Algoritma Bellman-Ford Sebagai Solusi Pencarian Akses Tercepat dalam Jaringan Komputer

Sri Handika Utami* - *NIM 13508006

Program Studi Teknik Informatika, Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha nomor 10, Bandung
e-mail: if18006@students.if.itb.ac.id

ABSTRAK

Pada saat ini, jaringan komputer merupakan salah satu bagian vital dalam kehidupan manusia. Jaringan ini mempermudah manusia untuk menyelasaikan pekerjaannya dan saling berbagi informasi tanpa perlu melakukan kontak langsung. Beberapa contoh akses tersebut adalah pengunduhan (*Download)* dan Pemuatan (*Upload*). Untuk melakukan berbagai akses tersebut tentunya terdapat batasan terhadap jumlah pengaksesan dan kecepatan akses pada tiap hubungan atau yang biasa dikenal dengan *bandwith*. Tentunya agar pengaksesan memiliki kecepatan yang maksimal dan efektif, dibutuhkan suatu kecerdasan jaringan untuk memilih dan mengarahkan pengguna melalui suatu jalur tertentu agar didapatkan kecepatan akses yang maksimal. Oleh karena itu, pada makalah ini, penulis akan menjabarkan tentang salah satu algoritma yang dapat membantu masalah tersebut. Algoritma yang penulis tawarkan yakni biasa disebut dengan algoritma Bellman-Ford.

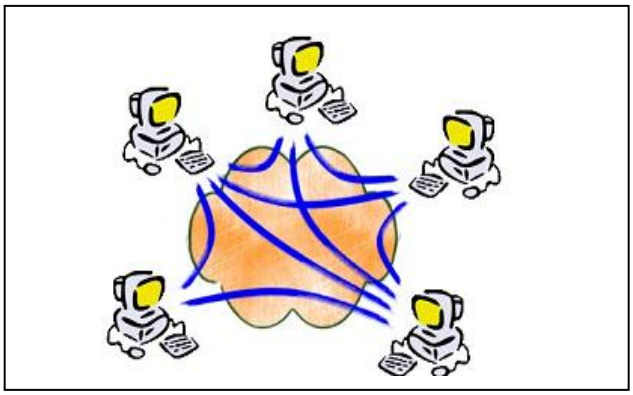
Kata kunci: Jaringan Komputer, Akses, *Bandwith*, Kecepatan Akses, Algoritma Bellman-Ford.

1. PENDAHULUAN

Saat ini, kecepatan merupakan prioritas setiap elemen profesi. Waktu menjadi faktor yang sangat diperhitungkan dalam melakukan berbagai hal.

Oleh karena itu, jaringan computer menjadi salah satu solusi dalam kecepatan ini. Penggunaan jaringan computer ini pun memiliki berbagai tingkatan, mulai dari yang paling sederhana hingga yang paling kompleks. Penggunaan jaringan komputer yang paling sederhana dapat dilihat pada *sharing* perangkat keras seperti yang biasa dilakukan dalam *home networking*. Sedangkan contoh yang cukup kompleks dapat dilihat pada penggunaan LAN dan *wireless* yang cukup populer akhir-akhir ini.

Berikut beberapa contoh ilustrasi dari topologi jaringan komputer.



Gambar 1. Contoh topologi jaringan peer to peer

2. TOPOLOGI JARINGAN DAN GRAF

Pada jaringan komputer, suatu komputer dengan computer lainnya dihubungkan dengan suatu kabel jaringan. Kabel inilah yang memungkinkan akses antar komputer di dalam suatu jaringan. Pada aplikasinya kabel ini bisa berupa kabel jaringan yang sebenarnya, dapat pula berupa "kabel semu". Keterhubungan yang seperti itu biasanya ditemui pada jaringan Nirkabel atau yang lebih populer dengan sebutan *wireless*.

Apabila digambarkan maka kita akan mendapatkan sebuah graf yang tentunya memiliki simpul dan sisi. Berikut dijelaskan tentang perumpamaan topologi jaringan ke dalam graf.

2.1 Simpul

Dalam topologi jaringan, simpul direalisasikan sebagai *client* ataupun *server* yang terlibat dalam jaringan tersebut. Karena di dalam makalah ini akan sering melibatkan istilah *client* dan *server*, maka penulis terlebih dahulu akan menjelaskan mengenai pengertian dari istilah tersebut.

Server merupakan computer yang dirancang untuk memroses permintaan dari *client* dan mengantarkan data

ke komputer lainnya. Sedangkan *client* merupakan komputer yang dirancang untuk melakukan permintaan file dari *client* lainnya ataupun dari suatu server dalam suatu jaringan. Namun, *client* hanya meiliki hak akses terbatas di dalam jaringan ini.

Dalam hal ini, *client*, *server*, ataupun *client-server* akan menjadi simpul dalam graf yang menggambarkan topologi jaringan ini.

2.2 Sisi

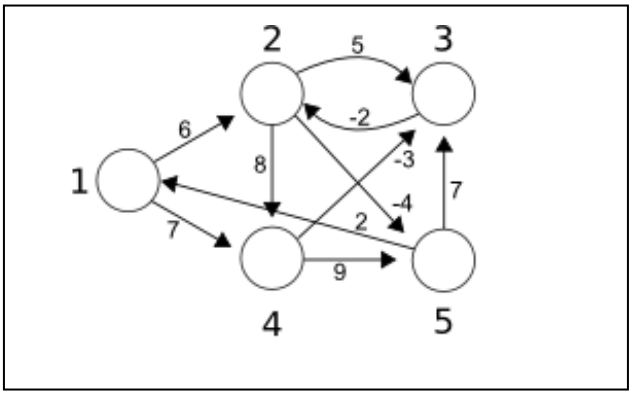
Sisi merupakan rute atau suatu jalur yang melambangkan hubungan saling keterkaitan antara dua simpul. Biasanya terdapat sisi berbobot, sisi berarah dan sisi sederhana, yakni sisi yang tidak berbobot dan tidak berarah.

Di dalam topologi jaringan, sisi direalisasikan sebagai *bandwith* yang menggambarkan kapasitas akses dan kecepatan akses antara dua computer yang berbeda. Terkadang *bandwith* juga dibedakan atas jalur *upload* ataupun jalur *download*.

3. ALGORITMA BELLMAN-FORD

Bellman-Ford merupakan salah satu algoritma yang menangani kasus pencarian lintasan dengan bobot terkecil. Algoritma ini memungknkan apabila di dalam system yang dibangun terdapat pencilan. Seperti yang sudah dicobakan sebelumnya, apabila simpul yang dituju ataupun simpul asal merupakan sebuah pencilan maka hasil yang didapatkan adalah *infinity*. Tidak hanya itu bahkan apabila ternyata tidak ada lintasan yang menghubungkan antara simpul awal dan simpul tujuan, maka bobot yang dihasilkan juga berupa *infinity*.

Keunggulan lain yang membuat algoritma ini lebih baik dari algoritma lainnya yaitu algoritma ini memungkinkan bobot pada sisi yang menghubungkan antara dua simpul berupa bilangan negatif. Hal tersebut seperti yang dijelaskan oleh contoh di bawah ini.



Gambar 2. Salah satu contoh graf dengan sisi bernilai negatif

Dalam algoritma Bellman-Fort, apabila ingin dicari lintasan dengan bobot paling sedikit dari satu ke dua, maka lintasannya adalah 1-4-3-2, sehingga bobot yang didapat adalah 7 -3 -2 = 2.

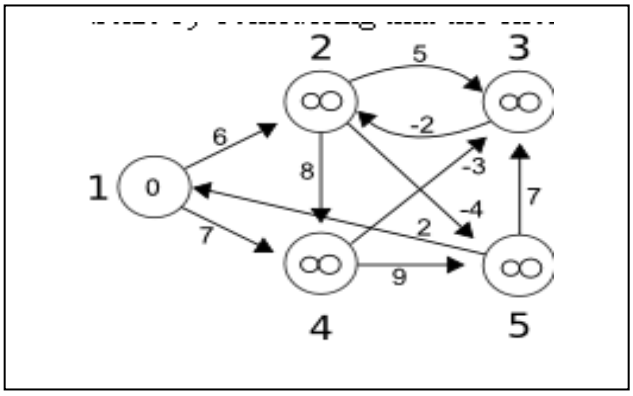
Sebenarnya landasan berpikir dari algoritma ini cukup sederhana. Berikut penulis akan menerangkan langkah-langkah pada algoritma ini. Untuk menjelaskan langkah ini kita akan tetap menggunakan contoh dari graf yang telah terdapat di gambar 2. Sebelum penulis menjelaskan tentang langkah kerja algoritma Bellman-Ford, berikut akan disajikanalgoroitma umum dari Bellman-Ford dalam notasi matematika.

$$D_i^{h+1} = \min_{j \in N(i)}[d(i,j)+D_j^h] \qquad i \neq 1$$

Pada persamaan ini berlaku $D_i^0=\infty$ dan $D_1^h=0$

Sebelum memulai perhitungan dan penganalisisan, terlebih dahulu yang harus dilakukan adalah menamai setiap simpul dan memberikan bobot dari tiap sisi. Hal yang perlu diingat apabila anda berniat untuk menggunakan algoritma yang akan saya berikan nanti, yang dkenal sebagai algoritma standar Bellman-Fort tanpa melakukan modifikasi terlebih dahulu, maka untuk sisi pada graf tak berarah anda harus mendefinisikannya sebanyak 2 kali, yakni dari titik pertam ke titik ke dua dan sebaliknya dengan nilai yang sama. Namun, apabila yang akan anda implementasikan adalah suatu graf yang berarah maka anda cukup mendefinisikannya sebanyak satu kali sesuai dengan arah graf.

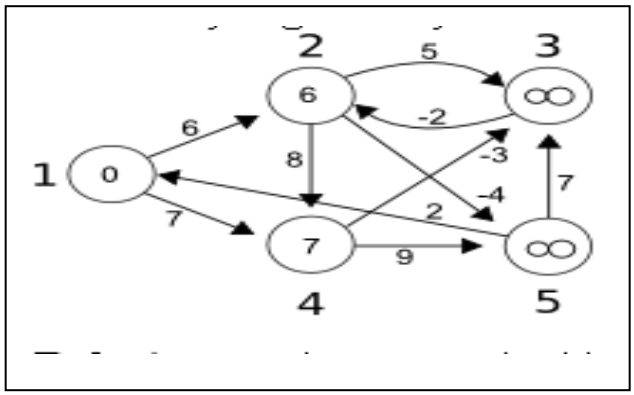
Langkah pertama yang harus dilakukan dalam analisis graf menggunakan algoritma Bellman-Ford adalah menentukan titik asal. Setelah menetapkan titik asal dari lintasan, lakukanlah penandaan simpul (*marking)*. Dalam hal ini, semua titik yang bukan titik asal harus ditandai dengan *infinity* (∞). Titik asal sendiri, sebagai titik pangkal dari lintasan yang akan dibentuk, ditandai dengan nol (0).



Gambar 3. Melakukan penandaan simpul

Selanjutnya kita harus melakukan *relaxing* pada simpul yang terdapat pada graf. Simpul yang di *relaxing* adalah

simpul selain simpul asal. Berikut akan ditunjukkan gambar yang menjelaskan salah satu dari proses *relaxing* dari graf di atas.

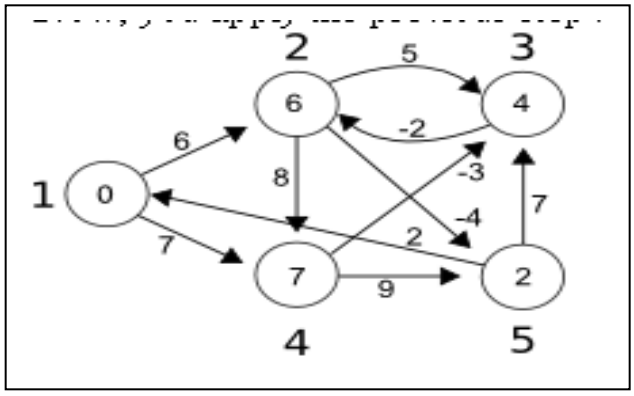


Gambar 4. *Relaxing* tahap 1

Penulis akan menjelaskan mengenai pengertian *relaxing* yang mungkin dari tadi telah membuat anda bingung. *Relaxing* di sini berarti membandingkan bobot suatu titik, dalam hal ini telah ditandai dengan *infinity*, dengan titik lain yang berada di sekitarnya yangmenghubungkannya dengan titik asal. Dalam *relaxing* tahap 1 yang ditunjukkan pada gambar di atas ditunjukkan bahwa simpul 2 langsung diberikan bobot 6. Hal ini terjadi Karena 6, besar bobot sisi yang menghubungkan antara simpul asal dengan simpul 2, lebih kecil daripada nilai sebelumnya, yakni *infinity*. Begitu pun yang terjadi pada simpul 4. Simpul 4 diberikan bobot 7 karena bobot sisi yang menghubungkan simpul 4 dengan simpul asal adalah 7 yang dalam hal ini lebih kecil jika dibandingkan dengan nilai sebelumnya (*infinity*).

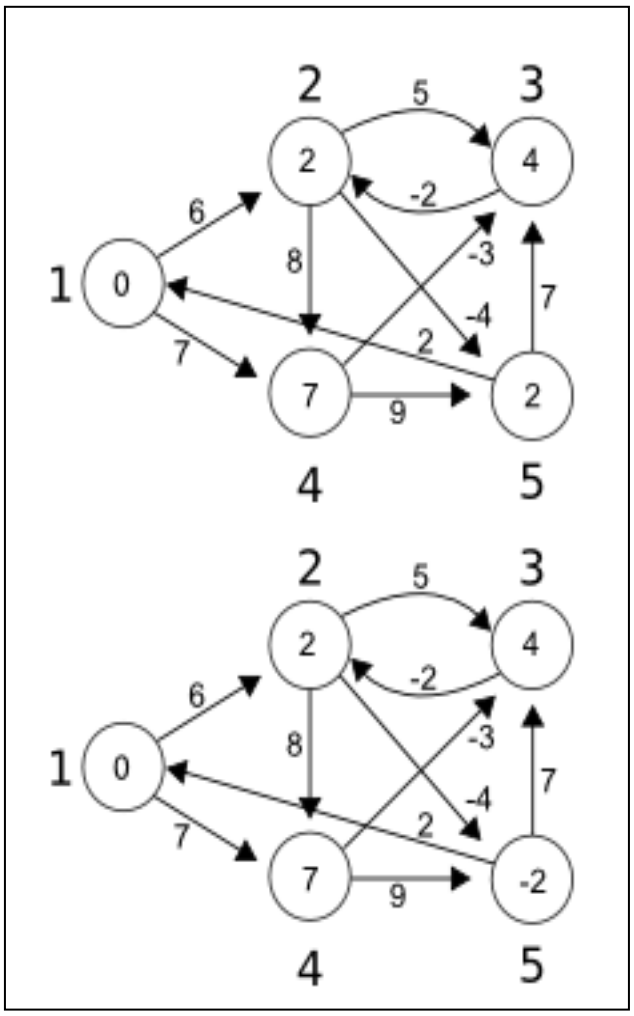
Proses relaxing ini dilakukan sebanyak n-1 kali, sehingga dapat disimpulkan bahwa kompleksitas algoritma ini adalah O(n) jika dilabangkan dengan notasi big-Oh.

Selanjutnya penulis akan menjelaskan mengenai relaxing yang terjadi berikutnya dari relaxing tahap 2 hingga relaxing tahap akhir, yang dalam kasus yang terdapat dalam contoh kita merupakan relaxing tahap 4.



Gambar 5. *Relaxing* tahap 2

Dari gambar di atas, simpul 3 bernilai 4 karena bobot simpul yang berhubungan dengan simpul 3 ditambah bobot sisi yang menghubungkannya yang paling kecil berasal dari simpul 4 yang nilainya adalah 4 (hasil penjumlahan 7 - 3). Simpul 5 berinilai 2 karena bobot hubunngan terkecil yang dimilikinya adalah keterhubungan dengan simpul 2 yakni 2 (hasil penjumlahan 6 – 4).



Gambar 6. *Relaxing* tahap 3 dan 4

Algoritma yang telah dijelaskan di atas apabila dirumuskan di dalam bahasa algoritmik akan menjadi sebagai berikut.

```
// Definisi tipe data dalam graf
record titik
{
   list sisi2
   real jarak
   titik sebelum
}
record sisi
{
   titik dari
   titik ke
   real bobot
```

```
}

function BellmanFord(list semuatitik, list semuasisi, titik dari)
    // Argumennya ialah graf, dengan bentuk daftar titik
    // and sisi. Algoritma ini mengubah titik-titik dalam
    // semuatitik sehingga atribut jarak dan sebelum
    // menyimpan jarak terpendek.

    // Persiapan
    for each titik v in semuatitik:
        if v is dari then
            v.jarak = 0
        else
            v.jarak := tak-hingga
            v.sebelum := null

    // Perulangan relaksasi sisi
    for i from 1 to size(semuatitik):
        for each sisi uv in semuasisi:
            u := uv.dari
            v := uv.ke          // uv adalah sisi dari u ke v
            if v.jarak > u.jarak + uv.bobot
                v.jarak := u.jarak + uv.bobot
                v.sebelum := u

    // Cari sirkuit berbobot(jarak) negatif
    for each sisi uv in semuasisi:
        u := uv.dari
        v := uv.ke
        if v.jarak > u.jarak + uv.bobot then
            error
            output("Graph mengandung siklus berbobot total
            negatif")
```

Telah banyak hal yang penulis paparkan mengenai kelebihan algoritma yang dirumuskan oleh Richard Bellman dan Lester Ford Jr ini. Namun, meskipun algoritma ini merupakan algoritma yang bagus untuk masalah lintasan terpendek pada graf, tentu saja ia juga memiliki beberapa kekurangan. Kekurangan tersebut antara lain, Algoritma ini apabila dibandigkan dengan Djikstra memiliki waktu eksekusi yang lebih lama. Selain itu, kesulitan yang juga mungkin dihadapi pengguna untuk algoritma ini adalah dalam modifikasinya apabila anda ingin menampilkan simpul yang dilalui oleh lintasan yang didapatkan.Hal ini terjadi karena algoritma ini tidak mengetahui simpul mana yang merupakan simpul terakhirnya sehingga algoritma ini harus memunculkan simpul baru untuk membentuk sebuah sirkuit dan melakukan *trace back* lagi.

4. ALGORITMA BELLMAN-FORD DALAM JARINGAN KOMPUTER

Data merupakan bagian yang terpenting dari jaringan komputer. Data memiliki ukuran dan perilaku yang beragam. Di dalam computer setiap *site* atau simpul melakukan *sharing file*. Tentunya di dalam melakukan kegiatan ini terdapat banyak keterbatasan dan aturan yang membatasinya.

Hal utama yang membatasi proses *sharing file* ini adalah masalah *bandwith*. *Bandwith* menunjukkan ukuran file yang dapat diproses tiap satuan waktu. Apabila pengaksesan pada suatu waktu hanya dilakukan oleh satu site, maka perhitungan kecepatan akses data akan lebih mudah. Namun, apabila pengaksesan dilakukan oleh lebih dari satu simpul, maka kecepatan pengaksesan data akan tidak konsisten dan cenderung lama.

Untuk pengaplikasian algoritma Bellman-Ford di dalam jaringan ini, perlu pengubahan nilai bobot sisi setiap terjadi penambahan jumlah akses pada suatu sisi. Hal ini tentu saja akan membantu algoritma Bellman-Ford untuk memutuskan jalur mana yang akan dipilih dan agar algoritma tidak melakukan kesalahan dalam perhitungan.

Pengaplikasian ini tentunya membutuhkan interpretasi dalam perancangannya. Dalam hal ini penulis menyarankan untuk menggunakan interpretasi matriks ketanggaan untuk mengolah data ini. Karena akan lebih mudah untuk memahami dan melakukan pengubahan pada matriks ketetanggaan dibandingkan jika kita memilih untuk menggunakan interpretasi lainnya.

Selain itu, hal lain yang perlu diperhatikan dalam hal ini adalah masalah *bandwith*. Pada algoritma yang diberikan penulis sebelumnya, algoritma Bellman-Ford mencari lintasan dengan bobot terkecil. Sedangkan apabila dipahami lebih lanjut, pada topologi jaringan kita justru mencari lintasa dengan bandwith yang terbesar. Oleh karena itu ada dua solusi yang ditawarkan oleh penulis.

Solusi yang pertama, untuk mendapatkan hasil yang diinginkan maka *bandwith* perlu diubah menjadi bentuk inversnya yakni nilai sisi menjadi 1 dibagi dengan nilai bandwith yang sebelumnya. Apabila anda memilih langkah ini, maka anda tidak perlu melakukan perubahan pada algoritma awal yang diberikan tadi. Sehingga kita tetap mencari lintasan dengan bobot terkecil.

Langkah kedua yang diberikan penulis adalah dengan melakukan pengubahan algoritma. Dalam hal ini, yang perlu dirubah adalah perbandingan saja. Namun, ada satu hal lagi yang harus diperhatikan. Apabila sebelumnya anda menetapkan infinity sebagai batas atas dari bobot simpul, maka apabila kita menerapkan cara ini kita harus mengubah definisi nilai *infinity* dari batas atas menjadi batas bawah.

Tentunya dalam setiap pilihan terdapat kekurangan dan kelebihan. Kelebihan algoritma ini dalam system jaringan adalah jalur yang didapatkan sebagai hasil dari

operasi ini memang benar-benar lintasan (*path)* yang terpendekj. Selain itu, algoritma Bellman-Ford juga memperhitungkan masalah buffer dan delay.

5. KESIMPULAN

Graf memiliki banyak realisasi dalam kehidupan sehari-hari. Jaringan adalah salah satu aplikasi dari graf tersebut. Karena jaringan dapat digambarkan sebagai sebuah graf, maka penyelesaian lintasan tersingkat juga dapat ditemukan dengan menggunakan algoritma-algoritma yang umum dalam graf. Bellman-Fort merupakan salah satu algoritma dalam menyelesaikan masalah ini. Namun, seperti algoritma lainnya, agoritma ini juga memiliki kelebihan dan kekurangannya tersendiri. Modifikasi merupakan salah satu solusi penyelesaian dalam massalah ini.

REFERENSI

.

[1] Cormein, Leiserson dan Rivest, "Introduction to Algorithm", McGraw-Hill,1990.

[2] Madhusudhan N, "Bellman-Ford's Algorithm", http://www.laynetworks.com/Bellman%20Ford%20Algorithm.htm, Tanggal akses: 17 Desmber 2009, pukul 10.00 WIB

[3] Munir, Rinaldi. Diktat Kuliah IF2153 Metemaika Diskrit Edisi Keempat. Program studi Teknik Informatika, Sekolah Teknik elektro dan Informatika, Institut Teknolosi Bandung.2006.

[4] Weisstein, Eric W. "Bellman-Ford Algorithm", http://mathworld.wolfram.com/Bellman-FordAlgorithm.html, Tanggal Akses: 19 Desember 2009, pukul 20.00

[1] Vasudev,C, "Graph Theory with Applications", New Age Internasional Publishers