Dasar Teori Graf (Bagian 4)

Masalah Lintasan Terpendek dan Algoritma Dijkstra, Masalah Pedagang Keliling, serta Masalah Tukang Pos Tiongkok

MZI

Fakultas Informatika
Telkom University

FIF Tel-U

Mei 2023

Acknowledgements

Slide ini disusun berdasarkan materi yang terdapat pada sumber-sumber berikut:

1. *Discrete Mathematics and Its Applications*, Edisi 8, 2019, oleh K. H. Rosen (acuan utama).
2. *Discrete Mathematics with Applications*, Edisi 5, 2018, oleh S. S. Epp.
3. *Mathematics for Computer Science*. MIT, 2010, oleh E. Lehman, F. T. Leighton, A. R. Meyer.
4. *Slide* kuliah Matematika Diskret 2 (2012) di Fasilkom UI oleh B. H. Widjaja.
5. Slide kuliah Matematika Diskret 2 di Fasilkom UI oleh Tim Dosen.
6. *Slide* kuliah Matematika Diskrit di Telkom University oleh B. Purnama dan rekan-rekan.

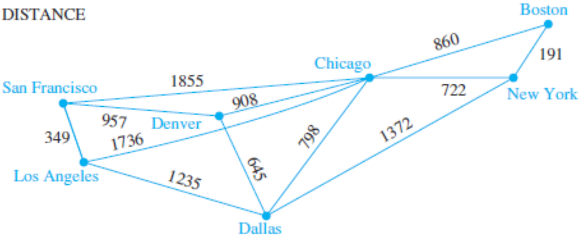
Beberapa gambar dapat diambil dari sumber-sumber di atas. *Slide* ini ditujukan untuk keperluan akademis di lingkungan FIF Telkom University. Jika Anda memiliki saran/ pendapat/ pertanyaan terkait materi dalam *slide* ini, silakan kirim email ke `<pleasedontspam>@telkomuniversity.ac.id`.

Bahasan

Bahasan

Motivasi: Penentuan Jarak Terpendek

Perhatikan graf yang setiap sisinya diberi keterangan jarak berikut.



Kita tertarik untuk menentukan sebuah lintasan terpendek (**dengan jarak minimum**) antar dua kota tertentu. Permasalahan penentuan lintasan dengan jarak minimum ini disebut sebagai masalah lintasan terpendek (*shortest path problem*).

Graf Berbobot

Untuk menentukan lintasan terpendek antara dua simpul, kita menggunakan model graf berbobot.

Definisi (Graf berbobot (*weighted graph*))

Misalkan $G = (V, E, f)$ adalah suatu graf tak berarah. Graf G dikatakan sebagai graf berbobot apabila setiap sisinya diberi label bobot. Biasanya bobot berupa suatu bilangan real positif. Bobot dari sisi $\{u, v\}$ dinyatakan dengan $w(u, v)$. Dalam hal ini $w(u, v) = w(v, u)$.

Berikut adalah contoh graf berbobot.

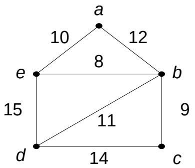
Graf Berbobot

Untuk menentukan lintasan terpendek antara dua simpul, kita menggunakan model graf berbobot.

Definisi (Graf berbobot (*weighted graph*))

Misalkan $G = (V, E, f)$ adalah suatu graf tak berarah. Graf G dikatakan sebagai graf berbobot apabila setiap sisinya diberi label bobot. Biasanya bobot berupa suatu bilangan real positif. Bobot dari sisi $\{u, v\}$ dinyatakan dengan $w(u, v)$. Dalam hal ini $w(u, v) = w(v, u)$.

Berikut adalah contoh graf berbobot.



Pada graf di atas, bobot dari sisi $\{a, b\}$ adalah $w(a, b) = w(b, a) =$

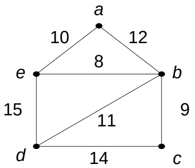
Graf Berbobot

Untuk menentukan lintasan terpendek antara dua simpul, kita menggunakan model graf berbobot.

Definisi (Graf berbobot (*weighted graph*))

Misalkan $G = (V, E, f)$ adalah suatu graf tak berarah. Graf G dikatakan sebagai graf berbobot apabila setiap sisinya diberi label bobot. Biasanya bobot berupa suatu bilangan real positif. Bobot dari sisi $\{u, v\}$ dinyatakan dengan $w(u, v)$. Dalam hal ini $w(u, v) = w(v, u)$.

Berikut adalah contoh graf berbobot.



Pada graf di atas, bobot dari sisi $\{a, b\}$ adalah $w(a, b) = w(b, a) = 12$. Bobot dari sisi $\{b, c\}$ adalah $w(b, c) = w(c, b) =$

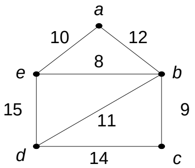
Graf Berbobot

Untuk menentukan lintasan terpendek antara dua simpul, kita menggunakan model graf berbobot.

Definisi (Graf berbobot (*weighted graph*))

Misalkan $G = (V, E, f)$ adalah suatu graf tak berarah. Graf G dikatakan sebagai graf berbobot apabila setiap sisinya diberi label bobot. Biasanya bobot berupa suatu bilangan real positif. Bobot dari sisi $\{u, v\}$ dinyatakan dengan $w(u, v)$. Dalam hal ini $w(u, v) = w(v, u)$.

Berikut adalah contoh graf berbobot.



Pada graf di atas, bobot dari sisi $\{a, b\}$ adalah $w(a, b) = w(b, a) = 12$. Bobot dari sisi $\{b, c\}$ adalah $w(b, c) = w(c, b) = 9$.

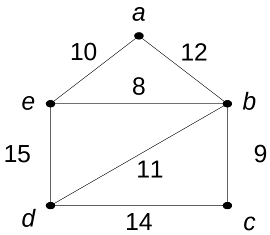
Matriks Jarak Graf Berbobot

Definisi

Misalkan $G = (V, E)$ adalah sebuah graf berbobot tak berarah tanpa sisi ganda maupun gelang. Jika bobot sisi $\{i, j\}$ adalah $w(i, j)$, maka matriks jarak untuk graf G adalah $\mathbf{D}_G$ yang didefinisikan sebagai

$$\mathbf{D}_G[i, j] = \begin{cases} w(i, j), & \text{jika simpul } i \text{ dan } j \text{ bertetangga,} \\ \infty, & \text{jika simpul } i \text{ dan } j \text{ tidak bertetangga.} \end{cases}$$

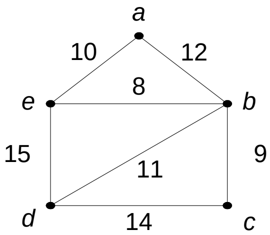
Sebagi contoh, untuk graf G berikut



Maka

$$\mathbf{D}_G =$$

Sebagi contoh, untuk graf G berikut



Maka

$$\mathbf{D}_G = \begin{array}{c} \\ a \\ b \\ c \\ d \\ e \end{array} \begin{array}{c} \begin{array}{ccccc} a & b & c & d & e \end{array} \\ \begin{bmatrix} \infty & 12 & \infty & \infty & 10 \\ 12 & \infty & 9 & 11 & 8 \\ \infty & 9 & \infty & 14 & \infty \\ \infty & 11 & 14 & \infty & 15 \\ 10 & 8 & \infty & 15 & \infty \end{bmatrix} \end{array}$$

Jenis-jenis Masalah Lintasan Terpendek

Ada beberapa jenis masalah lintasan terpendek yang mungkin ditinjau:

1. lintasan terpendek antara dua simpul tertentu,
2. lintasan terpendek antara semua pasangan simpul pada graf,
3. lintasan terpendek dari satu simpul tertentu ke semua simpul pada graf,
4. lintasan terpendek antara dua simpul yang harus melalui beberapa simpul tertentu.

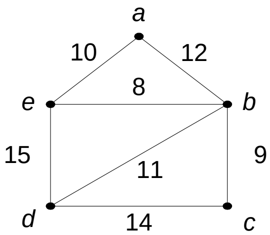
Bahasan

Bobot Lintasan

Definisi (Panjang lintasan)

Misalkan $G = (V, E, f)$ adalah suatu graf terhubung berbobot tak berarah dan $u, v \in V$. Panjang lintasan dari u ke v, $L(u, v)$, merupakan jumlah bobot yang terdapat pada sisi yang ada pada lintasan dari u ke v.

Sebagi contoh, pada graf berikut



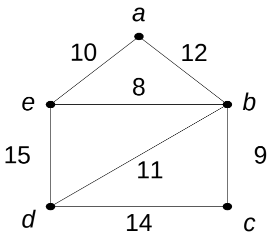
Panjang lintasan terpendek dari a ke c adalah

Bobot Lintasan

Definisi (Panjang lintasan)

Misalkan $G = (V, E, f)$ adalah suatu graf terhubung berbobot tak berarah dan $u, v \in V$. Panjang lintasan dari u ke v, $L(u, v)$, merupakan jumlah bobot yang terdapat pada sisi yang ada pada lintasan dari u ke v.

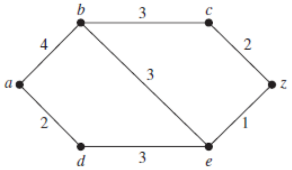
Sebagi contoh, pada graf berikut



Panjang lintasan terpendek dari a ke c adalah $12 + 9 = 21$, sehingga $L(a, c) = 21$ untuk lintasan $\langle a, b, c \rangle$.

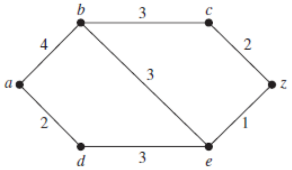
Contoh Penentuan Lintasan Terpendek Sederhana

Kita akan mencari lintasan terpendek dari a ke z pada graf berikut.



Contoh Penentuan Lintasan Terpendek Sederhana

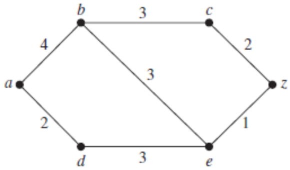
Kita akan mencari lintasan terpendek dari a ke z pada graf berikut.



1. Lintasan yang memuat dua simpul dengan a sebagai simpul awal ada dua, yaitu $\langle a, b\rangle$ dan $\langle a, d\rangle$. Perhatikan bahwa $L(a, b) = 4$ dan $L(a, d) = 2$. Oleh karena itu d merupakan simpul terdekat dari a.

Contoh Penentuan Lintasan Terpendek Sederhana

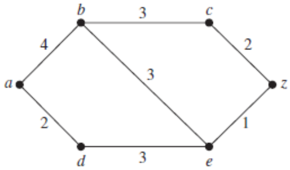
Kita akan mencari lintasan terpendek dari a ke z pada graf berikut.



1. Lintasan yang memuat dua simpul dengan a sebagai simpul awal ada dua, yaitu $\langle a, b\rangle$ dan $\langle a, d\rangle$. Perhatikan bahwa $L(a, b) = 4$ dan $L(a, d) = 2$. Oleh karena itu d merupakan simpul terdekat dari a.
2. Selanjutnya akan dicari simpul terdekat berikutnya. Perhatikan bahwa lintasan terpendek dari a ke b adalah $\langle a, b\rangle$ dengan $L(a, b) = 4$. Kemudian lintasan terpendek dari a ke e adalah $\langle a, d, e\rangle$, dengan $L(a, e) = 5$.

Contoh Penentuan Lintasan Terpendek Sederhana

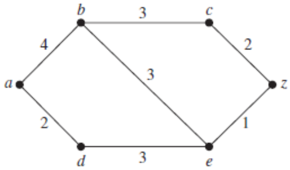
Kita akan mencari lintasan terpendek dari a ke z pada graf berikut.



1. Lintasan yang memuat dua simpul dengan a sebagai simpul awal ada dua, yaitu $\langle a, b\rangle$ dan $\langle a, d\rangle$. Perhatikan bahwa $L(a, b) = 4$ dan $L(a, d) = 2$. Oleh karena itu d merupakan simpul terdekat dari a.
2. Selanjutnya akan dicari simpul terdekat berikutnya. Perhatikan bahwa lintasan terpendek dari a ke b adalah $\langle a, b\rangle$ dengan $L(a, b) = 4$. Kemudian lintasan terpendek dari a ke e adalah $\langle a, d, e\rangle$, dengan $L(a, e) = 5$.
3. Lintasan terpendek dari a ke c adalah $\langle a, b, c\rangle$ dengan $L(a, c) = 7$. Lintasan terpendek dari a ke z adalah $\langle a, d, e, z\rangle$, dengan $L(a, z) = 6$.

Contoh Penentuan Lintasan Terpendek Sederhana

Kita akan mencari lintasan terpendek dari a ke z pada graf berikut.



1. Lintasan yang memuat dua simpul dengan a sebagai simpul awal ada dua, yaitu $\langle a, b\rangle$ dan $\langle a, d\rangle$. Perhatikan bahwa $L(a, b) = 4$ dan $L(a, d) = 2$. Oleh karena itu d merupakan simpul terdekat dari a.
2. Selanjutnya akan dicari simpul terdekat berikutnya. Perhatikan bahwa lintasan terpendek dari a ke b adalah $\langle a, b\rangle$ dengan $L(a, b) = 4$. Kemudian lintasan terpendek dari a ke e adalah $\langle a, d, e\rangle$, dengan $L(a, e) = 5$.
3. Lintasan terpendek dari a ke c adalah $\langle a, b, c\rangle$ dengan $L(a, c) = 7$. Lintasan terpendek dari a ke z adalah $\langle a, d, e, z\rangle$, dengan $L(a, z) = 6$.

Jadi lintasan terpendek dari a ke z adalah $\langle a, d, e, z\rangle$ dengan panjang 6.
Perhatikan bahwa tidak semua simpul pada graf G harus dilalui.

Inisialisasi Algoritma Dijkstra

- Algoritma Dijkstra yang akan dibahas berikut digunakan untuk mencari lintasan terpendek dari suatu simpul (yaitu a) ke simpul lain (yaitu z).
- Notasi $L_k(v)$ merupakan ringkasan dari notasi $L_k(a, v)$, yaitu panjang lintasan terpendek dari simpul a ke v setelah iterasi ke-k.
- Pada iterasi ke-0 (tahap inisialisasi), $L_0(a) = 0$ dan $L_0(v) = \infty$ untuk setiap simpul selain a.
- Notasi S_k menyatakan himpunan simpul yang diperiksa pada suatu lintasan setelah iterasi ke-k.
    1. Dalam tahap inisialisasi $S_0 = \emptyset$.
    2. Himpunan S_k diperoleh dari himpunan S_{k-1} dengan menambahkan simpul u yang tidak berada di S_{k-1} dengan sifat $L_{k-1}(u)$ minimal.

Sifat Penting Pada Algoritma Dijkstra

Algoritma Dijkstra menggunakan sifat penting berikut.

Misalkan $v \notin S_k$. Lintasan terpendek dari a ke v yang memuat simpul-simpul pada S_k adalah salah satu dari lintasan berikut

Sifat Penting Pada Algoritma Dijkstra

Algoritma Dijkstra menggunakan sifat penting berikut.

Misalkan $v \notin S_k$. Lintasan terpendek dari a ke v yang memuat simpul-simpul pada S_k adalah salah satu dari lintasan berikut

1. lintasan yang menghubungkan a ke v yang memuat setiap simpul pada S_{k-1}, atau

Sifat Penting Pada Algoritma Dijkstra

Algoritma Dijkstra menggunakan sifat penting berikut.

Misalkan $v \notin S_k$. Lintasan terpendek dari a ke v yang memuat simpul-simpul pada S_k adalah salah satu dari lintasan berikut

1. lintasan yang menghubungkan a ke v yang memuat setiap simpul pada S_{k-1}, atau
2. lintasan yang terdiri atas lintasan yang menghubungkan a ke u (untuk suatu simpul u pada S_{k-1}) dan sisi $\{u, v\}$.

Dari dua hal ini, diperoleh

$$L_k(a, v) =$$

Sifat Penting Pada Algoritma Dijkstra

Algoritma Dijkstra menggunakan sifat penting berikut.

Misalkan $v \notin S_k$. Lintasan terpendek dari a ke v yang memuat simpul-simpul pada S_k adalah salah satu dari lintasan berikut

1. lintasan yang menghubungkan a ke v yang memuat setiap simpul pada S_{k-1}, atau
2. lintasan yang terdiri atas lintasan yang menghubungkan a ke u (untuk suatu simpul u pada S_{k-1}) dan sisi $\{u, v\}$.

Dari dua hal ini, diperoleh

$$\begin{aligned} L_k(a,v) &= \min\{L_{k-1}(a,v), L_{k-1}(a,u) + w(u,v)\}, \text{ atau} \quad (1) \\ L_k(v) &= \end{aligned}$$

Sifat Penting Pada Algoritma Dijkstra

Algoritma Dijkstra menggunakan sifat penting berikut.

Misalkan $v \notin S_k$. Lintasan terpendek dari a ke v yang memuat simpul-simpul pada S_k adalah salah satu dari lintasan berikut

1. lintasan yang menghubungkan a ke v yang memuat setiap simpul pada S_{k-1}, atau
2. lintasan yang terdiri atas lintasan yang menghubungkan a ke u (untuk suatu simpul u pada S_{k-1}) dan sisi $\{u, v\}$.

Dari dua hal ini, diperoleh

$$\begin{aligned} L_k(a, v) &= \min\{L_{k-1}(a, v), L_{k-1}(a, u) + w(u, v)\}, \text{ atau} && (1) \\ L_k(v) &= \min\{L_{k-1}(v), L_{k-1}(u) + w(u, v)\}. && (2) \end{aligned}$$

Dalam formulasi (1) maupun (2), u merupakan suatu simpul dengan sifat

$$L_{k-1}(u) \leq L_{k-1}(x), \text{ untuk setiap } x \in S_{k-1}.$$

Dengan perkataan lain $L_{k-1}(u)$ menyatakan panjang lintasan terpendek dari a ke u setelah iterasi ke-$(k-1)$.

Pseudocode Algoritma Dijkstra

Algoritma Dijkstra: Inisialisasi

1. Masukan: $G = (V, E)$ yang terhubung sederhana dengan bobot-bobot sisi positif.
2. Simpul pada G adalah $a = v_0, v_1, v_2, \ldots, v_{n-1}, v_n = z$.
3. Inisialisasi $L_0(a) = 0$ dan $L_0(v) = \infty$ untuk setiap $v \neq a$, $v \in V$.
4. Inisialiasi himpunan simpul $S = \emptyset$. Himpunan S adalah himpunan simpul-simpul yang diperiksa. Salah satu himpunan bagian dari S adalah himpunan simpul-simpul yang terdapat pada lintasan terpendek dari a ke z. Jadi himpunan simpul-simpul yang terdapat pada lintasan tidak selalu sama dengan S.
5. `for` $i := 1$ `to` n
6. $\quad L(v_i) := \infty$
7. `end for`
8. $L(a) := 0$.

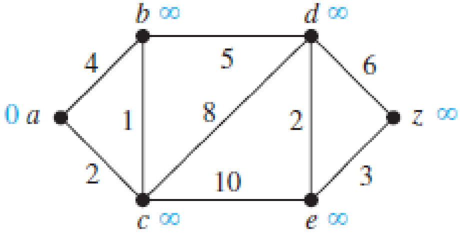
Algoritma Dijkstra: Iterasi

1. `while` $z \notin S$
2. `begin`
3. $u :=$ simpul yang tidak ada di S dengan $L(u)$ minimal
4. $S := S \cup \{u\}$
5. `for all` $v \notin S$
6. $L(v) := \min\{L(v), L(u) + w(u,v)\}$
   {proses ini menambah sebuah simpul ke S dengan label jarak minimal dan memperbarui label dari simpul yang tidak terdapat di S}
7. `end for`
8. `end while`
9. $L(z)$ adalah panjang lintasan terpendek dari a ke z.
10. Jika elemen-elemen $S = \{a = u_0, u_1, \ldots, u_k = z\}$ diurutkan, maka suatu himpunan bagian dari S dengan suatu urutan tertentu merupakan lintasan terpendek dari a ke z.

Bahasan

Contoh Penerapan Algoritma Dijkstra

Misalkan kita akan memakai algoritma Dijkstra untuk menentukan lintasan terpendek dari a ke z pada graf berikut.

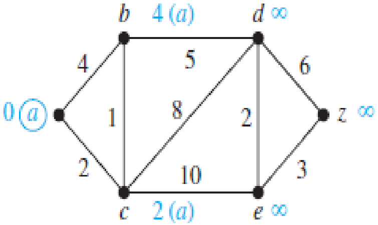


Graf G

Inisialisasi

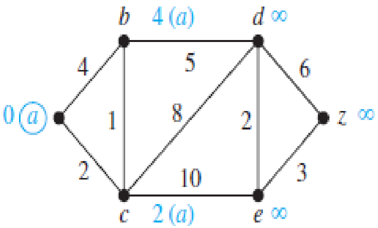
1. $S_0 = \emptyset$.
2. $L_0(a) = 0$.
3. $L_0(b) = L_0(c) = L_0(d) = L_0(e) = L_0(z) = \infty$.
4. Karena $z \notin S_0$ maka iterasi dilanjutkan.

Iterasi Pertama



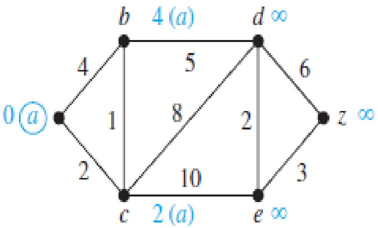
1. $u := a$ (karena $a \notin S_0$ dan $L_0(a)$ minimal).

Iterasi Pertama



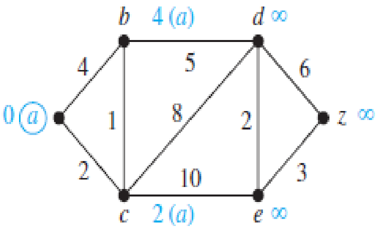
1. $u := a$ (karena $a \notin S_0$ dan $L_0(a)$ minimal).
2. $S_1 = S_0 \cup \{a\} = \emptyset \cup \{a\} = \{a\}$.
3. $L_1(b) = \min\{L_0(b), L_0(a) + w(a, b)\} = \min\{\infty, 0 + 4\} =$

Iterasi Pertama



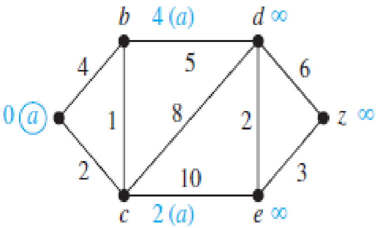
1. $u := a$ (karena $a \notin S_0$ dan $L_0(a)$ minimal).
2. $S_1 = S_0 \cup \{a\} = \emptyset \cup \{a\} = \{a\}$.
3. $L_1(b) = \min\{L_0(b), L_0(a) + w(a,b)\} = \min\{\infty, 0+4\} = 4$. Lintasan:

Iterasi Pertama



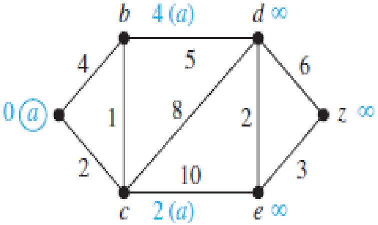
1. $u := a$ (karena $a \notin S_0$ dan $L_0(a)$ minimal).
2. $S_1 = S_0 \cup \{a\} = \emptyset \cup \{a\} = \{a\}$.
3. $L_1(b) = \min\{L_0(b), L_0(a) + w(a,b)\} = \min\{\infty, 0+4\} = 4$. Lintasan: $\langle a, b \rangle$.
   $L_1(c) = \min\{L_0(c), L_0(a) + w(a,c)\} = \min\{\infty, 0+2\} =$

Iterasi Pertama



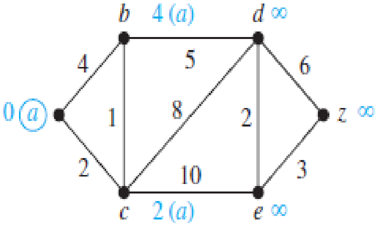
1. $u := a$ (karena $a \notin S_0$ dan $L_0(a)$ minimal).
2. $S_1 = S_0 \cup \{a\} = \emptyset \cup \{a\} = \{a\}$.
3. $L_1(b) = \min\{L_0(b), L_0(a) + w(a,b)\} = \min\{\infty, 0+4\} = 4$. Lintasan: $\langle a, b \rangle$.
   $L_1(c) = \min\{L_0(c), L_0(a) + w(a,c)\} = \min\{\infty, 0+2\} = 2$. Lintasan:

Iterasi Pertama



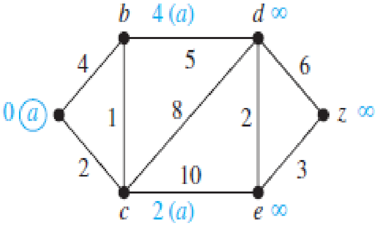
1. $u := a$ (karena $a \notin S_0$ dan $L_0(a)$ minimal).
2. $S_1 = S_0 \cup \{a\} = \emptyset \cup \{a\} = \{a\}$.
3. $L_1(b) = \min\{L_0(b), L_0(a) + w(a,b)\} = \min\{\infty, 0+4\} = 4$. Lintasan: $\langle a, b\rangle$.
   $L_1(c) = \min\{L_0(c), L_0(a) + w(a,c)\} = \min\{\infty, 0+2\} = 2$. Lintasan: $\langle a, c\rangle$.

Iterasi Pertama



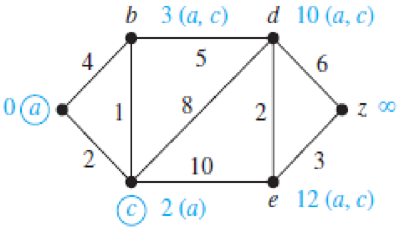
1. $u := a$ (karena $a \notin S_0$ dan $L_0(a)$ minimal).
2. $S_1 = S_0 \cup \{a\} = \emptyset \cup \{a\} = \{a\}$.
3. $L_1(b) = \min\{L_0(b), L_0(a) + w(a,b)\} = \min\{\infty, 0+4\} = 4$. Lintasan: $\langle a, b \rangle$.
   $L_1(c) = \min\{L_0(c), L_0(a) + w(a,c)\} = \min\{\infty, 0+2\} = 2$. Lintasan: $\langle a, c \rangle$.
   $L_1(d) = L_1(e) = L_1(z) = \infty$ karena $w(a,d) = w(a,e) = w(a,z) = \infty$.

Iterasi Pertama



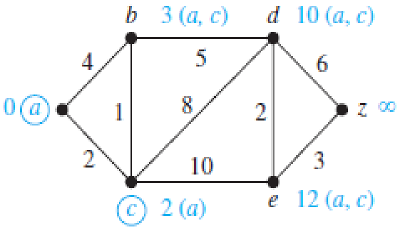
1. $u := a$ (karena $a \notin S_0$ dan $L_0(a)$ minimal).
2. $S_1 = S_0 \cup \{a\} = \emptyset \cup \{a\} = \{a\}$.
3. $L_1(b) = \min\{L_0(b), L_0(a) + w(a,b)\} = \min\{\infty, 0+4\} = 4$. Lintasan: $\langle a, b \rangle$.
   $L_1(c) = \min\{L_0(c), L_0(a) + w(a,c)\} = \min\{\infty, 0+2\} = 2$. Lintasan: $\langle a, c \rangle$.
   $L_1(d) = L_1(e) = L_1(z) = \infty$ karena $w(a,d) = w(a,e) = w(a,z) = \infty$.
4. Karena $z \notin S_1$ maka iterasi dilanjutkan.

Iterasi Ke-2



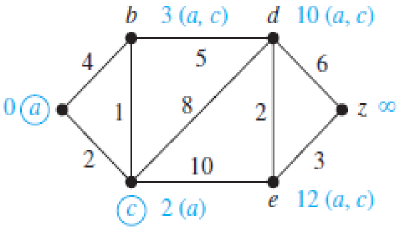
1. $u := c$ (karena $c \notin S_1$ dan $L_1(c)$ minimal).

Iterasi Ke-2



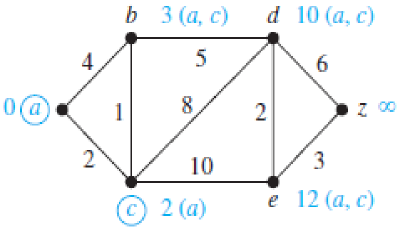
1. $u := c$ (karena $c \notin S_1$ dan $L_1(c)$ minimal).
2. $S_2 = S_1 \cup \{c\} = \{a\} \cup \{c\} = \{a, c\}$.
3. $L_2(b) = \min\{L_1(b), L_1(c) + w(c, b)\} = \min\{4, 2 + 1\} =$

Iterasi Ke-2



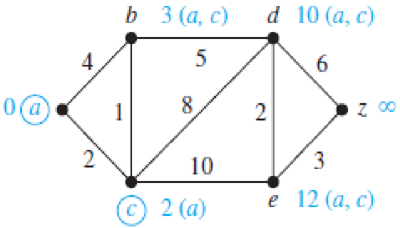
1. $u := c$ (karena $c \notin S_1$ dan $L_1(c)$ minimal).
2. $S_2 = S_1 \cup \{c\} = \{a\} \cup \{c\} = \{a, c\}$.
3. $L_2(b) = \min\{L_1(b), L_1(c) + w(c, b)\} = \min\{4, 2 + 1\} = 3$. Lintasan:

Iterasi Ke-2



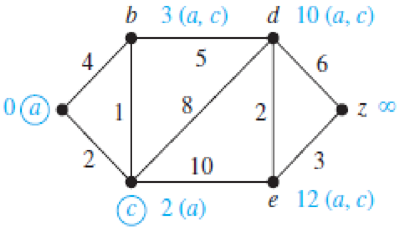
1. $u := c$ (karena $c \notin S_1$ dan $L_1(c)$ minimal).
2. $S_2 = S_1 \cup \{c\} = \{a\} \cup \{c\} = \{a, c\}$.
3. $L_2(b) = \min\{L_1(b), L_1(c) + w(c, b)\} = \min\{4, 2 + 1\} = 3$. Lintasan: $\langle a, c, b \rangle$.
   $L_2(d) = \min\{L_1(d), L_1(c) + w(c, d)\} = \min\{\infty, 2 + 8\} =$

Iterasi Ke-2



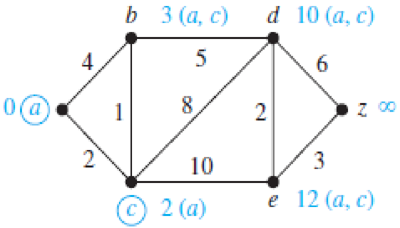
1. $u := c$ (karena $c \notin S_1$ dan $L_1(c)$ minimal).
2. $S_2 = S_1 \cup \{c\} = \{a\} \cup \{c\} = \{a, c\}$.
3. $L_2(b) = \min\{L_1(b), L_1(c) + w(c, b)\} = \min\{4, 2+1\} = 3$. Lintasan: $\langle a, c, b \rangle$.
   $L_2(d) = \min\{L_1(d), L_1(c) + w(c, d)\} = \min\{\infty, 2+8\} = 10$. Lintasan:

Iterasi Ke-2



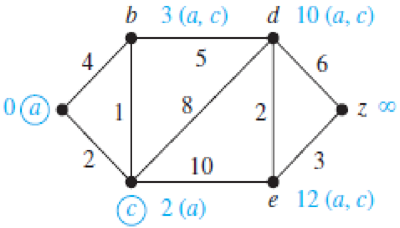
1. $u := c$ (karena $c \notin S_1$ dan $L_1(c)$ minimal).
2. $S_2 = S_1 \cup \{c\} = \{a\} \cup \{c\} = \{a, c\}$.
3. $L_2(b) = \min\{L_1(b), L_1(c) + w(c, b)\} = \min\{4, 2 + 1\} = 3$. Lintasan: $\langle a, c, b\rangle$.
   $L_2(d) = \min\{L_1(d), L_1(c) + w(c, d)\} = \min\{\infty, 2 + 8\} = 10$. Lintasan: $\langle a, c, d\rangle$.
   $L_2(e) = \min\{L_1(e), L_1(c) + w(c, e)\} = \min\{\infty, 2 + 10\} =$

Iterasi Ke-2



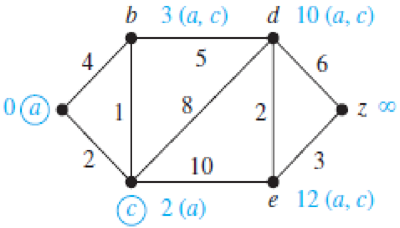
1. $u := c$ (karena $c \notin S_1$ dan $L_1(c)$ minimal).
2. $S_2 = S_1 \cup \{c\} = \{a\} \cup \{c\} = \{a, c\}$.
3. $L_2(b) = \min\{L_1(b), L_1(c) + w(c,b)\} = \min\{4, 2+1\} = 3$. Lintasan: $\langle a, c, b \rangle$.
   $L_2(d) = \min\{L_1(d), L_1(c) + w(c,d)\} = \min\{\infty, 2+8\} = 10$. Lintasan: $\langle a, c, d \rangle$.
   $L_2(e) = \min\{L_1(e), L_1(c) + w(c,e)\} = \min\{\infty, 2+10\} = 12$. Lintasan:

Iterasi Ke-2



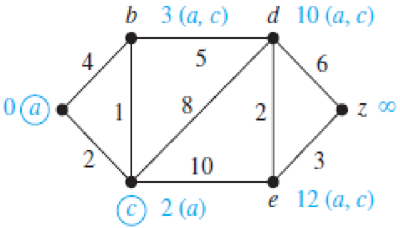
1. $u := c$ (karena $c \notin S_1$ dan $L_1(c)$ minimal).
2. $S_2 = S_1 \cup \{c\} = \{a\} \cup \{c\} = \{a, c\}$.
3. $L_2(b) = \min\{L_1(b), L_1(c) + w(c, b)\} = \min\{4, 2+1\} = 3$. Lintasan: $\langle a, c, b \rangle$.
   $L_2(d) = \min\{L_1(d), L_1(c) + w(c, d)\} = \min\{\infty, 2+8\} = 10$. Lintasan: $\langle a, c, d \rangle$.
   $L_2(e) = \min\{L_1(e), L_1(c) + w(c, e)\} = \min\{\infty, 2+10\} = 12$. Lintasan: $\langle a, c, e \rangle$.

Iterasi Ke-2



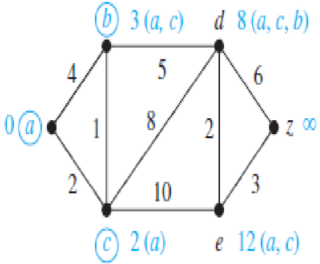
1. $u := c$ (karena $c \notin S_1$ dan $L_1(c)$ minimal).
2. $S_2 = S_1 \cup \{c\} = \{a\} \cup \{c\} = \{a, c\}$.
3. $L_2(b) = \min\{L_1(b), L_1(c) + w(c,b)\} = \min\{4, 2+1\} = 3$. Lintasan: $\langle a, c, b \rangle$.
   $L_2(d) = \min\{L_1(d), L_1(c) + w(c,d)\} = \min\{\infty, 2+8\} = 10$. Lintasan: $\langle a, c, d \rangle$.
   $L_2(e) = \min\{L_1(e), L_1(c) + w(c,e)\} = \min\{\infty, 2+10\} = 12$. Lintasan: $\langle a, c, e \rangle$.
   $L_2(z) = \min\{L_1(z), L_1(c) + w(c,z)\} = \infty$, karena $\{c, z\} \notin E$.

Iterasi Ke-2



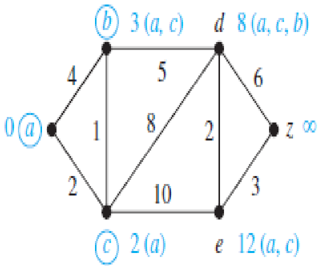
1. $u := c$ (karena $c \notin S_1$ dan $L_1(c)$ minimal).
2. $S_2 = S_1 \cup \{c\} = \{a\} \cup \{c\} = \{a, c\}$.
3. $L_2(b) = \min\{L_1(b), L_1(c) + w(c,b)\} = \min\{4, 2+1\} = 3$. Lintasan: $\langle a, c, b \rangle$.
   $L_2(d) = \min\{L_1(d), L_1(c) + w(c,d)\} = \min\{\infty, 2+8\} = 10$. Lintasan: $\langle a, c, d \rangle$.
   $L_2(e) = \min\{L_1(e), L_1(c) + w(c,e)\} = \min\{\infty, 2+10\} = 12$. Lintasan: $\langle a, c, e \rangle$.
   $L_2(z) = \min\{L_1(z), L_1(c) + w(c,z)\} = \infty$, karena $\{c, z\} \notin E$.
4. Karena $z \notin S_2$ maka iterasi dilanjutkan.

Iterasi Ke-3



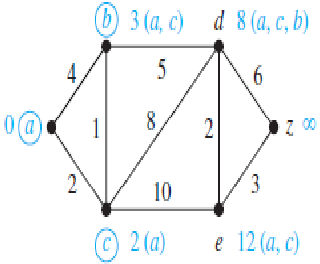
1. $u := b$ (karena $b \notin S_2$ dan $L_2\,(b)$ minimal).

Iterasi Ke-3



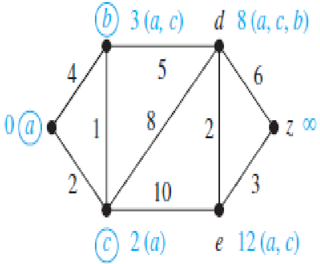
1. $u := b$ (karena $b \notin S_2$ dan $L_2(b)$ minimal).
2. $S_3 = S_2 \cup \{b\} = \{a, c\} \cup \{b\} = \{a, c, b\}$.
3. $L_3(d) = \min\{L_2(d), L_2(b) + w(b, d)\} = \min\{10, 3 + 5\} =$

Iterasi Ke-3



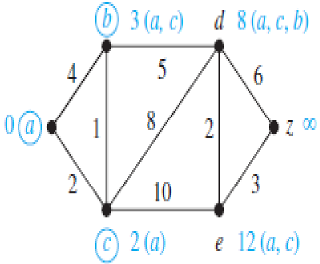
1. $u := b$ (karena $b \notin S_2$ dan $L_2(b)$ minimal).
2. $S_3 = S_2 \cup \{b\} = \{a, c\} \cup \{b\} = \{a, c, b\}$.
3. $L_3(d) = \min\{L_2(d), L_2(b) + w(b, d)\} = \min\{10, 3 + 5\} = 8$. Lintasan:

Iterasi Ke-3



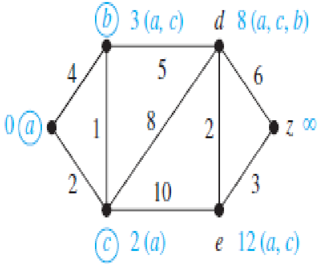
1. $u := b$ (karena $b \notin S_2$ dan $L_2(b)$ minimal).
2. $S_3 = S_2 \cup \{b\} = \{a, c\} \cup \{b\} = \{a, c, b\}$.
3. $L_3(d) = \min\{L_2(d), L_2(b) + w(b, d)\} = \min\{10, 3 + 5\} = 8$. Lintasan: $\langle a, c, b, d\rangle$.
   $L_3(e) = \min\{L_2(e), L_2(b) + w(b, e)\} = \min\{12, 3 + \infty\} =$

Iterasi Ke-3



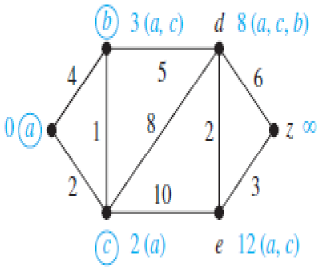
1. $u := b$ (karena $b \notin S_2$ dan $L_2(b)$ minimal).
2. $S_3 = S_2 \cup \{b\} = \{a, c\} \cup \{b\} = \{a, c, b\}$.
3. $L_3(d) = \min\{L_2(d), L_2(b) + w(b, d)\} = \min\{10, 3 + 5\} = 8$. Lintasan: $\langle a, c, b, d \rangle$.
   $L_3(e) = \min\{L_2(e), L_2(b) + w(b, e)\} = \min\{12, 3 + \infty\} = 12$. Lintasan:

Iterasi Ke-3



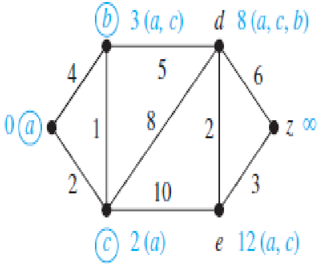
1. $u := b$ (karena $b \notin S_2$ dan $L_2(b)$ minimal).
2. $S_3 = S_2 \cup \{b\} = \{a, c\} \cup \{b\} = \{a, c, b\}$.
3. $L_3(d) = \min\{L_2(d), L_2(b) + w(b, d)\} = \min\{10, 3 + 5\} = 8$. Lintasan: $\langle a, c, b, d\rangle$.
   $L_3(e) = \min\{L_2(e), L_2(b) + w(b, e)\} = \min\{12, 3 + \infty\} = 12$. Lintasan: $\langle a, c, e\rangle$.

Iterasi Ke-3



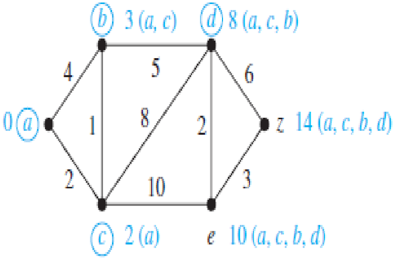
1. $u := b$ (karena $b \notin S_2$ dan $L_2(b)$ minimal).
2. $S_3 = S_2 \cup \{b\} = \{a, c\} \cup \{b\} = \{a, c, b\}$.
3. $L_3(d) = \min\{L_2(d), L_2(b) + w(b, d)\} = \min\{10, 3 + 5\} = 8$. Lintasan: $\langle a, c, b, d \rangle$.
   $L_3(e) = \min\{L_2(e), L_2(b) + w(b, e)\} = \min\{12, 3 + \infty\} = 12$. Lintasan: $\langle a, c, e \rangle$.
   $L_3(z) = \min\{L_2(z), L_2(b) + w(b, z)\} = \infty$, karena $\{b, z\} \notin E$.

Iterasi Ke-3



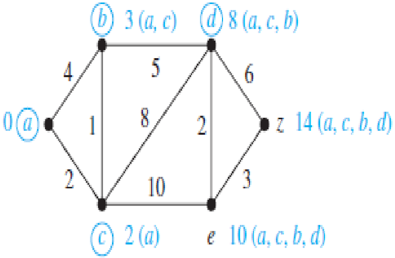
1. $u := b$ (karena $b \notin S_2$ dan $L_2(b)$ minimal).
2. $S_3 = S_2 \cup \{b\} = \{a, c\} \cup \{b\} = \{a, c, b\}$.
3. $L_3(d) = \min\{L_2(d), L_2(b) + w(b, d)\} = \min\{10, 3 + 5\} = 8$. Lintasan: $\langle a, c, b, d \rangle$.
   $L_3(e) = \min\{L_2(e), L_2(b) + w(b, e)\} = \min\{12, 3 + \infty\} = 12$. Lintasan: $\langle a, c, e \rangle$.
   $L_3(z) = \min\{L_2(z), L_2(b) + w(b, z)\} = \infty$, karena $\{b, z\} \notin E$.
4. Karena $z \notin S_3$ maka iterasi dilanjutkan.

Iterasi Ke-4



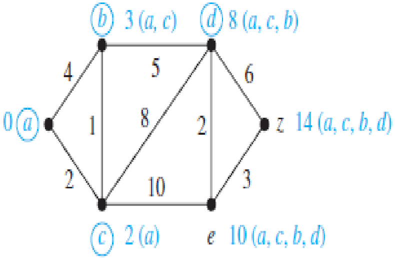
1. $u := d$ (karena $d \notin S_3$ dan $L_3\,(d)$ minimal).

Iterasi Ke-4



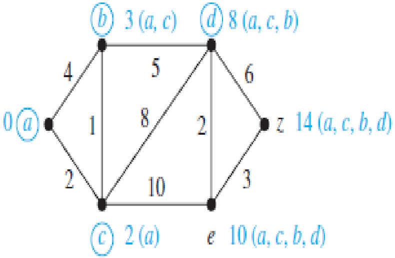
1. $u := d$ (karena $d \notin S_3$ dan $L_3(d)$ minimal).
2. $S_4 = S_3 \cup \{d\} = \{a, c, b\} \cup \{d\} = \{a, c, b, d\}$.
3. $L_4(e) = \min\{L_3(e), L_3(d) + w(d, e)\} = \min\{12, 8 + 2\} =$

Iterasi Ke-4



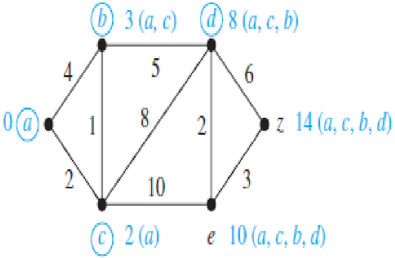
1. $u := d$ (karena $d \notin S_3$ dan $L_3(d)$ minimal).
2. $S_4 = S_3 \cup \{d\} = \{a, c, b\} \cup \{d\} = \{a, c, b, d\}$.
3. $L_4(e) = \min\{L_3(e), L_3(d) + w(d, e)\} = \min\{12, 8 + 2\} = 10$. Lintasan:

Iterasi Ke-4



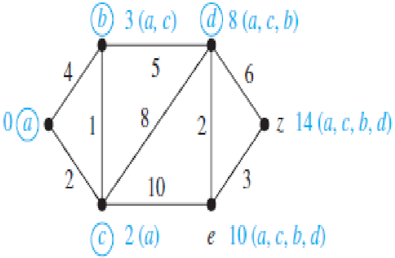
1. $u := d$ (karena $d \notin S_3$ dan $L_3(d)$ minimal).
2. $S_4 = S_3 \cup \{d\} = \{a, c, b\} \cup \{d\} = \{a, c, b, d\}$.
3. $L_4(e) = \min\{L_3(e), L_3(d) + w(d, e)\} = \min\{12, 8 + 2\} = 10$. Lintasan: $\langle a, c, b, d, e\rangle$.
   $L_4(z) = \min\{L_3(z), L_3(d) + w(d, z)\} = \min\{\infty, 8 + 6\} =$

Iterasi Ke-4



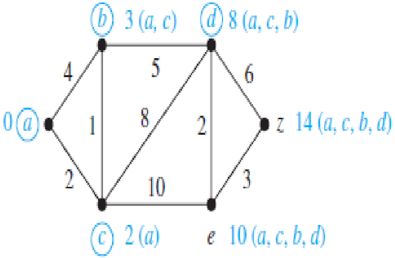
1. $u := d$ (karena $d \notin S_3$ dan $L_3(d)$ minimal).
2. $S_4 = S_3 \cup \{d\} = \{a, c, b\} \cup \{d\} = \{a, c, b, d\}$.
3. $L_4(e) = \min\{L_3(e), L_3(d) + w(d, e)\} = \min\{12, 8 + 2\} = 10$. Lintasan: $\langle a, c, b, d, e \rangle$.
   $L_4(z) = \min\{L_3(z), L_3(d) + w(d, z)\} = \min\{\infty, 8 + 6\} = 14$. Lintasan:

Iterasi Ke-4



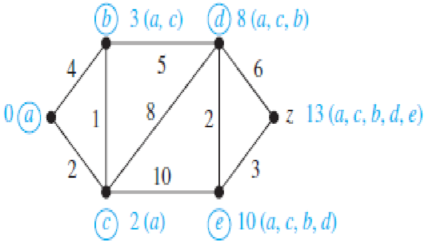
1. $u := d$ (karena $d \notin S_3$ dan $L_3(d)$ minimal).
2. $S_4 = S_3 \cup \{d\} = \{a, c, b\} \cup \{d\} = \{a, c, b, d\}$.
3. $L_4(e) = \min\{L_3(e), L_3(d) + w(d, e)\} = \min\{12, 8 + 2\} = 10$. Lintasan: $\langle a, c, b, d, e \rangle$.
   $L_4(z) = \min\{L_3(z), L_3(d) + w(d, z)\} = \min\{\infty, 8 + 6\} = 14$. Lintasan: $\langle a, c, b, d, z \rangle$.

Iterasi Ke-4



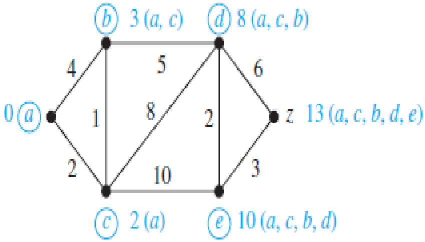
1. $u := d$ (karena $d \notin S_3$ dan $L_3(d)$ minimal).
2. $S_4 = S_3 \cup \{d\} = \{a, c, b\} \cup \{d\} = \{a, c, b, d\}$.
3. $L_4(e) = \min\{L_3(e), L_3(d) + w(d, e)\} = \min\{12, 8 + 2\} = 10$. Lintasan: $\langle a, c, b, d, e \rangle$.
   $L_4(z) = \min\{L_3(z), L_3(d) + w(d, z)\} = \min\{\infty, 8 + 6\} = 14$. Lintasan: $\langle a, c, b, d, z \rangle$.
4. Karena $z \notin S_4$ maka iterasi dilanjutkan.

Iterasi Ke-5



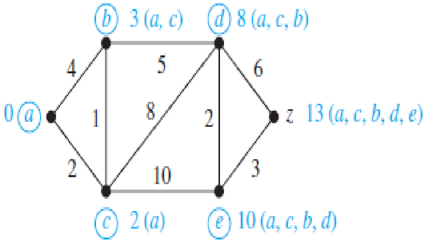
1. $u := e$ (karena $e \notin S_4$ dan $L_4(e)$ minimal).

Iterasi Ke-5



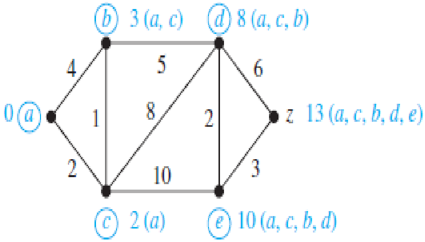
1. $u := e$ (karena $e \notin S_4$ dan $L_4(e)$ minimal).
2. $S_5 = S_4 \cup \{e\} = \{a, c, b, d\} \cup \{e\} = \{a, c, b, d, e\}$.
3. $L_5(z) = \min\{L_4(z), L_4(e) + w(e, z)\} = \min\{14, 10 + 3\} =$

Iterasi Ke-5



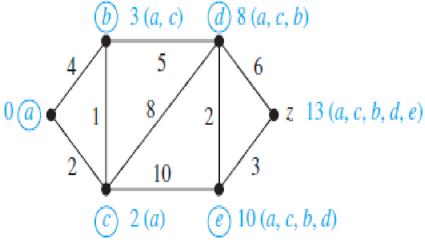
1. $u := e$ (karena $e \notin S_4$ dan $L_4(e)$ minimal).
2. $S_5 = S_4 \cup \{e\} = \{a, c, b, d\} \cup \{e\} = \{a, c, b, d, e\}$.
3. $L_5(z) = \min\{L_4(z), L_4(e) + w(e, z)\} = \min\{14, 10 + 3\} = 13$. Lintasan:

Iterasi Ke-5



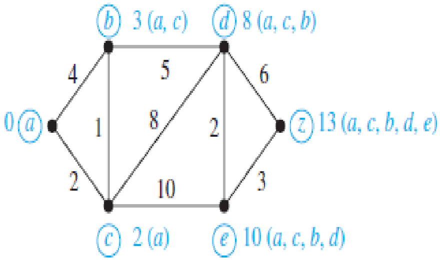
1. $u := e$ (karena $e \notin S_4$ dan $L_4(e)$ minimal).
2. $S_5 = S_4 \cup \{e\} = \{a, c, b, d\} \cup \{e\} = \{a, c, b, d, e\}$.
3. $L_5(z) = \min\{L_4(z), L_4(e) + w(e, z)\} = \min\{14, 10 + 3\} = 13$. Lintasan: $\langle a, c, b, d, e, z \rangle$.

Iterasi Ke-5



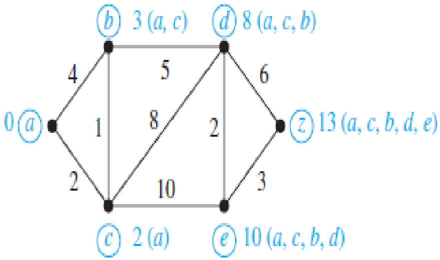
1. $u := e$ (karena $e \notin S_4$ dan $L_4(e)$ minimal).
2. $S_5 = S_4 \cup \{e\} = \{a, c, b, d\} \cup \{e\} = \{a, c, b, d, e\}$.
3. $L_5(z) = \min\{L_4(z), L_4(e) + w(e, z)\} = \min\{14, 10 + 3\} = 13$. Lintasan: $\langle a, c, b, d, e, z \rangle$.
4. Karena $z \notin S_5$ maka iterasi dilanjutkan.

Iterasi Ke-6



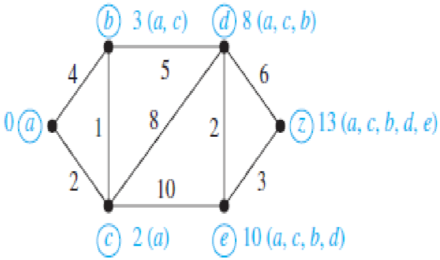
1. $u := z$ (karena $z \notin S_5$ dan $L_5(z)$ minimal).

Iterasi Ke-6



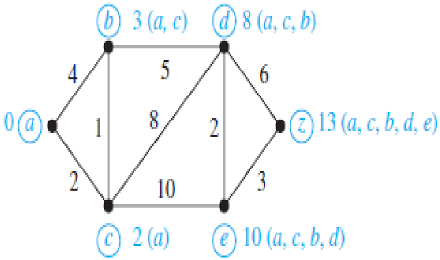
1. $u := z$ (karena $z \notin S_5$ dan $L_5(z)$ minimal).
2. $S_6 = S_5 \cup \{z\} = \{a, c, b, d, e\} \cup \{z\} = \{a, c, b, d, e, z\}$.

Iterasi Ke-6



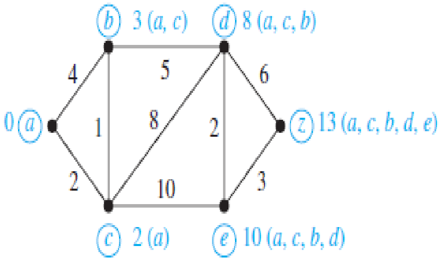
1. $u := z$ (karena $z \notin S_5$ dan $L_5(z)$ minimal).
2. $S_6 = S_5 \cup \{z\} = \{a, c, b, d, e\} \cup \{z\} = \{a, c, b, d, e, z\}$.
3. Tidak terdapat $v \in V$ dengan $v \notin S$. Akibatnya pengulangan/ *loop* `for all` $v \notin S$ pada baris 5, 6, dan 7 proses iterasi Algoritma Dijkstra tidak dijalankan.

Iterasi Ke-6



1. $u := z$ (karena $z \notin S_5$ dan $L_5(z)$ minimal).
2. $S_6 = S_5 \cup \{z\} = \{a, c, b, d, e\} \cup \{z\} = \{a, c, b, d, e, z\}$.
3. Tidak terdapat $v \in V$ dengan $v \notin S$. Akibatnya pengulangan/ *loop* `for all` $v \notin S$ pada baris 5, 6, dan 7 proses iterasi Algoritma Dijkstra tidak dijalankan.
4. Karena $z \in S_6$ maka pengulangan/ *loop* `while` dari Algoritma Dijkstra selesai.

Iterasi Ke-6



1. $u := z$ (karena $z \notin S_5$ dan $L_5(z)$ minimal).
2. $S_6 = S_5 \cup \{z\} = \{a, c, b, d, e\} \cup \{z\} = \{a, c, b, d, e, z\}$.
3. Tidak terdapat $v \in V$ dengan $v \notin S$. Akibatnya pengulangan/ *loop* `for all` $v \notin S$ pada baris 5, 6, dan 7 proses iterasi Algoritma Dijkstra tidak dijalankan.
4. Karena $z \in S_6$ maka pengulangan/ *loop* `while` dari Algoritma Dijkstra selesai.

Algoritma Dijkstra memberikan lintasan $\langle a, c, b, d, e, z \rangle$ sebagai lintasan terpendek dari a ke z dengan panjang (bobot) lintasan 13.

Bahasan

Masalah Pedagang Keliling

Masalah Pedagang Keliling (*Travelling Salesperson Problem*, TSP)

Misalkan $G = (V, E)$ adalah graf berbobot sederhana yang memodelkan beberapa kota dan jaraknya. Seorang pedangang keliling (*travelling salesperson*) **harus mengunjungi semua kota tersebut tepat satu kali, kecuali kota pertama dan terakhir** (kota pertama dan terakhir sama). Bagaimana cara menentukan perjalanan yang paling efisien untuk pedagang tersebut?

Masalah Pedagang Keliling

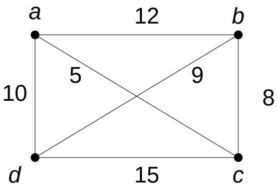
Masalah Pedagang Keliling (*Travelling Salesperson Problem*, TSP)

Misalkan $G = (V, E)$ adalah graf berbobot sederhana yang memodelkan beberapa kota dan jaraknya. Seorang pedangang keliling (*travelling salesperson*) **harus mengunjungi semua kota tersebut tepat satu kali, kecuali kota pertama dan terakhir** (kota pertama dan terakhir sama). Bagaimana cara menentukan perjalanan yang paling efisien untuk pedagang tersebut?

Masalah pedagang keliling adalah masalah penentuan sirkuit Hamilton pada sebuah graf berbobot sederhana dengan panjang lintasan (bobot) minimum.

Contoh Masalah Pedagang Keliling

Kita akan menentukan solusi dari TSP bila simpul awal dan simpul akhir yang dikunjungi adalah simpul a dari graf berikut.

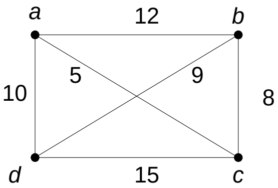


Karena graf di atas adalah graf lengkap, maka terdapat $\frac{(4-1)!}{2} = 3$ sirkuit Hamilton berbeda. Tinjau bahwa:

1. Sirkuit 1:

Contoh Masalah Pedagang Keliling

Kita akan menentukan solusi dari TSP bila simpul awal dan simpul akhir yang dikunjungi adalah simpul a dari graf berikut.

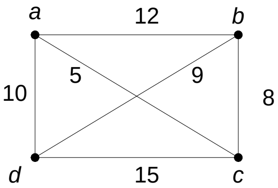


Karena graf di atas adalah graf lengkap, maka terdapat $\frac{(4-1)!}{2} = 3$ sirkuit Hamilton berbeda. Tinjau bahwa:

1. Sirkuit 1: $\langle a, b, c, d, a \rangle$ atau $\langle a, d, c, b, a \rangle$ memiliki panjang:

Contoh Masalah Pedagang Keliling

Kita akan menentukan solusi dari TSP bila simpul awal dan simpul akhir yang dikunjungi adalah simpul a dari graf berikut.

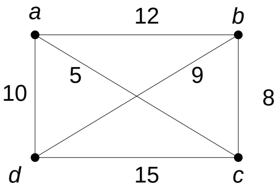


Karena graf di atas adalah graf lengkap, maka terdapat $\frac{(4-1)!}{2} = 3$ sirkuit Hamilton berbeda. Tinjau bahwa:

1. Sirkuit 1: $\langle a, b, c, d, a \rangle$ atau $\langle a, d, c, b, a \rangle$ memiliki panjang: $12 + 8 + 15 + 10 = 45$.
2. Sirkuit 2:

Contoh Masalah Pedagang Keliling

Kita akan menentukan solusi dari TSP bila simpul awal dan simpul akhir yang dikunjungi adalah simpul a dari graf berikut.

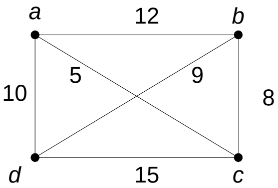


Karena graf di atas adalah graf lengkap, maka terdapat $\frac{(4-1)!}{2} = 3$ sirkuit Hamilton berbeda. Tinjau bahwa:

1. Sirkuit 1: $\langle a, b, c, d, a \rangle$ atau $\langle a, d, c, b, a \rangle$ memiliki panjang: $12 + 8 + 15 + 10 = 45$.
2. Sirkuit 2: $\langle a, c, d, b, a \rangle$ atau $\langle a, b, d, c, a \rangle$ memiliki panjang:

Contoh Masalah Pedagang Keliling

Kita akan menentukan solusi dari TSP bila simpul awal dan simpul akhir yang dikunjungi adalah simpul a dari graf berikut.

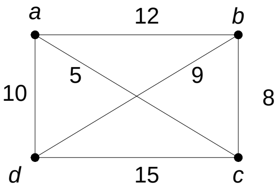


Karena graf di atas adalah graf lengkap, maka terdapat $\frac{(4-1)!}{2} = 3$ sirkuit Hamilton berbeda. Tinjau bahwa:

1. Sirkuit 1: $\langle a, b, c, d, a \rangle$ atau $\langle a, d, c, b, a \rangle$ memiliki panjang: $12 + 8 + 15 + 10 = 45$.
2. Sirkuit 2: $\langle a, c, d, b, a \rangle$ atau $\langle a, b, d, c, a \rangle$ memiliki panjang: $5 + 15 + 9 + 12 = 41$.
3. Sirkuit 3:

Contoh Masalah Pedagang Keliling

Kita akan menentukan solusi dari TSP bila simpul awal dan simpul akhir yang dikunjungi adalah simpul a dari graf berikut.

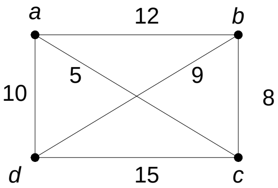


Karena graf di atas adalah graf lengkap, maka terdapat $\frac{(4-1)!}{2} = 3$ sirkuit Hamilton berbeda. Tinjau bahwa:

1. Sirkuit 1: $\langle a, b, c, d, a \rangle$ atau $\langle a, d, c, b, a \rangle$ memiliki panjang: $12 + 8 + 15 + 10 = 45$.
2. Sirkuit 2: $\langle a, c, d, b, a \rangle$ atau $\langle a, b, d, c, a \rangle$ memiliki panjang: $5 + 15 + 9 + 12 = 41$.
3. Sirkuit 3: $\langle a, c, b, d, a \rangle$ atau $\langle a, d, b, c, a \rangle$ memiliki panjang:

Contoh Masalah Pedagang Keliling

Kita akan menentukan solusi dari TSP bila simpul awal dan simpul akhir yang dikunjungi adalah simpul a dari graf berikut.



Karena graf di atas adalah graf lengkap, maka terdapat $\frac{(4-1)!}{2} = 3$ sirkuit Hamilton berbeda. Tinjau bahwa:

1. Sirkuit 1: $\langle a, b, c, d, a \rangle$ atau $\langle a, d, c, b, a \rangle$ memiliki panjang: $12 + 8 + 15 + 10 = 45$.
2. Sirkuit 2: $\langle a, c, d, b, a \rangle$ atau $\langle a, b, d, c, a \rangle$ memiliki panjang: $5 + 15 + 9 + 12 = 41$.
3. Sirkuit 3: $\langle a, c, b, d, a \rangle$ atau $\langle a, d, b, c, a \rangle$ memiliki panjang: $10 + 5 + 9 + 8 = 32$.

Jadi sirkuit yang dipilih adalah $\langle a, c, b, d, a \rangle$ dengan panjang 32.

Bahasan

Masalah Tukang Pos Tiongkok

Masalah Tukang Pos Tiongkok (*Chinese Postman Problem*)

Di Tiongkok, seorang tukang pos harus mengantar semua surat-surat yang ditugaskan kepadanya. Untuk menghemat waktu dan biaya, ia merencanakan rute perjalanan yang hanya melewati setiap jalan tepat sekali. Jika tempat awal dan tempat akhir keberangkatan tukang pos itu sama, bagaimana cara menentukan rute perjalanan yang paling efisien?

Masalah Tukang Pos Tiongkok

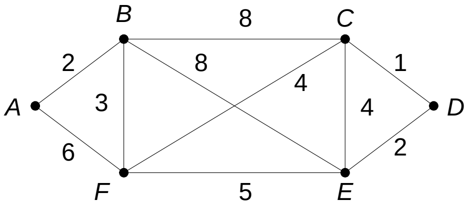
Masalah Tukang Pos Tiongkok (*Chinese Postman Problem*)

Di Tiongkok, seorang tukang pos harus mengantar semua surat-surat yang ditugaskan kepadanya. Untuk menghemat waktu dan biaya, ia merencanakan rute perjalanan yang hanya melewati setiap jalan tepat sekali. Jika tempat awal dan tempat akhir keberangkatan tukang pos itu sama, bagaimana cara menentukan rute perjalanan yang paling efisien?

Masalah ini kali pertama disampaikan oleh Guan Meigu pada tahun 1962. Pada dasarnya Masalah Tukang Pos Tiongkok (CPP) merupakan masalah pencarian sirkuit Euler pada sebuah graf.

Contoh Masalah Tukang Pos Tiongkok

Pada graf berikut, kita akan menentukan simpul awal dan simpul akhir untuk suatu lintasan Euler.



Salah satunya adalah: $\langle A, B, C, D, E, F, C, E, B, F, A \rangle$.