

Pencarian Jalur Terpendek Pada Sistem Jaringan Komputer

Menggunakan Algoritma Program Dinamis

Fadli Demitra (13511047)

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia

fadli_demitra@itb.ac.id

Abstrak—Program Dinamis adalah salah satu program yang cukup baik dan handal dalam melakukan pencarian jarak terpendek dari sebuah graf. Karena mampu menghasilkan solusi yang optimal dan dapat melakukan iterasi yang lebih mangkus dibandingkan dengan algoritma brute force.

Algoritma Program Dinamis ini dapat diterapkan pada jaringan komputer untuk mencari jarak tempuh terpendek dari router client ke router server. Dalam jaringan komputer, pencarian rute jalur terpendek ini sangat diperlukan untuk mempercepat pengiriman data.

Oleh sebab itu penulis mencoba untuk mengembangkan pencarian menggunakan program dinamis dalam melakukan pencarian jalur yang akan ditempuh oleh sebuah data sewaktu melakukan pengiriman data.

Istilah Index— Program Dinamis, Jaringan Komputer, Optimal, Jalur Terpendek.

I. PENDAHULUAN

Pada matakuliah Jaringan Komputer kita mengenal bahwa adanya algoritma routing yang berfungsi untuk menentukan ke arah mana sebuah paket harus diteruskan. Salah satu algoritma yang bisa digunakan adalah algoritma program dinamis ini.

Karena program dinamis ini mampu mencari semua kemungkinan solusi dari semua jalur yang ada dengan jumlah iterasi yang lebih minimum dengan algoritma brute force tentunya sehingga algoritma ini cukup baik digunakan pada pencarian jalur pengiriman paket pada studi jaringan komputer.

Banyak cara untuk melakukan penentuan jalur rute yang akan ditempuh sebuah data untuk sampai ke tempat tujuannya dalam melakukan pengiriman data atau melakukan koneksi internet. Salah satu diantaranya adalah menggunakan program dinamis ini.

Salah satu kemampuan program dinamis yang akan kita handalkan pada permasalahan kali ini adalah kemampuan program dinamis dalam melakukan pencarian solusi secara optimal. Diharapkan dengan ditemukannya solusi yang optimal pada sebuah jaringan internet ini dapat meningkatkan kecepatan transfer data dari satu tempat ke tempat yang lainnya atau dari client ke server.

II. LANDASAN TEORI

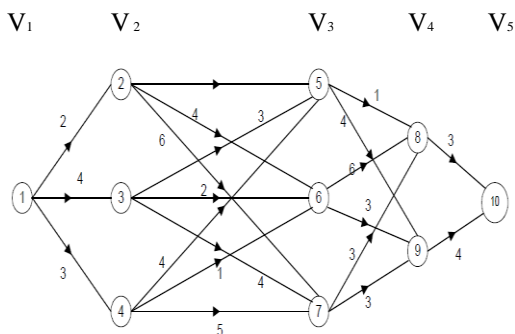
Program Dinamis merupakan sebuah metode untuk memecahkan masalah dengan menguraikan solusi yang

ada menjadi beberapa tahapan sehingga solusi bisa dipandang sebagai serangkaian keputusan yang saling berkaitan. Sehingga solusi dapat diselesaikan dengan menggunakan metode rekursifitas pada program.

Dengan menggunakan metode rekursifitas ini, akan membuat sebuah program berjalan lebih cepat karena membagi-bagi masalah menjadi bagian-bagian yang kecil. Selain itu hal itu juga yang akan membuat program ini lebih mangkus dari pada program yang lainnya yang mampu mencari solusi secara optimal.

Karakteristik Program Dinamis (lihat Munir, Rinaldi, *Diktat Kuliah IF2211 Strategi Algoritma*, Teknik Informatika ITB, 2004):

1. Persoalan dibagi menjadi beberapa tahap yang pada setiap tahap hanya diambil satu keputusan.
2. Masing-masing tahap terdiri dari sejumlah status yang berhubungan dengan tahap tersebut. Secara umum, status merupakan kemungkinan masukan yang ada pada tahap tersebut. Jumlah status bisa berhingga atau tidak berhingga. Gambar II.1 memperlihatkan perbedaan antara tahap dan status diberikan pada graf multistadium. Tiap simpul di dalam graf tersebut menyatakan status, sedangkan V_1, V_2, \dots menyatakan tahap.



Gambar II.1 Graf yang menyatakan tahap dan status

3. Hasil dari keputusan yang diambil pada setiap

tahap ditransformasikan dari status yang bersangkutan ke status berikutnya pada tahap berikutnya.

4. Ongkos(cost) pada suatu tahap meningkat secara teratur (steadily) dengan bertambahnya jumlah tahapan.
5. Ongkos pada suatu tahap bergantung kepada ongkos tahap-tahap yang sudah berjalan dan ongkos pada tahap itu.
6. Keputusan terbaik pada suatu tahap bersifat independen terhadap keputusan yang dilakukan pada tahap sebelumnya.
7. Adanya hubungan rekursif yang mengidentifikasi keputusan terbaik untuk setiap status pada tahap $k + 1$.
8. Prinsip optimalitas berlaku pada persoalan tersebut.

Langkah-langkah yang dilakukan untuk mengembangkan algoritma program dinamis :

1. Karakteristikan struktur solusi optimal.
2. Definisikan secara rekursif nilai solusi optimal.
3. Hitung nilai solusi optimal secara maju atau mundur.
4. Konstruksi solusi optimal.

Ada dua cara menyelesaikan masalah mencari lintasan terpendek dengan menggunakan program dinamis, yaitu dengan menggunakan program dinamis mundur atau program dinamis maju. Keduanya akan menghasilkan hasil yang sama karena keduanya dapat menghasilkan solusi yang optimal.

Program dinamis mundur memulai mencari jalur terpendek dari tempat tujuan ke posisi awal sedangkan

program dinamis maju melakukan hal yang sebaliknya. Tidak akan ada perbedaan dalam hasilnya.

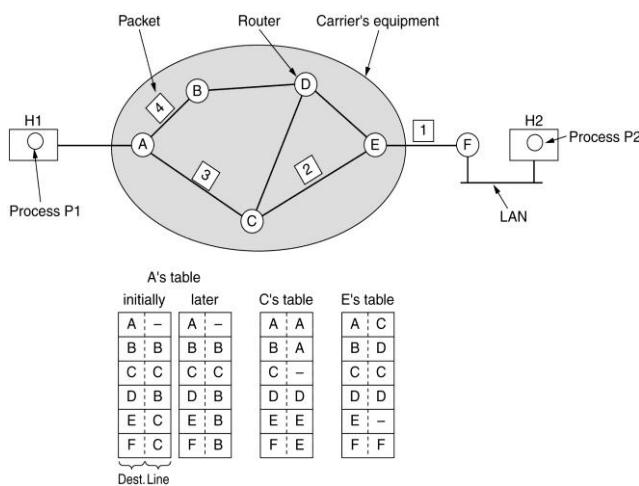
III. PEMBAHASAN DAN ANALISIS

Pada suatu jaringan seperti pada jaringan internet, sebuah paket data akan dikirimkan dari satu router ke router yang lainnya. Dimana di setiap router sudah mengetahui tetangganya masing-masing beserta informasi jarak tempuh untuk mencapai ke tetangga mereka.

Jika informasi jarak tempuh untuk sampai ke tetangga masing-masing belum diketahui maka kita harus membuat sebuah algoritma yang mampu mengeset nilai tetangga mereka masing-masing. Yang diperlukan disini hanyalah jarak antara satu router dengan router yang berdekatan.

Dengan asumsi bahwa setiap router telah mengetahui siapa tetangga mereka dan berapa jarak tempuh yang dibutuhkan hingga sampai ke tetangganya itu maka informasi dari router tersebut akan dimasukkan kedalam sebuah file eksternal berbentuk matrix.

Seperti pada gambar III.2 berikut :



Gambar III.2 Contoh pengiriman data dari Proses P1 ke Proses P2.

Pertama kita akan mulai menjelajahi router dari posisi kita saat ini. Ketika sampai pada sebuah router maka kita akan mengambil informasi siapa saja tetangga mereka dan berapa jaraknya dan disimpan ke dalam buffer dan jangan lupa mengambil nama router yang telah dilewati sebelum ditulis ke dalam file eksternal berupa matriks ketetanggan.

Telusuri terus tetangga dari router tersebut secara rekursif dan memasukkan semua data yang telah di dapat (berupa informasi jarak dan tetangganya) ke dalam buffer penampungan sementara yang berupa matriks ketetanggan hingga semuanya bertemu kepada posisi tujuan.

Jika sewaktu penelusuran dan pencarian data tidak sampai ketempat tujuan melainkan bertemu dengan router yang terputus dan tidak mencapai tujuan maka kembalikan semua jalur yang dilewati tadi ke posisi semula dan jika dia bertemu dengan posisi awal kita tadi maka lakukan hal yang sama karena hal itu tidak akan membuat kita sampai pada tujuan.

Yang kita cari disini adalah semua tetangga yang akan membawa kita sampai ke tempat tujuan, ketika di tempat tujuan jangan ambil informasi tetangga dan jaraknya. Selain dari itu jangan masukkan ke dalam file eksternal yang berupa matriks ketetanggaaan. Kemudian masukkan data yang sudah di dapat kedalam file eksternal.

Baris pertama pada file eksternal tersebut berisi jumlah router yang ada dan diambil informasinya. Atau panjang buffer tersebut, karena buffer tersebut menyimpan informasi dari tetangga yang dilewatinya satu persatu. Sehingga bisa dipastikan bahwa jumlah router yang dilewati sama dengan jumlah buffernya.

Baris selanjutnya berisi nilai ketetanggan dari buffer yang pertama kali atau tempat asal kita, kemudian mencari

ketetanggan dari setiap buffer tersebut dan pada setiap baris beri nilai masukan pada matriks dengan 0 jika tidak bertetangga dan nilai jaraknya jika mereka bertetangga. Misal buffer dengan indeks ke 0 bertetangga dengan buffer dengan indeks ke 4 dengan jarak 8, maka isi pada baris pertama kolom ke 5 dengan nilai 8.

Contoh file matriks ketetanggan :

```

10
0,0,0,0,8,0,0,0,0,0
2,0,0,0,0,0,0,0,0,0
4,0,0,0,0,0,0,0,0,0
3,0,0,0,0,0,0,0,0,0
0,7,3,4,0,0,0,0,0,0
0,4,2,1,0,0,0,0,0,0
0,6,4,5,0,0,0,0,0,0
0,0,0,0,1,6,3,0,0,0
0,0,0,0,4,3,3,0,0,0
0,0,0,0,0,0,0,0,0,0

```

Setelah selesai memasukkan semua data ketetanggan dari setiap router maka kita sudah selesai separuh jalan karena saat ini kita hanya tinggal membuat program dinamisnya saja.

Program dinamis yang akan kita buat adalah program dinamis maju atau mundur tergantung dengan masukan matriksnya. Tahap pertama yang kita lakukan dalam program dinamisnya adalah memasukkan semua nilai dari matriks tersebut ke dalam kelas yang bernama router dimana router tersebut mempunyai atribut table dari router tetangganya dan nilai jarak tempuh dari router itu sendiri.

Setelah tahap pertama selesai kita memulai pada tahap basis program dinamis dimana pada tahap basis berisi nilai dengan rumus (lihat Munir, Rinaldi, *Diktat Kuliah IF2211 Strategi Algoritma*, Teknik Informatika ITB, 2004):

$$F_k(s) = f_k(s, x_k)$$

Dan tahap rekurens yang berisi :

$$F_k(s) = \min\{c_{sxk} + f_{k+1}(x_k)\}, k = 1,2,3, \dots \text{ jumlah statenya.}$$

Keterangan :

- Peubah x_k menyatakan peubah keputusan pada tahap k
- c_{sxk} menyatakan bobot (cost) sisi dari s ke x_k
- $f_k(s, x_k)$ menyatakan total bobot lintasan dari s ke x_k
- $F_k(s)$ adalah nilai minimum dari $f_k(s, x_k)$

Contoh table pada tahap basis :

$$x_4 \in \{10\}; s \in \{8,9\}; f_4(8)=3; f_4(9)=4; x_4^*=10$$

s	Solusi Optimum	
	$f_4(s)$	x_4^*
8	3	10
9	4	10

Tabel III.1 Contoh Gambar Tahap Basis

Pada table tersebut terlihat bahwa pada tahap basis nilai dari jarak dari s yang ke 8 ke yang ke 10 dimasukkan langsung ke table pada kolom $f_k(s)$ dengan nilai jarak keduanya karena masih belum ada nilai tempuh terkecil dari 8 ke 10.

Pada kasus kita, nilai 8(8 adalah sebuah string atau nama kelas) merupakan bagian dari kelas router dimana dia akan mempunyai tetangga dan sekarang sudah mempunyai nilai yaitu 3 yang merupakan jarak terpendek dirinya ke tempat tujuan. Begitu juga dengan router yang mempunyai nama 9(9 adalah sebuah string atau nama kelas).

Dari dua kelas router kita tersebut kemudian kita akan mengambil semua tetangga mereka berdua jika mereka banyak maka ambil semua tetangganya dan masukkan ke table berikut :

$$x_3 \in \{8,9\}; s \in \{5,6,7\}; f_4(8)=3; f_4(9)=4$$

$s \setminus x_3$	$f_3(s, x_3) = c_{s, x_3} + f_4(s)$		Solusi Optimum	
	8	9	$f_3(s)$	x_3^*
5	$1+3=4$	$4+4=8$	4	8
6	$6+3=9$	$3+4=7$	7	9
7	$3+3=6$	$3+4=7$	6	8

Tabel III.2 Contoh Gambar Tahap Rekurens

Pada table diatas kita masukkan semua nilai tetangga pada kolom pertama kemudian kita akan membandingkan nilainya dengan router yang barusan kita ambil semua tetangganya. Jika hanya ada satu pembanding saja misalnya router 5 tidak bertetangga dengan router 9 maka nilai yang dimasukkan adalah nilai hasil penjumlahan pada kolom dengan nama router 8.

Dan nilai yang di dapat tersebut dijadikan sebagai nilai atribut dari router 5 dengan nilai jarak yang dia miliki saat ini merupakan nilai jarak terdekat ke tujuan. Jika router tersebut merupakan sebuah router yang akan di tuju maka dia dimasukkan ke dalam buffer1 yang merupakan tempat penampungan semua nilai jarak tempuh terdekat yang akan dibandingkan kemudian.

Dan jika itu bukan nilai tujuan maka tidak usah di simpan dalam buffer1 karena masih harus rekursif kembali hingga h=sampai ke posisi tujuannya. Dan jika semuanya sudah sampai ke posisi tujuannya maka semua nilai dari buffer tadi di bandingkan satu persatu, ingat kita tadi menyimpan semua nilai terdekatnya masing-masing.

Sekarang kita bandingkan semua nilai terdekatnya itu. Jika ada router dengan nilai atribut yang terkecil maka dialah yang merupakan jalur tersingkatnya. Dan jangan lupa kita juga menyimpan jalur yang telah buffer itu lewati dari awal hingga akhir. Sehingga kita bisa mendapatkan jalur yang paling optimal dari perbandingan yang terakhir kali tadi.

Setelah kita mendapatkan jalur yang paling optimal dari algoritma program dinamis tadi. Kita kemudian menambahkan informasi tersebut kepada paket yang akan melakukan perjalanan ke tempat tujuan. Sehingga paket tersebut mengetahui ke arah mana dia akan menempuh jalur hingga sampai ke tempat tujuan. Dan paketpun sampai ke tempat tujuan menggunakan jalur yang paling singkat.

IV. KESIMPULAN

Algoritma Program Dinamis juga mampu sebagai salah satu algoritma routing yang berfungsi untuk menentukan ke arah mana sebuah paket harus diteruskan. Dengan prinsip optimalitasnya maka algoritma ini akan membuat paket terkirim lebih cepat. Dan koneksi antar jaringan bisa berjalan dengan cepat.

Algoritma Program Dinamis mampu membantu pencarian solusi untuk menentukan arah yang harus dilewati oleh sebuah paket melalui jaringan.

REFERENCES

- [1] Munir, Rinaldi, *Diktat Kuliah IF2211 Strategi Algoritma*, Teknik Informatika ITB, 2004.
- [2] Horowitz, Ellis, & Sartaj Sahni, 1978, *Fundamental of Computer Algorithms*, Pitman Publishing Limited.
- [3] Neapolitan, Richard R., *Foundations of Algorithms*, D.C Heath and Company, 1996.
- [4] Levitin, Introduction to the Design & Analysis of Algorithms, 2003, Addison-Wesley, 2003. Website penerbit: www.aw.com/cssupport
- [5] Parsons, Thomas W., 1995, *Introduction to Algorithms in Pascal*, John Wiley and Sons.

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 20 Desember 2013

A handwritten signature in black ink, appearing to read 'Fadli', with a horizontal line underneath.

Fadli Demitra

13511047