

Penerapan Algoritma *Greedy* Untuk Pemantauan Jaringan Komputer Berbasis Rute (*Path-oriented*)

Charles Hariyadi (13505105)

Program Studi Teknik Informatika Institut Teknologi Bandung
Jl. Ganesha No.10, Bandung
charles@charles.web.id

ABSTRAK

Kebutuhan untuk suatu perangkat yang dapat melakukan pemantauan terhadap performa dan kinerja dari sebuah jaringan komputer semakin meningkat seiring pertumbuhan *Internet Service Providers (ISP)*. Perangkat yang digunakan untuk memantau jeda waktu komunikasi dan kesalahan-kesalahan yang terjadi pada jaringan komputer berbasis *Internet Protocol (IP)* menjadi suatu kebutuhan utama dalam melakukan manajemen terhadap jaringan komputer[1]. Saat ini telah berkembang berbagai jenis perangkat pemantau kinerja jaringan komputer. Tetapi perangkat pemantau yang sudah ada biasanya memiliki kekurangan, seperti harus diimplementasikan disetiap perangkat jaringan atau biasanya terbatas pada semua perangkat yang terdaftar pada jaringan lokal saja.

Karena itulah penulis melakukan kajian mengenai pendekatan dua fase yang menjadi acuan dalam melakukan pemantauan secara lengkap dan efisien terhadap jaringan komputer dengan menggunakan algoritma *greedy*. Pendekatan dua fase dilakukan dengan memilih *node* dari jaringan untuk menjadi stasiun pemantau (*monitoring station*) dengan menggunakan algoritma *greedy*. Dengan begitu diharapkan kajian dapat memberikan informasi mengenai kebutuhan *node* dan *monitoring station* untuk melakukan pemantauan terhadap jaringan secara optimal tanpa harus menghabiskan *resource* jaringan.

Kata kunci: Algoritma *Greedy*, *two-phased network monitoring*, *network monitor*, *monitoring station*.

1. Pendahuluan

Perangkat pemantau jaringan (*Network monitoring tools*) merupakan perangkat jaringan yang bertugas melakukan pemantauan terhadap kinerja dan performa jaringan. Perangkat ini digunakan untuk membantu seorang administrator jaringan melakukan pengawasan

dan manajemen terhadap keseluruhan perangkat yang terhubung ke jaringan komputer tersebut. Sebuah perangkat pemantau jaringan biasanya harus memenuhi dua hal berikut ini :

- Lingkup (*Coverage*)**, sebuah perangkat pemantau harus mampu melakukan pemantauan terhadap keseluruhan *link* dan rute pada jaringan komputer.
- Efisien**, Sebuah perangkat pemantau harus mampu meminimasi kebutuhan terhadap *resource* untuk melakukan pemantauan. *Resource* dalam hal ini dapat berupa perangkat komputer, *modem*, *router* ataupun perangkat-perangkat jaringan lainnya.

Untuk memenuhi kebutuhan akan sebuah alat pemantau jaringan yang mampu memenuhi kedua persyaratan diatas, maka pendekatan pemantauan jaringan secara 2 fase (*two-phased network monitoring*) dapat digunakan. Dalam penerapannya, pemantauan jaringan 2 fase menggunakan algoritma *greedy*. Algoritma *greedy* digunakan sebagai algoritma utama dalam kedua fase tersebut. Dengan begitu diharapkan algoritma *greedy* dapat menjadi

2. Algoritma Greedy

Algoritma *greedy* merupakan salah satu algoritma yang dapat menyelesaikan bermacam-macam permasalahan termasuk permasalahan mengoptimalkan (minimum atau maksimum) hasil yang didapat. Algoritma *greedy* menerapkan metode pencarian terkontrol dengan melakukan pilihan yang memberikan hasil optimal sementara[2]. Dewasa ini, terdapat berbagai jenis algoritma yang telah diperkenalkan, seperti A bintang (*A-star*) atau algoritma genetik dimana inti dari semua algoritma ini adalah algoritma *greedy*. Memilih nilai optimal (maksimum atau minimum) dari sekumpulan pilihan merupakan konsep dasar dari algoritma *greedy*. Algoritma *greedy* dapat dideskripsikan sebagai berikut :

Procedure Greedy (partial solution *S*, sub-problem *P*)

Begin

Bangkitkan semua kemungkinan pilihan sebagai sebuah list *L* untuk sub-problem *P*;

```

while (L tidak kosong OR belum berakhir)

```

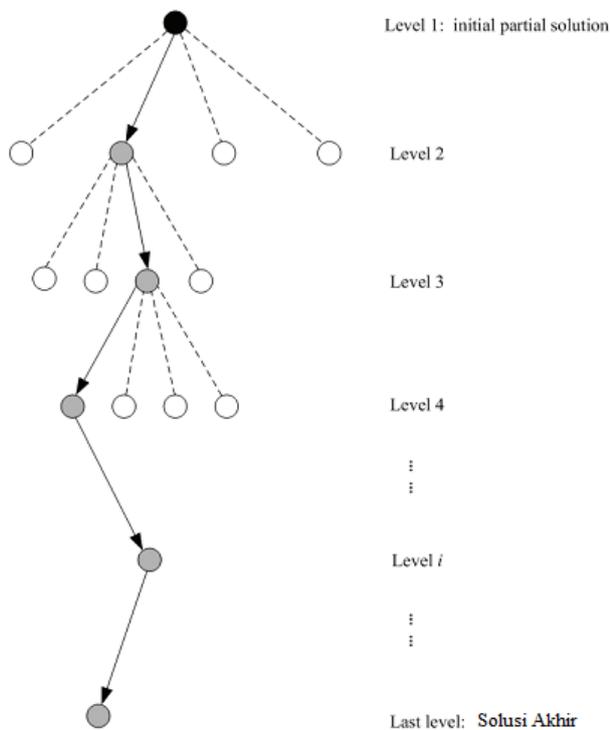
```

    Hitung besar nilai untuk setiap pilihan di L;
    Lakukan perubahan pada S dan P dengan memilih anggota L yang memberi nilai optimal;
    Ubah L sesuai dengan S dan P;
end while;
return solusi berupa pilihan yang paling optimal;

```

End.

Algoritma diatas diulangi sebanyak jumlah sub-problem sampai ditemukan hasil optimal.



Gambar 1 Representasi dari Algoritma Greedy

- Kelebihan utama dari algoritma *greedy* adalah :
- Mudah diimplementasikan, algoritma *greedy* sangat mudah untuk diimplementasikan pada persoalan-persoalan yang ada.
 - Efisien, algoritma *greedy* merupakan algoritma yang sangat efisien karena algoritma ini selalu berusaha memilih hasil optimal.

Kedua hal ini menjadikan algoritma *greedy* terkadang mampu memberikan hasil optimal, seperti untuk permasalahan *activity-selection*, *fractional knapsack* dan *minimum spanning tree* [3].

2. Network Monitoring

Network monitoring merupakan proses pemantauan terhadap kinerja, kondisi dan performa dari jaringan. *Network monitoring* biasanya digunakan pada jaringan berskala besar untuk mempermudah kerja dari seorang administrator jaringan. Terdapat dua metode pemantauan jaringan yang telah diterapkan:

- Node-oriented tools* dimana setiap perangkat pada jaringan masing-masing memiliki aplikasi pemantau jaringan. Sehingga dapat memberikan informasi dengan tingkat akurasi yang tinggi untuk setiap perangkat tersebut. Kekurangan utama dari metode ini adalah tidak dapat melakukan pemantauan parameter yang melibatkan lebih dari satu perangkat jaringan.
- Path-oriented tools* dilakukan dengan memanfaatkan teknik-teknik pemantauan jaringan seperti *ping*, *traceroute*, *telnet*, dan sebagainya. Metode ini memberikan kemampuan lebih untuk melakukan pemantauan terhadap beberapa perangkat jaringan tanpa harus melakukan instalasi aplikasi pemantau disetiap perangkat jaringan. Kelemahan utama dari metode ini adalah waktu tenggang yang sangat lama jika penempatan *node* pemantau (*node station*) tidak pada tempat yang optimal.

Kajian ini dilakukan dengan memanfaatkan *path-oriented tools* dan menerapkan skema 2 fase untuk mengoptimalkan hasil pemantauan.

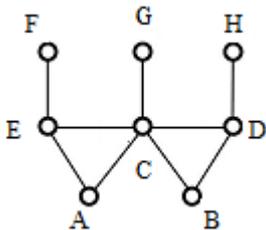
2.1 Pemodelan Jaringan Pada Kajian

Penulis memodelkan jaringan komputer sebagai graf $G(V,E)$, dimana *node* V/v pada graf merupakan representasi dari *router* atau perangkat komputer dan sisi E/e pada jaringan merepresentasikan komunikasi yang menghubungkan setiap *node*. V'/v' merupakan representasi dari *node* yang bertetangga dengan sisi E/e dan E'/e' merupakan representasi dari sisi yang bertetangga dengan V'/v' . Jumlah dari *node* dan sisi pada graf dinotasikan dengan $|V|$ dan $|E|$. Pst merupakan rute dimana sebuah paket IP berjalan dari sumber s dengan tujuan t . Semua paket data diteruskan menggunakan protokol standard dalam pengiriman paket data[4]. *Routing tree* (RT) dari s merupakan semua kemungkinan rute yang dapat dituju oleh s pada V . Setiap rute (sisi) pada pemodelan ini memiliki nilai yang nantinya berfungsi untuk memilih nilai optimal.

2.2 Cara Kerja

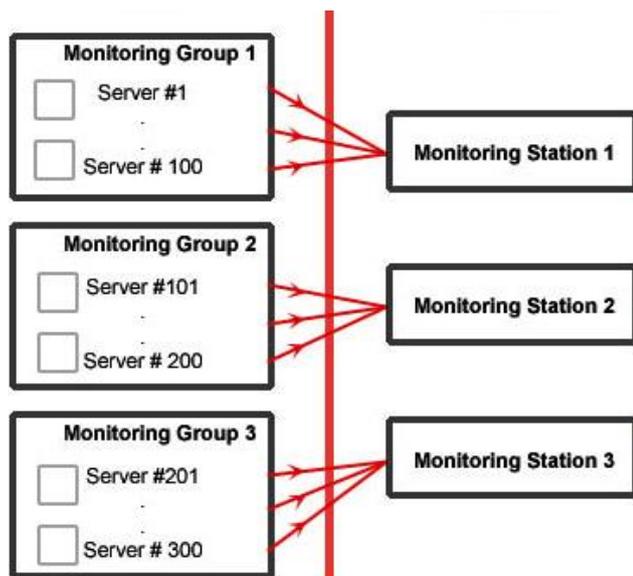
Pada bagian ini penulis akan menjelaskan secara ringkas cara pemantauan jaringan dengan menggunakan metode *path-oriented* yang diterapkan dalam kajian. Pertama kali,

akan dilakukan test untuk menentukan *monitoring station* dari RTs, yaitu dengan melakukan pengecekan terhadap *node* yang ada pada RTs dan hubungannya terhadap *node* lain. Pengecekan dilakukan dengan cara mengirimkan pesan terhadap setiap *node* yang ada pada jaringan. Seperti pada gambar 2, pengecekan akan dimulai dari *node* A yang mengirimkan pesan ke semua *node* yang ada di RTs tersebut (BCDEFGH), dan seterusnya sampai ditemukan *node* yang memiliki delay minimal ke semua *node* yang ada di RTs.



Gambar 2 Pemilihan *Monitoring Station*

Jika *monitoring station* telah ditemukan, maka selanjutnya akan dilakukan fase ke-2. Pada fase ke-2, sebuah *monitoring station* RTs yang ingin melakukan pemantauan terhadap jaringan, akan mengirimkan pesan ke semua *node* yang ada pada RTs melalui *node* s (sebagai *monitoring station*). Kemudian *node-node* yang ada pada RTs akan mengirimkan kembali pesan secara langsung kepada s. *Monitoring station* akan melakukan analisis terhadap pesan-pesan yang dibalas oleh *node* yang adapada RTs sesuai dengan protokol TCP/IP [4]. Hal ini berlaku untuk setiap *monitoring station* yang ada pada jaringan (Gambar 3).



Gambar 3 Fase 2, Melakukan Pemantauan Terhadap Setiap *node*

Terdapat beberapa ketentuan untuk kedua fase ini :

- a. Sebuah *monitoring station* harus mampu mencakup semua *link* dan *node* yang ada pada RTs-nya (*monitoring group*)
- b. Fase kedua memastikan bahwa setiap *link* yang ada pada jaringan dipantau oleh paling sedikit satu buah *monitoring station*.

Dengan menggunakan skema dua fase, permasalahan pemasangan dan perbaikan terhadap *monitoring station* dan juga pengiriman pesan terhadap banyak *node* dapat dihindari. Karena pada setiap fase ini dicari solusi optimal untuk mengoptimalkan kinerja dari pemantauan jaringan, yaitu dengan mengurangi sebanyak mungkin *monitoring station* yang mungkin muncul dan juga mengoptimasi jumlah maksimal *node* yang mungkin ditangani oleh sebuah *monitoring station*. Bagian analisis dan distribusi informasi pesan tidak ditangani oleh kajian ini.

3. Penerapan Algoritma *Greedy* dalam Fase 2 Tahap

Dalam pengembangannya tentu saja diperlukan algoritma yang efektif dan efisien sehingga fase ini dapat memberikan hasil yang optimal. Berikut ini akan dijelaskan penerapan algoritma *greedy* dalam fase 2 tahap.

3.1 Algoritma *Greedy* untuk Penentuan *Monitoring station*

Dengan merepresentasikan permasalahan sebagai graf $G(V,E)$ penulis dapat mendefinisikan permasalahan *set cover* (SC). Sekumpulan sisi E , mendefinisikan semua elemen yang ada pada graf Z . Kumpulan *set* Q termasuk *subset*-nya $Q_v = \{e \mid e \in T_v\}$ untuk semua *node* $v \in V$, dimana *cost / weight* dari setiap sisi e pada *subset* didefinisikan sebagai w_v . Algoritma *greedy* merupakan algoritma yang memanfaatkan iterasi dimana setiap iterasi dilakukan pemilihan terhadap nilai paling optimal. Kita misalkan $C \subseteq Z$ sebagai kumpulan dari elemen yang belum termasuk dalam *monitoring station*. Sebagai tambahan, $n_v = \{Q_v \cap C\}$ sebagai jumlah dari elemen C untuk setiap $v \in V$ pada setiap permulaan iterasi.

Cara kerja algoritma ini adalah dengan memilih *set* Q_v pada setiap iterasi. Kemudian dilakukan seleksi terhadap rasio minimum dan menghilangkan setiap elemen dari C sampai C menjadi kosong. Kompleksitas waktu optimal dari penggunaan algoritma ini adalah $O(\ln |V|)$ dan kasus terburuk $O(|V|^3)$.

```

 $S \leftarrow \emptyset, C \leftarrow Z$ 
While  $C \neq \emptyset$  do
  For every  $Q_v \in Q - S$  do
     $n_v \leftarrow |Q_v \cap C|$ 
     $Q \leftarrow \arg \min_{Q_v, Q_v \notin S} \frac{w_v}{n_v}$ 
     $S \leftarrow S \cup \{Q\}$ 
     $C \leftarrow C - Q$ 
End While
Return  $S$ 

```

Algoritma ini mampu memberikan solusi mendekati optimal untuk permasalahan pencarian *monitoring station*. Dalam setiap iterasinya dilakukan pencarian *node* yang memiliki delay paling kecil terhadap semua *node* yang ada pada jaringan.

3.2 Algoritma Greedy untuk Penentuan Penugasan Probe/Node

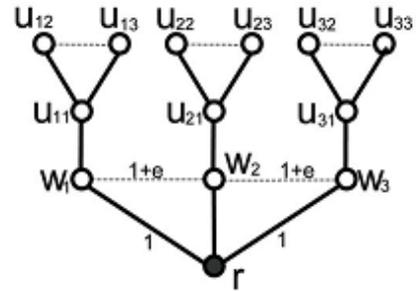
Berdasarkan hasil penelitian, untuk penugasan *probe* diperlukan pengiriman 2 pesan [5]. Pesan itu digunakan untuk melakukan pemantauan tenggang waktu terhadap *link* yang ditentukan. Algoritma ini mendefinisikan hubungan *monitoring station* dan sisi sebagai (s, e) , $e \in T_s$, dan dalam setiap iterasi algoritma dilakukan pemilihan terhadap pasangan (s', e') dengan nilai paling minimal dan menambahkan pesan sebagai penugasan untuk melakukan analisis. Jika terdapat pasangan yang memiliki nilai yang sama, mak dipilih berdasarkan jumlah *hop* yang dilalui untuk mencapai pasangan tersebut. Algoritma ini dilakukan sampai semua sisi telah termonitor. Berikut algoritma *greedy* yang diterapkan.

```

 $\mathcal{A} \leftarrow \emptyset$ 
 $L \leftarrow E$ 
For every  $s \in S$  and for every  $e = (u, v) \in T_s$  do
   $w_{s,e} \leftarrow c_{s,u} + c_{s,v}$ 
While  $L \neq \emptyset$  do
  Let  $(s', e')$  be the pair with minimal  $w_{s,e}$  value
   $\mathcal{A} \leftarrow \mathcal{A} \cup \{(s', u'), (s', v') | e' = (u', v')\}$ 
   $L \leftarrow L - \{e'\}$ 
  For every  $\tilde{e} = (x, y) \in T_{s'} \wedge y \in \{u', v'\}$  do
     $w_{s',\tilde{e}} \leftarrow c_{s',x}$ 
  End For
End While
Return  $\mathcal{A}$ 

```

Algoritma diatas dimulai dengan melakukan penugasan terhadap *node* yang paling dekat dengan *monitoring station*. Kemudian penugasan akan dilakukan terhadap *link* yang berdekatan dengan *node* sebelumnya. Hal ini dilakukan untuk menghindari situasi dimana terdapat *node* yang ditangani oleh dua *monitoring station* yang berbeda.



Gambar 4 r sebagai *monitoring station* melakukan penugasan

4. Hasil Percobaan

Pada bagian ini akan ditampilkan hasil percobaan untuk mengetahui jumlah *monitoring station* yang efisien dalam melakukan pemantauan jaringan dengan menggunakan algoritma *greedy*. Percobaan dilakukan berdasarkan topologi jaringan yang dibangkitkan menggunakan *Waxman Model* [6]. Model ini merupakan model yang sedang populer digunakan oleh para peneliti jaringan. Topologi ini dibangkitkan berdasarkan tiga parameter berikut :

- n, Jumlah dari *node* pada topologi jaringan komputer
- α , Parameter yang mengatur hubungan antara sisi yang pendek pada jaringan.
- β , Parameter yang mengatur derajat dari sebuah *node*.

Tabel 1 Hasil Percobaan Penggunaan Algoritma Greedy Untuk Memonitor Jaringan

JUMLAH SISI	JUMLAH MONITOR	JUMLAH PESAN PROBES
2200	10	2275
5500	24	5650
8947	41	9150

Untuk melakukan perhitungan terhadap jumlah *monitoring station* yang dibutuhkan, digunakan algoritma *greedy* yang telah dijelaskan di Bab 3. *Monitoring station* ini nantinya akan melakukan perhitungan terhadap jumlah pesan yang harus dikirimkan terhadap *node* yang ada. Kemudian hasil yang didapat akan digunakan untuk proses penugasan *probe* tersebut. Hasil percobaan ini akan menampilkan jumlah *monitoring station* yang diperlukan untuk melakukan pemantauan terhadap jaringan dalam skala besar. Dengan menggunakan *node* sebanyak 1000 buah dan jumlah *link* antar *station* yang berbeda-beda.

5. KESIMPULAN

Makalah ini melakukan kajian untuk pemantauan jaringan berbasis *Internet Protokol* dengan pendekatan algoritma *greedy*. Kajian yang diterapkan menggunakan skema *two-phased monitoring* yang memastikan setiap *link* dan rute pada jaringan terjangkau dan juga meminimalisasi penggunaan *resource* yang berlebihan pada saat ingin melakukan pemantauan terhadap jaringan. Pada fase pertama dilakukan komputasi terhadap *node* yang akan melakukan pemantauan (*monitoring station*) dengan menggunakan algoritma *greedy*. Kemudian pada fase kedua komputasi dilakukan untuk menghitung jumlah pesan minimal yang harus dikirimkan oleh setiap *monitoring station* agar tidak terdapat tenggang waktu yang berjumlah besar. Dengan menggunakan pendekatan algoritma *greedy* dapat ditemukan hasil yang cukup mendekati optimal, baik untuk permasalahan pencarian *monitoring station* maupun penugasan terhadap setiap *probe* jaringan.

REFERENSI

- [1] R. L. Carter and M. E. Crovella. 1997. "Server Selection Using Dynamic Path Characterization in Wide-Area Networks", In Proceedings of IEEE INFOCOM'99. Kobe, Japan.
- [2] Stalling Williams. 1999. "SNMP, SNMPv2, SNMPv3, and RMON 1 and 2", (Third Edition). Addison-Wesley Longman Inc.
- [3] Munir Rinaldi. 2005. Diktat Kuliah IF2251 Strategi Algoritmik, Bandung.
- [4] S. R. Stevens. 1994. "TCP/IP illustrated". Addison-Wesley Publishing Company.
- [5] Bejerano Yigal, Rastogi Rajeev. "A Greedy Scheme For Designing Delay Monitoring System of IP Network". Yahoo-Inc. USA.
- [6] B. M. Waxman. 1988. "Routing of Multipoint Connections". IEEE Journal on Selected Areas in Communications, 6(9):1617-1622.