

# SIMULASI ALGORITMA DIJKSTRA PADA PROTOKOL ROUTING OPEN SHORTEST PATH FIRST

Eman Suherman

Agung Budi Prasetijo,ST,MIT

Ir.Sudjadi,MT

## ABSTRAK

*Perkembangan teknologi perutean data pada jaringan komputer pada dasarnya bertujuan untuk mengoptimalkan kemampuan memilih rute terbaik pada sekumpulan jaringan komputer. Teknologi perutean data yang digunakan di Internet saat ini adalah protokol routing Open Shortest Path First.*

*Jenis algoritma routing yang digunakan protokol routing open shortest path first adalah routing keadaan link, dimana routing keadaan link membentuk peta jaringan dalam tiga tahap. Tahap pertama setiap router mengenali seluruh tetangganya. Tahap berikutnya, router-router saling bertukar informasi keadaan link, dan tahap terakhir setiap router menghitung jarak terpendek ke setiap tujuan menggunakan algoritma Dijkstra*

*Tugas akhir ini mensimulasikan algoritma Dijkstra dalam menentukan (menghitung) rute terpendek ke setiap host tujuan dalam suatu topologi jaringan.*

*Dalam menentukan (menghitung) rute terpendek ke setiap host tujuan dalam suatu topologi jaringan, hasil simulasi dibandingkan dengan algoritma Dijkstra secara teori.*

## 1. Pendahuluan

Kebutuhan untuk memperoleh dan mengirim informasi dengan cepat dan mudah telah memacu perkembangan teknologi telekomunikasi. Berbagai macam jaringan telekomunikasi yang menghubungkan berbagai perangkat komunikasi telah dirancang dan diimplementasikan untuk memenuhi kebutuhan akan komunikasi yang cepat dan efisien.

Salah satu teknologi tersebut adalah teknologi jaringan komputer dan internet. Dalam dunia komunikasi data komputer, dikenal istilah protokol yang mengatur bagaimana sebuah komputer berkomunikasi dengan komputer yang lain TCP/IP (Transmission Control Protocol/Internet Protocol) adalah sekelompok protokol yang mengatur komunikasi data di Internet. Komputer-komputer yang terhubung ke Internet berkomunikasi dengan protokol ini. Protokol TCP/IP terdiri dari beberapa protokol yang masing-masing bertanggung jawab atas bagian tertentu dari komunikasi data.

Untuk menghubungkan suatu komputer dengan komputer lain, suatu node harus mencari jalur komunikasi yang tepat dari sejumlah jalur yang ada. Proses pencarian jalur komunikasi dikenal dengan istilah routing sedangkan jalur komunikasi yang diperoleh disebut rute. Protokol yang mengatur proses pencarian jalur komunikasi atau routing ini disebut protokol routing. Dengan adanya protokol routing, penyampaian data dari satu komputer ke komputer lain menjadi cepat dan efisien.

Salah satu protokol *routing* yang digunakan pada jaringan TCP/IP saat ini adalah protokol routing *Open Shortest Path First* (OSPF). Protokol *routing* ini menggunakan algoritma *Dijkstra* dalam menentukan rute terpendek pada proses pencarian jalur komunikasinya

### 1.1 Tujuan

Tujuan dari tugas akhir ini adalah mempelajari penentuan rute terpendek algoritma *Dijkstra* pada protokol routing *open shortest path first*.

### 1.2 Batasan Masalah

- 1 Menentukan rute terpendek pada suatu topologi jaringan menggunakan algoritma *Dijkstra*
- 2 Simulasi dibuat berdasarkan kode sumber terbuka dibuat oleh Carla Laffra, Maret 1996, diperoleh dari internet dengan alamat web site: <http://carbon.cudenver.edu/~hgreenbe/sessions/dijkstra/DijkstraApplet.html>
- 3 Tidak mensimulasikan proses kesepakatan semua router tentang informasi rute yang terpendek (*konvergen*) dan kemampuan beradaptasi terhadap perubahan kondisi jaringan.

## 2. Perancangan Simulasi

### 2.1 Algoritma Dijkstra

Prinsip kerja *algoritma Dijkstra* dapat dijelaskan sebagai berikut

```
Procedure SPF;  
begin  
(1) S:={1};  
(2) for i := 2 to n do  
(3)   D[i]:= C[1,j]; {inisialisai D}  
(4) for i := 1 to n-1 do begin  
(5) Cari vertex w dalam V - S yang  
    mempunyai D[w] minimum  
(6)   Tambahkan w ke S;  
(7)   for tiap vertex v di V - S do  
(8)     D[v]:= min(D[v], D[w] + C[w,v])  
    end  
end;  
end;
```

Misalkan sebuah simpul sumber s di sebuah jaringan, ingin memperoleh jarak yang tersingkat dan termurah ke simpul lainnya yang ada di jaringan tersebut, misalnya ke simpul i. Label jarak : d(i) menunjukkan jarak antara simpul s dan simpul i. Simpul-simpul ini dikelompokkan kedalam dua jenis simpul, yaitu simpul dengan label jarak permanen dan simpul dengan label jarak sementara. Kemudian mencari nilai biaya (*cost*) yang paling kecil diantara simpul-simpul yang terhubung dan menjadikannya label permanen. Dan untuk mencari jalur terpendek berikutnya yang masih belum menjadi label permanen dengan membandingkan nilai biaya kumulatif langsung menuju node tersebut atau dengan melalui node yang telah memiliki label permanen jalur terpendek. Langkah berhenti bila semua label sudah merupakan label permanen

### 2.2 Perancangan Perangkat Lunak Simulasi Algoritma Dijkstra

Pada Tugas Akhir ini penulis membuat perangkat lunak untuk memberikan gambaran simulasi algoritma penentuan jalur terpendek. Algoritma ini digunakan pada protokol routing *Open Shortest Path First* (OSPF) di jaringan TCP/IP untuk melakukan pencarian jalur terbaik dari sistem awal ke sistem tujuan. Algoritma ini pertama kali dikenalkan oleh Edsger W. Dijkstra pada tahun 1959, oleh karenanya dikenal juga dengan sebutan "*Algoritma Dijkstra*".<sup>[3,5,6]</sup>

Proses pada routing *Open Shortest Path First* (OSPF) dilakukan dalam tiga langkah; proses *adjacency*, proses *pembanjiran (flooding)* dan penghitungan tabel routing. Pada program simulasi ini proses *adjacency*, dan proses *pembanjiran* tidak diikutsertakan. Yang dilakukan pada program ini

adalah pada penghitungan tabel routing, mensimulasikan penentuan rute terbaik tiap router berdasarkan topologi jaringan yang ada.

Program dibuat menggunakan JavaApplet dengan bahasa pemrograman JAVA.

Untuk mempermudah dalam mendapatkan gambaran keluaran topologi jaringannya, penulis menetapkan maksimal jumlah node yang akan disimulasikan sebanyak 20 node. Dan besar Biaya (*cost*) node satu ke node lainnya antara 0 sampai dengan 100.

Ada 2 masukan program yang akan disimulasikan pada perangkat lunak ini, yaitu:

1. Masukan topologi jaringan yang digambarkan sendiri oleh pengguna simulasi
2. Dari perangkat lunak ini juga diharapkan menyediakan contoh topologi jaringan.

Pada proses perhitungan jalur terpendek, perangkat lunak akan mensimulasikan dengan 2 cara, yaitu:

1. Algoritma Dijkstra langsung dijalankan
2. Algoritma Dijkstra dijalankan per langkah.

Perangkat lunak akan menampilkan setiap perubahan dari masukan topologi jaringan, dan juga akan menampilkan keterangan yang merupakan penjelasan dari perubahan topologi jaringan masukan.

### 2.3 Flowchart Perangkat Lunak Simulasi Algoritma Dijkstra

Algoritma *Dijkstra* diterapkan pada Grafik satu arah (*directed graphs*) dan biaya (*cost*) jalur yang non-negatif, artinya algoritma akan bekerja dengan benar jika biaya (*cost*) lebih besar dari nol

Proses perhitungan jalur terpendek menggunakan algoritma *Dijkstra* diilustrasikan oleh diagram alir pada Gambar 2.1

## 3. Implementasi Simulasi

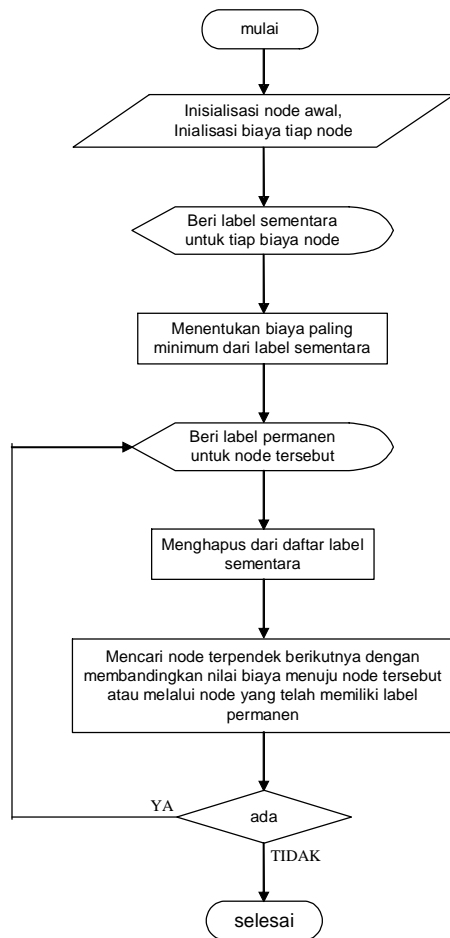
Perangkat lunak simulasi algoritma *Dijkstra* diimplementasikan dengan tahapan sebagai berikut:

1. Membuat daftar program menggunakan perangkat lunak pengolah kata EditPlus versi 2.11 dan disimpan dengan ekstensi file \*.Java. Dalam Tugas Akhir ini Penulis

menyimpannya dengan nama file AlgoritmaSPF.java

2. Mengkompilasi file GraphAlgorithm.java menggunakan perangkat lunak Java Development Kit versi 1.4.2. Proses kompilasi ini menghasilkan file-file class, yaitu:

- AlgoritmaSPF.class
- Judul.class
- Pilihan.class
- Keterangan.class
- InfoSimulasi.class
- PilihanKeterangan.class
- TeksKeterangan.class
- TampilanGrafik.class



Gambar 2.1 Diagram alir Algoritma Dijkstra

3. Membuat file HTML, penulis membuatnya dengan nama file SPFApplet.htm dengan TAG berikut:

```

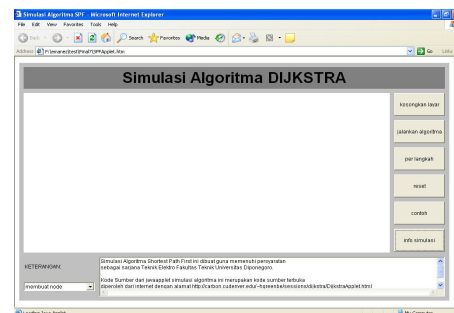
<title>Simulasi Algoritma SPF</title>
<applet code="AlgoritmaSPF.class" width="100%"
height="100%">
</applet>
</center>
  
```

### 3.1 Tampilan Utama Simulasi

Tampilan utama simulasi diimplementasikan oleh AlgoritmaSPF.class, merupakan panel class BorderLayout yang berisi empat buah komponen yaitu:

- TampilanGrafik.class; pada bagian tengah (*center*) panel, pada bagian ini menampilkan simulasi algoritma dijkstra
- Keterangan.class; pada bagian selatan (*south*) panel, sebelah bawah tampilan. Keterangan.class menampilkan semua dokumentasi tata cara menjalankan simulasi algoritma dijkstra. Pada bagian ini juga menampilkan keterangan atau penjelasan untuk mempermudah memahami algoritma dijkstra ketika simulasi berjalan
- Judul.class; pada bagian utara (*north*) panel, sebelah atas tampilan. Judul.class menampilkan label simulasi algoritma dijkstra
- Pilihan.class; pada bagian timur (*east*), sebelah kanan tampilan. Pilihan.class menampilkan tombol-tombol utama untuk menjalankan simulasi algoritma dijkstra

Tampilan utama simulasi ditunjukkan pada Gambar 3.1



Gambar 3.1 Tampilan utama simulasi

### 3.2 Dokumentasi Simulasi (Keterangan.class)

Dokumentasi simulasi direpresentasikan pada class keterangan dikompilasi dan disimpan dengan nama file Keterangan.class; Merupakan BorderLayout baru yang diletakkan pada bagian selatan

(*south*) tampilan, sebelah bawah tampilan. Keterangan.class terdiri dari:

1. PilihanKeterangan.class; menampilkan pilihan dokumentasi penggunaan simulasi algoritma Dijkstra
2. TeksKeterangan.class; merupakan masukan teks keterangan PilihanKeterangan.class yang akan di tampilkan pada area teks. TeksKeterangan.class ini akan menampilkan keterangan sesuai dengan pilihan dokumentasi yang diinginkan pengguna
3. InfoSimulasi.class; menampilkan identitas pembuat simulasi.

### 3.3 Pilihan Simulasi (Pilihan.class)

Pilihan simulasi direpresentasikan pada class pilihan dikompilasi dan disimpan dengan nama file Pilihan.class; Merupakan GridLayout baru yang diletakkan pada Border Layout bagian timur (*east*) tampilan, sebelah kanan tampilan.

Metode GridLayout membagi area layar menjadi: 6 baris dan 1 kolom dengan jarak 10 antar baris. Masing-masing ditempati 6 buah tombol yang digunakan untuk menjalankan simulasi, yaitu:

1. Tombol [kosongkan layar]
2. Tombol [jalankan algoritma]
3. Tombol [per langkah]
4. Tombol [reset]
5. Tombol [contoh]
6. Tombol [info simulasi].

Tombol-tombol ini merupakan objek bagi pengguna untuk berinteraksi dengan aplikasi melalui antarmuka MouseListener yang akan menangkap kejadian yang ditimbulkan oleh mouse.

Fungsi masing-masing tombol dijelaskan sebagai berikut:

1. Tombol [kosongkan layar]

Jika tombol [kosongkan layar] diklik, program akan mengosongkan tampilan grafik pada layar, menampilkan semua keterangan pada daerah teks keterangan dan memberi label per langkah pada tombol b3.

2. Tombol [jalankan algoritma]

Algoritma *Dijkstra* dijalankan melalui tombol ini. Masukan node-node, hubungannya dan besar biaya tiap node ditampilkan pada layer tampilan grafik. Algoritma dijalankan secara otomatis dengan selang waktu 2 detik per langkah.:

3. Tombol [contoh]

Menampilkan contoh node-node, hubungannya dan besar biaya tiap node yang akan dicari jalur terpendeknya menggunakan algoritma *dijkstra*.

4. Tombol [per langkah]

Tombol [per langkah] mempunyai fungsi yang sama dengan tombol [jalankan algoritma], perbedaannya algoritma *dijkstra* dijalankan per

langkah sehingga pengguna dapat mengetahui proses pencarian rute terpendek pada simulasi ini.

5. Tombol [reset]

Jika tombol [reset] diklik, program akan mereset hasil tampilan grafik pada layar, menampilkan semua keterangan pada daerah teks keterangan dan memberi label per langkah pada tombol b3.

6. Tombol [info simulasi].

Menampilkan teks info pembuat simulasi pada area teks keterangan.

### 4. Pengujian Pencarian Rute terbaik Menggunakan Aplikasi Simulasi Algoritma Dijkstra

Aplikasi simulasi algoritma *dijkstra* dijalankan dengan memanggil file SPFApplet.html melalui perangkat lunak browser. Penulis menjalankannya menggunakan perangkat lunak internet explorer 5.5 pada Microsoft Windows XP Professional Edition.

Tampilan utama ketika pertama kali file SPFApplet.html dijalankan ditunjukkan pada Gambar 3.1:

Pengguna simulasi dapat langsung menjalankan simulasi berdasarkan petunjuk pilihan keterangan terdapat pada bagian bawah tampilan. Pilihan keterangan ini dihasilkan oleh PilihanKeterangan.class. Daftar pilihan keterangan ini merupakan link untuk menampilkan informasi yang lebih detail pada daerah teks keterangan.

Tombol-tombol pilihan dihasilkan oleh Pilihan.class. Tombol-tombol pilihan ini merupakan menu utama untuk menjalankan simulasi yang akan ditampilkan pada panel

Contoh node-node, hubungannya dan besar biaya tiap node yang akan dicari jalur terpendeknya ditampilkan pada Gambar 3.1: Untuk mengetahui perubahan biaya yang terjadi pada tiap node, simulasi dijalankan per langkah

### 5 Kesimpulan

Dari pembahasan tugas akhir yang berjudul Simulasi Algoritma *Dijkstra*, Pada Protokol Routing *Open Shortest Path First* maka dapat diambil kesimpulan:

1. Perangkat lunak simulasi algoritma *Dijkstra* ini mampu mensimulasikan penentuan rute terbaik tiap node berdasarkan topologi jaringan yang ada.

2. Dengan masukan jumlah *hop* topologi jaringan dan biaya masing-masing rute pada topologi jaringan yang diberikan, algoritma *dijkstra* optimal menentukan rute terpendek tiap-tiap node pada topologi jaringan tersebut.
3. Perangkat lunak simulasi algoritma *Dijkstra* ini tidak mensimulasikan proses kesepakatan semua router tentang informasi rute yang terpendek (*konvergen*) dan kemampuan beradaptasi terhadap perubahan kondisi jaringan.
13. ..., *Cisco – OSPF Design Guide*, URL: <http://www.cisco.com>, 15 Desember 2002
14. ..., *Documentasi Java*, URL: <http://java.sun.com/j2se/1.4.2/download.html#docs>, 15 Desember 2002
15. ..., *Idkf on Bogor.Net Website*, URL: <http://www.bogor.net/idkf/idkf-1/network>, 15 Desember 2002.

## DAFTAR PUSTAKA

1. Alfred. V. Aho, John E. Hopcroft, Jeffrey D. Ullman., *Data Structures and Algorithms*, Addison-Wesley Publishing Company, 1995.
2. Andrew S. Tanenbaum, *Jaringan Komputer Edisi Bahasa Indonesia dari Computer Networks 3e*, Jilid 1, Prenhallindo, Jakarta, 1997.
3. Andrew S. Tanenbaum, *Jaringan Komputer Edisi Bahasa Indonesia dari Computer Networks 3e*, Jilid 2, Prenhallindo, Jakarta, 1997
4. DC Green, *Komunikasi Data*, Cetakan Kedua, Penerbit ANDI, Yogyakarta, 1998.
5. Drew Heywood, *Konsep dan penerapan Microsoft TCP/IP*, Penerbit ANDI, Yogyakarta, 2001.
6. Hunt. Craig, *TCP/IP Network Administration Second Edition*, O'Reilly & Associates, USA 1997.
7. Isak Riyanto, ST, *Dasar Pemrograman Berorientasi Objek dengan JAVA2 (JDK1.4)*, Penerbit ANDI, Yogyakarta, 2002
8. Onno W. Purbo, *TCP/IP Standar, Desain, dan Implementasi*, Cetakan Keenam, Elex Media Komputindo, Jakarta, 2001
9. Patrick Naughton, *JAVA HANDBOOK Konsep Dasar Pemrograman Java*, Penerbit ANDI, Yogyakarta, 2002.
10. Rangsang Purnama, *Tuntunan Pemrograman JAVA*, Jilid 1, Cetakan Pertama, Prestasi Pustaka Publisher, Jakarta, 2003.
11. Rangsang Purnama, *Tuntunan Pemrograman JAVA*, Jilid 2, Cetakan Pertama, Prestasi Pustaka Publisher, Jakarta, 2003.
12. Sidnie Feit, *TCP/IP Architecture, Protocols, and Implementation*, McGRAW-HILL International Edition, USA, 1993.