Penggunaan Algoritma Greedy dan Program Dinamis dalam Pemecahan *Bridge and Torch Problem*

Jeffry / 13516156
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13516156@std.stei.itb.ac.id

***Abstract*—Dalam makalah ini, akan dibahas pemecahan teka-teki *Bridge and Torch Problem*, dimana merupakan variansi dari teka-teki *River Crossing Puzzle*. Untuk memecahkan teka-teki ini, akan coba digunakan dua algoritma pemrograman, yakni algoritma Greedy dan program dinamis. Pada makalah ini, akan dijelaskan dasar teori dan langkah-langkah pemecahan teka-teki tersebut dengan algoritma Greedy dan program dinamis.**

Keywords*— Algoritma Greedy, *Bridge and Torch*, Program Dinamis, *River Crossing Puzzle

I. PENDAHULUAN

River Crossing Puzzle merupakan teka-teki yang memberikan tantangan dimana beberapa orang atau objek perlu menyeberangi sungai, dengan berbagai aturan/ketentuan. Disebutkan bahwa teka-teki pertama kali dicatat dalam buku *Proporsitiones ad acuendos iuvenes* (dalam Bahasa Inggris: "Problems to sharpen the young"), ciptaan Alcuin of York sekitar 800 Masehi.

Bridge and Torch Problem merupakan salah satu variansi dari *River Crossing Puzzle*. Pada variansi ini, sungai direpresentasikan sebagai jembatan, dan beberapa orang harus menyeberangi jembatan itu dalam waktu tertentu secara bergiliran. Untuk menyeberangi jembatan tersebut, diperlukan juga suatu alat penerangan. Pada makalah ini, kasus yang akan digunakan adalah sebagai berikut.

Pada suatu gunung, terdapat suatu laboratorium. Pada suatu malam, kepala laboratorium tersebut secara tidak sengaja melepaskan segerombolan zombie dari tempat penyimpanan laboratorium. Dia perlu melarikan diri bersama asisten lab, penjaga lab, dan professor tua renta dari laboratorium tersebut. Satu-satunya cara untuk meninggalkan gunung itu ialah melalui sebuah jembatan gantung yang sempit. Mereka perlu menyeberangi jembatan itu dalam waktu singkat, dan memotong jembatan itu agar zombie tidak menyerang hingga kota-kota. Untungnya, mereka berhasil sampai ke jembatan terlebih dahulu dibandingkan para zombie.

Gambar 1. Para pekerja lab perlu meninggalkan gunung melalui jembatan, dan sebelum zombie mengejar[4]

Agar mereka dapat melarikan diri dengan selamat, terdapat beberapa kondisi:

- Mereka semua harus berhasil menyeberangi jembatan dalam waktu tidak lebih dari 17 menit
- Kepala laboratorium dapat menyeberangi jembatan dalam 1 menit, asisten laboratorium dapat menyeberangi jembatan dalam 2 menit, penjaga laboratorium dapat menyeberangi jembatan dalam 5 menit, dan professor dapat menyeberangi jembatan dalam 10 menit.
- Hanya 2 orang yang boleh menyeberangi jembatan dalam waktu bersamaan. Selain itu, kedua orang itu akan melewati jembatan selama waktu yang diperlukan orang terlambat di antara kedua orang itu (Contoh, bila kepala laboratorium dan professor menyeberangi jembatan secara bersamaan, maka mereka akan berhasil menyeberangi jembatan setelah 10 menit).
- Untuk menyeberangi jembatan, diperlukan senter untuk menerangi jalan. Akibatnya, setelah sepasang orang telah menyeberangi jembatan, maka harus ada salah seorang yang balik untuk memberikan senter ke pasangan selanjutnya yang akan menyeberang jembatan.

Gambar 2. Waktu menyeberangi jembatan yang diperlukan masing-masing pekerja laboratorium[4]

Dari paparan di atas, kita dapat menyimpulkan bahwa persoalan ini termasuk ke dalam persoalan optimasi, sebab persoalan ini menginginkan waktu minimal yang diperlukan untuk menyeberangkan semua pekerja laboratorium dan juga tidak boleh melebihi 17 menit. Dengan demikian, kita akan mencoba memecahkan persoalan ini dengan menggunakan dua algoritma yang biasa untuk pemecahan persoalan optimasi, yakni Algoritma Greedy dan Program Dinamis.

II. TEORI DASAR

A. *Algoritma Greedy*

Algoritma Greedy adalah algoritma yang biasa digunakan untuk memecahkan persoalan optimasi. Algoritma Greedy mencari solusi langkah per langkah, dan solusi tersebut tidak dapat diubah lagi. Pada setiap langkah, akan :

- Mengambil pilihan yang terbaik yang dapat diperoleh pada saat itu (optimum lokal), tanpa memerdulikan konsekuensi ke depannya.
- Berharap bahwa dengan memilih optimum lokal pada setiap langkah, akan berakhir dengan optimum global

Elemen-elemen yang dimiliki algoritma Greedy antara lain:

1. Himpunan Kandidat
   Berisi elemen-elemen pembentuk solusi
2. Himpunan Solusi
   Berisi kandidat-kandidat yang terpilih sebagai solusi persoalan
3. Fungsi Seleksi
   Untuk memilih kandidat yang paling memungkinkan mencapai solusi optimal
4. Fungsi Kelayakan
   Untuk memeriksa apakah suatu kandidat yang dipilih merupakan solusi yang layak, yakni kandidat tersebut bersama himpunan solusi yang sudah ada tidak melanggar kendala yang ditetapkan.
5. Fungsi Objektif
   Untuk memaksimumkan atau meminimumkan nilai solusi

B. *Program Dinamis*

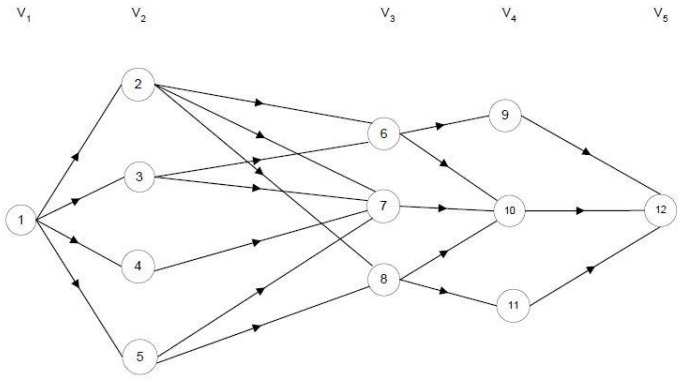
Program Dinamis (*dynamic programming)* adalah suatu metode pemecahan masalah dengan cara menguraikan solusi menjadi beberapa tingkat. Sehingga, solusi dari setiap tingkat dapat dianggap sebagai serangkaian keputusan yang saling bersangkutan.

Dalam menyelesaikan persoalan dengan program dinamis, terdapat beberapa karakterik sebagai berikut

- Terdapat sejumlah pilihan aksi dan solusi yang dapat dilakukan
- Solusi dari masing-masing tingkat dibangun dari solusi dari tingkat sebelumnya
- Menggunakan prinsip optimalisasi dan suatu pembatas untuk membatasi sejumlah pilihan yang harus dipertimbangkan pada suatu tingkat tertentu

Prinsip optimalisasi merupakan suatu prinsip yang menentukan suatu pilihan berdasarkan solusi total. Menurut prinsip ini, bila suatu solusi total optimal, maka solusi pada tingkat ke-n juga akan optimal. Dengan begitu, untuk mencari solusi optimal dari tahap ke-n+1, maka kita dapat menggunakan solusi optimal dari tahap ke-n, tanpa harus mengulang dari awal.

Program dinamis mirip dengan Algoritma Greedy, sebab sama-sama mencari nilai optimum pada setiap langkah. Namun, terdapat perbedaan yang mencolok, yakni pada Algoritma Greedy, hanya memeriksa satu kemungkinan pilihan saja, sehingga memungkinkan solusi optimum lokal yang didapat, bukan merupakan solusi optimum global. Sedangkan, pada program dinamis, akan dilakukan pemeriksaaan pada semua kemungkinan pilihan yang terjadi, sehingga pasti mendapatkan solusi optimum global.



Gambar 3. Grafik program dinamis, yang terdiri atas beberapa tahap dan status[1]

Suatu persoalan dapat diselesaikan dengan menggunakan program dinamis, bila persoalan tersebut memiliki karakteristik-karakteristik berikut ini.

1. Persoalan dapat dibagi menjadi beberapa tahap/tingkat (*stage*), yang pada setiap tahap hanya diambil satu keputusan.
2. Masing-masing tahap terdiri dari sejumlah status (*state*) yang berhubungan dengan tahap tersebut. Secara umum, status merupakan bermacam kemungkinan solusi yang dibangkitkan pada tahap tersebut.
3. Hasil dari keputusan solusi yang diambil dari setiap tahap ditransformasikan dari status yang bersangkutan ke status berikutnya pada tahap selanjutnya.
4. Ongkos (*cost*) pada suatu tahap meningkat seiring bertambahnya jumlah tahapan yang telah dilewati.
5. Ongkos total pada suatu tahap bergantung pada ongkos tahap-tahap yang sudah dilewati dan ongkos yang diperlukan untuk mencapai tahap ini dari tahap sebelumnya.
6. Keputusan terbaik pada suatu tahap bersifat independen terhadap keputusan yang dilakukan pada tahap sebelumnya.
7. Adanya hubungan rekursif yang mengidentifikasikan keputusan terbaik untuk setiap status pada tahap *n*, untuk kemudian keputusan tersebut diberikan ke setiap kasus pada tahap *n*+1.
8. Prinsip optimalitas berlaku pada persolan ini. Ini dikarenakan salah satu tujuan dari program dinamis adalah solusi dari suatu persoalan dapat ditemukan dan merupakan solusi yang optimal.

Misalkan x_1, x_2, x_3,..., x_n menyatakan peubah/variabel keputusan yang harus dibuat masing-masing untuk tahap 1, 2, 3, ..., n. Terdapat 2 macam pendekatan yang digunakan dalam program dinamis, antara lain:

- Program dinamis maju (*forward* atau *up-down*). Program dinamis ini akan bergerak mulai dari tahap 1, lalu maju ke tahap 2, 3, dan seterusnya hingga tahap ke-*n*. Runtutan peubah keputusan adalah x_1, x_2, x_3,..., x_n. Prinsip optimalitas pada program dinamis maju yakni:
  Ongkos pada tahap *k*+1 = (Ongkos yang dihasilkan pada tahap *k*) + (Ongkos dari tahap *k* ke tahap *k*+1)
  k = 1, 2, 3, ..., n-1
- Program dinamis mundur (*backward* atau *bottom-up*). Program dinamis ini akan bergerak mulai dari tahap *n*, lalu mundur ke tahap *n*-1, *n*-2, dan seterusnya hingga tahap 1. Runtutan peubah keputusan adalah x_n, x_{n-1}, x_{n-2}, ... , x_1.
  Prinsip optimalitas pada program dinamis mundur:
  Ongkos pada tahap *k* = (Ongkos yang dihasilkan pada tahap *k*+1) + (Ongkos dari tahap *k*+1 ke tahap *k*)
  k = *n*-1, *n*-2, ..., 1

Langkah-langkah pengembangan algoritma program dinamis untuk suatu persoalan, yakni:

1. Menentukan pendekatan yang akan digunakan untuk mencari solusi optimal, apakah maju atau mundur
2. Definisikan secara rekursif nilai solusi optimal
3. Hitung nilai solusi optimal (maju atau mundur berdasarkan pendekatan program dinamis yang digunakan)
4. Konstruksi solusi optimal, dalam bentuk jalur/aksi-aksi yang menghasilkan nilai solusi optimal.

III. PENYELESAIAN TEKA-TEKI BRIDGE AND TORCH PROBLEM

A. DENGAN ALGORITMA GREEDY

Sebelum memulai pemecahan persoalan perlu dicari terlebih dahulu elemen-elemen algoritma Greedy. Elemen-elemen algoritma Greedy untuk kasus ini adalah:

1. Himpunan Kandidat : himpunan waktu penyeberangan yakni 1, 2, 5, dan 10
2. Himpunan Solusi : himpunan nilai dari waktu pasangan orang menyeberang jembatan (misal (1,2) diambil 2 dan (10,-) diambil 10. – merepresentasikan bahwa hanya satu orang yang menyeberangi jembatan)
3. Fungsi Seleksi : pilih 2 waktu terkecil dari himpunan kandidat yang tersisa. Selain itu, untuk mengembalikan senter, perlu diambil 1 waktu terkecil dari himpunan solusi untuk dikembalikan ke himpuanan kandidat (balik ke tempat semula)
4. Fungsi Layak : memeriksa apakah total dari waktu pada himpunan solusi tidak melebihi 17
5. Fungsi Objektif : jumlah waktu sudah minimal

Pertama, akan diambil (1,2) dikarenakan merupakan 2 waktu terkecil di himpunan kandidat. Karena perlu mengembalikan senter, maka 1 akan kembali ke himpunan kandidat. Lalu, akan diambil (1,5) dikarenakan merupakan 2 waktu terkecil di himpunan kandidat sekarang. Setelah itu, 1 akan kembali ke himpunan kandidat lagi. Sampai langkah ini, total waktu adalah max(1,2) + 1 + max(1,5) + 1 = 9 menit. Untuk mempermudah penjelasan, langkah-langkah pengerjaan algoritma Greedy dapat dilihat pada Gambar 4 pada Bab 4 di bawah ini.

Kemudian, akan diambil (1,10) dikarenakan merupakan 2 anggota terakhir pada himpunan kandidat. Namun, solusi ini tidak layak, dikarenakan total waktu (9 + max(1,10) = 19) telah melebihi batas waktu (17 menit). Dengan demikian, algoritma Greedy gagal untuk memecahkan persoalan ini.

B. DENGAN PROGRAM DINAMIS

Sebelum memulai menghitung nilai solusi optimal, pertama perlu ditentukan terlebih dahulu jenis pendekatan program dinamis yang akan digunakan. Dikarenakan kita perlu mencari urutan pasangan orang yang menyeberang jembatan, maka akan digunakan pendekatan program dinamis maju untuk menentukan urutan pasangan orang yang akan menyeberangi jembatan tersebut.

Setelah menentukan pendekatan program dinamis yang

digunakan, selanjutnya perlu menentukan status dan juga ongkos dari satu tahap ke tahap selanjutnya. Status akan digunakan untuk merepresentasikan orang-orang yang telah sampai di seberang, sehingga status akhir merupakan tujuan kita, yakni semua orang berada di seberang. Dari status ini, kita juga bisa memperoleh informasi orang-orang yang masih belum di seberang, sehingga dapat menentukan pasangan orang yang akan menyeberangi jembatan kemudian.

Ongkos dari satu tahap ke tahap selanjutnya ditentukan berdasarkan maksimum waktu (dalam menit) yang diperlukan untuk menyeberangi jembatan antara A dan B, dan juga waktu orang A/B menyeberang jembatan untuk balik ke tempat semula agar senter dikembalikan. Ongkos dihitung sedemikian rupa agar mengurangi pembangkitan status yang berulang maupun yang tidak berguna. Jadi, ongkos pada satu tahap ke-*k* dihitung dengan rumus sebagai berikut.

Ongkos tahap ke-*k* = (Ongkos yang dihasilkan pada tahap ke *k*-1) + (maks(waktu pindah A, waktu pindah B)) + (Waktu pindah A atau B)

k = 2, 3, 4, ..., *n*

Untuk mencari solusi optimal dari suatu tahap, dapat menggunakan relasi rekurens sebagai berikut.

$f_1(s) = c_{x1s}$ (basis)

$f_k(s) = \min \{c_{xks} + f_{k-1}(x_k)\}$ (rekurens)

Keterangan :

- x_k = peubah keputusan pada tahap k
- c_{xks} = ongkos sisi dari s ke x_k
- $f_k(x_k, s)$ = total ongkos lintasan dari $f_k(x_k, s)$
- $f_k(s)$ = nilai minimum dari $f_k(x_k, s)$

Dari Gambar 5 pada Bab 4, dapat dilihat bahwa untuk kasus ini, terdapat 3 tahap, sehingga kita perlu mencari 3 solusi optimal.

- Tahap 1 :

$f_1(s) = c_{x1s}$

S	Solusi Optimum	
	$f_1(s)$	x_1*
2	3	1
3	6	1
4	11	1

- Tahap 2 :

$f_2(s) = \min \{c_{x2s} + f_1(x_2)\}$

x2 / s	$f_2(x_2, s) = c_{x2s} + f_1(x_2)$			Solusi Optimum	
	2	3	4	$f_2(s)$	x_2*
5	9	9	28	9	2 atau 3
6	14	21	14	14	2 atau 4
7	15	17	17	15	2

- Tahap 3 :

- $f_3(s) = \min \{c_{x3s} + f_2(x_3)\}$

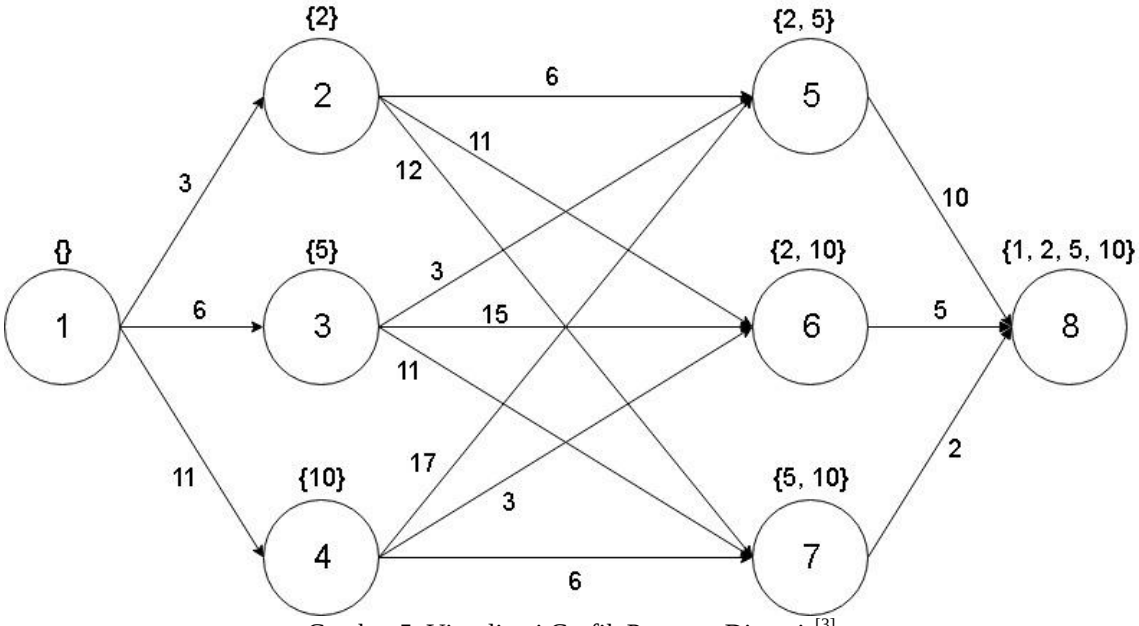
x3 / s	$f_3(x_3, s) = c_{x3s} + f_2(x_3)$			Solusi Optimum	
	5	6	7	$f_3(s)$	x_3*
8	19	19	17	17	7

Dari hasil pengerjaan di atas, dapat dilihat bahwa didapatkan nilai solusi optimal yang sesuai dengan batas waktu yang telah ditentukan, yakni 17 menit. Solusi optimal yang didapatkan yakni 1 → 2 → 7 → 8. Berdasarkan Gambar 5 pada Bab 4, maka skenario untuk mendapatkan solusi optimal adalah sebagai berikut.

1. Kepala laboratorium dan asisten laboratorium terlebih dahulu menyeberang. Setelah itu, kepala laboratorium balik ke tempat semula.
2. Penjaga laboratorium dan profesor menyebereang bersamaan. Setelah sampai, asistem laboratorium akan balik ke tempat semula.
3. Kepala laboratorium dan asisten laboratorium menyeberang jembatan tersebut untuk kedua kalinya. Dengan demikian, maka semua pekerja laboratorium sudah berada di seberang dalam waktu 17 menit.

IV. GAMBAR TAMBAHAN

Gambar 4. Langkah-langkah pengerjaan Algoritma Greedy[3]



Gambar 5. Visualisasi Graf Program Dinamis[3]

V. KESIMPULAN

Dari hasil pengerjaan di atas, dapat disimpulkan bahwa dengan algoritma Greedy tidak dapat dinemukan solusi optimum, sedangkan dengan program dinamis, dapat ditemukan. Ini disebabkan algoritma Greedy yang hanya memeriksa satu kemungkinan pilihan, tidak seperti program dinamis yang mengecek semua skenario yang mungkin terjadi. Dengan demikian, program dinamis merupakan algoritma pemececahan persoalan optimasi yang lebih bagus dari algoritma Greedy.

Selain untuk memecahkan teka-teki, algoritma Greedy dan program dinamis dapat juga digunakan dalam berbagai bidang kehidupan, seperti menentukan jalan terpendek, menghitung keuntungan terbesar, dan menentukan waktu tersingkat dalam mengerjakan beberapa hal sekaligus. Selain algoritma, teka-teki *Bridge and Torch Problem* dapat juga digunakan sebagai materi pengujian untuk suatu bahasa pemrograman, misal untuk mengetes kemampuan pemecahan masalah antar bahasa Haskell dan Prolog[10].

VI. UCAPAN TERIMA KASIH

Pertama-tama, penulis berterima kasih dan bersyukur kepada Tuhan Yang Maha Esa atas berkat-Nya sehingga penulis dapat menyelesaikan makalah ini dengan lancar dan baik. Penulis juga mengucapkan terima kasih kepada kedua orang tau penulis yang selalu memberikan bantuan secara moral, materi, dan juga doa. Penulis turut mengucapkan terima kasih kepada Bapak Dr. Ir. Rinaldi Munir, M.T., Ibu Dr. Masayu Leylia Khodra, dan Ibu Dr. Nur Ulfa Maulidevi, S.T, M.Sc. selaku dosen pengajar mata kuliah IF2211 Strategi Algortima atas segala bimbingan serta segala ilmu yang telah diberikan kepada penulis, terutama dalam mengajarkan materi mengenai logika dan pohon.

REFERENSI

[1] R. Munir, *Bahan Kuliah ke-3*. Bandung: Departemen Teknik Informatika Institut Teknologi Bandung, 2004.
[2] R. Munir, *Bahan Kuliah ke-13*. Bandung: Departemen Teknik Informatika Institut Teknologi Bandung, 2004.
[3] draw.io (keperluan menggambar grafik)
[4] https://www.youtube.com/watch?v=7yDmGnA8Hw0&index=5&list=PLJicmE8fK0EiFRt1Hm5a_7SJFaikIFW30 . Diakses pada 12 Mei 2018
[5] https://ed.ted.com/lessons/can-you-solve-the-bridge-riddle-alex-gendler#digdeeper . Diakses pada 12 Mei 2018
[6] Jack Brimberg, Bill Hurley, (2007) *Solving the U2 Brainteaser with Integer and Dynamic Programming*. INFORMS Transactions on Education 7(3):223-227. https://doi.org/10.1287/ited.7.3.223 diakses pada 13 Mei 2018
[7] http://www-history.mcs.st-and.ac.uk/HistTopics/Alcuin_book.html. Diakses pada 13 Mei 2018
[8] G. Rote (2002). "Crossing the Bridge at Night". *Bulletin of the European Association for Theoretical Computer Science*. 78. pp. 241–246.
[9] T. Sillke (2001). "Crossing the Bridge in an Hour".
[10] M. Erwig (2004). "Escape from Zurg". Journal of Functional Programming, Vol. 14, No. 3. Pp. 253-261.

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, 13 Mei 2018

Jeffry
13516156