MATEMATIKA DISKRIT

GRAF BERLABEL

Wike Handini

POHON RENTANG MINIMUM

Definisi

- Graf berlabel (*weighted graph*) adalah suatu graf tanpa garis paralel dimana setiap garisnya berhubungan dengan suatu bilangan riil tak negatif yang menyatakan bobot garis tersebut.
- Bobot garis e biasanya diberi simbol w(e). Jumlah bobot semua garis disebut Total Bobot.
- Matriks yang bersesuaian dengan graf label G adalah matriks hubung A = (a_{ij}) dengan a_{ij} = bobot garis yang menghubungkan titik v_i dengan titik v_j. Jika titik v_i tidak berhubungan langsung dengan titik v_j, maka $a_{ij} = \infty$, dan a_{ij} = 0 jika i = j.

CONTOH 30

Dalam suatu propinsi, ada 8 kota (v_1, ..., v_8) yang akan dihubungkan dengan jaringan listrik. Biaya pemasangan jaringan listrik yang mungkin dibuat antara 2 kota adalah sebagai 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
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

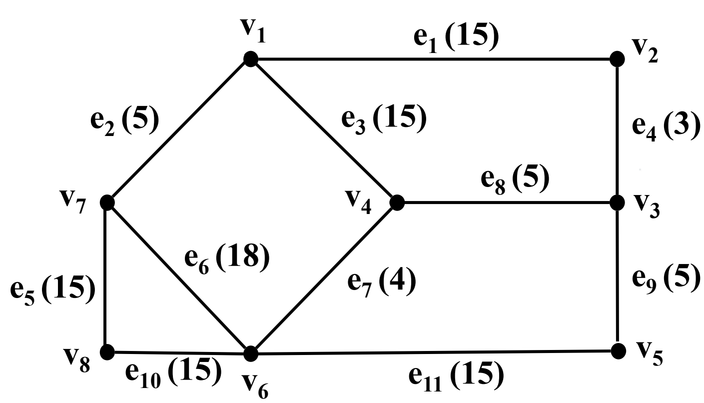
a. Nyatakan masalah tersebut dalam graf berlabel.

b. Buatlah matriks hubung yang sesuai untuk menyatakan masalah tersebut.

CONTOH 30 → Penyelesaian

a. Graf berlabel dapat digambarkan sebagai berikut:

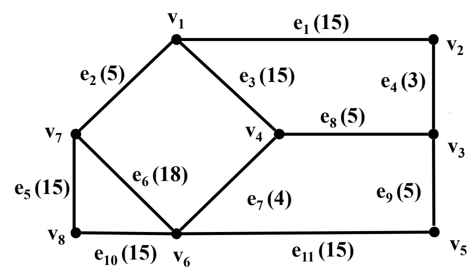
Garis	Kota yang dihubungkan	Biaya per satuan
e_4	$v_2 - v_3$	3
e_7	$v_4 - v_6$	4
e_2	$v_1 - v_7$	5
e_8	$v_3 - v_4$	5
e_9	$v_3 - v_5$	5
e_1	$v_1 - v_2$	15
e_3	$v_1 - v_4$	15
e_{10}	$v_6 - v_8$	15
e_5	$v_7 - v_8$	15
e_{11}	$v_5 - v_6$	15
e_6	$v_6 - v_7$	18



CONTOH 30 → Penyelesaian

b. Matriks hubung yang menyatakan graf berlabel adalah sebagai berikut:

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



POHON RENTANG MINIMUM

- Dalam program komputer, sel dengan nilai ∞ diisi dengan bilangan yang harganya jauh lebih besar dari elemen-elemen yang bukan ∞.
- Aplikasi yang sering digunakan dalam graf berlabel adalah mencari pohon rentang dengan total bobot seminimum mungkin (sering disebut pohon rentang minimum).
- Pada tahun 1956, Kruskal dan Prim yang bekerja secara terpisah, masing-masing berhasil menyusun algoritma yang membuat pohon rentang minimum secara efisien.

ALGORITMA KRUSKAL

- Pertama, semua garis dalam graf G diurutkan berdasarkan bobotnya dari kecil ke besar. Kemudian pilih garis dengan bobot terkecil, langkah berikutnya pilih garis dengan bobot terkecil (diantara garis-garis sisa yang belum dipilih). Pada setiap langkah, dipilih garis dengan bobot terkecil, tetapi tidak membentuk loop dengan garis-garis yang sudah dipilih sebelumnya.
- Misalkan G adalah graf mula-mula dengan n titik, T adalah pohon rentang minimum.
- E adalah himpunan semua garis G.
- Secara formal, algoritma yang ditemukan Kruskal dapat dinyatakan sebagai berikut:

7

ALGORITMA KRUSKAL

1. Isi T dengan semua titik – titik G tanpa garis
2. m = 0
3. Selama m < (n–1) lakukan:
   a. Tentukan garis $e \in E$ dengan bobot minimum. Jika ada beberapa e dengan sifat tersebut, pilih salah satu secara sembarang
   b. Hapus e dari E
   c. Jika e ditambahkan ke T tidak menghasilkan sirkuit, maka:
      i. tambahkan e ke T
      ii. m = m + 1

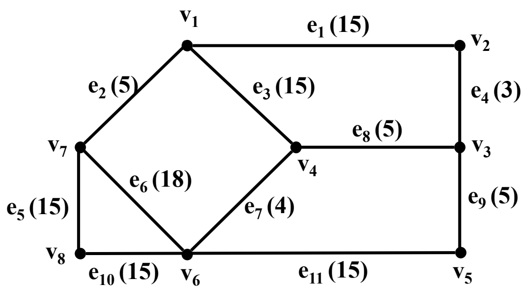
8

CONTOH 31

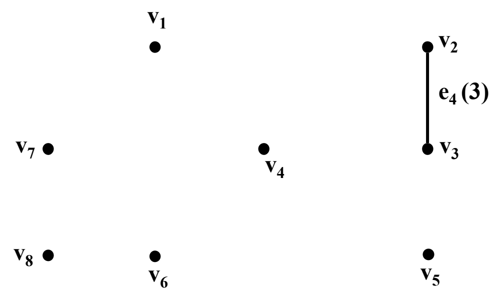
Carilah pohon rentang minimum dari contoh 30 dengan menggunakan algoritma Kruskal. Hitunglah bobot minimum totalnya.

Penyelesaian

Dari contoh 30 didapatkan graf:

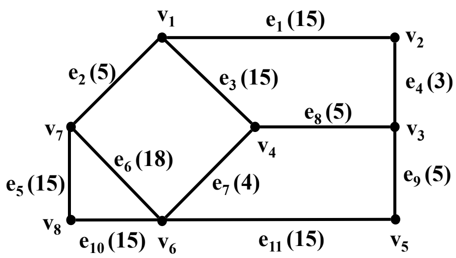


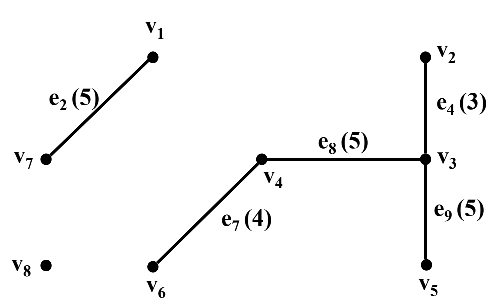
- Ambil garis dengan bobot minimum, maka pohon rentang T yang semula tanpa garis akan menjadi:



CONTOH 31 → Penyelesaian

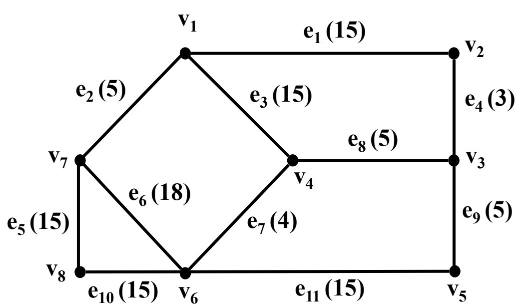
- Iterasi berikutnya, ditambahkan garis satu persatu pada T selama penambahan garis tersebut tidak membentuk loop dengan garis sebelumnya.
  Dengan demikian ditambahkan garis e_7.
  Selanjutnya ada 3 garis dengan bobot terkecil (5) yaitu e_2, e_8 dan e_9. Oleh karena penambahan ketiga garis tersebut tidak menghasilkan loop, maka ketiganya dapat ditambahkan dalam graf T.

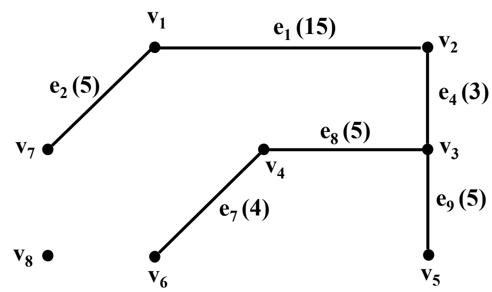




CONTOH 31 → Penyelesaian

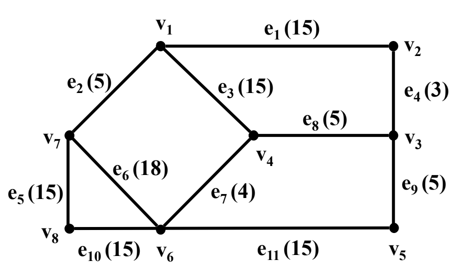
- Berikutnya ada 5 garis dengan bobot yang sama yaitu 15, Pilih satu secara sembarang, misalnya dipilih e_1, maka graf T menjadi:

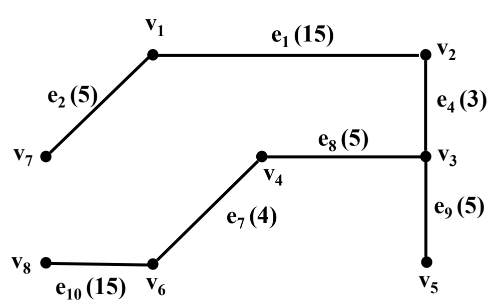




CONTOH 31 → Penyelesaian

- Dari 4 garis dengan bobot 15 yang tersisa (e_3, e_5, e_{10}), dipilih garis yang tidak membentuk loop dengan garis sebelumnya, yaitu e_{10}, sehingga garf T menjadi:





- Bobot total = 3 + 4 + 5+ 5 +5 +15 + 15 = 52

ALGORITMA PRIM

- Jika algoritma Kruskal dimulai dengan graf tanpa garis, maka algoritma Prim dimulai dari graf yang kosong sama sekali.
- Pertama pilih satu titik sembarang (misalnya v_1). Kemudian tambahkan 1 garis berbobot paling minimum yang berhubungan dengan v_1 (misalnya e_1) dan titik ujung lainnya ke T. Langkah selanjutnya, dipilih garis dalam E(G) yang bukan anggota E(T) dengan sifat:
  a. Garis tersebut berhubungan dengan salah satu titik $\in$ V(T).
  b. Garis tersebut memiliki bobot terkecil.
- Langkah tersebut diulang-ulang sehingga diperoleh (n – 1) garis dalam E(T) dimana n adalah jumlah titik dalam G.
- Misalkan G adalah graf berlabel dengan n titik dan T adalah pohon rentang minimum. Secara formal, algoritma Prim adalah sebagai berikut:

13

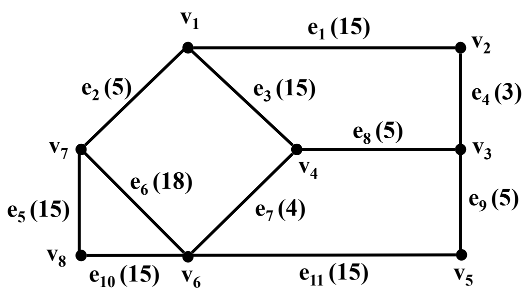
ALGORITMA PRIM

0. Inisialisasi: mula-mula T adalah graf kosong
1. Ambil sembarang v $\in$ V(G). Masukkan v ke dalam V(T)
2. V(G) = V(G) – (v)
3. Untuk i = 1, 2, ..., n – 1, lakukan:
   a. Pilih garis e $\in$ E(G) dan e $\notin$ E(T) dengan syarat:
      i. e berhubungan dengan salah satu titik dalam T
      ii. e memiliki bobot terkecil dibandingkan dengan semua garis yang berhubungan dengan titik-titik dalam T (misalkan titik w adalah titik ujung e yang tidak berada dalam T)
   b. Tambahkan e ke E(T) dan w ke V(T)
   c. V(G) = V(G) – (w)

14

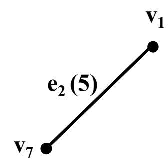
CONTOH 32

Gunakan algoritma Prim untuk mencari pohon rentang minimum dari contoh 30.



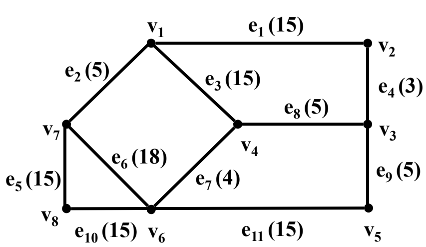
Penyelesaian

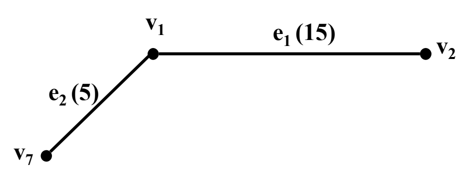
- Misalkan semula diambil titik v_1 sehingga $V(T) = [v_1]$ dan $E(T) = [\ ]$.
  Kemudian pada iterasi pertama, pilih garis yang terhubung v_1 dengan bobot terkecil yaitu e_2.
  Sehingga $V(T) = [v_1, v_7]$ dan $E(T) = [e_2]$



CONTOH 32 → Penyelesaian

- Iterasi kedua, pilih garis $e_i \in E(G)$ dan $e_i \notin E(T)$ yang terhubung dengan titik-titik dalam V(T) dengan bobot terkecil. Ada 3 titik yang berbobot sama yaitu e_1, e_3 dan e_5. Misalnya dipilih e_1. Sehingga $V(T) = [v_1, v_7, v_2]$ dan $E(T) = [e_2, e_1]$

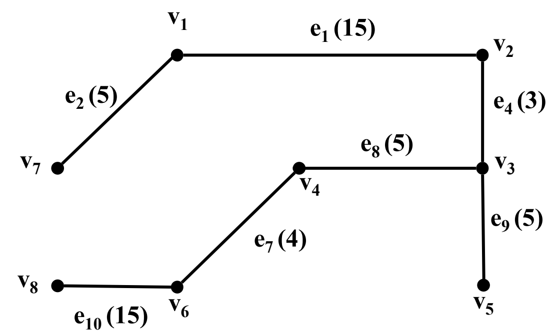




- Proses iterasi yang sama diulang-ulang hingga V(T) mememuat semua titik dalam G (atau jumlah iterasi adalah (n – 1), dengan n jumlah titik dalam G). Didapatkan hasil seperti pada tabel.

CONTOH 32 → Penyelesaian

Iterasi	Garis yang dipilih	Titik yang di-tambahkan	Keterangan
0	-	v_1	-
1	e_2 (5)	v_7	-
2	e_1 (15)	v_2	Pilih antara e_1, e_3, e_5
3	e_4 (3)	v_3	-
4	e_8 (5)	v_4	Pilih antara e_8 dan e_9
5	e_7 (4)	v_6	
6	e_9 (5)	v_5	
7	e_{10} (15)	v_8	Pilih antara e_5 dan e_{10}



- Bobot total baik dengan algoritma Kruskal maupun algoritma Prim adalah sama, yaitu 52

17

PATH MINIMUM

- Bobot yang berhubungan dengan suatu garis pada graf dapat diaplikasikan pada graf berarah dengan prinsip yang sama.
- Salah satu aplikasi graf berarah berlabel yang sering dipakai adalah mencari path terpendek antara 2 titik.
- Elemen-elemen dari matriks hubung W yang digunakan untuk menyatakan graf berarah berlabel menyatakan bobot garis.
- Secara umum, matriks hubung untuk graf berarah berlabel tidak simetris karena bobot garis dari titik v_i ke v_j (= W(i, j) belum tentu sama dengan bobot garis dari titik v_j ke v_i (= W(j, i), bahkan mungkin hubungan antara ke dua titik tersebut hanya searah.
- W(i, i) = ∞ untuk semua i.

18

PATH MINIMUM → Algoritma Warshall

- Algoritma ini merupakan algoritma sederhana dan mudah implementasinya.
- Masukan algoritma Warshall adalah matriks hubung graf berarah berlabel dan keluarannya adalah path terpendek dari semua titik ke semua titik.
- Dalam usaha untuk mencari path terpendek, iterasi dimulai dari titik awalnya yang kemudian memperpanjang path dengan mengevaluasi titik demi titik hingga mencapai titik tujuan dengan jumlah bobot seminimum mungkin.
- Misalkan W_0 adalah matriks hubung graf berlabel mula-mula.
- W^* adalah matriks hubung minimal dengan W_{ij}^* = path terpendek dari titik v_i ke v_j.
- Algoritma Warshall untuk mencari path terpendek adalah sebagai berikut:

19

PATH MINIMUM → Algoritma Warshall

1. $W = W_0$
2. Untuk k = 1 hingga n, lakukan:

    Untuk i = 1 hingga n, lakukan:

        Untuk j = 1 hingga n, lakukan:

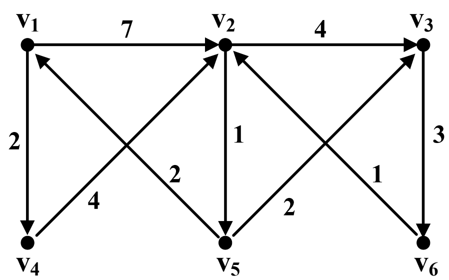
            Jika W[i, j] > W[i, k] + W[k, j] maka

                tukar W[i, j] dengan W[i, k] + W[k, j]
3. $W^* = W$

20

CONTOH 33

Carilah path terpendek dari titik v_i ke v_j (i, j = 1, 2, ..., 6) graf berarah berlabel pada gambar.



Penyelesaian

Matriks hubung dari graf adalah sebagai berikut:

$$W = W_0 = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \end{array} \begin{array}{c} \begin{array}{cccccc} v_1 & v_2 & v_3 & v_4 & v_5 & v_6 \end{array} \\ \begin{bmatrix} \infty & 7 & \infty & 2 & \infty & \infty \\ \infty & \infty & 4 & \infty & 1 & \infty \\ \infty & \infty & \infty & \infty & \infty & 3 \\ \infty & 4 & \infty & \infty & \infty & \infty \\ 2 & \infty & 2 & \infty & \infty & \infty \\ \infty & 1 & \infty & \infty & \infty & \infty \end{bmatrix} \end{array}$$

CONTOH 33 → Penyelesaian

Iterasi untuk k = 1

Untuk setiap sel matriks W dicek apakah W[i, j] > W[i, 1] + W[1, j]. Jika ya, maka W[i, j] diganti dengan W[i, 1] + W[1, j]. Contoh:

- W[1, 2] = 7, sedangkan W[1, 1] + W[1, 2] = ∞ + 7 = ∞.
- Karena W[1, 2] $\not>$ W[1, 1] + W[1, 2], maka harga W[1, 2] tidak berubah.
- W[5, 4] = ∞, sedangkan W[5, 1] + W[1, 4] = 2 + 2 = 4.
- Karena W[5, 4] > W[5, 1] + W[1, 4], maka harga W[5, 4] diubah menjadi 4.
- Ini berarti ada path dari v_5 ke v_4 melalui v_1 yang memiliki bobot lebih kecil (yaitu path v_5 v_1 v_4 dengan jumlah bobot 4)

CONTOH 33 → Penyelesaian

Dengan cara yang sama, nilai W[i, j] dihitung untuk setiap i dan j, sehingga didapatkan matriks:

$$W_1 = \begin{array}{cc} & \begin{array}{cccccc} v_1 & v_2 & v_3 & v_4 & v_5 & v_6 \end{array} \\ \begin{array}{c} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \end{array} & \begin{bmatrix} \infty & 7 & \infty & 2 & \infty & \infty \\ \infty & \infty & 4 & \infty & 1 & \infty \\ \infty & \infty & \infty & \infty & \infty & 3 \\ \infty & 4 & \infty & \infty & \infty & \infty \\ 2 & 9 & 2 & 4 & \infty & \infty \\ \infty & 1 & \infty & \infty & \infty & \infty \end{bmatrix} \end{array}$$

Iterasi untuk k = 2

Untuk setiap sel matriks W dicek apakah W[i, j] > W[i, 2] + W[2, j]. Jika ya, maka W[i, j] diganti dengan W[i, 1] + W[1, j]. Contoh:

- W[6, 5] = ∞, sedangkan W[6, 2] + W[2, 5] = 1 + 1 = 2.
- Karena W[6, 5] > W[6, 2] + W[2, 5] , maka harga W[6, 5] diubah menjadi 2.

CONTOH 33 → Penyelesaian

Dengan cara yang sama, nilai W[i, j] dihitung untuk setiap i dan j, sehingga didapatkan matriks:

$$W_2 = \begin{array}{cc} & \begin{array}{cccccc} v_1 & v_2 & v_3 & v_4 & v_5 & v_6 \end{array} \\ \begin{array}{c} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \end{array} & \begin{bmatrix} \infty & 7 & 11 & 2 & 8 & \infty \\ \infty & \infty & 4 & \infty & 1 & \infty \\ \infty & \infty & \infty & \infty & \infty & 3 \\ \infty & 4 & 8 & \infty & 5 & \infty \\ 2 & 9 & 2 & 4 & 10 & \infty \\ \infty & 1 & 5 & \infty & 2 & \infty \end{bmatrix} \end{array}$$

Iterasi untuk k = 3

Dengan cara yang sama, diperoleh matriks:

$$W_3 = \begin{array}{cc} & \begin{array}{cccccc} v_1 & v_2 & v_3 & v_4 & v_5 & v_6 \end{array} \\ \begin{array}{c} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \end{array} & \begin{bmatrix} \infty & 7 & 11 & 2 & 8 & 14 \\ \infty & \infty & 4 & \infty & 1 & 7 \\ \infty & \infty & \infty & \infty & \infty & 3 \\ \infty & 4 & 8 & \infty & 5 & 11 \\ 2 & 9 & 2 & 4 & 10 & 5 \\ \infty & 1 & 5 & \infty & 2 & 8 \end{bmatrix} \end{array}$$

CONTOH 33 → Penyelesaian

Iterasi untuk k = 4

Dengan cara yang sama, diperoleh matriks:

$$W_4 = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \end{array} \begin{array}{c} \begin{array}{cccccc} v_1 & v_2 & v_3 & v_4 & v_5 & v_6 \end{array} \\ \begin{bmatrix} \infty & 6 & 10 & 2 & 7 & 13 \\ \infty & \infty & 4 & \infty & 1 & 7 \\ \infty & \infty & \infty & \infty & \infty & 3 \\ \infty & 4 & 8 & \infty & 5 & 11 \\ 2 & 8 & 2 & 4 & 9 & 5 \\ \infty & 1 & 5 & \infty & 2 & 8 \end{bmatrix} \end{array}$$

Iterasi untuk k = 5

Dengan cara yang sama, diperoleh matriks:

$$W_5 = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \end{array} \begin{array}{c} \begin{array}{cccccc} v_1 & v_2 & v_3 & v_4 & v_5 & v_6 \end{array} \\ \begin{bmatrix} 9 & 6 & 9 & 2 & 7 & 12 \\ 3 & 9 & 3 & 5 & 1 & 6 \\ \infty & \infty & \infty & \infty & \infty & 3 \\ 7 & 4 & 7 & 9 & 5 & 10 \\ 2 & 8 & 2 & 4 & 9 & 5 \\ 4 & 1 & 4 & 6 & 2 & 7 \end{bmatrix} \end{array}$$

25

CONTOH 33 → Penyelesaian

Iterasi untuk k = 6

Dengan cara yang sama, diperoleh matriks:

$$W^* = W_6 = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \end{array} \begin{array}{c} \begin{array}{cccccc} v_1 & v_2 & v_3 & v_4 & v_5 & v_6 \end{array} \\ \begin{bmatrix} 9 & 6 & 9 & 2 & 7 & 12 \\ 3 & 7 & 3 & 5 & 1 & 6 \\ 7 & 4 & 7 & 9 & 5 & 3 \\ 7 & 4 & 7 & 9 & 5 & 10 \\ 2 & 6 & 2 & 4 & 7 & 5 \\ 4 & 1 & 4 & 6 & 2 & 7 \end{bmatrix} \end{array}$$

Catatan:

Jika pada W* ada w_{ij} dengan nilai ∞, hal ini berarti tidak ada path dari v_i ke v_j baik langsung maupun tidak langsung

26

PATH MINIMUM → Algoritma Dijkstraa

- Misalkan G adalah graf berarah berlabel dengan titik-titik V(G) = [v_1, v_2, ..., v_n] dan path terpendek yang dicari adalah dari v_1 ke v_n. Algoritma Dijkstraa dimulai dari titik v_1. Dalam iterasinya, algoritma akan mencari satu titik yang jumlah bobotnya dari titik 1 terkecil. Titik-titik yang terpilih dipisahkan, dan titik-titik tersebut tidak diperhatikan lagi dalam iterasi berikutnya.
- Misalkan:

V(G)	=	[v_1, v_2, ..., v_n]
L	=	Himpunan titik-titik $\in$ V(G) yang sudah terpilih dalam jalur path terpendek.
D(j)	=	Jumlah bobot path terkecil dari v_1 ke v_j
w(i, j)	=	Bobot garis dari titik v_1 ke v_j
w*(1, j)	=	Jumlah bobot path terkecil dari v_1 ke v_j

27

PATH MINIMUM → Algoritma Dijkstraa

1. L = { }; V = {v_2, v_3, ..., v_n}
2. Untuk i = 2, ..., n, lakukan D(i) = W(1,i)
3. Selama $v_n \notin$ L, lakukan:
   a. Pilih titik $v_k \in$ V–L dengan D(k) terkecil.
      L = L $\cup$ {v_k}
   b. Untuk setiap $v_j \in$ V–L, lakukan:
      Jika D(j) > D(k) + W(k, j) maka ganti D(j) dengan D(k) + W(k, j)
4. Untuk setiap $v_j \in$ V, w*(1, j) = D(j)

Catatan: Menurut algoritma diatas, path terpendek dari titik v_1 ke v_n adalah melalui titik-titik dalam L secara berurutan, dan jumlah bobot path terkecilnya adalah D(n).

28