

## 10.2 - Algoritma pada Graf (part 2)

[KOMS124404]

Desain dan Analisis Algoritma (2024/2025)

Dewi Sintiar

Prodi S1 Ilmu Komputer  
Universitas Pendidikan Ganesha

Week 10 (Mei 2025)

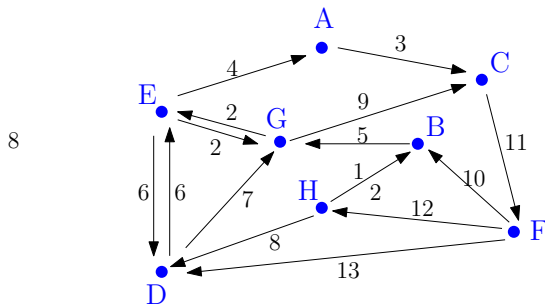
# Daftar isi

- Shortest path problem (masalah jarak terdekat)
  - ▶ 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  $G$ .

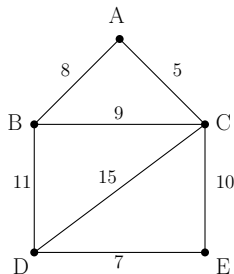


Figure: Graf  $G$

# 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  $G$ .

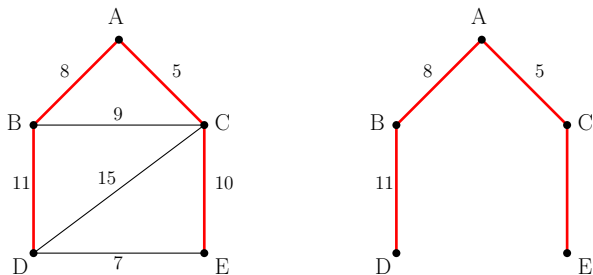
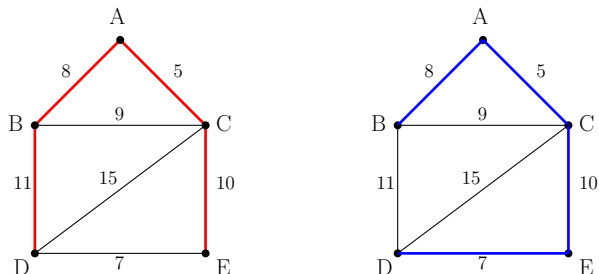


Figure: Yang berwarna merah adalah pohon jalur terpendek dari  $G$  berakar pada  $A$

## 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 simpul sumber  $s$  ke suatu simpul akhir  $t$
- 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

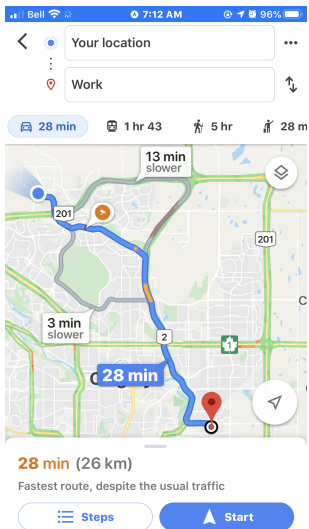


Figure: Mencari rute terpendek pada Google maps

# Penerapan dari shortest path



Figure: Sistem navigasi pada mobil

# Bagian 2. Algoritma Dijkstra

# Algoritma Dijkstra

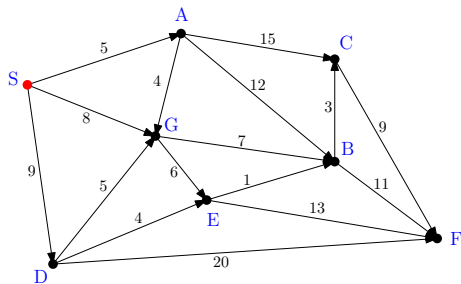


Figure: Edsger W. Dijkstra

Single-source shortest path:

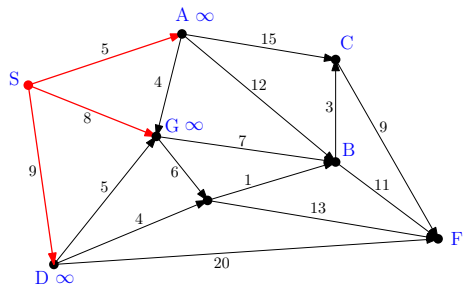
**Tujuan:** Temukan jalur terpendek dari sumber  $s$  ke setiap simpul lainnya.

# Algoritma Dijkstra



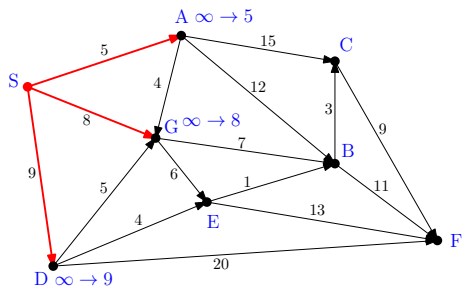
$v$	distTo[ ]	edgeTo[ ]
S	0	-
A	5.0	S → A
B	14.0	E → B
C	17.0	B → C
D	9.0	S → D
E	13.0	D → E
F	25.0	B → F
G	8.0	S → G

# Algoritma Dijkstra



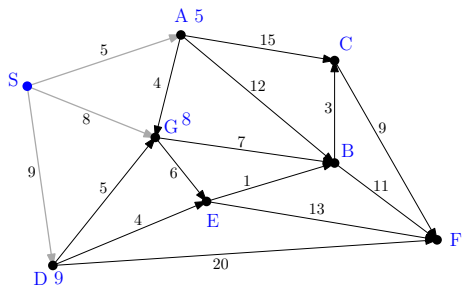
$v$	distTo[ ]	edgeTo[ ]
S	0	-
A	5.0	S $\rightarrow$ A
B	14.0	E $\rightarrow$ B
C	17.0	B $\rightarrow$ C
D	9.0	S $\rightarrow$ D
E	13.0	D $\rightarrow$ E
F	25.0	B $\rightarrow$ F
G	8.0	S $\rightarrow$ G

# Algoritma Dijkstra



$v$	distTo[ ]	edgeTo[ ]
S	0	-
A	5.0	S $\rightarrow$ A
B	14.0	E $\rightarrow$ B
C	17.0	B $\rightarrow$ C
D	9.0	S $\rightarrow$ D
E	13.0	D $\rightarrow$ E
F	25.0	B $\rightarrow$ F
G	8.0	S $\rightarrow$ G

# Algoritma Dijkstra

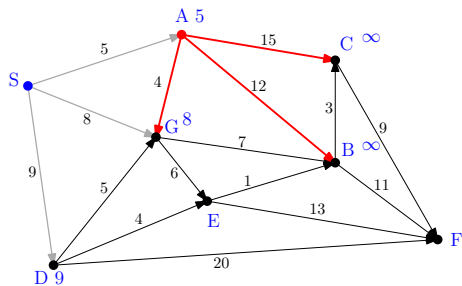


$v$	distTo[ ]	edgeTo[ ]
S	0	-
A	5.0	S → A
B	14.0	E → B
C	17.0	B → C
D	9.0	S → D
E	13.0	D → E
F	25.0	B → F
G	8.0	S → G



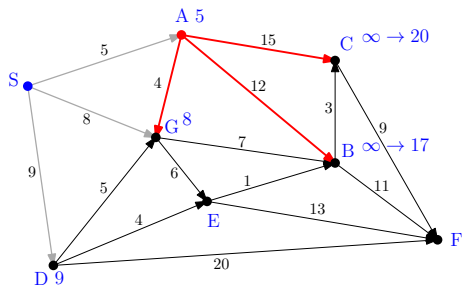


# Algoritma Dijkstra



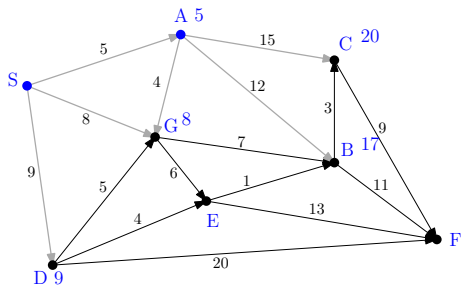
$v$	distTo[ ]	edgeTo[ ]
S	0	-
A	5.0	S → A
B	14.0	E → B
C	17.0	B → C
D	9.0	S → D
E	13.0	D → E
F	25.0	B → F
G	8.0	S → G

# Algoritma Dijkstra



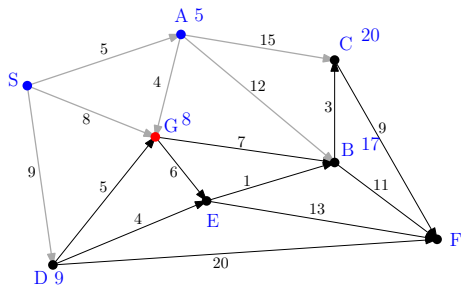
$v$	distTo[ ]	edgeTo[ ]
S	0	-
A	5.0	S → A
B	14.0	E → B
C	17.0	B → C
D	9.0	S → D
E	13.0	D → E
F	25.0	B → F
G	8.0	S → G

# Algoritma Dijkstra



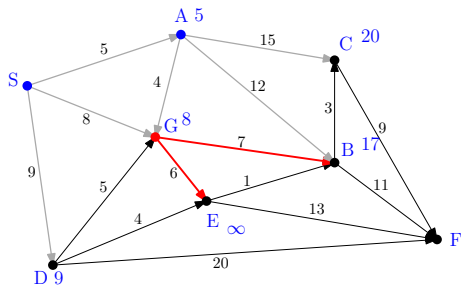
$v$	distTo[ ]	edgeTo[ ]
S	0	-
A	5.0	S → A
B	14.0	E → B
C	17.0	B → C
D	9.0	S → D
E	13.0	D → E
F	25.0	B → F
G	8.0	S → G

# Algoritma Dijkstra



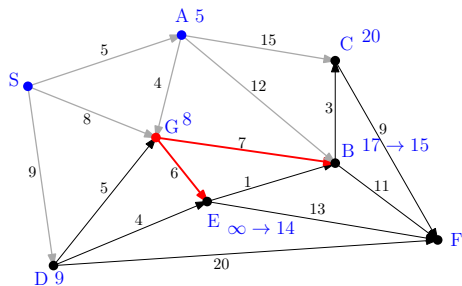
$v$	distTo[ ]	edgeTo[ ]
S	0	-
A	5.0	S → A
B	14.0	E → B
C	17.0	B → C
D	9.0	S → D
E	13.0	D → E
F	25.0	B → F
G	8.0	S → G

# Algoritma Dijkstra



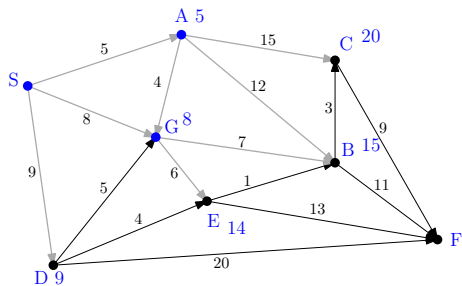
$v$	distTo[ ]	edgeTo[ ]
S	0	-
A	5.0	S $\rightarrow$ A
B	14.0	E $\rightarrow$ B
C	17.0	B $\rightarrow$ C
D	9.0	S $\rightarrow$ D
E	13.0	D $\rightarrow$ E
F	25.0	B $\rightarrow$ F
G	8.0	S $\rightarrow$ G

# Algoritma Dijkstra



$v$	distTo[ ]	edgeTo[ ]
S	0	-
A	5.0	S → A
B	14.0	E → B
C	17.0	B → C
D	9.0	S → D
E	13.0	D → E
F	25.0	B → F
G	8.0	S → G

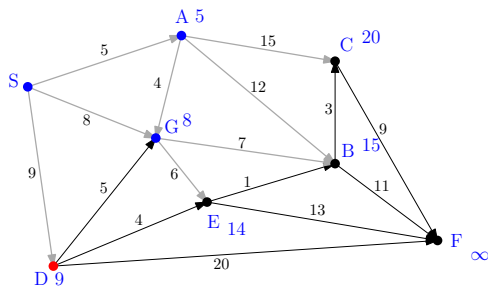
# Algoritma Dijkstra



$v$	distTo[ ]	edgeTo[ ]
S	0	-
A	5.0	S → A
B	14.0	E → B
C	17.0	B → C
D	9.0	S → D
E	13.0	D → E
F	25.0	B → F
G	8.0	S → G

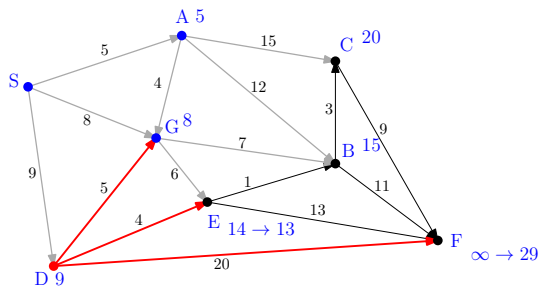


# Algoritma Dijkstra



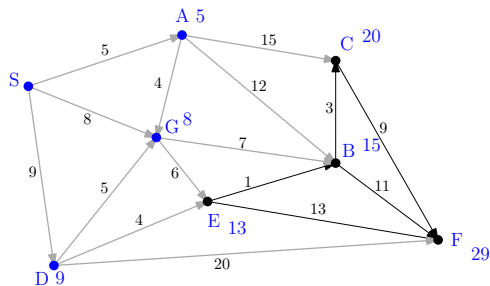
$v$	distTo[ ]	edgeTo[ ]
S	0	-
A	5.0	S → A
B	14.0	E → B
C	17.0	B → C
D	9.0	S → D
E	13.0	D → E
F	25.0	B → F
G	8.0	S → G

# Algoritma Dijkstra



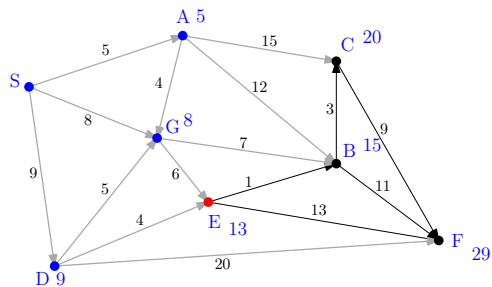
$v$	distTo[ ]	edgeTo[ ]
S	0	-
A	5.0	S $\rightarrow$ A
B	14.0	E $\rightarrow$ B
C	17.0	B $\rightarrow$ C
D	9.0	S $\rightarrow$ D
E	13.0	D $\rightarrow$ E
F	25.0	B $\rightarrow$ F
G	8.0	S $\rightarrow$ G

# Algoritma Dijkstra



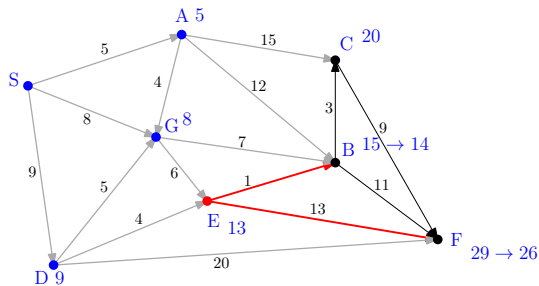
$v$	distTo[ ]	edgeTo[ ]
S	0	-
A	5.0	S → A
B	14.0	E → B
C	17.0	B → C
D	9.0	S → D
E	13.0	D → E
F	25.0	B → F
G	8.0	S → G

# Algoritma Dijkstra



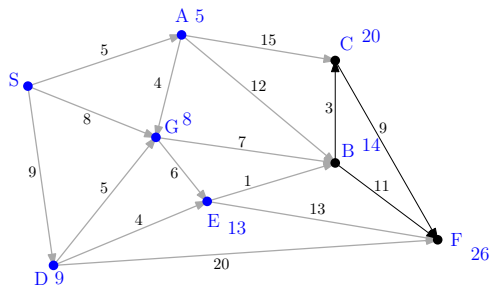
$v$	distTo[ ]	edgeTo[ ]
S	0	-
A	5.0	S → A
B	14.0	E → B
C	17.0	B → C
D	9.0	S → D
E	13.0	D → E
F	25.0	B → F
G	8.0	S → G

# Algoritma Dijkstra



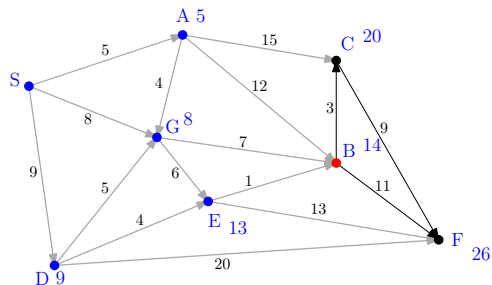
$v$	distTo[ ]	edgeTo[ ]
S	0	-
A	5.0	S → A
B	14.0	E → B
C	17.0	B → C
D	9.0	S → D
E	13.0	D → E
F	25.0	B → F
G	8.0	S → G

# Algoritma Dijkstra



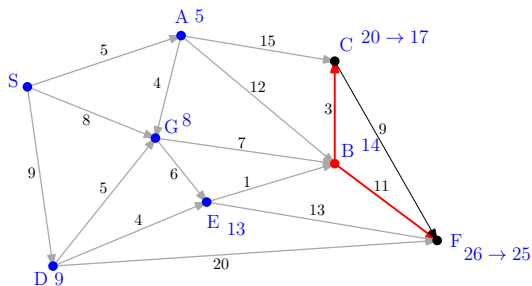
$v$	distTo[ ]	edgeTo[ ]
S	0	-
A	5.0	S → A
B	14.0	E → B
C	17.0	B → C
D	9.0	S → D
E	13.0	D → E
F	25.0	B → F
G	8.0	S → G

# Algoritma Dijkstra



$v$	distTo[ ]	edgeTo[ ]
S	0	-
A	5.0	S → A
B	14.0	E → B
C	17.0	B → C
D	9.0	S → D
E	13.0	D → E
F	25.0	B → F
G	8.0	S → G

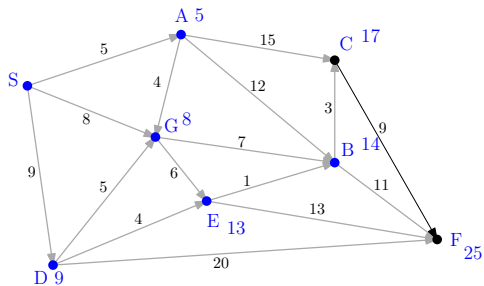
# Algoritma Dijkstra



$v$	distTo[ ]	edgeTo[ ]
S	0	-
A	5.0	S → A
B	14.0	E → B
C	17.0	B → C
D	9.0	S → D
E	13.0	D → E
F	25.0	B → F
G	8.0	S → G

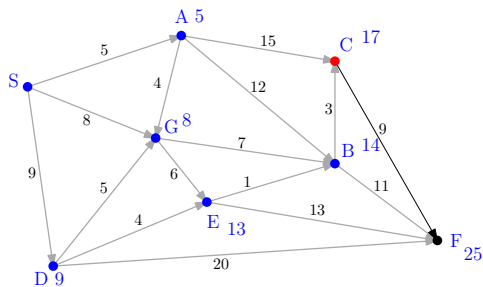


# Algoritma Dijkstra



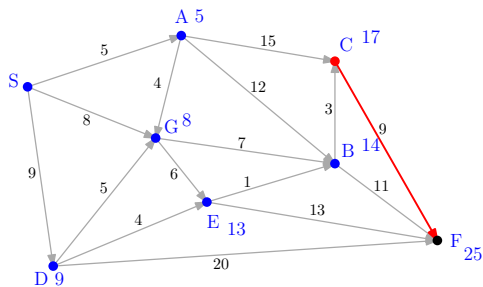
$v$	distTo[ ]	edgeTo[ ]
S	0	-
A	5.0	S → A
B	14.0	E → B
C	17.0	B → C
D	9.0	S → D
E	13.0	D → E
F	25.0	B → F
G	8.0	S → G

# Algoritma Dijkstra



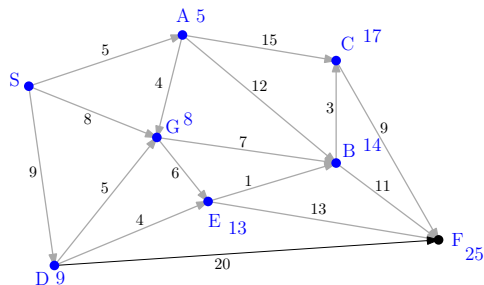
$v$	distTo[ ]	edgeTo[ ]
S	0	-
A	5.0	S → A
B	14.0	E → B
C	17.0	B → C
D	9.0	S → D
E	13.0	D → E
F	25.0	B → F
G	8.0	S → G

# Algoritma Dijkstra



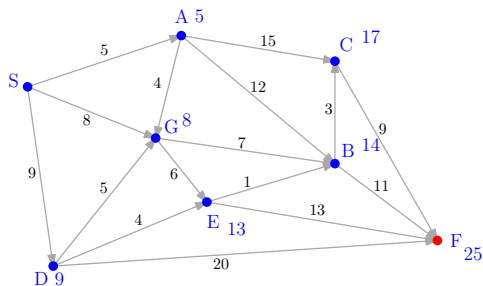
$v$	distTo[ ]	edgeTo[ ]
S	0	-
A	5.0	S → A
B	14.0	E → B
C	17.0	B → C
D	9.0	S → D
E	13.0	D → E
F	25.0	B → F
G	8.0	S → G

# Algoritma Dijkstra



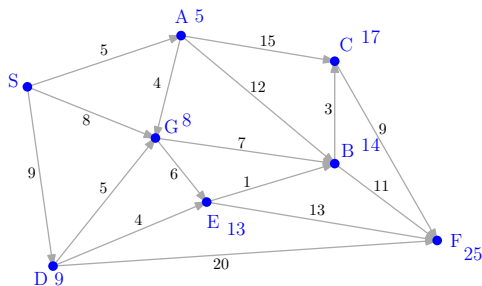
$v$	distTo[ ]	edgeTo[ ]
S	0	-
A	5.0	S → A
B	14.0	E → B
C	17.0	B → C
D	9.0	S → D
E	13.0	D → E
F	25.0	B → F
G	8.0	S → G

# Algoritma Dijkstra



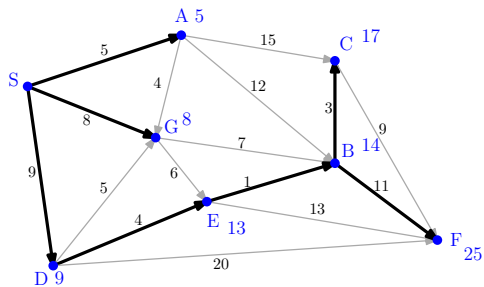
$v$	distTo[ ]	edgeTo[ ]
S	0	-
A	5.0	S → A
B	14.0	E → B
C	17.0	B → C
D	9.0	S → D
E	13.0	D → E
F	25.0	B → F
G	8.0	S → G

# Algoritma Dijkstra



$v$	distTo[ ]	edgeTo[ ]
S	0	-
A	5.0	S → A
B	14.0	E → B
C	17.0	B → C
D	9.0	S → D
E	13.0	D → E
F	25.0	B → F
G	8.0	S → G

# Algoritma Dijkstra



$v$	distTo[ ]	edgeTo[ ]
S	0	-
A	5.0	S → A
B	14.0	E → B
C	17.0	B → C
D	9.0	S → D
E	13.0	D → E
F	25.0	B → F
G	8.0	S → G

*end of slide...*