

# Penerapan Branch and Bound dalam 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 memperbaiki algoritma yang umum digunakan untuk menyelesaikan *playing with wheels* yakni BFS dengan menggunakan algoritma *branch and bound*.

**Kata kunci** - *Graph Traversal, BFS, Branch and Bound, Playing With Wheels.*

## I. PENDAHULUAN

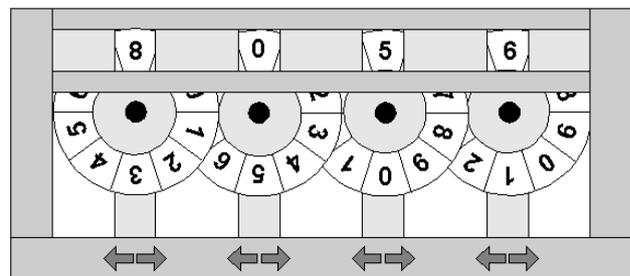
Di kalangan programmer, developer, atau tim olimpiade. Permasalahan *playing with wheels* pasti sudah sering di dengar. Ini merupakan salah satu permasalahan yang seringkali muncul pada olimpiade, dan permasalahan yang menarik untuk dibahas karena ada banyak hal yang dapat dilakukan untuk memperbaiki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

Permasalahan ini umumnya diselesaikan dengan menggunakan abstraksi traversal pada sebuah graf. Umumnya, banyak yang menggunakan algoritma pencarian menyebar atau BFS untuk menyelesaikannya. Pada kesempatan ini penulis hendak mencoba untuk memperbaiki algoritma BFS untuk lebih menambah lagi kemangkusan penggunaan algoritma tersebut untuk menyelesaikan permasalahan ini. Penulis mencoba memperbaikinya dengan menggunakan algoritma *branch and bound*. Algoritma yang menambahkan fungsi pembatas pada algoritma BFS.

Semoga dengan terciptanya makalah ini, pembaca dari kalangan manapun dapat dengan cepat mempertimbangkan langkah yang efisien untuk menyelesaikan permasalahan *playing with wheels*, atau mungkin permasalahan lain yang serupa, maupun implementasinya.

## 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.



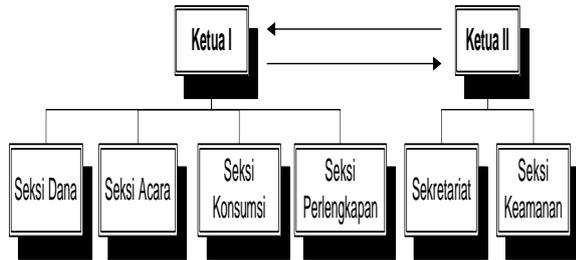
Gambar 1. Ilustrasi *Playing with Wheels*

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), katakana  $F_1F_2F_3F_4$  ( $1 \leq i \leq n$ ). 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.

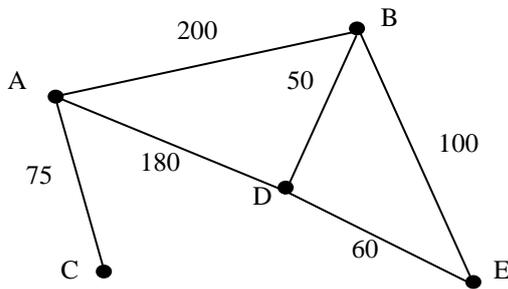
## III. TEORI GRAF

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 2. Ilustrasi Sebuah Graf



Gambar 3. 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

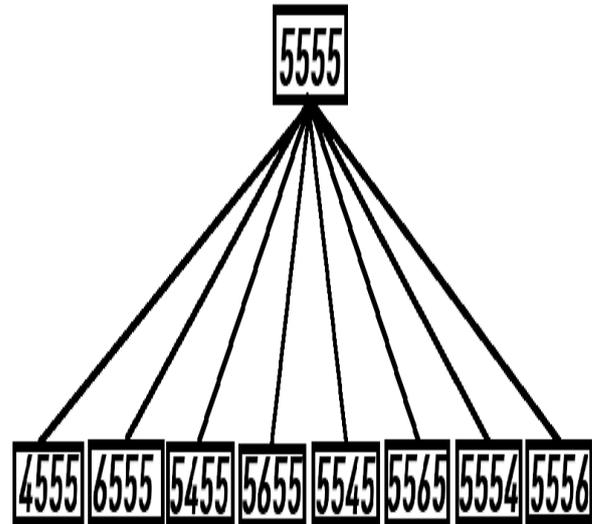
#### IV. APLIKASI TEORI GRAF UNTUK PERMASALAHAN PLAYING WITH WHEELS

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 4. Hasil Translasi Playing with Wheels Menjadi Bentuk Graf

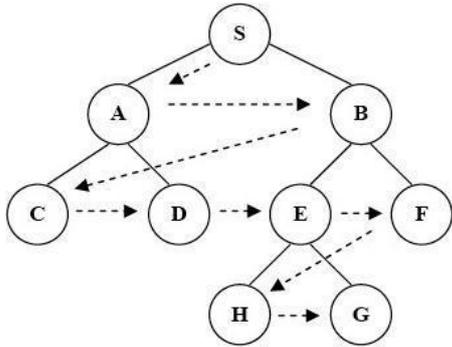
Setelah status-status pada roda-roda diubah menjadi bentuk graf. Barulah algoritma DFS, BFS, runut balik, dan branch and bound dapat digunakan.

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.

#### V. 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.



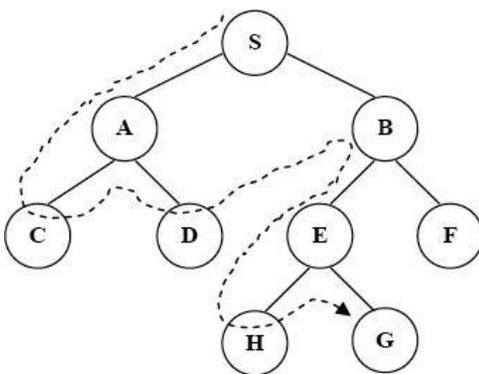
Gambar 5. Ilustrasi Traversal dengan DFS

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

## VI. 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 6. 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. BRANCH AND BOUND

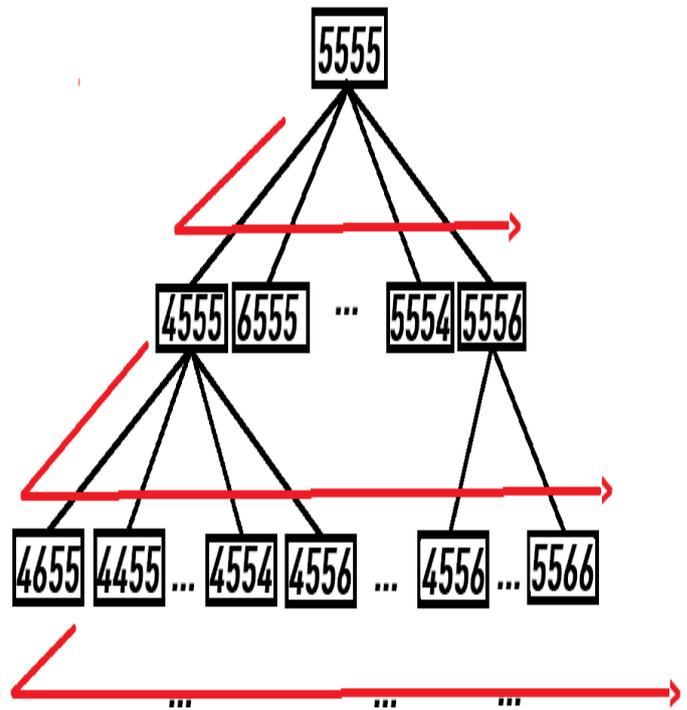
Algoritma *Branch and Bound* (B&B) juga merupakan metode pencarian di dalam ruang solusi secara sistematis.

Algoritma B&B → skema BFS

Untuk mempercepat pencarian ke simpul solusi, maka 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*).

## VII. PENERAPAN ALGORITMA BREADTH FIRST SEARCH UNTUK PERMASALAHAN PLAYING WITH WHEELS

Dalam menyelesaikan permasalahan playing with wheels, algoritma BFS dilakukan dengan cara membuat seluruh kemungkinan “path” kedalam bentuk graf lalu mencari dimana di dalam graf tersebut solusi yang dicari. Setelah solusi ditemukan, graf dicari mundur untuk mendapatkan solusi yang terbaik. BFS merupakan solusi yang cukup mangkus namun menghabiskan banyak karenamemori . BFS mencari seluruh path per “level”, memasukannya ke memori sampai solusi optimal ditemukan. Setelah ditemukan, graf akan di “trace” mundur untuk menjabarkan solusinya yang paling optimal.

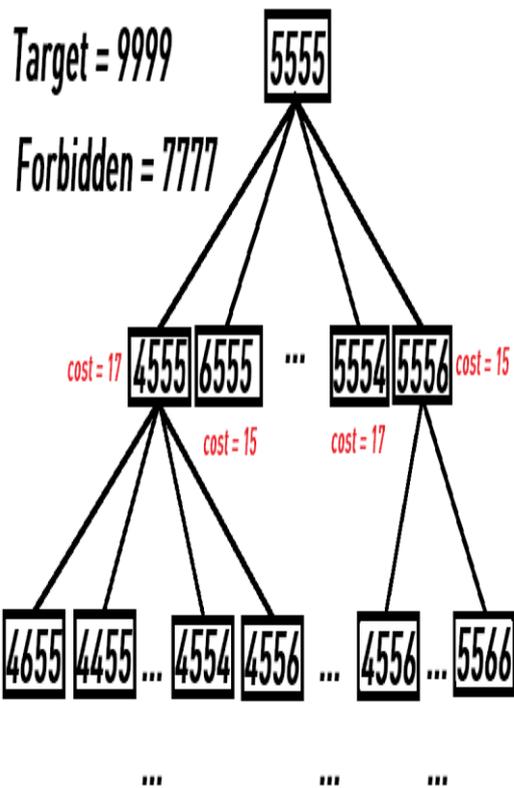


Gambar 7. Traversal Graf Playing With Wheels dengan BFS

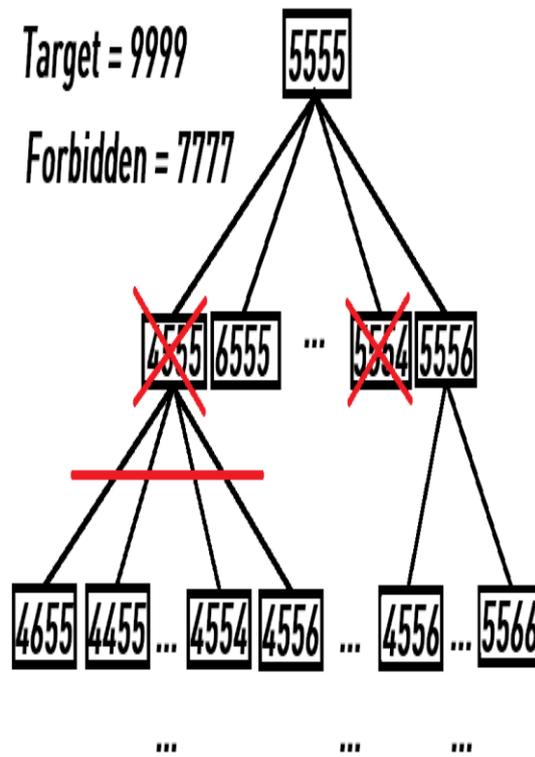
### VIII. PENERAPAN ALGORITMA BRANCH AND BOUND UNTUK MEMPERBAIKI KEMANGKUSAN PENYELESAIAN SECARA BFS

Branch and Bound memperbaiki efisiensi BFS dengan menambahkan cost “pergeseran digit” pada tiap simpul pada graf representasi playing with wheels. Cost dihitung dari mesti sebanyak apa lagi digit-digit pada suatu simpul mesti digunakan agar bilangan pada simpul tertentu menuju simpul target. Apabila setelah secara BFS suatu level graf diketahui seluruh cost menuju simpul targetnya, dibuatlah “bound” pembatas untuk menyatakan simpul tersebut berhenti dicari anaknya., atau terlebih dahulu dicari simpul lain yang costnya lebih rendah.

Selain itu algoritma ini juga membuat “bound” atau pembatas pada digit-digit terlarang (forbidden digit). Apabila sebuah traversal menemukan sebuah simpul dengan digit yang terlarang, simpul tersebut akan diberi “bound” atau batas sehingga simpul tersebut tidak akan diteruskan kembali traversalnya.



Gambar 8. Perhitungan cost untuk membuat bound, bound dihasilkan dari berapa kali bilangan mesti diubah untuk dapat mencapai digit target



Gambar 9. Penandaan Simpul yang Diberi Bound Karena Memiliki Cost yang Tinggi

### IX. PERBANDINGAN EFISIENSI

Dari segi efisiensi, jelas Branch and Bound jauh lebih mangkus disbanding hanya sekedar BFS. Terutama untuk permasalahan playing with wheels, akan banyak sekali simpul yang di “bound” selama traversal berlangsung. Dari tiap simpul yang mengandung masing-masing 8 simpul anak, dapat lebih dari 4 simpul yang dibound karena dinyatakan terlalu jauh dari simpul target dan merupakan sebuah simpul terlarang yang mengandung digit terlarang. Dengan demikian, efisiensi BFS yang di “upgrade” dengan Branch and Bound bisa menjadi 2 kali lebih efisien.

### X. KESIMPULAN

Algoritma-algoritma traversal graf yang digunakan untuk menyelesaikan berbagai permasalahan beragam, matematis maupun implementatif. Sebelum algoritma tersebut dapat digunakan, permasalahan tersebut mesti diterjemahkan terlebih dahulu kedalam graf status. Namun algoritma-algoritma tertentu memiliki keunggulannya terhadap suatu masalah masing-masing, ada yang mangkus terhadap suatu masalah dan tidak untuk yang lain. Untuk masalah Playing With Wheels, branch and bound merupakan algoritma yang menurut penulis paling mangkus dibandingkan dengan algoritma-algoritma traversal graf lain. Karena dapat membuat efisiensi 2 kali lebih besar dibandingkan dengan algoritma BFS biasa.

## DAFTAR REFERENSI

- [1] <http://adit279.wordpress.com/2008/12/04/implementasi-algoritma-pada-permasalahan-optimasi-solusi-dynamic-water-jug/>. 21 Desember 2012 - 11:00
- [2] <http://belbesy.wordpress.com/2011/06/13/uva-10067-playing-with-wheels/>. 21 Desember 2012 - 11:15
- [3] <http://progspectia.blogspot.com/2011/05/10067-playing-with-wheels.html>. 21 Desember 2012 11:30
- [4] <http://uva.onlinejudge.org/external/100/10067.html>. 21 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>. 21 Desember 2012 - 12:30
- [7] <http://informatika.stei.itb.ac.id/~rinaldi.munir>. 21 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, 21 Desember 2012



Tino Eka Krisna Sambora/13510062