BAB II

LANDASAN TEORI

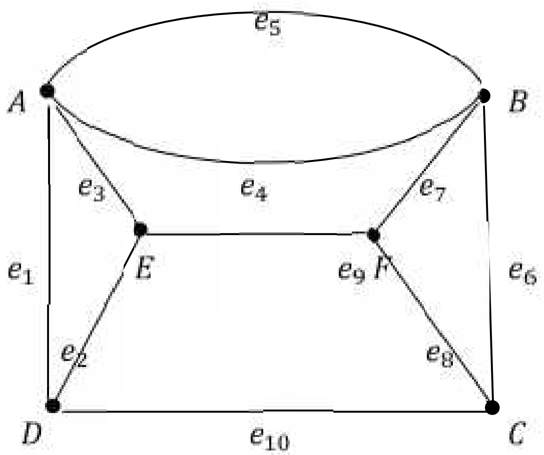
2.1 Terminologi graf

Tereminologi termasuk istilah yang berkaitan dengan graf. Di bawah ini akan dijelaskan beberapa definisi yang sering dipakai terminologi.

2.1.1 Graf

Definisi 2.1 (Munir, Rinaldi :2001) Suatu graf G terdiri atas himpunan yang tidak kosong dari elemen–elemen yang disebut titik, dan suatu daftar pasangan titik yang tidak terurut disebut sisi. Himpunan titik dari suatu graf G dinotasikan dengan V, dan daftar himpunan sisi dari graf tersebut dinotasikan dengan E. Untuk selanjutnya suatu graf G dapat dinotasikan dengan $G = (V, E)$.

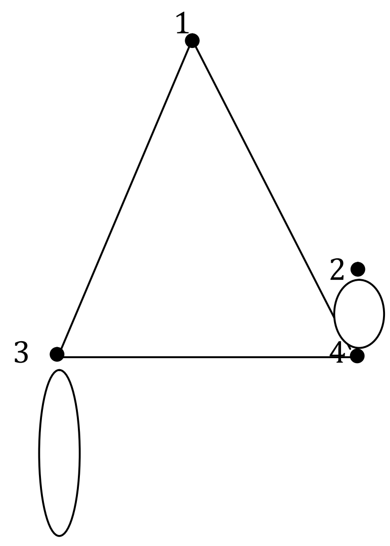
Perhatikan Gambar 2.1 di bawah ini :



Gambar 2.1 Graf G

Gambar 2.1, menunjukkan graf G dengan $V = \{A, B, C, D, E, F\}$ dan $E = \{e_1, e_2, e_3, \ldots e_{10}\}$.

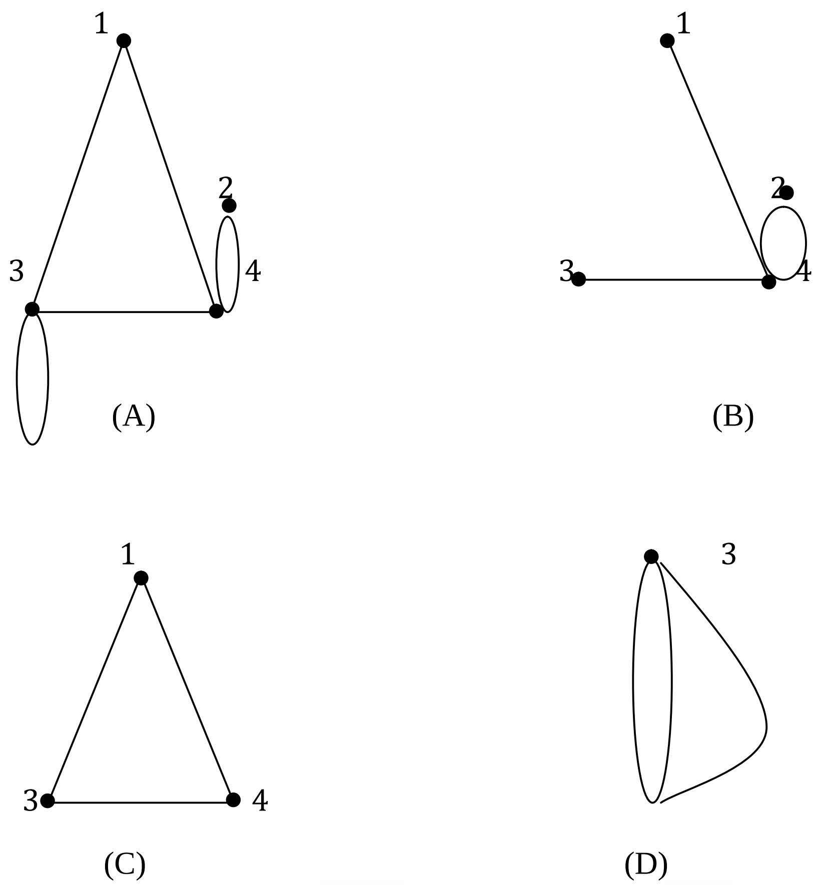
Definisi 2.2 (Munir, Rinaldi :2001) Sisi atau lebih yang menghubungkan pasangan titik yang sama disebut sisi ganda, dan sebuah sisi yang mengubungkan sebuah titik ke dirinya sendiri disebut loop.



Gambar 2.2 Sisi Ganda dan Loop

Definisi 2.3 (Munir, Rinaldi :2001) Misal G suatu graf dengan himpunan titik V dan himpunan sisi E. Suatu subgraf G' adalah suatu himpunan pasangan berurutan (V', E') dimana V' merupakan himpunan bagian dari V dan E' adalah himpunan bagian dari E. Dengan kata lain, subgraf dari G adalah suatu graf yang semua titiknya anggota V dan semua sisinya anggota E.

Jika G suatu graf terhubung seperti pada Gambar 2.2, dengan $V = \{1, 2, 3, 4\}$ dan $E = \{(1,3), (1,4), (2,4), (3,3), (3,4), (4,2)\}$. Berikut contoh dari subgraf G' diunjukkan pada Gambar 2.3 di bawah ini.



Gambar 2.3 (B) dan (C) *Subgraf* dari (A) dan (D) Bukan *Subgraf* dari (A)

Berdasarkan Gambar 2.3 (B) merupakan subgraf G' dari graf G, dengan himpunan titik $V' = \{1, 2, 3, 4\}$ yang merupakan himpunan bagian dari V dan himpunan sisi $E' = \{(1,3), (1,4), (2,4), (3,3), (3,4), (4,2)\}$ yang merupakan himpunan bagian dari E. Gambar 2.3 (C) adalah graf sederhana yang merupakan subgraf G' dari graf G dengan himpunan titik $V' = \{1, 3, 4\}$ dan himpunan sisi $E' = \{(1,3), (1,4), (3,4)\}$ yang masing–masing merupakan himpunan bagian dari V dan E. Gambar 2.3 (D) bukan subgraf dari (A).

2.1.2 Beberapa Istilah yang Berkaitan dengan Graf

1. *Loop*

Loop adalah suatu sisi yang menghubungkan suatu titik dengan dirinya sendiri.

2. Lintasan

Lintasan yang panjang n dari titik awal v_0 ke titik tujuan v_n di dalam graf G ialah barisan berselang–seling titik-titik dan sisi yang berbentuk $v_0, e_1, v_1, e_2, v_2, \dots, v_{n-1}, e_n, v_n$ sedemikian hingga $e_1 = (v_0, v_1), e_2 = (v_1, v_2), \dots, e_n = (v_{n-1}, v_n)$ adalah sisi dari graf G.

3. Lintasan tertutup atau sirkuit

Lintasan yang berawal dan berakhir pada titik yang sama disebut lintasan tertutup.

Subgraf G' dari graf G dengan himpunan titik $V' = \{1\}$ dan himpunan sisi $E' = \{(3)\}$ yang masing–masing merupakan himpunan bagian dari V dan E.

Definisi 2.4 (Munir, Rinaldi :2001) Sebuah graf G berisikan dua himpunan yaitu himpunan hingga tak kosong $V(G)$ yang elemennya disebut titik atau *vertek* dan himpunan (mungkin kosong) $E(G)$ yang elemen-elemennya disebut sisi, sedemikian hingga setiap sisi e di $E(G)$ adalah sebuah pasangan tak berurutan dari titik-titik di $V(G)$.

2.2 Jenis-jenis Graf

Definisi 2.7 (Munir, Rinaldi : 2001) Ada tidaknya gelang atau sisi ganda pada suatu graf, maka graf di golongkan menjadi dua jenis:

2.2.1 Graf Sederhana

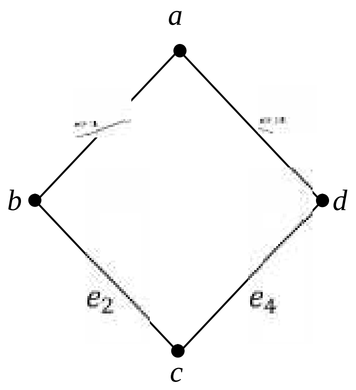
Graf yang tidak mengandung gelang maupun sisi ganda dinamakan graf sederhana. Pada graf sederhana, sisi adalah pasangan tak-terurut (*unordered pairs*). Jadi menuliskan sisi (u, v) sama saja dengan (v, u).

Contoh 2.1

Gambarlah sebuah graf sederhana dengan titik $V(G) = \{A, B, C, D\}$ dan garis $E(G) = \{e_1, e_2, e_3, e_4\}$:

Penyelesaian:

Berdasarkan titik graf sederhana dapat kita peroleh gambarnya sebagai berikut:



Gambar 2.4 Graf Sederhana

2.2.2 Graf tak Sederhana (*Unsimpel - Graph*)

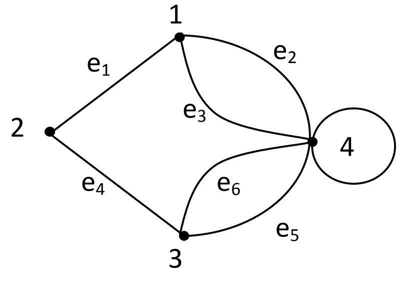
Graf yang mengandung sisi ganda atau gelang disebut graf tak sederhana. Ada dua macam graf tak sederhana, yaitu graf ganda (*multigraph*) dan graf semu (*pseudograph*). Graf ganda adalah graf yang mengandung sisi ganda. Sisi ganda yang menghubungkan sepasang titik bisa lebih dari dua. Graf semu adalah graf yang mengandung gelang (*loop*).

Contoh 2.2

Gambarlah sebuah graf dengan titik $V(G) = \{1,2,3,4\}$ dan sisi $E(G) = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7\}$ dengan titik ujung $e_1 = (2,1), e_2 = (1,4), e_3 = (1,4)$, $e_4 = (2,3), e_5 = (3,4)$, $e_6 = (3,4), e_7 = (4)$.

Penyelesaian:

Berdasarkan graf tak sederhana yang sudah diketahui titiknya, maka kita memperoleh gambar sebagai berikut:



Gambar 2.5 Graf tak Sederhana

2.2.3. Bertetangga (*adjacent*) dan Bersisian (*inscident*)

Bertetangga adalah suatu graf sederhana dua buah titik bertetangga apabila keduanya dihubungkan oleh satu sisi yang bersisian.

Definisi 2.5 (Lipscuts: 2002) Didefinisikan bertetangga dan bersisian dalam sebuah graf. Misalkan $e = \{(u, v)\}$ adalah sebuah sisi dalam G, yaitu u dan v adalah titik-titik ujung dari e titik u dikatakan bertetangga terhadap titik v dan sisi e dikatakan bersisian atau terhubung dengan pada u dan v.

2.2.3 Derajat (*Degree*)

Derajat suatu titik adalah jumlah sisi yang bersisian dengan titik tersebut, sehingga didapatkan definisi dibawah ini.

Definisi 2.6 (Siang, 2006) Misalkan v adalah titik dalam suatu graf G. Derajat titik v (titik $d(v)$) adalah jumlah garis (sisi) yang berhubungan dengan simpul v dan sisi gelang (*loop*) dihitung dua kali. Derajat total G adalah jumlah derajat semua titik dengan G.

Derajat suatu titik pada graf tak berarah adalah jumlah sisi yang bersisian dengan titik tersebut. Siang (2006) menyebutkan bahwa derajat pada graf berarah dibagi menjadi dua, yaitu $d_{in}(v)$ dan $d_{out}(v)$ yang dalam hal ini:

a) $d_{in}(v)$ adalah derajat-masuk (*in-degree*) yaitu jumlah busur yang masuk ke titik v .
b) $d_{out}(v)$ adalah derajat-keluar (*out-degree*) yaitu jumlah busur yang keluar dari titik v, dengan:

$$d(v) = d_{in(v)} + d_{out(v)}$$

Contoh 2.3 :

Berdasarkan Gambar 2.5 di atas tentukanlah derajat tiap titiknya sebagai berikut:

Penyelesaian:

Berdasarkan Gambar 2.5 pembahasan graf tak sederhana diatas kita bisa menentukan berapa titik yang diperoleh. Maka kita bisa menggunakan penyelesaian dibawah ini:

$d(v_1) = 3, d(v_2) = 2, d(v_3) = 3, dand(v_4) = 5.$

Sehingga,

Derajat total = $\sum_{i=1}^{5} d(v_i) = 3 + 2 + 3 + 5 = 13$

2.2.5 Graf Berbobot (*Weighted Graph*)

Graf yang setiap sisinya diberi sebuah harga (bobot), bobot pada setiap sisi dapat berbeda-beda bergantung pada masalah yang dimodelkan dengan graf. Bobot dapat dinyatakan jarak antara dua buah kota, biaya perjalanan antara dua buah kota, waktu tempuh pesan (*massage*) dari buah simpul komunikasi ke jaringan komputer, ongkos produksi.

Definisi 2.7 (Munir, Rinaldi : 2009) Bila sisi e dalam graf G dikaitkan dengan sebuah bilangan real $W(e)$ disebut bobot (*weight*) dari e. bobot dari sebuah graf G, dinotasikan dengan $w(G)$. Sebuah graf yang setiap sisinya dikaitkan dengan bilangan real disebut graf bobot.

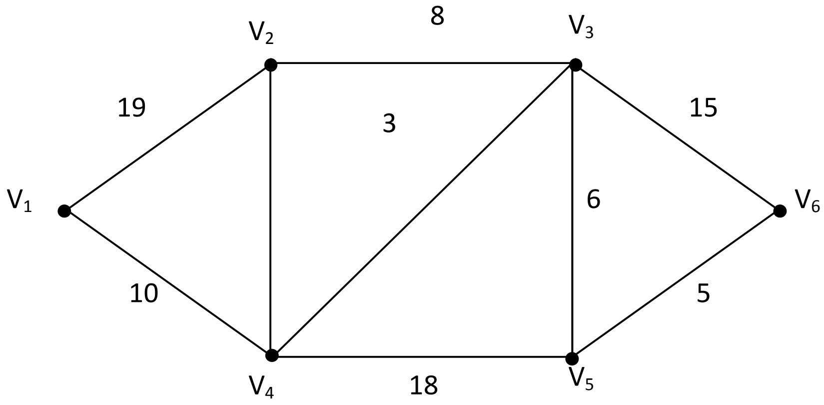
Contoh 2.6

Berikut ini adalah graf berbobot dengan ketentuan sebagai berikut:

$$(v_1, v_2) = 19, (v_2, v_3) = 8, (v_1, v_4) = 10, (v_2, v_4) = 4, (v_4, v_5) = 1{,}8, (v_4, v_3) = 3, (v_3, v_5) = 6, (v_3, v_6) = 15, (v_5, v_6) = 5.$$

Penyelesaian:

Berdasarkan titik yang sudah di ketahui maka kita dapat menggambarkan graf berbobot sebagai berikut:



Gambar 2.6 Graf Berbobot

2.3 Representasi Graf dalam Matriks

(Munir, Rinaldi : 2006) Menyebutkan matriks dapat digunakan untuk menyatakan suatu graf. Berikut ini terdapat beberapa representasi graf dalam matriks**:**

1. Matriks ketetanggaan (*adjacency matriks*)

Matriks ketetanggaan adalah representasi graf paling umum. Misalkan $G = (V, E)$ adalah graf dengan n titik, $n \geq 1$. Bila matriks $A = [a_{ij}]$ maka $[a_{ij}] = 1$ jika titik i dan j bertengga. Begitu juga dengan graf berbobot, a_{ij} menyatakan bobot tiap sisi yang menghubungkan titik i dengan titik j. tanda "$_{\infty}$" menyatakan bahwa tidak ada sisi dari titik i ke titik j atau dari titik i ke titik i itu sendiri, sehingga diberi nilai tak berhingga.

Notasi :

$$a_{ij} = \begin{cases} 1, \text{ jika ada sisi } (v_i, v_j) \\ 0, \text{ jika tidak bertetangg a } (v_i, v_j) \end{cases}$$

2. Matriks bersisian (*incidency matrix*)

Adapun matriks ketetanggaan menyatakan ketetanggaan titik-titik di dalam graf, maka matriks bersisian menyatakan bersisian titik dengan sisi. Misalkan $G = (V, E)$ adalah graf dengan n titik dan m buah sisi. Matriks bersisian G adalah matriks *Dwimatra* yang berukuran $m \times n$. Baris menunjukkan label titik, sedangkan kolom menunjukkan label sisinya. Apabila matriks tersebut dinamakan $A = [m_{ij}]$, maka m_{ij} = 1 jika titik i bersisian dengan sisi j, sebaliknya m_{ij} = 0 jika títik i tidak bersísían dengan titik j.

Notasi:

$$m_{ij} = \begin{cases} 1, \text{jika ada sisi } (v_i, v_j) \\ 0, \text{jika tidak ada sisi } (v_i, v_j) \end{cases}$$

2.4 Lintasan Terpendek

Bobot yang berhubungan dengan suatu garis pada graf juga dapat diaplikasikan pada graf berarah. Prinsip dan arti bobot pada graf berarah sama dengan pada bobot graf tak berarah, yaitu menyatakan seberapa kuat hubungan antara dua titik yang arahnya ditunjukkan dengan arah garis. Salah satu aplikasi graf berlabel yang sering dipakai adalah mencari lintasan terpendek di antara dua titik.

Sebagai contoh, terdapat banyak jalan yang menghubungkan kota Yogyakarta ke Jakarta. Pertanyaan yang sering muncul adalah jalur mana yang paling pendek? Jika kota-kota di Jawa Tengah dan Jawa Barat dinyatakan sebagai titik-titik jalan yang menghubungkan kota-kota tersebut dinyatakan sebagai garis bobot garis, maka masalah mencari jalur yang paling dekat antara dua kota adalah bersangkutan. Apabila masalahnya adalah mencari jalur tercepat (jalur terpendek belum tentu cepat), lintasan terpendek tetap dapat digunakan dengan cara mengganti bobot garis sehingga dinyatakan waktu perjalanan antar kota-kota yang dinyatakan sebagai titik-titik di ujung garis. Cara yang sama dapat dipakai apabila dikehendaki untuk mencari jalur termudah.

2.4.1 Jalan (*Walk*)

Definisi 2.8 (Munir, Rinaldi : 2006) Suatu jalan (*walk*) dalam graf G adalah barisan titik–titik dan sisi–sisi yang dimulai dan diakhiri oleh suatu titik. Panjang suatu jalan dihitung berdasarkan jumlah sisi dalam jalan tersebut.

Jalan juga dapat diartikan sebagai suatu perjalanan (dalam sebuah graf) dari titik satu ke titik lain yang terhubung dengan suatu sisi.

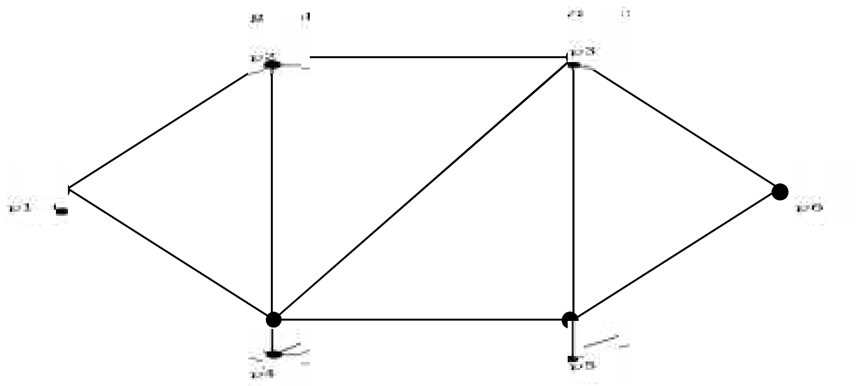
2.4.2 Lintasan (*path*)

Berdasarkan lintasan untuk menetukan panjangnya n dari titik awal, maka definisi didapatkan di bawah ini.

Definisi 2.9 (Munir, Rinaldi : 2006) Lintasan yang panjangnya n dari titik awal v_0 ketitik tujuan di dalam graf G ialah barisan berselang-seling titik-titik dan sisi-sisi yang berbentuk $v_0, e_1, v_1, e_2, \dots v_{n-1}, e_n, v_n$ sedemikian sehingga $e_1 = (v_0, v), e_2(v_1, v_2), \dots\ e_n(v_{n-1}, v_n)$ adalah sisi-sisi dari graf G.

Contoh2.4

Graf berikut ini merupakan gambaran suatu daerah, yaitu:



Gambar 2.7 Lintasan pada Graf

Berdasarkan Gambar 2.7 diatas dapat dimisalkan bahwa titik-titik dalam graf tersebut mewakili desa, sisi mewakili jalan antara dua desa. Misalkan seseorang berada di desa v_1 dan ingin berpergian dengan mobil ke desa v_6. Ada beberapa lintasan yang bisa ditempuh. Tunjukanlah lintasan-lintasan tersebut.

Penyelesaian:

Berdasarkan penyelesaian diatas, maka dapat kita simpulkan lintasan yang terpendek itu ada pada lintasan v_1, v_3, dan v_6.

a. $p_1 = (v_1, v_2, v_3, v_6)$
b. $p_2 = (v_1, v_4, v_2, v_3, v_6)$
c. $p_3 = (v_1, v_4, v_5, v)$
d. $p_4 = (v_1, v_4, v_2, v_3, v_5, v_6)$

e. $p_5 = (v_1, v_2, v_4, v_3, v_5, v_6)$

f. $p_6 = (v_1, v_4, v_3, v_6)$

g. $p_7 = (v_1, v_4, v_5, v_3, v_6)$

2.5 Algoritma Floyd Warshall

Algoritma yang ditemukan oleh Floyd Warshall untuk mencari lintasan terpendek merupakan algoritma yang sederhana dan mudah implementasinya. Masukan Algoritma Floyd Warshall adalah matriks hubung graf berarah berlabel, dan keluarannya adalah lintasan terpendek dari semua titik ke semua titik.

Dalam usaha untuk mencari lintasan terpendek, Algoritma Warshall memulai iterasi dari titik awalnya kemudian memperpanjang lintasan dengan mengevaluasi titik demi titik hingga mencapai titik tujuan dengan jumlah bobot yang seminimum mungkin.

Misalkan W_0 adalah matriks hubung graf berarah berlabel mula-mula. W^* adalah matriks hubung minimal dengan W_{ij}^* = lintasan terpendek dari v_i titik ke v_j.

Adapun langkah-langkah Algoritma Floyd Warshall sebagai berikut:

a. $W = W_0$.

b. Untuk $k = 1$, hingga n lakukan: Untuk $i = 1$, hingga n lakukan; Jika $W[i,j] > W[i,k] = W[k,j]$ maka tukar $W[i,j]$ dengan $W[i,k] + W[k,j]$.

c. $W^* = W$

Algoritma Floyd-Warshall memiliki input graf berarah dan berbobot (V, E), yang berupa daftar titik (*node/*titik V) dan daftar sisi (sisi E). Jumlah bobot sisi-sisi pada sebuah jalur adalah bobot jalur tersebut sisi pada E diperbolehkan memiliki bobot negatif, akan tetapi tidak diperbolehkan bagi graf ini untuk memiliki siklus dengan bobot negatif. Algoritma ini membandingkan semua kemungkinan lintasan pada graf untuk setiap sisi dari semua titik, hal tersebut bisa terjadi karena adanya perkiraan pengambilan keputusan (pemiliihan jalur pendek) pada setiap tahap antara dua titik hingga perkiraan tersebut diketahui sebagai nilai optimal. Implementasi algoritma ini berupa graf yang dipresentasikan sebagai matriks keterhubungan, yang isinya adalah bobot/jarak sisi yang menghubungkan tiap pasangan titik, dilambangkan dengan indeks baris dan kolom. Ketiadaan sisi yang menghubungkan sebuah pasangan dilambangkan dengan tak hingga.

Berdasarkan iterasinya lintasan terpendek Algoritma Floyd Warshall membentuk n matriks, sesuai dengan iterasi-k. Ini menyebabkan waktu prosesnya lambat, terutama untuk n yang sangat besar. Meskipun waktu prosesnya bukanlah yang tercepat, Algoritma Floyd Warshall sering dipergunakan untuk menghitung lintasan terpendek karena kesederhanaan algoritmanya. Program implementasi Algoritma Floyd Warshall sangat mudah dibuat.