

**TUGAS 6 PRAKTIKUM  
MINIMUM SPANNING TREE**

**MATA KULIAH ANALISIS ALGORITMA  
D10G.4205 & D10K.0400601**



**Disusun Oleh :**

Muhammad Ariq Farhansyah Mutyara (140810170053)

**PROGRAM STUDI S1 TEKNIK INFORMATIKA FAKULTAS  
MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS PADJADJARAN  
JATINANGOR  
2019**

## Definisi

Pohon rentangan adalah total weight (bobot) semua edge dalam pohon dan tidak jarang pohon rentangan bisa berjumlah lebih dari satu. Istilah Minimum spanning tree merujuk kepada pohon rentangan yang memiliki kompleksitas terendah di antara semua pohon rentangan. Sama halnya dengan pohon rentangan pada umumnya, (pohon rentangan minimum) juga bisa berjumlah lebih dari satu.

## Algoritma Prim

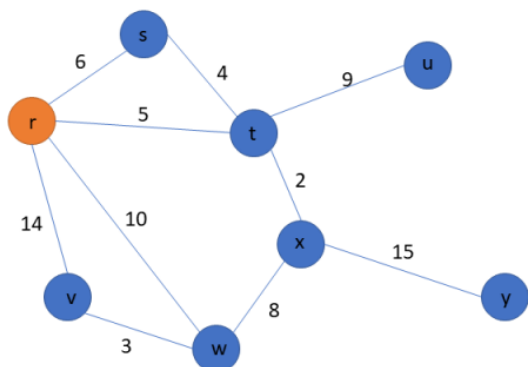
Algoritma Prim adalah algoritma untuk mencari *Minimum Spanning Tree*. Algoritma ini juga menggunakan pendekatan greedy untuk menemukan *Minimum Spanning Tree*. Dalam algoritma Prim, kita mengembangkan spanning tree dari posisi asal. Bagian yang berbeda dari algoritma Kruskal adalah ditambahkannya edge ke growing spanning tree; sedangkan pada algoritma Prim, verteks yang ditambahkan ke growing spanning tree.

## Langkah Algoritma

1. Terdapat dua disjoint set verteks. Salah satu disjoint set berisi verteks dalam growing spanning tree, dan yang lainnya berisi verteks yang tidak dalam growing spanning tree.
2. Pilih verteks paling ringan yang terhubung dengan growing spanning tree dan yang tidak dalam growing spanning tree lalu tambahkan ke growing spanning tree. Hal ini dapat dilakukan menggunakan Priority Queue. Sisipkan verteks yang ada dalam growing spanning tree ke dalam Priority Queue.
3. Pastikan tidak ada siklus yang terjadi. Untuk melakukan hal tersebut, tandai node yang sudah dipilih dan hanya sisipkan node yang tidak bertanda ke dalam Priority Queue.

### TAHAP 1

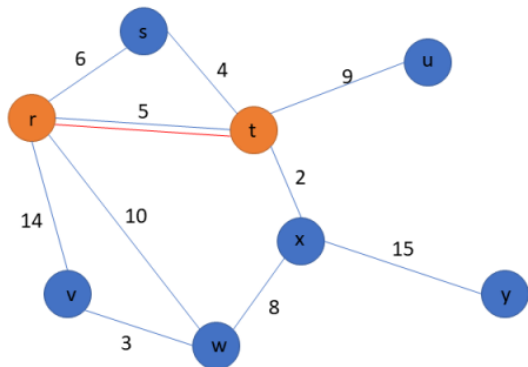
Simpul r sebagai simpul start.



VERTEX	KEY	PARENT
r	0	
s		
t		
u		
v		
w		
x		
y		

## TAHAP 2

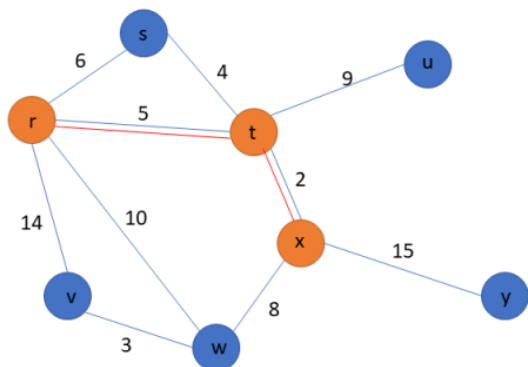
Memilih simpul t karena memiliki nilai jarak terkecil dibandingkan simpul lainnya yang terhubung ke simpul r.



VERTEX	KEY	PARENT
r	0	
s		
t	5	r
u		
v		
w		
x		
y		

## TAHAP 3

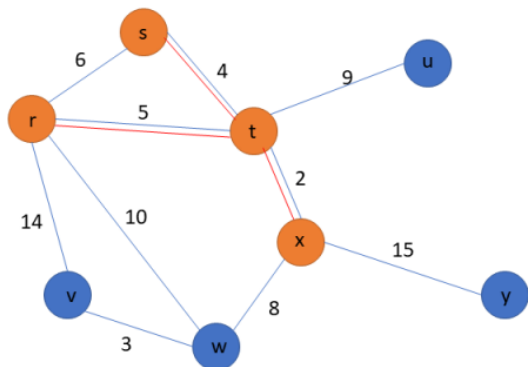
Memilih simpul x karena memiliki nilai jarak terkecil dibandingkan simpul lainnya yang terhubung ke simpul t.



VERTEX	KEY	PARENT
r	0	
s		
t	5	r
u		
v		
w		
x	2	t
y		

## TAHAP 4

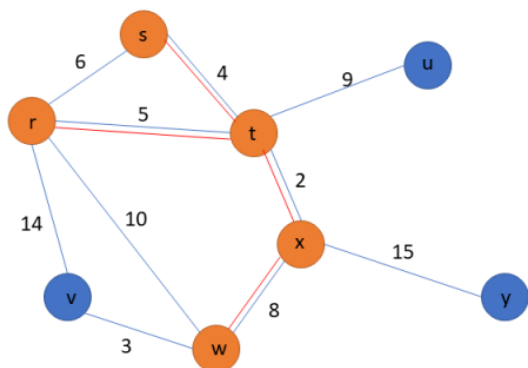
Memilih simpul s karena memiliki nilai jarak terkecil dibandingkan simpul lainnya yang terhubung ke simpul t atau x.



VERTEX	KEY	PARENT
r	0	
s	4	t
t	5	r
u		
v		
w		
x	2	t
y		

#### TAHAP 5

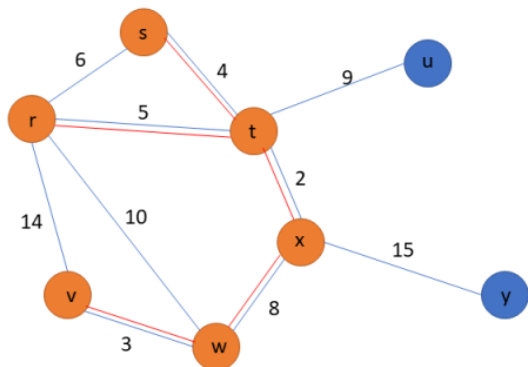
Memilih simpul w karena memiliki nilai jarak terkecil dibandingkan simpul lainnya yang terhubung ke simpul t atau x.



VERTEX	KEY	PARENT
r	0	
s	4	t
t	5	r
u		
v		
w	8	x
x	2	t
y		

#### TAHAP 6

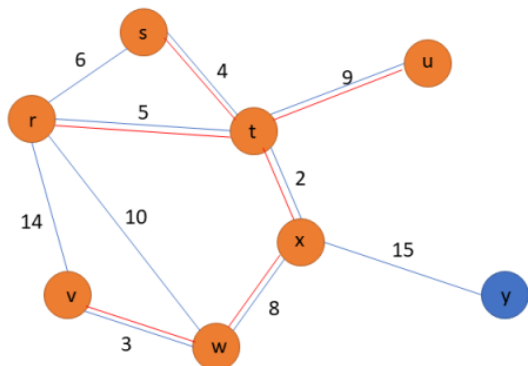
Memilih simpul v karena memiliki nilai jarak terkecil dibandingkan simpul lainnya yang terhubung ke simpul t, x, atau w.



VERTEX	KEY	PARENT
r	0	
s	4	t
t	5	r
u		
v	3	w
w	8	x
x	2	t
y		

#### TAHAP 7

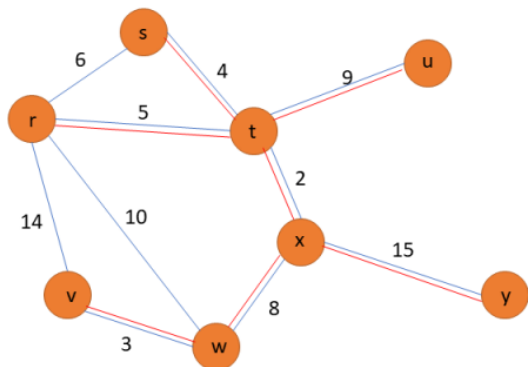
Memilih simpul u karena memiliki nilai jarak terkecil dibandingkan simpul lainnya yang terhubung ke simpul t atau x.



VERTEX	KEY	PARENT
r	0	
s	4	t
t	5	r
u	9	t
v	3	w
w	8	x
x	2	t
y		

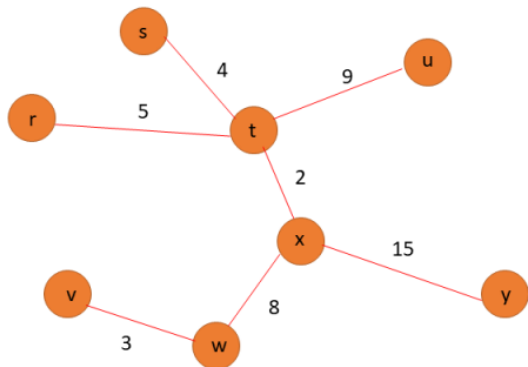
#### TAHAP 8

Memilih simpul y karena memiliki nilai jarak terkecil dibandingkan simpul lainnya yang terhubung ke simpul x.



VERTEX	KEY	PARENT
r	0	
s	4	t
t	5	r
u	9	t
v	3	w
w	8	x
x	2	t
y	15	x

HASIL



VERTEX	KEY	PARENT
r	0	
s	4	t
t	5	r
u	9	t
v	3	w
w	8	x
x	2	t
y	15	x

## Referensi

<https://www.informatikawan.com/minimum-spanning-tree-algoritma-prim/>