MODIFIKASI ALGORITMA *DJIKSTRA* DENGAN METODE *FORD* UNTUK MENCARI *SHORTEST PATH* PADA GRAF YANG MEMILIKI BOBOT NEGATIF

Alfan Farizki Wicaksono - 13506067

Teknik Informatika, Institut Teknologi Bandung
Jalan Ganesha Nomor 10, Kota Bandung
e-mail: alfan_fw@students.itb.ac.id

ABSTRAK

Seperti kita ketahui bahwa masalah mengenai *shortest path* merupakan masalah yang sangat penting dalam kehidupan kita. Misalnya saja pada penentuan jalan antara kota A dan kota B, kita dapat memodelkan kedua kota tersebut dengan sebuah graf dimana simpul menyatakan kota dan sisi menyatakan hubungan terdapat jalan antara kedua kota terbsebut. sisi juga mempunyai tanda berupa bobot yang artinya jarak antar kedua buah kota. Model graf tersebut dapat kita gunakan untuk menyelesaikan masalah pencarian jalan terpendek (*shortest path*). Untuk mencari jalan terpendek antara dua simpul kita membutuhkan sebuah algoritma yang bisa bekerja pada suatu model graf. Algoritma Djikstra merupakan algoritma yang paling banyak digunakan untuk penyelesaian masalah ini. Algoritma ini termasuk algoritma yang cukup mangkus dalam penyelesaian masalah *shortest path*. Tetapi agoritma ini secara murni tidak dapat menangani kasus dimana ada bobot sisi yang negatif. Hal ini sangat disayangkan sekali, mengingat pemodelan graf dengan bobot negatif banyak sekali digunakan dalam kehidupan kita. Oleh karena itu, kita harus mencari solusi untuk menyelesaikan masalah ini. Salah satu jalan yang dapat ditempuh adalah dengan memodifikasi algoritma Djikstra ini supaya dapat menangani kasus graf dengan bobot negatif. Salah satu metode yang digunakan dalam memodifikasi algoritma djikstra adalah metode Ford. Metode ini memodifikasi algoritma Djikstra sehingga dapat menangani kasus dengan graf berbobot negatif. Modifikasi yang dilakukan meliputi pengulangan dalam menandai simpul yang sudah pernah dikunjungi. Akibat dari modifikasi ini tingkat kompleksitas algoritma bertambah secara polinomial. Walaupun begitu metode ini cukup hebat dalam penanganan kasus dengan graf yang mempunyai sisi negatif.

Kata kunci: Djikstra, Metode Ford, Ford, *Shortest Path*, Bobot Negatif.

1. PENDAHULUAN

Kita mulai pembahasan kita dengan membahas mengenai algoritma djikstra itu sendiri. Sampai saat ini, sudah banyak algoritma mencari lintasan terpendek yang pernah ditulis orang. Algoritma lintasan terpendek *(shortest path)* yang paling terkenal adalah algoritma djikstra (sesuai dengan nama penemunya, Edsger Wybe Djikstra). Dari naskah aslinya, algoritma Djikstra diterapkan untuk mencari lintasan terpendek pada graf berarah. Namun, algoritma ini tetap benar untuk graf yang tak-berarah.

Algoritma Djikstra menacari lintasan terpendek dalam sejumlah langkah. Algoritma ini menerapkan strategi greedy dalam pengerjaannya. Penerapan strategi greedy dalam algoritma Djikstra terlihat pada deskripsi berikiut:

Pada setiap langkah, ambil sisi yang berbobot minimum yang menghubungkan sebuah simpul yang telah dipilih dengan sebuah simpul lain yang belum terpilih. Lintasan dari simpul asal ke simpul yang baru haruslah merupakan lintasan yang terpendek diantara semua lintasannya ke simpul-simpul yang belum terpilih [2].

Misalkan kita tentukan S adalah simpul awal dan T adalah simpul akhir, kita akan mencari lintasan terpendek *(shortest path)* antara simpul S dan simpul T.Dari deskripsi diatas langkah – langkah yang digunakan oleh algoritma Djikstra adalah sebagai berikut:

Langkah 1. Inisiasi awal, semua simpul dan sisi belum ditandai. Isi semua nilai d(x) untuk x adalah setiap simpul pada graf dengan aturan sebagai berikut. Untuk simpul awal S yang merupakan awal titik perjalanan *shortest path* kita beri nilai d(S) = 0, untuk simpul lain $d(X) = \infty$, $X \neq S$. Kemudian kita ambil Y = S.

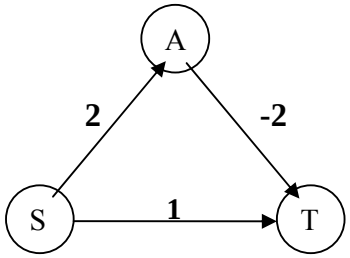
Langkah 2. Untuk setiap simpul yang belum ditandai, definisikan ulang d(X) dengan rumus berikut

$$d(X) = \text{Min}\{d(X), d(Y) + a(Y, X)\} \quad (1)$$

dimana Y adalah simpul lain yang dapat dikunjungi oleh X serta belum ditandai sebelumnya, maksud a(Y, X) adalah bobot sisi yang dibentuk oleh simpul Y ke simpul X. Jika $d(X) = \infty$ untuk semua simpul yang belum ditandai, berhentilah karena tidak ada jalan lagi dari simpul S ke simpul yang lainnya. Jika ada, tandai simpul yang mempunyai nilai d(X) terkecil, juga tandai simpul yang terbentuk antara simpul S ke simpul X tersebut. Kemudian kita ambil Y = X.

Langkah 3. Jika simpul akhir T sudah ditandai berhentilah, karena lintasan terpendek *(shortest path)* dari S ke T sudah ditemukan. Jika simpul T belum ditandai, ulangi langkah 2.

Untuk kasus dimana terdapat graf yang mempunyai bobot negatif, misalkan graf tersebut merupakan graf pada gambar 1.



Gambar 1. Graf dengan salah satu bobot sisinya negatif

Misalkan kita tentukan simpul awal adalah simpul S dan simpul akhir adalah simpul T. Secara mudah kita dapat melihat bahwa lintasan terpendek yang dapat ditemukan antara kedua simpul ini adalah (S, A), (A, T) dimana panjang jalannya adalah 2 + (-2) = 0. Jika kita terapkan algoritma Djikstra pada kasus ini akan terdapat kekeliruan, algoritma Djikstra akan mengambil sisi (S, T) yang berjarak 1 sebagai lintasan terpendek antara simpul S dan T. Dari hal ini sudah dipastikan bahwa tidak ada jaminan algoritma Djikstra akan menemukan hasil yang benar untuk masalah *shortest path* jika ada bobot sisi yang negatif [1].

2. METODE

Untuk menangani kasus seperti pada gambar 1. Kita memerlukan sebuah metode yang dapat menyelesaikannya, oleh karena itu lahirlah sebuah metode Ford (1956) dimana metode ini tidak lain merupakan sebuah hasil modifikasi dari algoritma Djikstra. Metode melakukan tiga buah perubahan kecil pada langkah algoritma Djikstra yang sudah dituliskan pada bagian pendahuluan [1].

2.1 Algoritma

Adapun tiga perubahan yang dilakukan pada langkah algoritma Djikstra adalah sebagai berikut :

1. Pada langkah 2, persamaan 1 diberlakukan pada semua simpul dan sisi baik itu sudah ditandai maupun belum belum ditandai, kalau algoritma Djikstra biasa, persamaan 1 hanya berlaku pada simpul atau sisi yang belum ditandai. Hal ini dilakukan karena pada suatu saat bisa jadi nilai dari sisi yang sudah ditandai berubah nilainya menjadi lebih kecil. Nilai yang lebih kecil inilah yang akan diambil untuk proses selanjutnya.
2. Jika sisi yang sudah ditandai berubah nilainya menjadi lebih kecil, hilangkan tanda sisi tersebut.
3. Hentikan algoritma jika semua sisi dan simpul sudah ditandai dan pada langkah 2 gagal untuk melahirkan kembali sisi yang sudah ditandai dengan nilai yang lebih kecil lagi.

Dengan demikian, algoritma Ford dapat dinyatakan dalam langkah berikut ini :

Langkah 1. Inisiasi awal, semua simpul dan sisi belum ditandai. Isi semua nilai d(x) untuk x adalah setiap simpul pada graf dengan aturan sebagai berikut. Untuk simpul awal S yang merupakan awal titik perjalanan *shortest path* kita beri nilai d(S) = 0, untuk simpul lain $d(X) = \infty$, $X \neq S$. Kemudian kita ambil Y = S.

Langkah 2. Untuk setiap simpul kecuali simpul pada Y, definisikan ulang d(X) dengan rumus berikut pada persamaan 1. dimana Y adalah simpul lain yang dapat dikunjungi oleh X serta belum ditandai sebelumnya, maksud a(Y, X) adalah bobot sisi yang dibentuk oleh simpul Y ke simpul X. Jika $d(X) = \infty$ untuk semua simpul yang belum ditandai, berhentilah karena tidak ada jalan lagi dari simpul S ke simpul yang lainnya. Jika ada, tandai simpul yang mempunyai nilai d(X) terkecil, juga tandai simpul yang terbentuk antara simpul S ke simpul X tersebut. Kemudian kita ambil Y = X. Jika tidak ada satu pun sisi berarah yang keluar dari simpul Y = X, kita ganti pilihan kita, jangan memilih Y = X tetapi simpul yang terkecil berikutnya misal A. Dengan demikian Y = A. Pada langkah ini juga jangan lupa menerapkan butir kedua pada daftar perubahan yang dilakukan yang ada pada bagian awal subbab ini.

Langkah 3. Jika simpul akhir T sudah ditandai dan pada langkah kedua gagal untuk mendapatkan nilai sisi yang lebih kecil lagi, berhentilah, karena lintasan terpendek *(shortest path)* dari S ke T sudah ditemukan. Jika simpul T belum ditandai, ulangi langkah 2.

2.2. Studi Kasus

Kita akan lebih mendalami pemahaman mengenai metode ini dengan mencoba menerapkannya pada sebuah contoh kasus. Kembali lagi kita akan menggunakan gambar 1 sebagai contoh penerapan metode Ford dalam penyelesaian masalah lintasan terpendek (*shortest path*) pada graf tersebut. Tujuannya adalah untuk memberikan gambaran sederhana mengenai metode ini.

Kita mulai dengan mendefinisikan sebuah titik awal yaitu simpul S dan titik akhir yaitu simpul T. Kita akan menerapkan metode Ford dalam mencari lintasan terpendek antara simpul S dan T pada graf di gambar 1.

Adapun langkah – langkah yang digunakan dengan menggunakan metode Ford adalah sebagai berikut :

Langkah 1. Inisial awal, hanya simpul S yang ditandai, $d(S) = 0$, $d(A) = \infty$, $d(T) = \infty$.

Langkah 2. Y = S

$d(A) = \text{Min} \{ d(A), d(S) + a(S, A) \} = \text{Min}\{\infty,0+2\} = 2$

$d(T) = \text{Min} \{ d(T), d(S) + a(S, T) \} = \text{Min}\{\infty,0+1\} = 1$

karena $d(T) = \text{Min}\{d(A), d(T)\}$, jadi simpul T ditandai dan sisi (S, T) juga ditandai. Lintasan terpendek sementara kali ini adalah (S, T).

Langkah 3. karena belum semua simpul dan sisi ditandai. Maka iterasi kedua dijalankan yaitu kembali ke langkah 2.

Langkah 2. Y = T

Karena tidak ada sisi yang meninggalkan simpul T, jadi simpul T diganti dengan simpul A. Simpul A kemudian ditandai dan sisi (S, A) juga ditandai. Jadi, lintasan terpendek sementara sekarang adalah (S, T), (S, A).

Langkah 3. Kembali ke langkah 2 untuk mendapatkan kemungkinan nilai sisi yang lebih kecil lagi.

Langkah 2. Y = A

$d(T) = \text{Min}\{d(T), d(A) + a(A, T)\}=\text{Min}\{1,2-2\}=0$

$d(S) = \text{Min}\{d(S), d(A) + a(A, S)\}=\text{Min}\{0,2+\infty\}=0$

Karena d(T) mengecil nilainya dari 1 ke 0, jadi simpul T dan sisi (S,T) dicabut lagi label tertandanya menjadi belum ditandai. Jadi lintasan pendek sementara sekarang adalah (S,A). Simpul T adalah satu-satu simpul yang belum ditandai akibat dicabut pada proses sebelumnya. Akibatnya simpul T dan sisi (A,T) harus ditandai kembali. Jadi lintasan terpendek sekarang adalah (S, A), (A, T).

Langkah 3. Kembali ke langkah 2 untuk Y = T

Langkah 2. Y = T

Karena tidak ada sisi yang keluar dari T dan tidak ada simpul yang bisa dikecilkan nilainya, serta tidak ada pula sisi yang belum ditandai. Langsung ke Langkah 3.

Langkah 3. karena semua simpul dan sisi sudah ditandai dan tidak ada lagi sisi yang nilainya bisa dikurangi, dengan begitu berhantilah algoritma Ford ini.

Jadi, Lintasan terpendek terakhir adalah solusinya yaitu (S,A), (A,T). Dimana nilainya adalah 2 + (-2) = 0.

3. ANALISIS

Untuk menangani kasus pencarian jalan terpendek pada suatu graf yang mempunyai bobot sisi yang negatif, metode Ford memang merupakan metode yang cukup hebat untuk penyelesaian kasus ini. Hal ini disebabkan metode ini hanya mengubah beberapa bagian pada algoritma Djikstra yang sudah banyak dikenal kalangan orang.

Akan tetapi metode masih mempunyai kelemahan yang cukup berarti. Metode mempunyai kelemahan jika pada suatu graf mempunyai sirkuit yang total semua bobot sisi yang membentuknya berjumlah nilai negatif atau kurang dari nol [2]. Misalkan pada suatu graf terdapat sirkuit S dimana S terdiri dari kumpulan sisi (x1, x2), (x2, x3), (x3, x4), ..., (xn, x1). Jika kita jumlahkan bobot semua sisi yang ada pada sirkuit tersebut adalah kurang dari nol. Kemudian jika kita terapkan metode Ford, kita akan menemukan bahwa perulangan pengubahan nilai d(X) dimana X adalah simpul – simpul pembentuk sirkuit akan dilakukan terus menerus tanpa henti. Jadi dalam hal seperti ini bisa dikatakan kalau metode Ford gagal menyelesaikan masalah pencarian jalan terpendek.

Jika kita tidak yakin pada suatu graf tertentu terdapat sirkuit dengan jumlah total bobot sisi pembentuknya adalah negatif, kita dapat menerapkan langsung metode Ford pada graf yang kita tentukan tersebut. Dalam proses, kita hitung berapa banyak jumlah salah satu simpul pada graf dikunjungi, jika ada salah satu simpul yang dikunjungi lebih besar sama dengan N kali, dimana N adalah jumlah sisi pada graf, hentikan algoritma tersebut, karena pada graf tersebut terdapat sirkuit yang sedang kita bahas. Jika kita teruskan, kita akan terus mengunjungi simpul tersebut sampai tidak terbatas kali.

Hal lain yang umum dibahas jika kita membicarakan mengenai algoritma adalah mengenai kompleksitasnya.

Seperti kita ketahui sebelumnya bahwa metode Ford ini merupakan sebuah metode yang diterapkan pada algoritma Djikstra. Tetapi jika algoritma Djikstra sudah dimodifikasi dengan metode ini maka kompleksitasnya bukan menurun melainkan naik secara polinomial.

Pada algoritma Djikstra yang murni, pada iterasi yang pertama, sebanyak N – 1 sisi yang belum ditandai harus dikunjungi. Dari persamaan (1), proses ini membutuhkan N – 1 Penambahan, N – 1 proses minimisasi, dan menyeleksi sebanyak N – 1 angka. Jadi pada iterasi yang pertama akan ada 3(N – 1) proses. Proses yang sama dilakukan pada iterasi selanjutnya, karena jumlah sisi yang dikunjunngi berukuran satu jadi total proses yang dilakukan adalah 3(N – 2) proses. Sehingga proses total yang dilakukan adalah sebagai berikut :

$$\Sigma_{i=1}^{i=N} 3(N-i) = 3N(N-1)/2 = 1\tfrac{1}{2}N(N-1) \quad (2)$$

Dari persamaan (2) diatas dapat kita simpulkan bahwa kompleksitas algoritma dari algoritma Djikstra adalah $O(1\ \tfrac{1}{2}\ N^2)$.

Jika algoritma Djikstra dimodifikasi dengan metode Ford, kompleksitas dari algoritma ini akan meningkat secara polinomial. Hal ini disebabkan setiap simpul pada graf maksimum akan dikunjungi sebanyak N – 1 kali. Akibat dari hal ini adalah peningkatan kompleksitas sebanyak N – 1 kali. Jadi, $O(1\ \tfrac{1}{2}\ N^2)\ X\ (N-1) = O(1\ \tfrac{1}{2}\ N^3)$. Jadi setelah algoritma Djikstra dimodifikasi dengan menggunakan metode Ford, kompleksitasnya adalah $O(1\ \tfrac{1}{2}\ N^3)$.

4. KESIMPULAN

Jika kita menemukan sebuah permasalahan *shortest path* atau pencarian jalan terpendek dalam kehidupan kita. Kita dapat memecahkan masalah tersebut dengan menggunakan algoritma Djikstra, karena algoritma tersebut cukup mangkus dalam menyelesaikan masalah tersebut. Kompleksitas dari algoritma Djikstra tersbut adalah $O(1\ \tfrac{1}{2}\ N^2)$, notasi O-besar ini menyatakan bahwa algoritma Djikstra ini cukup mampu dihandalkan dalam masalah ini.

Permasalahan lain timbul ketika kita mempunyai masalah mengenai pencarian jalan terpendek dengan memperhatikan nilai negatif. Permasalahan ini cukup penting dalam kehidupan kita. Sayangnya, algoritma Djikstra biasa tidak dapat menangani hal ini, salah satu solusinya adalah dengan memodifikasi algoritma Djikstra tersebut dengan menggunakan metode Ford. Dengan metode ini kita dapat menyelesaikan masalah yang telah disebutkan pada awal paragraf ini.

Metode yang kita gunakan ini bukanlah tanpa masalah. Metode ini mempunyai kelemahan jika pada suatu graf terdapat sebuah sirkuit yang total bobonya adalah negatif atau kurang dari nol. Jika terdapat kasus ini, algoritma yang sudah dimodifikasi dengan menggunakan metode ini akan terus berulang tanpa henti pada simpul – simpul di sekitar sirkuit. Jadi, penggunaan metode ini terbatas hanya pada graf yang tidak mempunyai sirkuit dengan total bobot negatif.

REFERENSI

[1] Rinaldi Munir, "Strategi Algoritmik", ITB, 2007.

[2] Edward Minieka, "Optimization Algorithms for Networks and Graphs", Marcel Dekker, Inc, 1978.