

# Penerapan Teori Graf untuk Penyelesaian Permasalahan Playing with Wheels

Tino Eka Krisna Sambora - 13510062  
Program Studi Teknik Informatika  
Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia  
13510062@std.stei.itb.ac.id

**Abstract**— *Playing with wheels* merupakan permasalahan matematis yang cukup terkenal di kalangan para programmer. Permasalahan ini seringkali muncul pada olimpiade pemrograman. Aplikasi dari permasalahan *playing with wheels* pun beragam, mulai dari permasalahan matematis seperti kombinatorial sampai dengan implementasinya pada teknologi seperti elektronika. Pada kesempatan ini, penulis hendak menjabarkan keterlibatan teori graf dalam mentranslasikan permasalahan ini agar dapat ditangani lebih lanjut oleh algoritma-algoritma traversal graf.

**Kata kunci** - *Teori Graf, Traversal Graf, DFS, BFS, Brute Force, A\*, Playing With Wheels.*

## I. PRAKATA

Di kalangan programmer, developer, atau tim olimpiade. Permasalahan *playing with wheels* pasti sudah sering di dengar. Ini merupakan salah satu permasalahan yang seringkali muncul pada olimpiade, dan permasalahan yang menarik untuk dibahas karena ada banyak hal yang dapat dilakukan untuk memperbaiki kemangkusan penyelesaian permasalahan ini.

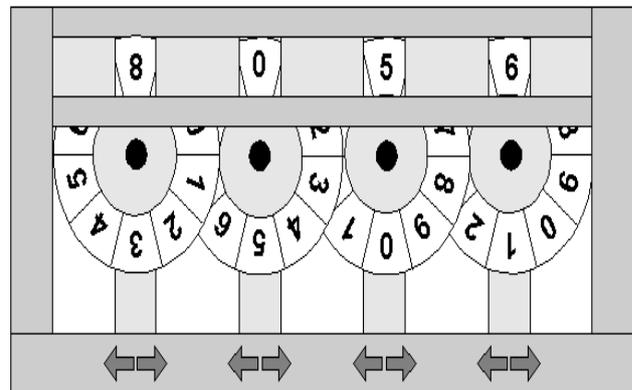
Permasalahan ini umumnya diselesaikan dengan menggunakan abstraksi traversal pada sebuah graf, namun sebelumnya, permasalahan *playing with wheels* tersebut mesti terlebih dahulu ditranslasikan kedalam graf status.

Setelah permasalahan *playing with wheels* ditranslasikan menjadi graf status. Barulah algoritma-algoritma traversal graf seperti DFS, BFS, runut balik, dan branch and bound dapat diterapkan untuk menyelesaikan permasalahan ini.

Pada kesempatan kali ini, penulis hendak memperkenalkan algoritma traversal graf untuk menyimpulkan betapa berperannya teori graf untuk mentranslasikan berbagai permasalahan matematis menjadi bentuk graf status agar dapat ditemukan solusinya dengan algoritma-algoritma traversal graf.

## II. PLAYING WITH WHEELS

Ada empat buah roda, masing-masing memiliki 10 juring yang dinumerasi dari 0 sampai dengan 9. Digit yang paling atas membentuk sebuah 4-bit integer. Sebagai contoh, pada gambar dibawah roda-roda angka membentuk bilangan 8056. Tiap-tiap roda memiliki dua buah tombol yang berasosiasi dengan pergerakan bilangan roda, ke kiri atau ke kanan. menuju bilangan yang lebih kecil atau menuju bilangan yang lebih besar.

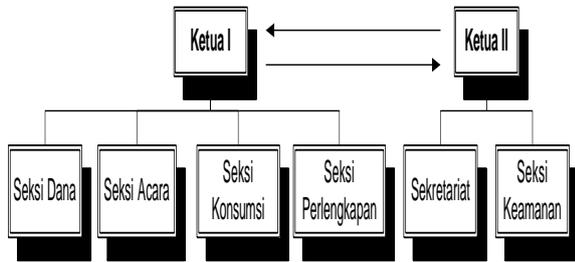


Gambar 1. Ilustrasi *Playing with Wheels 1*

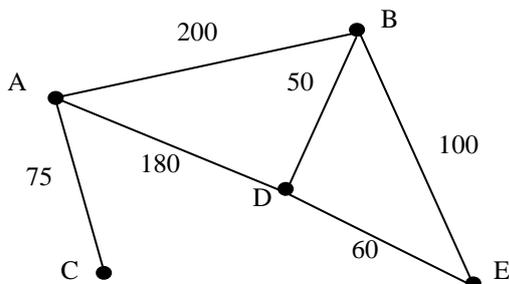
Permainan dimulai dengan konfigurasi awal pada roda-roda. Katakan konfigurasi digit mula-mula membentuk integer  $S_1S_2S_3S_4$ . Lalu anda akan diberikan sejumlah  $n$  konfigurasi yang terlarang (forbidden), katakanlah  $F_1F_2F_3F_4$  ( $1 \leq i \leq n$ ). Lalu pengguna akan diberi konfigurasi target atau tujuan empat digit yang mesti dicapai, katakanlah  $T_1T_2T_3T_4$ . Anda ditantang untuk menulis program yang dapat menghitung langkah minimum dari tombol-tombol yang ditekan yang dibutuhkan untuk mengubah konfigurasi mula-mula menuju konfigurasi target tanpa melalui konfigurasi terlarang.

### III. TEORI GRAF

Graf merupakan suatu diagram yang memuat informasi tertentu jika diinterpretasikan secara tepat. Dalam kehidupan sehari-hari, graf digunakan untuk menggambarkan berbagai macam struktur yang ada. Tujuannya adalah sebagai visualisasi obyek-obyek agar lebih mudah dimengerti. Beberapa contoh graf yang sering dijumpai dalam kehidupan sehari-hari antara lain: struktur organisasi, bagan alir pengambilan mata kuliah, peta, rangkaian listrik, dan lain-lain. Graf struktur sebuah organisasi dan peta beberapa daerah tampak pada Gambar 6 dan Gambar 7.



Gambar 6. Ilustrasi Sebuah Graf



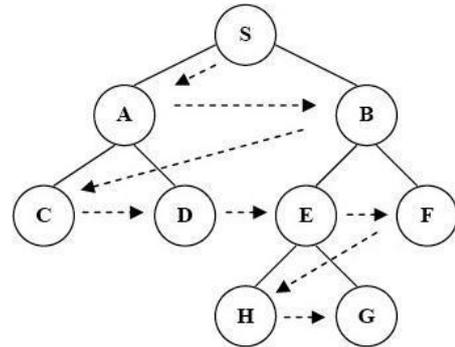
Gambar 7. Ilustrasi Sebuah Graf

Setiap diagram memuat obyek-obyek serta garis-garis yang menghubungkan obyek-obyek tersebut. Garis bisa berarah ataupun tidak berarah. Garis yang berarah biasanya digunakan untuk menyatakan hubungan yang mementingkan urutan antar objek-objek. Urut-urutan objek akan mempunyai arti yang lain jika arah garis diubah. Sebagai contoh adalah garis komando yang menghubungkan titik-titik struktur sebuah organisasi. Sebaliknya, garis yang tidak berarah digunakan untuk menyatakan hubungan antar objek-objek yang tidak mementingkan urutan. Sebagai contoh adalah garis untuk menyatakan jarak hubung 2 kota pada Gambar 2. Jarak dari kota A ke kota B sejauh 200 km akan sama dengan jarak dari kota B ke kota A. Apabila

jarak 2 tempat tidak sama jika dibalik (misalnya karena harus melalui jalan memutar), maka garis yang digunakan haruslah garis yang berarah

### IV. BREADTH FIRST SEARCH

Breadth-First Search (BFS) atau pencarian melebar merupakan salah satu metoda traversal di dalam graf yang mengunjungi simpul-simpul di dalam graf secara sistematis. Pada metode Breadth-First Search, semua node pada tingkat n akan dikunjungi terlebih dahulu sebelum mengunjungi node-node pada level n+1. Pencarian dimulai dari node akar terus ke level ke-1 dari kiri ke kanan, kemudian berpindah ke level berikutnya demikian pula dari kiri ke kanan sampai solusi persoalan yang direpresentasikan oleh graf tersebut ditemukan. Keuntungan BFS adalah tidak akan menemui jalan buntu. Namun, BFS membutuhkan banyak memori karena menyimpan semua node dalam satu pohon.



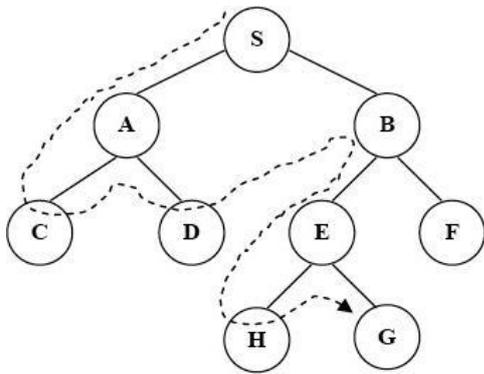
Gambar 2. Ilustrasi Traversal dengan BFS

Algoritma BFS:

1. Kunjungi simpul n,
2. Kunjungi semua simpul yang bertetangga dengan simpul n terlebih dahulu.
3. Kunjungi simpul yang belum dikunjungi dan bertetangga dengan simpul-simpul yang tadi dikunjungi, demikian seterusnya sampai solusi ditemukan

### V. DEPTH FIRST SEARCH

Depth-first search (DFS) atau pencarian mendalam merupakan salah satu metoda traversal di dalam graf yang mengunjungi simpul-simpul di dalam graf secara sistematis. Pada depth-first search, proses pencarian akan dilakukan pada anaknya sebelum pencarian ke tetangga yang setingkat dengan simpul tersebut dilakukan. Pencarian dimulai dari node akar ke level yang lebih tinggi. Proses ini diulangi terus hingga ditemukan solusi. DFS hanya akan mendapatkan 1 solusi pada setiap pencarian, namun membutuhkan memori yang relatif kecil



Gambar 3. Ilustrasi Traversal dengan DFS

Algoritma DFS:

1. Kunjungi simpul n,
2. Kunjungi simpul w yang bertetangga dengan simpul n.
3. Ulangi DFS mulai dari simpul w.

## VI. ALGORITMA RUNUT-BALIK

Algoritma runut-balik merupakan algoritma yang memperbaiki DFS untuk menemukan solusi persoalan dengan lebih efisien dan merupakan perbaikan dari algoritma lempang, Algoritma ini memotong simpul-simpul yang tidak mengarah ke solusi.

Properti metode runut balik merupakan solusi persoalan, fungsi pembangkit, dan fungsi pembatas.

### 1. Solusi persoalan

Solusi dinyatakan sebagai vektor n-tuple:

$x=(x_1, x_2, \dots, x_n)$ ,  $x_i$  anggota himpunan berhingga  $S_i$ .

Mungkin saja  $S_1 = S_2 = \dots = S_n$ .

Contoh:  $S_i = \{0, 1\}$ .  $S_i = 0$  atau 1

### 2. Fungsi pembangkit nilai $x_k$

Dinyatakan sebagai  $T(k)$

$T(k)$  membangkitkan nilai untuk  $x_k$ , yang merupakan komponen vektor solusi

### 3. Fungsi pembatas (fungsi kriteria)

Dinyatakan sebagai  $B(x_1, x_2, \dots, x_k)$ . Fungsi pembatas menentukan apakah  $(x_1, x_2, \dots, x_k)$  mengarah ke solusi. Jika ya, maka pembangkitan nilai untuk  $x_{k+1}$  dilanjutkan, tetapi jika tidak, maka  $(x_1, x_2, \dots, x_k)$  dibuang dan tidak dipertimbangkan lagi dalam pencarian solusi.

Langkah-langkah pencarian solusi pada pohon ruang status secara dinamik dengan algoritma runut balik adalah sebagai berikut:

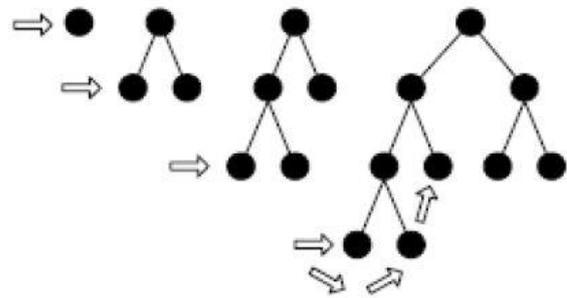
1. Penyelesaian dicari dengan membentuk jalur dari akar ke daun. Aturan pembentukan yang dipakai adalah mengikuti DFS. Simpul-simpul yang sudah dibuat dinamakan "simpul hidup". Simpul hidup yang sedang diperluas dinamakan simpul-E (expand node). Simpul

denumerasi dari atas ke bawah sesuai dengan urutan kelahirannya.

2. Tiap kali simpul-E diperluas, jalur yang dibangun olehnya tentu saja bertambah panjang. Jika jalur yang sedang dibentuk tidak menuju solusi, maka simpul-E "dimatikan" sehingga menjadi simpul mati. Fungsi yang digunakan untuk membunuh simpul-E adalah dengan menerapkan fungsi pembatas (bounding function). Simpul yang sudah mati tidak akan diperluas lagi.

3. Jika pembentukan jalur berakhir dengan simpul yang mati, maka proses pencarian diteruskan dengan membangkitkan anak lainnya. Bila tidak ada lagi simpul anak yang dapat dibangkitkan maka pencarian solusi dilanjutkan dengan melakukan runut-balik ke simpul terdekat. Selanjutnya simpul ini menjadi simpul-E yang baru. Jalur baru dibangun kembali sampai solusi ditemukan.

4. Pencarian dihentikan bila kita telah menemukan solusi atau tidak ada lagi simpul hidup untuk di runut-balik.



Gambar 4. Ilustrasi Traversal dengan Runut Balik

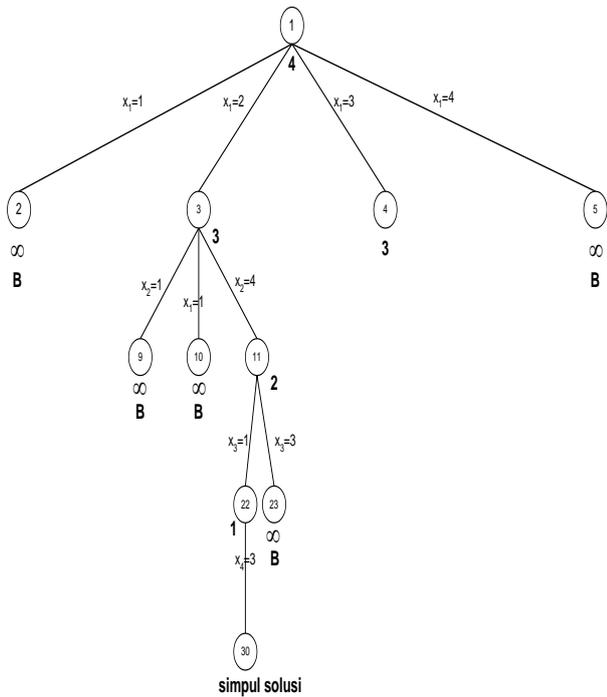
## VII. ALGORITMA BRANCH AND BOUND

Branch and Bound (B&B) pun adalah algoritma pencarian di dalam ruang solusi secara sistematis.

Algoritma runut-balik  $\rightarrow$  skema DFS

Algoritma B&B  $\rightarrow$  skema BFS

Untuk membuat pencarian menuju simpul solusi, menjadi lebih efisien, setiap simpul diberi sebuah nilai ongkos (*cost*). Simpul berikutnya yang akan diekspansi tidak lagi berdasarkan urutan pembangkitannya (sebagaimana pada BFS murni), tetapi simpul yang memiliki ongkos yang paling kecil (*least cost search*).



Gambar 5 Ilustrasi Traversal dengan B&B

### VIII. ALGORITMA A\*

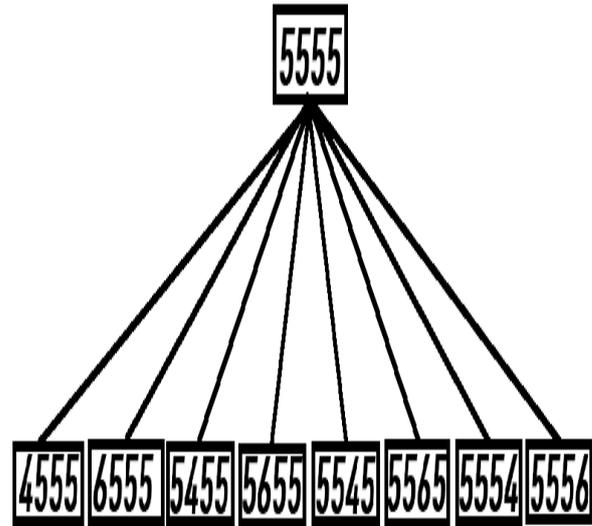
Algoritma A\* menyelesaikan masalah yang menggunakan graf untuk perluasan ruang statusnya. Dengan menerapkan suatu heuristik, algoritma ini membuang langkah-langkah yang tidak perlu dengan pertimbangan bahwa langkah-langkah yang dibuang sudah pasti merupakan langkah yang tidak akan mencapai solusi yang diinginkan. Algoritma A\* membangkitkan simpul yang paling mendekati solusi. Simpul ini kemudian disimpan suksesornya ke dalam list sesuai dengan urutan yang paling mendekati solusi terbaik. Kemudian, simpul pertama pada list diambil, dibangkitkan suksesornya dan kemudian suksesor ini disimpan ke dalam list sesuai dengan urutan yang terbaik untuk solusi-solusi.

### IX. APLIKASI TEORI GRAF UNTUK PERMASALAHAN PLAYING WITH WHEELS

Untuk menerapkan algoritma-algoritma diatas agar dapat digunakan untuk menyelesaikan permasalahan playing with wheels. Abstraksi roda-roda tersebut harus ditranslasikan ke dalam bentuk graf. Jika sudah dalam bentuk draft, algoritma-algoritma traversal graf dapat dilakukan

Simpul pada graf dapat dianalogikan sebagai status roda-roda yang sedang diacu. Selanjutnya, daun-daun dari sebuah simpul merupakan seluruh status next-step dari suatu status. Bisa jadi angka pertama digeser mundur atau maju, dan seterusnya.

Berikut merupakan contoh simpul yang berisi status pada roda-roda:



Gambar 8. Hasil Translasi Playing with Wheels Menjadi Bentuk Graf

Setelah status-status pada roda-roda diubah menjadi bentuk graf. Barulah algoritma DFS, BFS, runut balik, dan branch and bound dapat digunakan.

Yang harus dilakukan dengan teliti adalah mentranslasikan status-status pada permasalahan playing-with wheels ini menjadi graf status yang tepat. Dalam kasus ini tiap simpul memiliki anak sebanyak 8. Jangan sampai ada simpul dengan state yang sama yang diulang kembali.

### X. KESIMPULAN

Banyak sekali algoritma traversal graf yang digunakan untuk menyelesaikan berbagai permasalahan, matematis maupun implementatif. Sebelum algoritma tersebut dapat digunakan, permasalahan tersebut mesti diterjemahkan terlebih dahulu kedalam graf status. Pada bagian itulah teori graf berperan. Untuk mentranslasikan permasalahan-permasalahan kedalam graf status dengan baik. Agar setelah ditranslasikan menjadikan graf, permasalahan-permasalahan yang ada dapat diselesaikan dengan berbagai algoritma traversal graf secara efektif.

## DAFTAR REFERENSI

- [1] <http://adit279.wordpress.com/2008/12/04/implementasi-algoritma-pada-permasalahan-optimasi-solusi-dynamic-water-jug/>. 13 Desember 2012 - 11:00
- [2] <http://belbesy.wordpress.com/2011/06/13/uva-10067-playing-with-wheels/>. 13 Desember 2012 - 11.15
- [3] <http://progspectia.blogspot.com/2011/05/10067-playing-with-wheels.html>. 13 Desember 2012 11:30
- [4] <http://uva.onlinejudge.org/external/100/10067.html>. 13 Desember 2012 11:45
- [5] <http://online-judge.uva.es/board/viewtopic.php?f=9&t=9566&start=15>.
- [6] <http://www.questtosolve.com/browse.php?pid=10067>. 13 Desember 2012 - 12:30
- [7] <http://informatika.stei.itb.ac.id/~rinaldi.munir>. 13 Desember 2012 - 16:00

## 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, 18 Desember 2012



Tino Eka Krisna Sambora/13510062