

# Aplikasi Graf Pada Topologi Sort

Yulianti Oenang 13510059  
Program Studi Teknik Informatika  
Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia  
yulianti@students.itb.ac.id

*Abstrak—Makalah ini berisi tentang aplikasi graf pada topologi sort, yang biasanya berguna untuk mengurutkan berbagai macam hal berdasarkan prioritasnya. Didalam makalah ini juga akan dijelaskan bagaimana kita mengurutkan berbagai macam hal tersebut berdasarkan prioritasnya dengan menggunakan graf*

**Kata kunci—DAG, Graf, topologi sort, penjadwalan**

## I. PENDAHULUAN

Aplikasi dari graf sangat banyak kita temui dalam kehidupan sehari-hari. Antara lain penggunaan graf dalam bidang pengurutan misalnya: proses pemilihan mata kuliah yang membutuhkan syarat jika ingin mengambil mata kuliah tertentu, atau pada games, banyak sekali kita temui game-game yang membutuhkan syarat-syarat untuk menyelesaikan sebuah game. Misalnya untuk menyelesaikan sebuah game, kita perlu mendapatkan bebek emas. Bebek emas didapat setelah kita memasuki ruang raja. Untuk memasuki ruang raja, kita memerlukan kunci. Kunci tersebut ada di sebuah desa. Aplikasi lainnya adalah untuk mengecek apakah terjadi deadlocking. Untuk mengatur kerja dalam computer, evaluasi formula dalam spreadsheet. Oleh karena itu saya tertarik untuk membuat tulisan tentang itu. Dalam makalah ini, aplikasi topologi sort yang ingin dibahas adalah dalam bidang penjadwalan terutama dalam sebuah proyek yang biasa digunakan pada sebuah program yang bernama PERT. Selain PERT, masih ada program lain yang dapat melakukan penjadwalan juga yaitu CPM (Critical Path Method).

### 1.1 GRAF

Graf digunakan untuk merepresentasikan objek-objek diskrit dan hubungan antara objek-objek tersebut. Representasi visual dari graf adalah dengan menyatakan objek sebagai noktah, bulatan atau titik. Sedangkan hubungan antarobjek dinyatakan dengan garis. Secara matematis, graf didefinisikan sebagai pasangan himpunan  $(V,E)$  yang dalam hal ini:

$V =$  Kumpulan titik kosong dari simpul-simpul (*vertices* atau *node*)  $=\{v_1, v_2, \dots, v_n\}$

$E =$  himpunan sisi (edges atau arcs) yang menghubungkan sepasang simpul  $=\{e_1, e_2, \dots, e_n\}$

Dapat ditulis dengan notasi  $G=(V,E)$ .

Berdasarkan orientasi arahnya, graf dapat dibagi menjadi:

1. Graf tak-berarah (undirected graf)  
Graf yang sisinya tidak mempunyai orientasi arah disebut graf tak berarah.
2. Graf berarah (directed graf)  
Graf yang setiap sisinya diberikan orientasi arah disebut graf berarah. Sisi yang berarah disebut busur. Untuk busur  $(V_a, V_b)$ , simpul  $V_a$  disebut simpul asal sedangkan  $V_b$  disebut simpul terminal.

Istilah-istilah yang digunakan pada graf adalah:

1. Bertetangga (Adjacent)  
Dua simpul dikatakan bertetangga jika keduanya terhubung langsung dengan sebuah sisi.
2. Bersisian (Incident)  
Untuk sembarang sisi  $e=(V_j, V_k)$ , sisi  $e$  dikatakan bersisian dengan simpul  $V_j$  dan simpul  $V_k$ .

### 1.2 TOPOLOGI SORT

Topologi sort adalah mengurutkan objek-objek tertentu berdasarkan prioritasnya.

Contohnya: A-B-C-D

Berarti A harus dikerjakan lebih dahulu baru B dapat dikerjakan. C baru dapat selesai jika B telah selesai dikerjakan. Jika C selesai dikerjakan barulah D dapat dikerjakan.

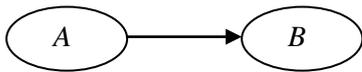
## II. CARA MENGURUTKAN TOPOLOGI SORT

Sangat pusing jika kita harus mengurutkan disaat yang harus diurutkan banyak. Misalnya:

1. A harus dikerjakan setelah D dikerjakan
2. C baru dapat dikerjakan jika A dikerjakan
3. B dikerjakan jika A dikerjakan
4. E dikerjakan jika D dan B selesai dikerjakan
5. F dikerjakan jika A dan C selesai dikerjakan.

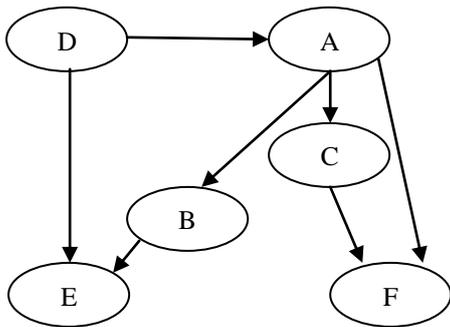
Untuk memudahkan persoalan diatas kita menggunakan graf untuk mempermudah penyelesaian persoalan diatas.

Caranya adalah membuat graf. Simpul menyatakan kegiatannya sedangkan sisinya menggambarkan prioritasnya. Misal



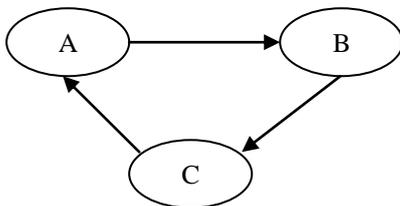
**Gambar 2-1**

Gambar diatas menunjukkan bila A harus dikerjakan sebelum B dikerjakan. Untuk contoh persoalan diatas, maka gambar grafnya adalah:



**Gambar 2-2**

Perlu diperhatikan untuk memecahkan persoalan pengurutan, kita memerlukan DAG (Directed Acyclic Graf), artinya graf yang kita gambarkan haruslah graf yang berarah dan tidak sirkuler. Kenapa tidak boleh sirkuler? Perhatikan contoh berikut:



**Gambar 2-3**

Jika gambar grafnya seperti itu, maka langkah apa yang harus kita kerjakan terlebih dahulu? A? Tidak bisa, karena A harus dikerjakan sebelum C. C? Tidak bisa, karena harus dikerjakan sebelum B. Begitu seterusnya, sehingga kita tidak dapat menentukan A atau B atau C yang terlebih dahulu dapat kita kerjakan. Oleh karena itu, dalam persoalan Topologi Sort kita harus menggunakan DAG.

Kembali ke gambar 1.2-2, persoalannya adalah apa yang harus kita lakukan terlebih dahulu jika kita melihat gambar seperti itu. Cara termudah adalah dengan melihat simpul mana yang tidak memiliki arah panah kedalamnya, alias simpul mana yang tidak memerlukan simpul lain untuk dikerjakan. Dari gambar kita mempunyai satu pilihan, yaitu dari simpul D.

Yang harus kita lakukan selanjutnya adalah dengan mengambil simpul D, lalu memutuskan hubungan dengan simpul-simpul yang bertetangga dengannya. Pada persoalan ini, kita memutuskan hubungan D ke A dan D ke E. Selanjutnya kita tinggal mengulangi langkah seperti diatas, yaitu melihat simpul mana yang sudah

tidak memiliki arah panah kedalamnya. Dari persoalan diatas kita hanya memiliki satu alternatif jawaban yakni simpul A. Begitu selanjutnya hingga kita mendapatkan sebuah simpul yang tidak memiliki hubungan sama sekali.

Dari contoh, kita memperoleh beberapa alternatif jawaban yaitu: D-A-B-C-F-E, D-A-B-E-C-F, D-A-C-F-B-E. Dari jawaban tersebut kita dapat menarik kesimpulan bahwa pengurutan dalam topologi sort solusinya tidak tunggal. Tergantung dari pilihan mana yang kita ambil. Namun yang terpenting urutan dari prioritasnya tetap kita jaga.

Dengan graf kita dapat menyelesaikan, persoalan dalam topologi sort.

### III. APLIKASI TOPOLOGI SORT DALAM PERT

PERT adalah singkatan dari *Program Evaluation Review Technique*. PERT digunakan untuk membantu penjadwalan dalam sebuah proyek. Selain untuk penjadwalan, PERT juga dapat digunakan untuk mengatur dan mengkoordinasi bagian-bagian pekerjaan yang ada didalam suatu proyek.

PERT menggunakan 6 langkah dasar yaitu:

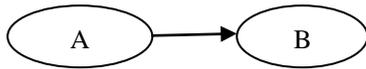
1. Membuat spesifikasi dari proyek yang harus dikerjakan, memecahkan proyek tersebut dalam pecahan pengerjaan.
2. Membuat hubungan antar setiap pekerjaan, dan memutuskan pekerjaan mana yang harus dikerjakan terlebih dahulu dan mana yang harus dikerjakan setelah pekerjaan yang lain dikerjakan.
3. Menggambarkan graf yang menghubungkan setiap pekerjaan dalam proyek tersebut.
4. Menetapkan perkiraan waktu atau biaya untuk setiap pekerjaan
5. Menghitung lintasan terpanjang yang ada dalam graf tersebut. Lintasan ini disebut jalur kritis.
6. Dengan menggunakan graf untuk membantu perencanaan, penjadwalan dan pengendalian proyek.

#### 3.1 Graf pada PERT

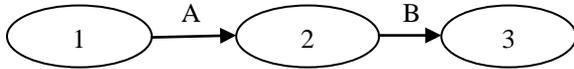
Graf pada PERT menunjukkan hubungan antara satu kegiatan dengan kegiatan yang lainnya.

Ada dua cara untuk menggambarkan graf dalam PERT yaitu dengan AON dan AOA. AON merupakan singkatan dari Activity On Node dan AOA adalah Activity on arrow. Pada konvensi AON, titik menunjukkan kegiatan, sedangkan pada AOA, panah menunjukkan kegiatan.

Misal kita mempunyai kegiatan A dan B, kegiatan B dilakukan setelah kegiatan A dilakukan. Maka representasi AON dan AOA masing-masing adalah:

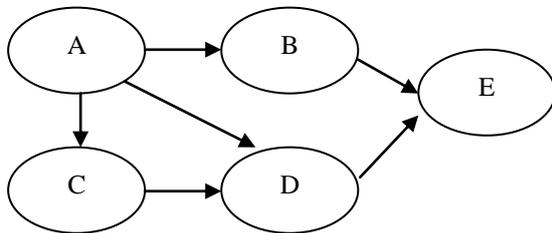


Gambar 3.1-1



Gambar 3.2-2

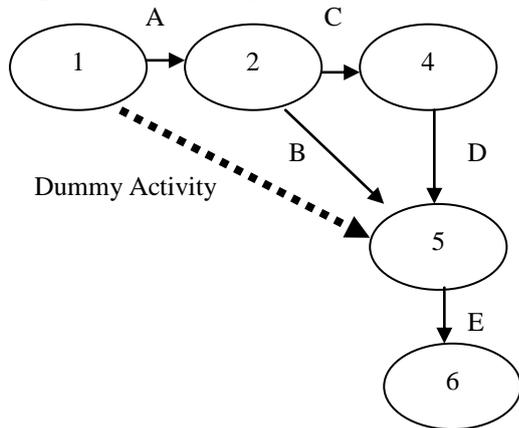
Dengan adanya representasi AOA dalam PERT menyebabkan adanya dummy activity. Dummy activity adalah kegiatan yang tidak harus ada. Contohnya adalah sebagai berikut:



Gambar 3.1-3

Gambar diatas merupakan representasi AON yang berarti. B harus dikerjakan sebelum A, C harus dikerjakan sebelum A, D harus dikerjakan sebelum A dan C, E harus kerjakan sebelum B dan D.

Reperesentasi AOA-nya adalah:



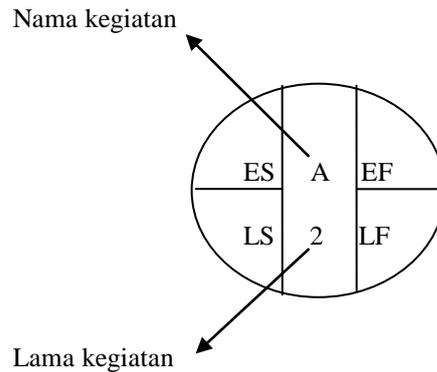
Gambar 3.1-4

### 3.2 Penjadwalan Kegiatan

Untuk menjadwalkan kegiatan-kegiatan yang kita gunakan pada proyek. Kita perlu tahu kapan waktu suatu kegiatan dimulai dan kapan suatu kegiatan selesai dikerjakan. Kita menggunakan proses *two-pass*, terdiri atas *forward pass* dan *backward pass* untuk menentukan jadwal waktu untuk tiap kegiatan. ES (earliest start) dan EF (earliest finish) selama *forward pass*. LS (latest start)

dan LF (latest finish) ditentukan selama *backward pass*.

Representasi fisik dari komponen diatas adalah:



Gambar 3.2-1

Forward pass merupakan identifikasi waktu-waktu sebelumnya. Aturan ES adalah sebagai berikut:

1. Sebelum suatu kegiatan dimulai, maka kegiatan pendahulu langsungnya harus sudah selesai. Misalnya pada gambar 3.1-3 jika B ingin dimulai maka kegiatan A harus sudah selesai
2. Jika suatu kegiatan hanya memiliki sebuah pendahulu, maka  $ES=EF$ . Misalnya gambar 3.1-1, Early start dari B adalah Early Finish dari kegiatan A.
3. Jika suatu kegiatan memiliki sebuah pendahulu, maka ES adalah nilai maksimum dari semua EF pendahulunya. Misalnya: pada gambar 3.1-3, ES untuk E kan bisa dua, yaitu EF dari B atau EF dari D. Kita mengambil EF dari kegiatan dengan EF maksimal. Misalnya EF dari B 8 sedangkan EF dari D adalah 2. Maka ES dari E adalah 8.

Aturan FS adalah:

Waktu FS dari suatu kegiatan adalah jumlah dari ES dan waktu kegiatannya.  $EF=ES+Waktu\ kegiatan$

Backward pass merupakan identifikasi waktu-waktu terlama dari pengerjaan sebuah kegiatan. Aturan menentukan LF sebagai berikut:

1. Jika suatu kegiatan memiliki pendahulu langsung dari suatu kegiatan selanjutnya, maka LF dari kegiatan yang menjadi pendahulu tersebut=LS dari kegiatan yang didahului oleh kegiatan tadi. Misalnya dari gambar 3.1-4. Kegiatan A mendahului kegiatan C, maka LF dari A merupakan LS C.
2. Jika suatu kegiatan adalah pendahulu langsung bagi lebih dari satu kegiatan, maka LF adalah minimum dari seluruh LS dari kegiatan-kegiatan yang secara langsung mengikutinya.

Aturan menentukan LS adalah:

LS=LF-waktu kegiatan

### 3.3 Hambatan Kegiatan(Slack activity) dan Jalur Kritis(critical path)

Waktu slack adalah waktu bebas setiap kegiatan tanpa menimbulkan keterlambatan dari proyek secara keseluruhan

Slack Time atau Total Slack (TS) = LS – ES atau LF – EF

Jalur kritis adalah jalur dengan kegiatan yang tidak mempunyai waktu slack. Kegiatan dengan waktu slack=0 disebut kegiatan kritis. Dengan kata lain jalur kritis adalah jalur dengan waktu pengerjaan terlama.

### 3.4 Kemungkinan Waktu Penyelesaian Aktivitas

Dibagi menjadi tiga yaitu:

1. Waktu optimis(a)
2. Waktu Pesimis(b)
3. Waktu realistik(m)

Dengan waktu yang diharapkan dari pengerjaan sebuah kegiatan adalah

$$T=(a+4m+b)/6$$

Variansi dari waktu tersebut adalah

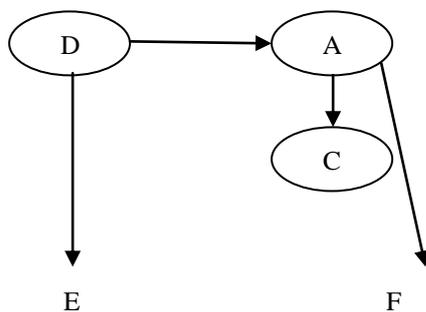
$$V=[(b-a)/6]^2$$

Variansi dari proyek (s) adalah=jumlah dari variansi kegiatan pada jalur kritis.

Nilai deviasi normal (Z) = (batas waktu-waktu penyelesaian yang diharapkan)/s

## IV. ANALISIS KASUS

Misalkan kita mempunyai persoalan sebagai berikut: A,B,C,D,E,F melambangkan kegiatan-kegiatan pada proyek dan keterhubungan antar simpul melambangkan kegiatan mana yang harus dilakukan terlebih dahulu. Pertama-tama kita



**Gambar 4-1**

Misalkan waktu pengerjaan masing-masing kegiatan adalah sebagai berikut:

**Tabel 4.1**

Kegiatan	Waktu(Jam)
A	1

B	3
C	9
D	5
E	2
F	8

Dengan menggunakan topologi sort, kita dapat menentukan pilihan mana saja yang mungkin untuk menyelesaikan problem diatas. Pilihannya adalah D-E-F atau D-A-F atau D-A-C-F. Dengan memasukkan waktu-waktu pengerjaan diatas. Kita dapat mengetahui critical path dari gambar diatas adalah D-A-C-F. Dengan mengetahui critical path kita dapat menentukan peluang sebuah proyek terlaksana jikadiketahui berapa tenggang waktu dari pelaksanaan proyek tersebut.

## V. KESIMPULAN

Banyak sekali penggunaan Graf. Salah satunya adalah dalam memecahkan persoalan topologi sort yang nantinya dapat berguna untuk aplikasi PERT.

Aplikasi PERT digunakan untuk mengurutkan bagaimana proses supaya sebuah proyek dapat maksimal dikerjakan. Namun sayang, PERT ini mempunyai keterbatasan antara lain, kita harus sudah dapat memastikan bagaimana hubungan antar satu kegiatan dengan kegiatan yang lain atau dengan kata lain hubungan tersebut harus stabil. Perkiraan waktu juga cenderung subjektif.

## DAFTAR PUSTAKA

- [1] <http://translate.google.co.id/translate?hl=id&langpair=en|id&u=http://www.scribd.com/doc/50381953/238/Uses-of-Topological-Sort> Tanggal akses 12-12-2011.
- [2] <http://translate.google.co.id/translate?hl=id&sl=en&tl=id&u=http%3A%2F%2Fwww.cs.purdue.edu%2Fhomes%2Fcs251%2FLectureNotes%2FSpring2006%2Fweek12-2%2F&anno=2>. Tanggal akses 12-12-2011.
- [3] [http://translate.google.co.id/translate?hl=id&sl=en&tl=id&u=http%3A%2F%2Fwww.enotes.com%2Ftopic%2FTopological\\_sorting&anno=2](http://translate.google.co.id/translate?hl=id&sl=en&tl=id&u=http%3A%2F%2Fwww.enotes.com%2Ftopic%2FTopological_sorting&anno=2). Tanggal akses 12-12-2011.
- [4] <http://www.cs.princeton.edu/~rs/AlgsDS07/13DirectedGraphs.pdf>. tanggal akses 12-12-2011
- [5] Munir, Rinaldi, "Struktur Diskrit edisi keempat" *Informatika Bandung*. Agustus 2008.
- [6] <http://id.shvoong.com/business-management/management/1698755-perbedaan-cpm-dengan-pert/> tanggal akses 12-12-2011
- [7] [http://www.google.co.id/url?sa=t&rct=j&q=CPM+dan+pert&source=web&cd=1&ved=0CBkQFjAA&url=http%3A%2F%2Fsyukronali.files.wordpress.com%2F2010%2F05%2Fbab-9-cpm-pert.ppt&ei=eizlTqHwGsXJrAe45YvjCg&usq=AFQjCNGy\\_L7dUKeSEMJMZqYvDto4qON0eg&cad=rja](http://www.google.co.id/url?sa=t&rct=j&q=CPM+dan+pert&source=web&cd=1&ved=0CBkQFjAA&url=http%3A%2F%2Fsyukronali.files.wordpress.com%2F2010%2F05%2Fbab-9-cpm-pert.ppt&ei=eizlTqHwGsXJrAe45YvjCg&usq=AFQjCNGy_L7dUKeSEMJMZqYvDto4qON0eg&cad=rja) tanggal akses 12-12-2011

## 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, 12 Desember 2010

A handwritten signature in black ink, consisting of several overlapping loops and a long vertical stroke at the bottom.

Yulianti Oenang 13510059