

9.2 - Algoritma pada Graf (part 2)

[KOMS120403]

Desain dan Analisis Algoritma (2022/2023)

Dewi Sintiar

Prodi S1 Ilmu Komputer
Universitas Pendidikan Ganesha

Week 12 (April 2023)

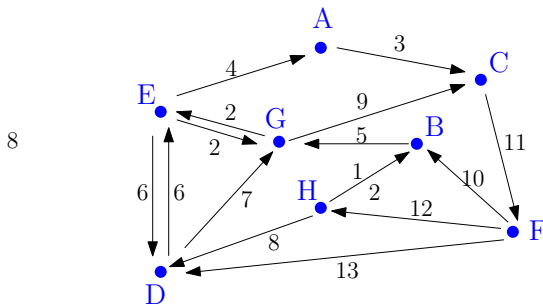
Daftar isi

- Shortest path problem (masalah jarak terpendek)
 - ▶ Algoritma Dijkstra

Bagian 1. Shortest Path

Shortest path

Masalah: diberikan graf berarah berbobot sisi. Temukan jalur terpendek dari s ke t .



Shortest path

Pohon jalur terpendek (*shortest-path tree*) yang berakar pada simpul v adalah pohon rentang (*spanning tree*) T dari graf G , sedemikian sehingga jarak jalur dari akar v ke simpul lain mana pun u di T adalah jalur terpendek jarak dari v ke u di G .

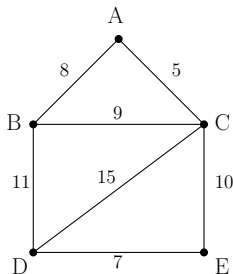


Figure: Graf G

Shortest path

Pohon jalur terpendek (*shortest-path tree*) yang berakar pada simpul v adalah pohon rentang (*spanning tree*) T dari graf G , sedemikian sehingga jarak jalur dari akar v ke simpul lain mana pun u di T adalah jalur terpendek jarak dari v ke u di G .

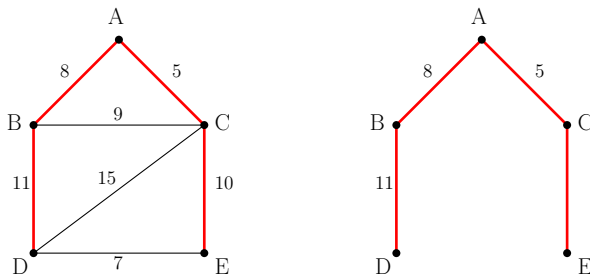


Figure: Yang berwarna merah adalah pohon jalur terpendek dari G berakar pada A

Shortest path

Pohon jalur terpendek (*shortest-path tree*) yang berakar pada simpul v adalah pohon rentang (*spanning tree*) T dari graf G , sedemikian sehingga jarak jalur dari akar v ke simpul lain mana pun u di T adalah jalur terpendek jarak dari v ke u di G .

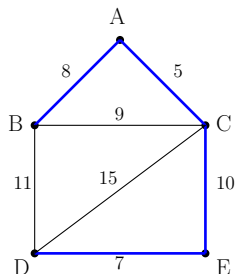
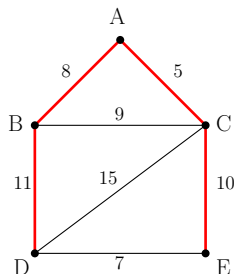


Figure: Yang berwarna merah merupakan pohon jalur terpendek, dan yang biru merupakan MST dari G

Shortest path

Varian dari shortest path

Berdasarkan titik

- Sumber tunggal (*single source*): dari satu simpul s ke setiap simpul lainnya
- Source-sink: dari satu simpul s ke t lainnya
- Semua pasangan: antara semua pasangan simpul

Batasan pada bobot sisi?

- Bobot yang tak-negatif
- Bobot yang merupakan jarak Euclid
- Bobot sebarang

Sirkuit?

- Tanpa sirkuit berarah
- Tanpa “sirkuit negatif”

Penerapan dari shortest path

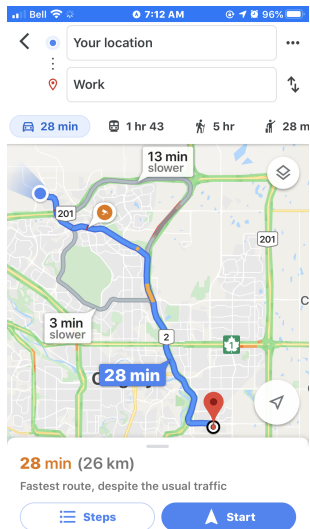


Figure: Mencari rute terpendek pada Google maps

Penerapan dari shortest path



Figure: Sistem navigasi pada mobil

Bagian 2. Algoritma Dijkstra

Algoritma Dijkstra

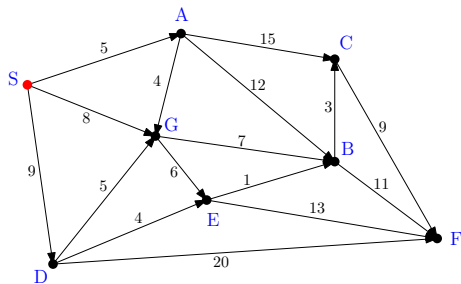


Figure: Edsger W. Dijkstra

Single-source shortest path:

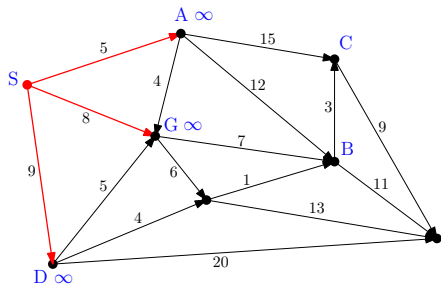
Tujuan: Temukan jalur terpendek dari sumber s ke setiap simpul lainnya.

Algoritma Dijkstra



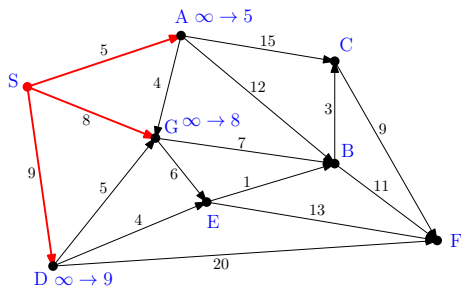
v	distTo[]	edgeTo[]
0	0	-
1	5.0	$0 \rightarrow 1$
2	14.0	$5 \rightarrow 2$
3	17.0	$2 \rightarrow 3$
4	9.0	$0 \rightarrow 4$
5	13.0	$4 \rightarrow 5$
6	25.0	$2 \rightarrow 6$
7	8.0	$0 \rightarrow 7$

Algoritma Dijkstra



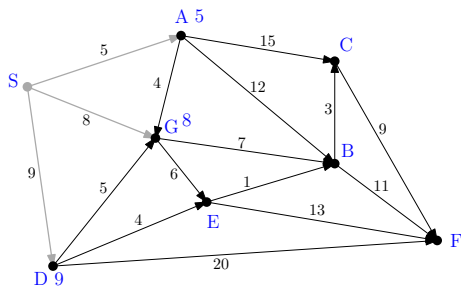
v	distTo[]	edgeTo[]
0	0	
1	5.0	
2	14.0	
3	17.0	
4	9.0	
5	13.0	
6	25.0	
7	8.0	

Algoritma Dijkstra



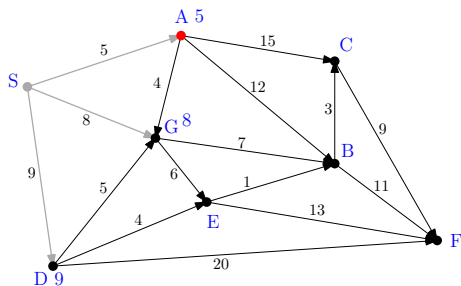
v	distTo[]	edgeTo[]
0	0	-
1	5.0	$0 \rightarrow 1$
2	14.0	$5 \rightarrow 2$
3	17.0	$2 \rightarrow 3$
4	9.0	$0 \rightarrow 4$
5	13.0	$4 \rightarrow 5$
6	25.0	$2 \rightarrow 6$
7	8.0	$0 \rightarrow 7$

Algoritma Dijkstra



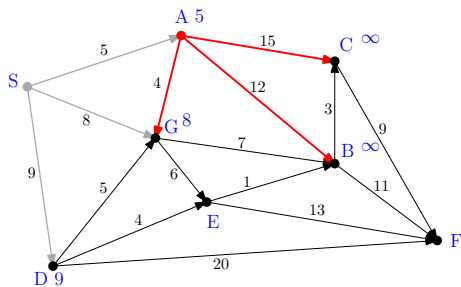
v	distTo[]	edgeTo[]
0	0	-
1	5.0	$0 \rightarrow 1$
2	14.0	$5 \rightarrow 2$
3	17.0	$2 \rightarrow 3$
4	9.0	$0 \rightarrow 4$
5	13.0	$4 \rightarrow 5$
6	25.0	$2 \rightarrow 6$
7	8.0	$0 \rightarrow 7$

Algoritma Dijkstra



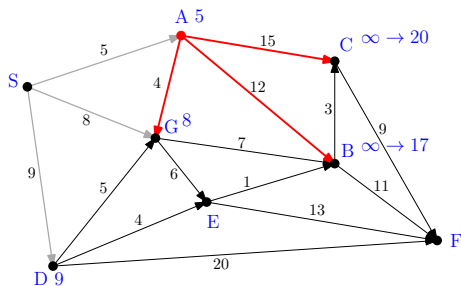
v	distTo[]	edgeTo[]
0	0	-
1	5.0	$0 \rightarrow 1$
2	14.0	$5 \rightarrow 2$
3	17.0	$2 \rightarrow 3$
4	9.0	$0 \rightarrow 4$
5	13.0	$4 \rightarrow 5$
6	25.0	$2 \rightarrow 6$
7	8.0	$0 \rightarrow 7$

Algoritma Dijkstra



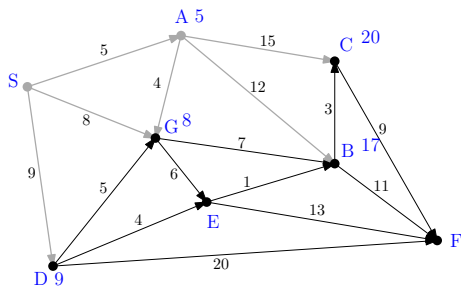
v	distTo[]	edgeTo[]
0	0	-
1	5.0	$0 \rightarrow 1$
2	14.0	$5 \rightarrow 2$
3	17.0	$2 \rightarrow 3$
4	9.0	$0 \rightarrow 4$
5	13.0	$4 \rightarrow 5$
6	25.0	$2 \rightarrow 6$
7	8.0	$0 \rightarrow 7$

Algoritma Dijkstra



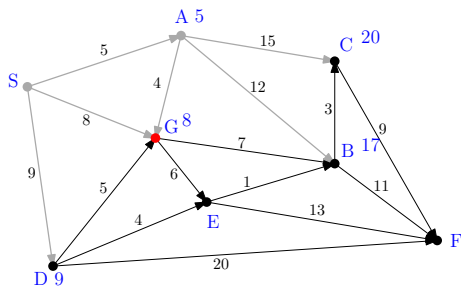
v	distTo[]	edgeTo[]
0	0	-
1	5.0	$0 \rightarrow 1$
2	14.0	$5 \rightarrow 2$
3	17.0	$2 \rightarrow 3$
4	9.0	$0 \rightarrow 4$
5	13.0	$4 \rightarrow 5$
6	25.0	$2 \rightarrow 6$
7	8.0	$0 \rightarrow 7$

Algoritma Dijkstra



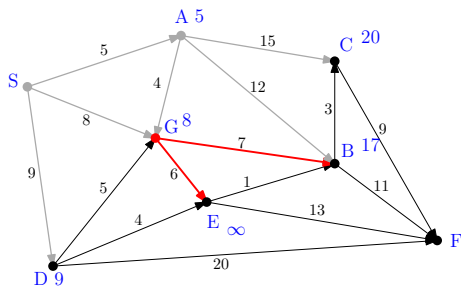
v	distTo[]	edgeTo[]
0	0	-
1	5.0	$0 \rightarrow 1$
2	14.0	$5 \rightarrow 2$
3	17.0	$2 \rightarrow 3$
4	9.0	$0 \rightarrow 4$
5	13.0	$4 \rightarrow 5$
6	25.0	$2 \rightarrow 6$
7	8.0	$0 \rightarrow 7$

Algoritma Dijkstra



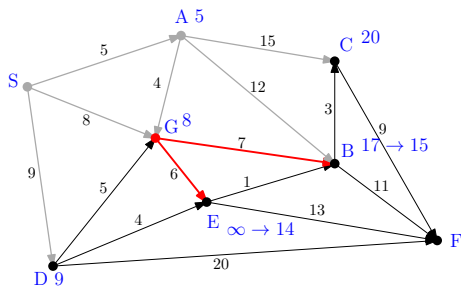
v	distTo[]	edgeTo[]
0	0	-
1	5.0	$0 \rightarrow 1$
2	14.0	$5 \rightarrow 2$
3	17.0	$2 \rightarrow 3$
4	9.0	$0 \rightarrow 4$
5	13.0	$4 \rightarrow 5$
6	25.0	$2 \rightarrow 6$
7	8.0	$0 \rightarrow 7$

Algoritma Dijkstra



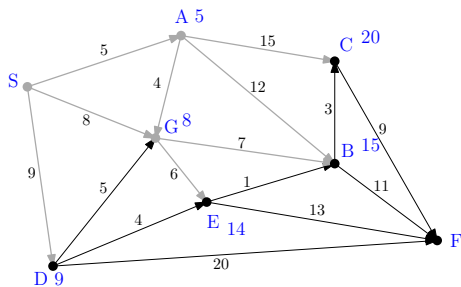
v	distTo[]	edgeTo[]
0	0	-
1	5.0	$0 \rightarrow 1$
2	14.0	$5 \rightarrow 2$
3	17.0	$2 \rightarrow 3$
4	9.0	$0 \rightarrow 4$
5	13.0	$4 \rightarrow 5$
6	25.0	$2 \rightarrow 6$
7	8.0	$0 \rightarrow 7$

Algoritma Dijkstra



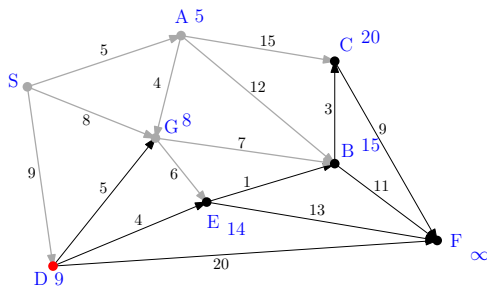
v	distTo[]	edgeTo[]
0	0	-
1	5.0	$0 \rightarrow 1$
2	14.0	$5 \rightarrow 2$
3	17.0	$2 \rightarrow 3$
4	9.0	$0 \rightarrow 4$
5	13.0	$4 \rightarrow 5$
6	25.0	$2 \rightarrow 6$
7	8.0	$0 \rightarrow 7$

Algoritma Dijkstra



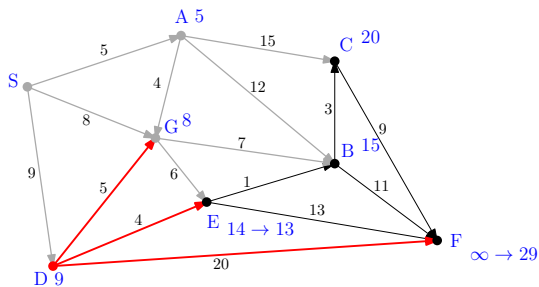
v	distTo[]	edgeTo[]
0	0	-
1	5.0	$0 \rightarrow 1$
2	14.0	$5 \rightarrow 2$
3	17.0	$2 \rightarrow 3$
4	9.0	$0 \rightarrow 4$
5	13.0	$4 \rightarrow 5$
6	25.0	$2 \rightarrow 6$
7	8.0	$0 \rightarrow 7$

Algoritma Dijkstra



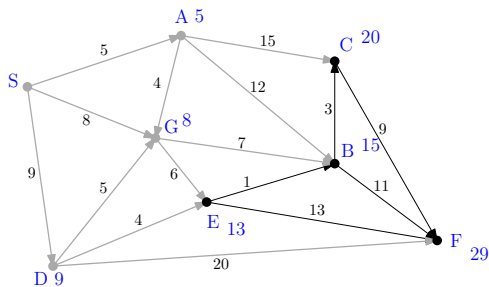
v	distTo[]	edgeTo[]
0	0	-
1	5.0	$0 \rightarrow 1$
2	14.0	$5 \rightarrow 2$
3	17.0	$2 \rightarrow 3$
4	9.0	$0 \rightarrow 4$
5	13.0	$4 \rightarrow 5$
6	25.0	$2 \rightarrow 6$
7	8.0	$0 \rightarrow 7$

Algoritma Dijkstra



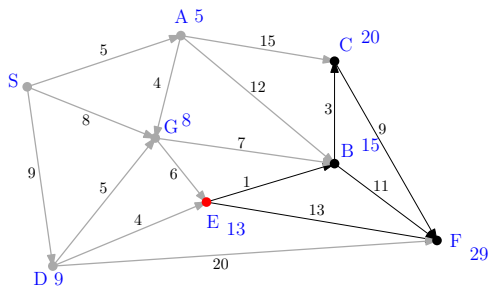
v	distTo[]	edgeTo[]
0	0	-
1	5.0	0 \rightarrow 1
2	14.0	5 \rightarrow 2
3	17.0	2 \rightarrow 3
4	9.0	0 \rightarrow 4
5	13.0	4 \rightarrow 5
6	25.0	2 \rightarrow 6
7	8.0	0 \rightarrow 7

Algoritma Dijkstra



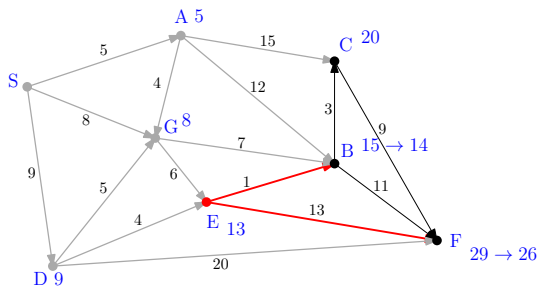
v	distTo[]	edgeTo[]
0	0	-
1	5.0	$0 \rightarrow 1$
2	14.0	$5 \rightarrow 2$
3	17.0	$2 \rightarrow 3$
4	9.0	$0 \rightarrow 4$
5	13.0	$4 \rightarrow 5$
6	25.0	$2 \rightarrow 6$
7	8.0	$0 \rightarrow 7$

Algoritma Dijkstra



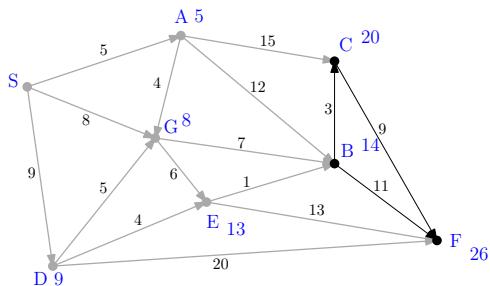
v	distTo[]	edgeTo[]
0	0	
1	5.0	
2	14.0	
3	17.0	
4	9.0	
5	13.0	
6	25.0	
7	8.0	

Algoritma Dijkstra



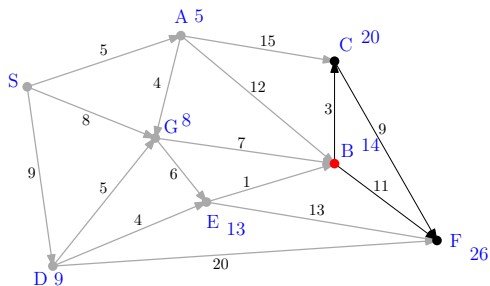
v	distTo[]	edgeTo[]
0	0	
1	5.0	
2	14.0	
3	17.0	
4	9.0	
5	13.0	
6	25.0	
7	8.0	

Algoritma Dijkstra



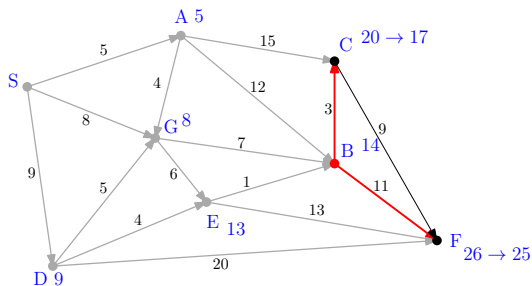
v	distTo[]	edgeTo[]
0	0	
1	5.0	
2	14.0	
3	17.0	
4	9.0	
5	13.0	
6	25.0	
7	8.0	

Algoritma Dijkstra



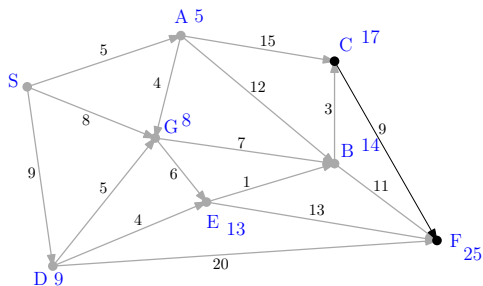
v	distTo[]	edgeTo[]
0	0	
1	5.0	
2	14.0	
3	17.0	
4	9.0	
5	13.0	
6	25.0	
7	8.0	

Algoritma Dijkstra



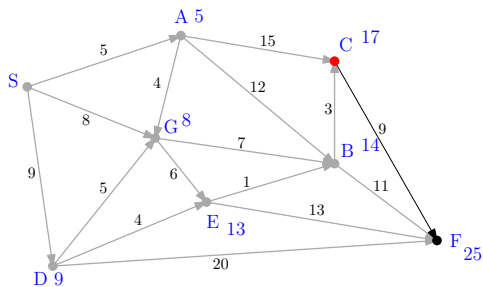
v	distTo[]	edgeTo[]
0	0	
1	5.0	
2	14.0	
3	17.0	
4	9.0	
5	13.0	
6	25.0	
7	8.0	

Algoritma Dijkstra



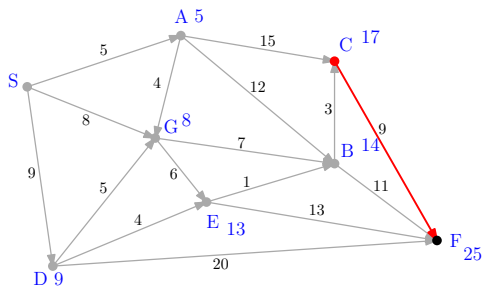
v	distTo[]	edgeTo[]
0	0	
1	5.0	
2	14.0	
3	17.0	
4	9.0	
5	13.0	
6	25.0	
7	8.0	

Algoritma Dijkstra



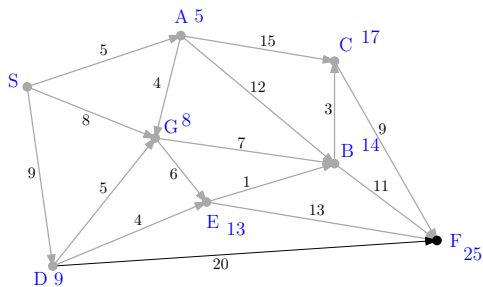
v	distTo[]	edgeTo[]
0	0	
1	5.0	
2	14.0	
3	17.0	
4	9.0	
5	13.0	
6	25.0	
7	8.0	

Algoritma Dijkstra



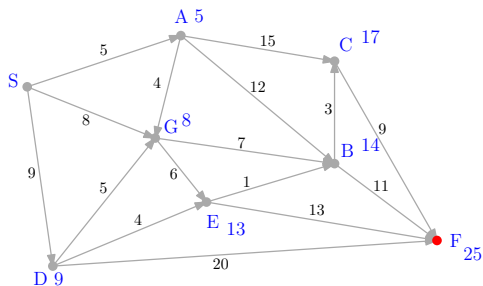
v	distTo[]	edgeTo[]
0	0	
1	5.0	
2	14.0	
3	17.0	
4	9.0	
5	13.0	
6	25.0	
7	8.0	

Algoritma Dijkstra



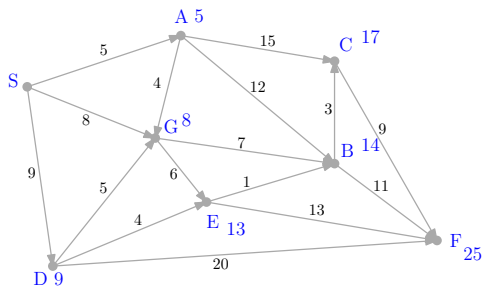
v	distTo[]	edgeTo[]
0	0	
1	5.0	
2	14.0	
3	17.0	
4	9.0	
5	13.0	
6	25.0	
7	8.0	

Algoritma Dijkstra



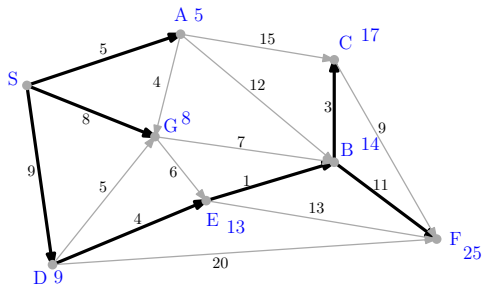
<i>v</i>	distTo[]	edgeTo[]
0	0	
1	5.0	
2	14.0	
3	17.0	
4	9.0	
5	13.0	
6	25.0	
7	8.0	

Algoritma Dijkstra



v	distTo[]	edgeTo[]
0	0	
1	5.0	
2	14.0	
3	17.0	
4	9.0	
5	13.0	
6	25.0	
7	8.0	

Algoritma Dijkstra



v	distTo[]	edgeTo[]
0	0	-
1	5.0	$0 \rightarrow 1$
2	14.0	$5 \rightarrow 2$
3	17.0	$2 \rightarrow 3$
4	9.0	$0 \rightarrow 4$
5	13.0	$4 \rightarrow 5$
6	25.0	$2 \rightarrow 6$
7	8.0	$0 \rightarrow 7$

end of slide...