) adalah suatu pohon dimana ada satu titik akar (root) yang dikhususkan dari yang lain.
- Tingkat (level) suatu titik adalah banyaknya garis antara titik tersebut dengan akar.
- Tinggi (height) pohon adalah tingkat maksimum yang dimiliki oleh titik-titik pohon.
- Anak (children) dari titik v adalah semua titik yang berhubungan langsung dengan v, tetapi memiliki tingkat yang lebih tinggi dari v.
- Jika w adalah anak dari v, maka v disebut sebagai orang tua (pareng) dari w.
- Dua titik yang memiliki orang tua yang sama disebut saudara (sibling)

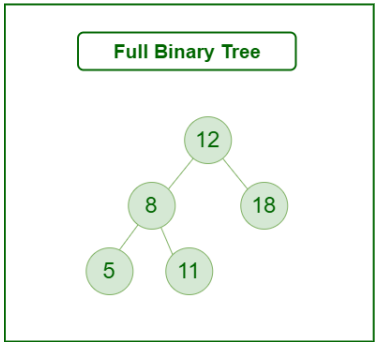
Pohon Berakar

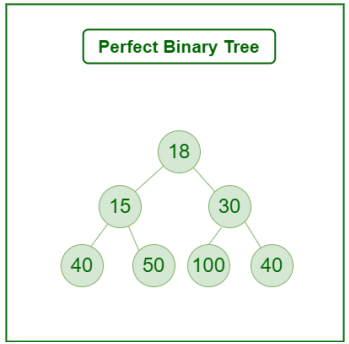


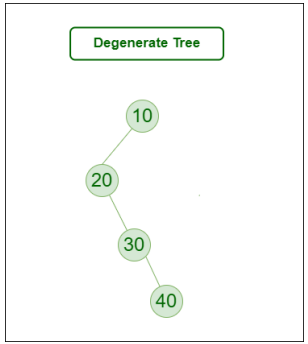


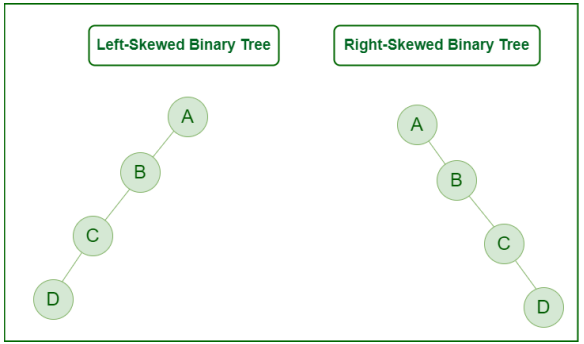
Pohon Biner (Binary Tree)

- Pohon Biner (Binary Tree) adalah pohon berakar yang setiap titiknya memiliki paling banyak 2 anak, yang disebut anak kiri (left child) dan anak kanan (right child).
- Pohon biner penuh (full binary tree) adalah pohon biner yang setiap titiknya memiliki tepat 2 anak.
- Pohon biner banyak digunakan untuk ekspresi aljabar maupun pencarian dan pengurutan data (searching and sorting).



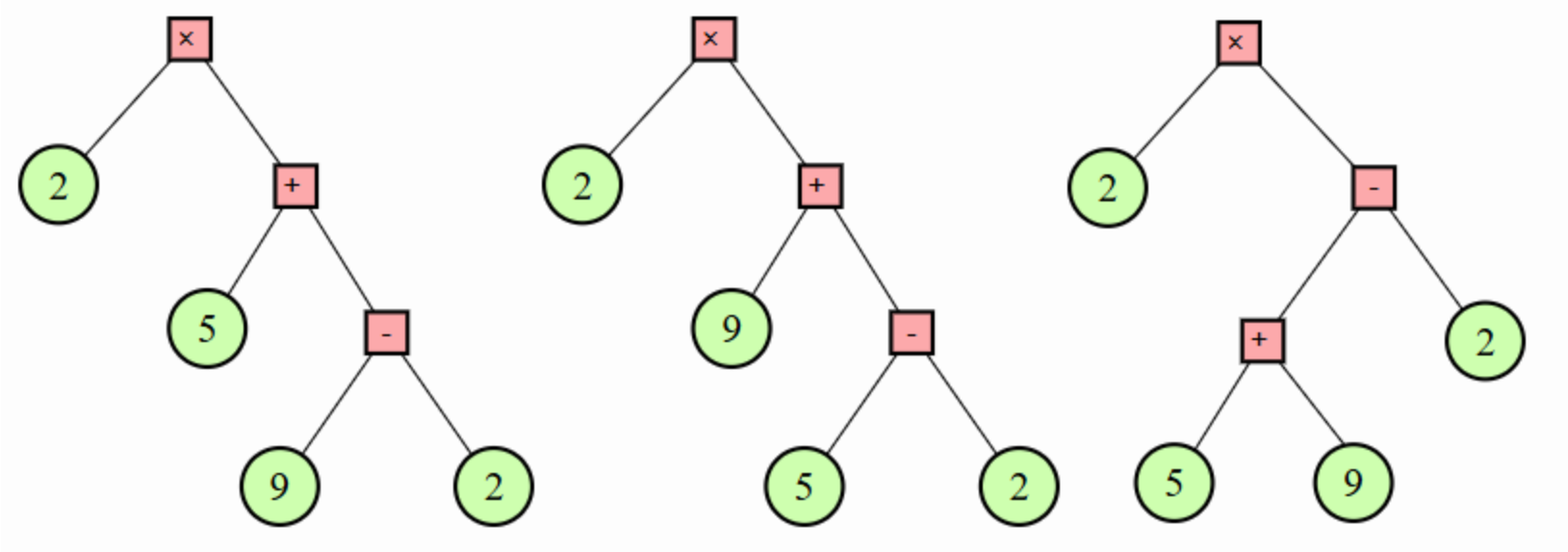






IDB BALI
INSTITUT DESAIN & BISNIS BALI

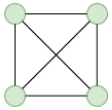
Pohon Biner

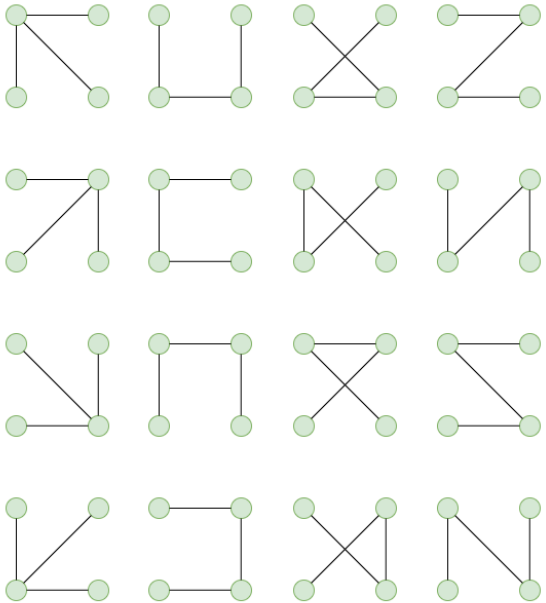


Pohon Rentang (Spanning Tree)

- Pohon rentang adalah bagian dari pohon yang memiliki keseluruhan titik dari pohon utama.

Graph =





Graf Berlabel

Graf Berlabel

- Label dalam graf dapat disebut sebagai jarak, biaya, panjang, dan sejenisnya.
- Graf berlabel adalah suatu graf tanpa garis pararel di mana setiap garisnya berhubungan dengan suatu bilangan riil tidak negatif yang menyatakan bobot garis tersebut.
- Graf berlabel secara umum dapat dibagi menjadi dua, yaitu:
  - Pohon rentang minimum
  - Path Minimum

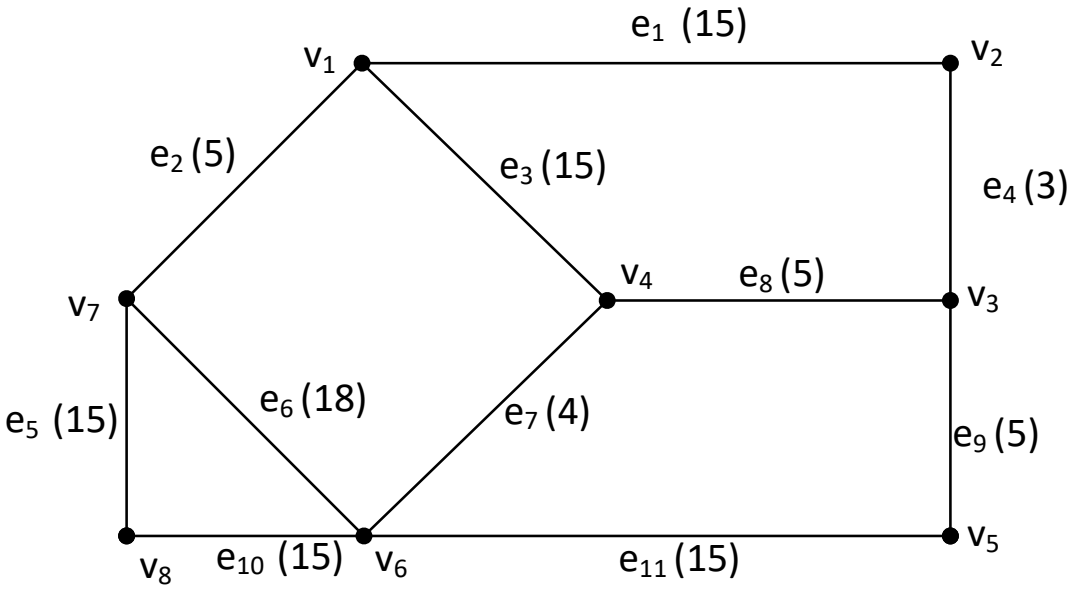
Pohon Rentang Minimum

- Dalam suatu provinsi terdapat 8 kota yang akan dihubungkan dengan jaringan listrik. Biaya pemasangan antar 2 kota terdapat pada tabel berikut.

Garis	Kota yang dihubungkan	Biaya per satuan
e_4	v_2 - v_3	3
e_7	$v_4 - v_6$	4
e_2	$v_1 - v_7$	5
e_8	$v_3 - v_4$	5
e_9	$v_3 - v_5$	5
e_1	$v_1 - v_2$	15

Garis	Kota yang dihubungkan	Biaya per satuan
e_3	$v_1 - v_4$	15
e_{10}	$v_6 - v_8$	15
e_5	$v_7 - v_8$	15
e_{11}	$v_5 - v_6$	15
e_6	$v_6 - v_7$	18

Pohon Rentang Minimum

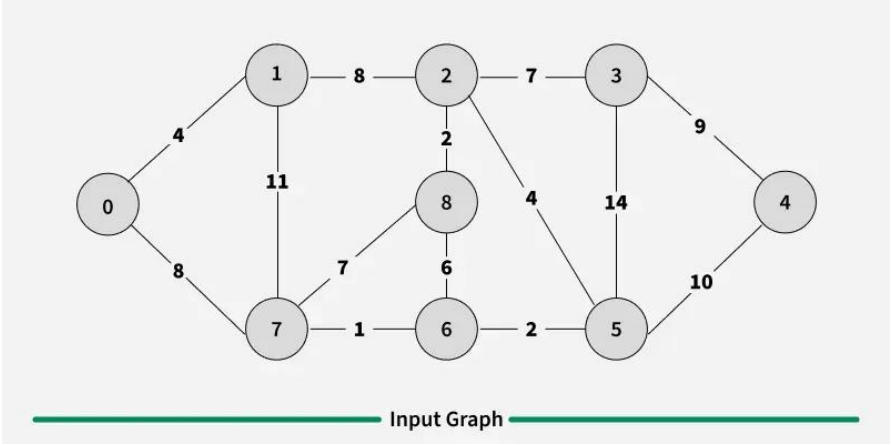


Pohon Rentang Minimum

$$
\begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \\ v_7 \\ v_8 \end{array}
\begin{array}{c}
\begin{array}{cccccccc} v_1 & v_2 & v_3 & v_4 & v_5 & v_6 & v_7 & v_8 \end{array} \\
\begin{pmatrix}
0 & 15 & \infty & 15 & \infty & \infty & 5 & \infty \\
15 & 0 & 3 & \infty & \infty & \infty & \infty & \infty \\
\infty & 3 & 0 & 5 & 5 & \infty & \infty & \infty \\
15 & \infty & 5 & 0 & \infty & 4 & \infty & \infty \\
\infty & \infty & 5 & \infty & 0 & 15 & \infty & \infty \\
\infty & \infty & \infty & 4 & 15 & 0 & 18 & 15 \\
\infty & \infty & \infty & \infty & \infty & 18 & 0 & 15 \\
\infty & \infty & \infty & \infty & \infty & 15 & 15 & 0
\end{pmatrix}
\end{array}
$$

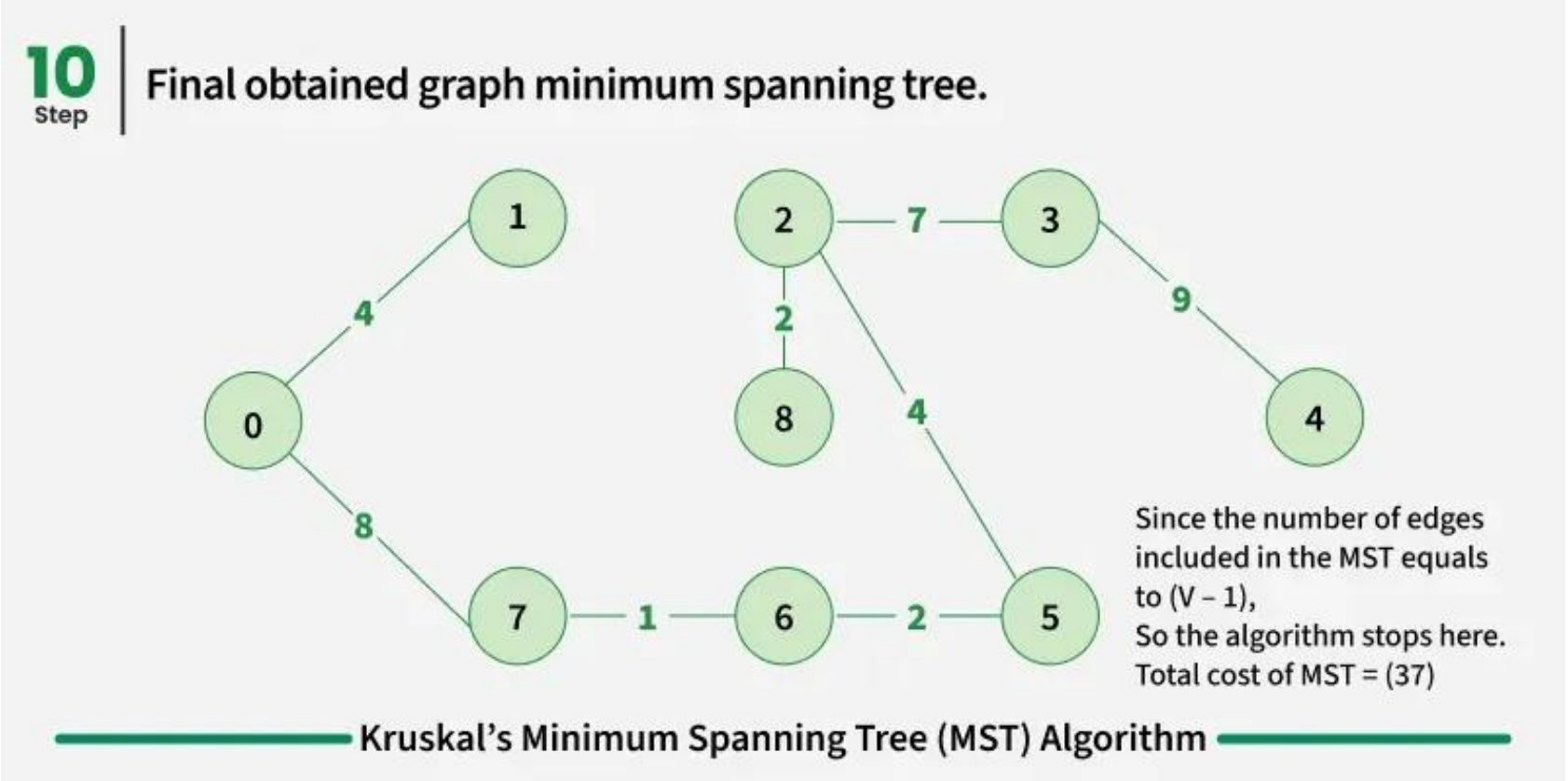
Pohon Rentang Minimum (Kruskal)

1. Algoritma Kruskal dimulai dengan menentukan garis dengan beban terkecil.
2. Bila garis dengan beban terkecil tidak membentuk sirkuit, maka garis tersebut akan dimasukkan ke dalam pohon.
3. Apabila garis dengan beban terkecil membentuk sirkuit, maka garis tersebut dibuang dan dilanjutkan ke garis terkecil berikutnya
4. Langkah 1-3 diulang hingga seluruh garis telah dilewati.



Sumber: https://www.geeksforgeeks.org/kruskals-minimum-spanning-tree-algorithm-greedy-algo-2/

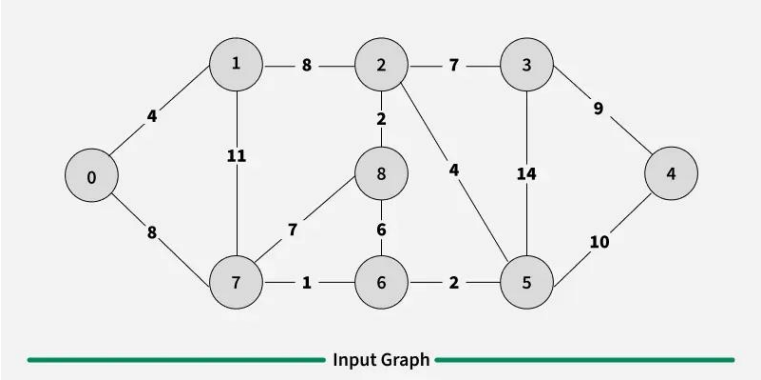
Pohon Rentang Minimum (Kruskal)



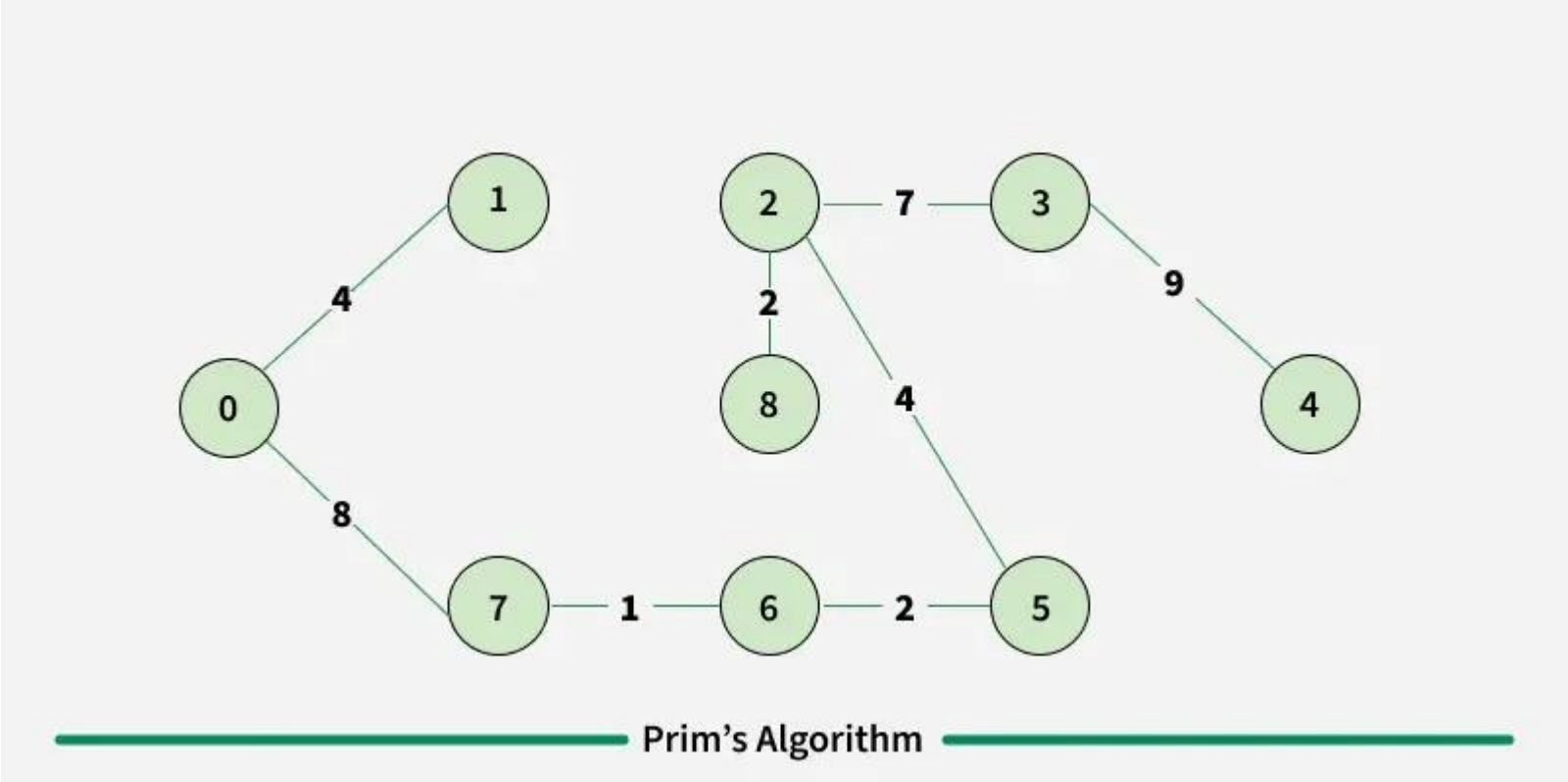
Sumber: https://www.geeksforgeeks.org/kruskals-minimum-spanning-tree-algorithm-greedy-algo-2/

Pohon Rentang Minimum (Prim)

1. Tentukan sebuah titik secara acak awal pohon.
2. Ikuti langkah 3 hingga 5 sampai semua titik sudah termasuk dalam MST (titik yang belum termasuk dikenal sebagai *fringe vertex*).
3. Temukan garis yang menghubungkan setiap simpul pohon (*tree vertex*) dengan *fringe vertices*.
4. Temukan garis dengan bobot minimum di antara garis-garis tersebut.
5. Tambahkan garis yang dipilih ke dalam pohon.

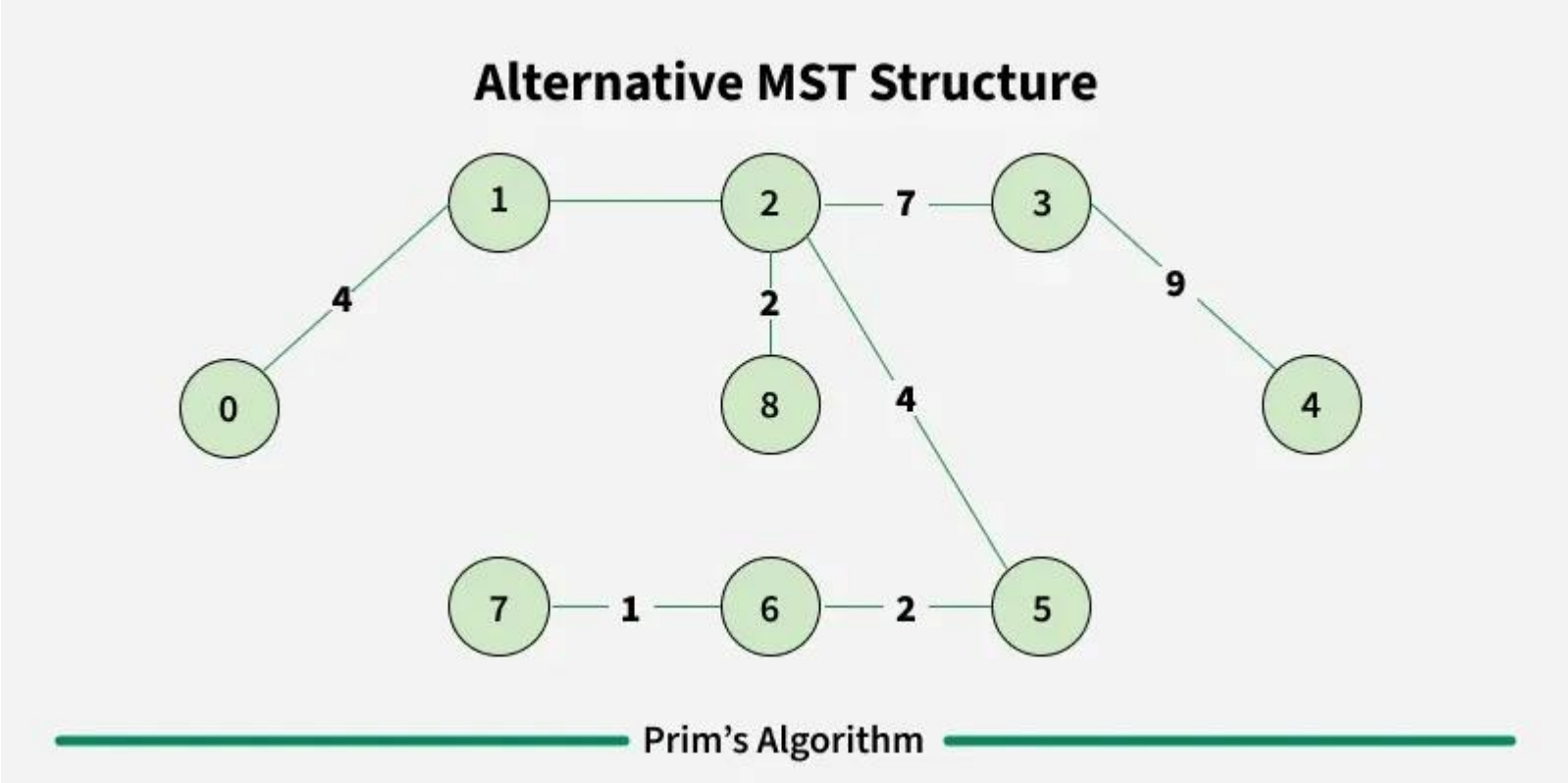


Pohon Rentang Minimum (Prim)



Sumber: https://www.geeksforgeeks.org/prims-minimum-spanning-tree-mst-greedy-algo-5/

Pohon Rentang Minimum (Prim)



Sumber: https://www.geeksforgeeks.org/prims-minimum-spanning-tree-mst-greedy-algo-5/

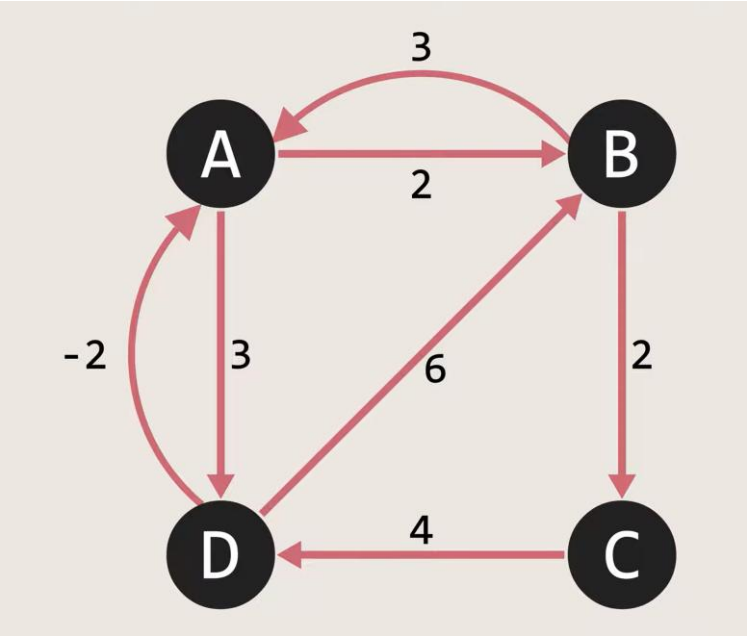
Path Minimum

- Sesuai namanya, path minimum memiliki arti untuk memilih jalur dengan beban terpendek dari titik awal ke titik tujuan.
- Beberapa algoritma untuk mencari path minimum yang sering digunakan yaitu:
  - Algoritma Warshall
  - Algoritma Djikstraa

Path Minimum (Warshall)

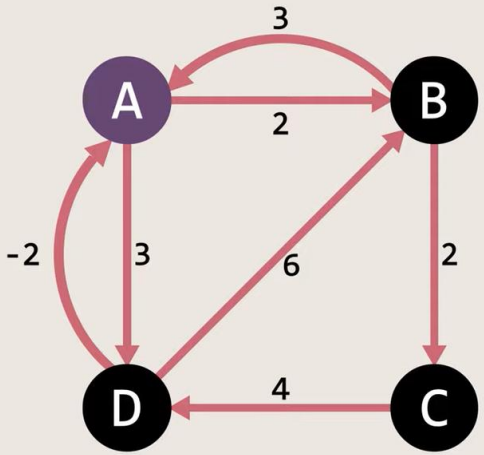
1. Mulailah dengan memperbarui matriks jarak
2. Lakukan iterasi melalui setiap simpul, satu per satu. Untuk setiap simpul
3. Saat kita memilih simpul nomor k sebagai simpul perantara, kita sudah mempertimbangkan simpul (0,1,...,k-1) sebagai simpul perantara sebelumnya
4. Untuk setiap pasangan simpul sumber dan tujuan (i,j) ada dua kemungkinan kasus:
    1. k **bukanlah** simpul perantara dalam jalur terpendek dari i ke j. Kita mempertahankan nilai dari dist[i][j] apa adanya.
    2. k **merupakan** simpul perantara dalam jalur terpendek dari i ke j. Kita memperbarui nilai dist[i][j] menjadi dist[i][k] + dist[k][j], jika nilai dist[i][j] lebih besar dari dist[i][k] + dist[k][j].
5. Ulangi proses ini untuk setiap simpul k hingga semua kemungkinan perantara telah dipertimbangkan.

Path Minimum (Warshall)



Sumber: https://www.youtube.com/watch?v=sLg6Leb-xt0&ab_channel=ByteQuest

Path Minimum (Warshall)



M_0

	A	B	C	D
A	0	2	∞	3
B	3	0	2	∞
C	∞	∞	0	4
D	-2	6	∞	0

T_0

	A	B	C	D
A	-	A	-	A
B	B	-	B	-
C	-	-	-	C
D	D	D	-	-

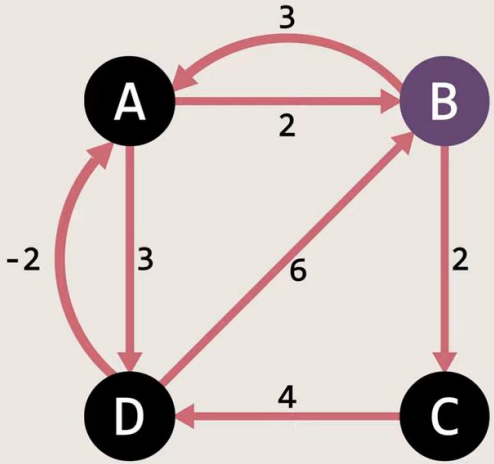
$$\infty \approx -2 + \infty$$

	A	B	C	D
A	0	2	∞	3
B	3	0	2	6
C	∞	∞	0	4
D	-2	0	∞	0

	A	B	C	D
A	-	A	-	A
B	B	-	B	A
C	-	-	-	C
D	D	A	-	-

Sumber: https://www.youtube.com/watch?v=sLg6Leb-xt0&ab_channel=ByteQuest

Path Minimum (Warshall)



M_1

	A	B	C	D
A	0	2	∞	3
B	3	0	2	6
C	∞	∞	0	4
D	-2	0	∞	0

T_1

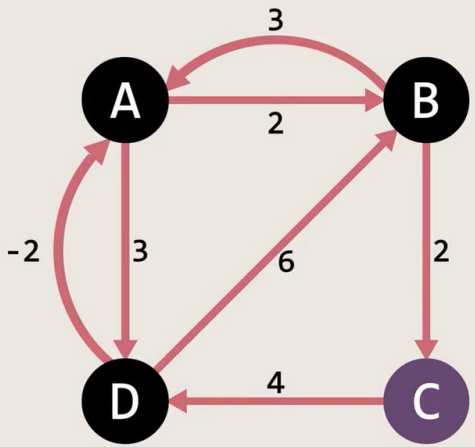
	A	B	C	D
A	-	A	-	A
B	B	-	B	A
C	-	-	-	C
D	D	A	-	-

	A	B	C	D
A	0	2	4	3
B	3	0	2	6
C	∞	∞	0	4
D	-2	0	2	0

	A	B	C	D
A	-	A	B	A
B	B	-	B	A
C	-	-	-	C
D	D	A	B	-

Sumber: https://www.youtube.com/watch?v=sLg6Leb-xt0&ab_channel=ByteQuest

Path Minimum (Warshall)



M_2

	A	B	C	D
A	0	2	4	3
B	3	0	2	6
C	∞	∞	0	4
D	-2	0	2	0

T_2

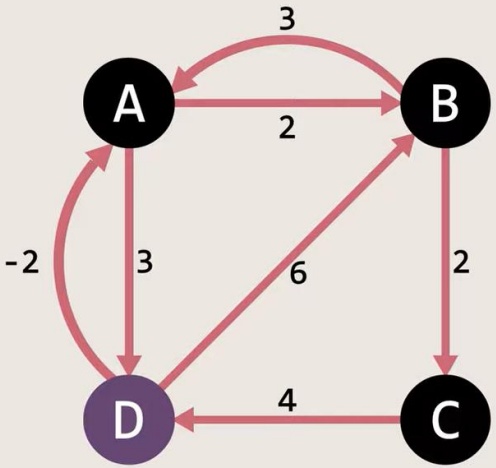
	A	B	C	D
A	-	A	B	A
B	B	-	B	A
C	-	-	-	C
D	D	A	B	-

	A	B	C	D
A	0	2	4	3
B	3	0	2	6
C	∞	∞	0	4
D	-2	0	2	0

	A	B	C	D
A	-	A	B	A
B	B	-	B	A
C	-	-	-	C
D	D	A	B	-

Sumber: https://www.youtube.com/watch?v=sLg6Leb-xt0&ab_channel=ByteQuest

Path Minimum (Warshall)



M_3

	A	B	C	D
A	0	2	4	3
B	3	0	2	6
C	∞	∞	0	4
D	-2	0	2	0

T_3

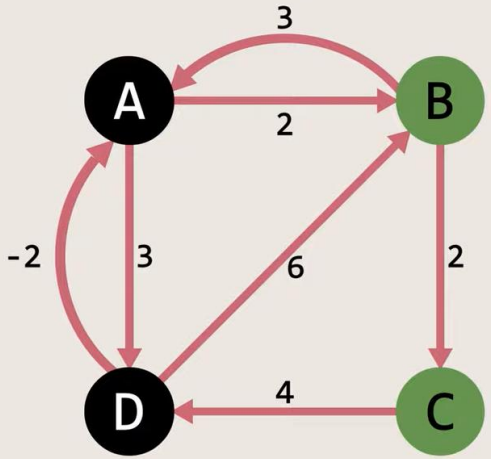
	A	B	C	D
A	-	A	B	A
B	B	-	B	A
C	-	-	-	C
D	D	A	B	-

	A	B	C	D
A	0	2	4	3
B	3	0	2	6
C	2	4	0	4
D	-2	0	2	0

	A	B	C	D
A	-	A	B	A
B	B	-	B	A
C	D	A	-	C
D	D	A	B	-

Sumber: https://www.youtube.com/watch?v=sLg6Leb-xt0&ab_channel=ByteQuest

Path Minimum (Warshall)



M_4

	A	B	C	D
A	0	2	4	3
B	3	0	2	6
C	2	4	0	4
D	-2	0	2	0

T_4

	A	B	C	D
A	-	A	B	A
B	B	-	B	A
C	D	A	-	C
D	D	A	B	-

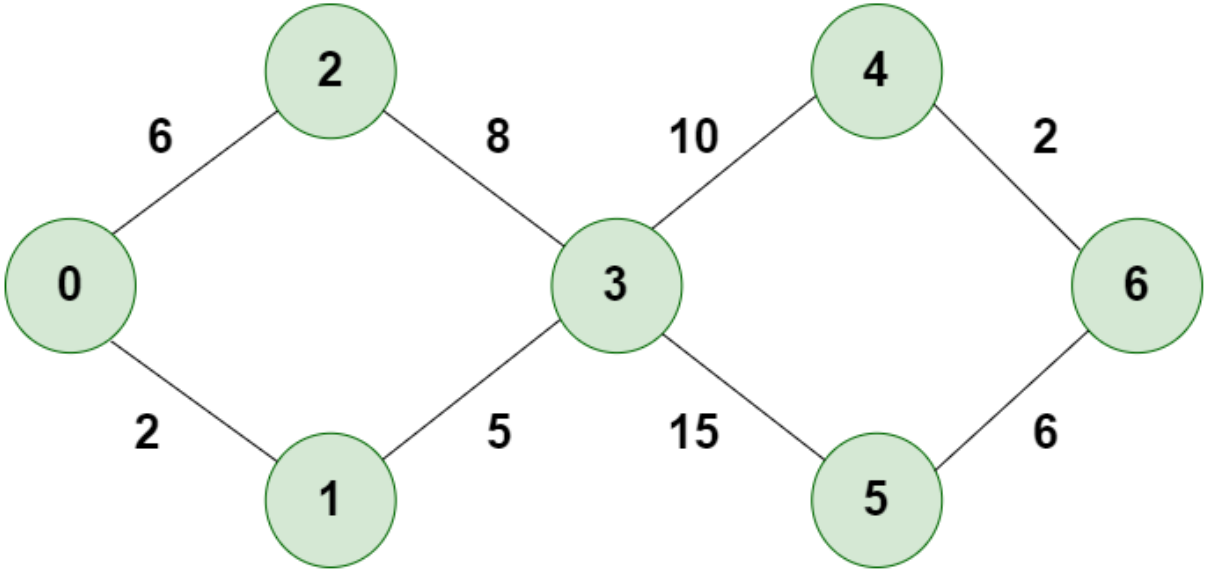
C → D → A → B

Sumber: https://www.youtube.com/watch?v=sLg6Leb-xt0&ab_channel=ByteQuest

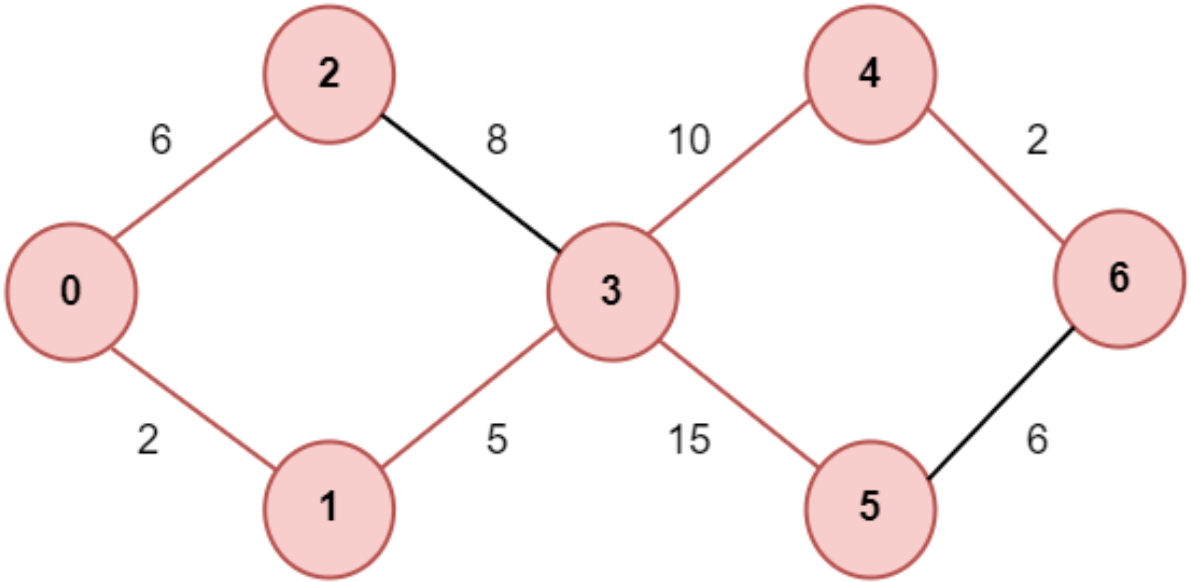
Path Minimum (Djikstraa)

- Algoritma Djikstraa merupakan salah satu algoritma utuk mencari rute terpendek dalam suatu graf.
- Pemilihan rute terpendek dengan cara membandingkan bobot dari dua garis yang sejajar.
    1. Tandai titik sumber (source node) dengan jarak 0 dan simpul lainnya dengan jarak tidak terhingga (infinity).
    2. Pilih titik yang belum dikunjungi dengan jarak terkecil, lalu jadikan titik tersebut sebagai simpul saat ini.
    3. Untuk setiap tetangga (N) dari simpul saat ini, hitung total jarak baru dengan menambahkan jarak simpul saat ini dengan bobot rusuk yang menghubungkannya. Jika total jarak ini lebih kecil dari jarak yang tercatat pada N, perbarui jarak N dengan total jarak baru tersebut.
    4. Tandai titik saat ini sebagai telah dikunjungi.
    5. Ulangi dari langkah kedua jika masih ada titik yang belum dikunjungi.

Path Minimum (Djikstraa)



Path Minimum (Djikstraa)



Unvisited Nodes

{0,1,2,3,4,5,6}

Distance:

0: 0 ✓
1: 2 ✓
2: 6 ✓
3: 7 ✓
4: 17 ✓
5: 22 ✓
6: 19 ✓

Source: https://www.geeksforgeeks.org/introduction-to-dijkstras-shortest-path-algorithm/

Terima Kasih!

Thank you!