

PEMAKETAN TUR KLUB SEPAKBOLA DENGAN ALGORITMA BRANCH AND BOUND

Bayu Hendra W. (13504138)¹, Wisnu Manupraba (13504148)²

*Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
Jl. Ganesha 10, Bandung*

E-mail : if14138@students.if.itb.ac.id¹, if14148@students.if.itb.ac.id²

Abstrak

Kondisi geografis Indonesia yang terbentang dari Sabang sampai Merauke menyebabkan biaya tur klub sepakbola yang berkompetisi di Liga Indonesia sangat mahal. Untuk itu diperlukan suatu keputusan menentukan jadwal tur suatu klub yang bertanding. Diharapkan dengan penentuan rute tur yang tepat akan mengurangi biaya pengeluaran yang harus ditanggung. Penentuan rute tur dilaksanakan dengan memilih 2 kota yang berdekatan sebagai satu paket tur. Untuk itu diperlukan suatu algoritma yang bisa mengelompokkan kota-kota markas klub sepakbola menjadi paket-paket tur dengan meminimumkan total jarak keseluruhan. Salah satu algoritma yang bisa digunakan untuk menyelesaikan masalah ini adalah algoritma *branch and bounds*.

Kata kunci : rute tur, jarak minimum, *branch and bound*

1. Pendahuluan

Penentuan paket tur yang tepat akan membantu setiap klub yang berlaga untuk menghemat pengeluaran transportasi. Jika paket yang terdiri dari dua buah kota yang saling berdekatan dikelompokkan menjadi satu maka ongkos bisa ditekan jika dibandingkan harus menempuh jarak yang lebih jauh.

Tujuan penulisan makalah:

1. Menemukan suatu cara yang bisa digunakan untuk menentukan paket-paket tur yang setiap musim berubah dengan adanya kejadian degradasi dan promosi klub.
2. Mengaplikasikan ke dalam kehidupan nyata algoritma *branch and bounds*.

Dokumen ini memuat algoritma yang digunakan dan contoh penerapannya dalam menentukan jadwal paket tur yang bisa digunakan.

2. Ruang Lingkup

Ruang lingkup masalah ini adalah bagaimana mengelompokkan beberapa tempat menjadi kelompok-kelompok paket yang terdiri dari dua buah tempat. Dalam pembahasan kali ini, batasan jumlah tempat per-paket adalah dua buah dan jumlah kota yang akan dikelompokkan merupakan kelipatan dua.

3. Pemecahan masalah

Pemecahan masalah ini adalah dengan membentuk pohon dengan node-nya berupa pasangan tempat yang berdekatan.

Dimulai dengan membentuk node semua kota yang berdekatan dengan kota pertama. Pencarian dimulai dari pasangan dengan jarak minimum dengan kota pertama. Block semua baris dan kolom yang telah terpilih pada pasangan. Lakukan hal yang sama sampai semua kota telah terpilih. Hingga kota terakhir, akan didapat *the best solution so far*.

Matikan semua node yang memiliki jumlah lebih besar dari *the best solution so far*, kemudian trace kembali pasangan yang masih memiliki kemungkinan untuk memiliki jumlah yang lebih kecil dari *the best solution so far*. Jika itu terjadi, maka *the best solution so far* diganti dengan jarak yang lebih kecil tadi.

Kompleksitas penyelesaian masalah ini adalah merupakan kombinatorial 2 dari n (${}_nC_2$).

$$\begin{aligned} {}_nC_2 &= n!/((n-2)! \cdot 2!) \\ &= n \cdot (n-1)/2 \\ &= (n-1) \cdot (n-3) \end{aligned}$$

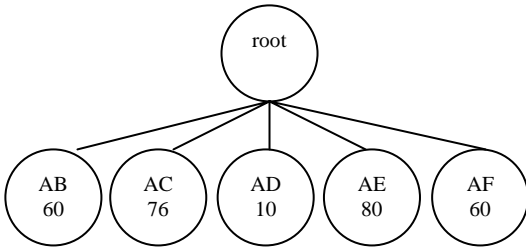
$$O(n) = n^2$$

4. Contoh kasus

Diketahui matrik dua dimensi yang menyatakan jarak antara dua buah kota sebagai berikut :

	A	B	C	D	E	F
A	-	60	76	10	80	60
B	60	-	75	18	15	20
C	76	75	-	20	25	20
D	10	18	20	-	100	80
E	80	15	25	100	-	5
F	60	20	20	80	5	-

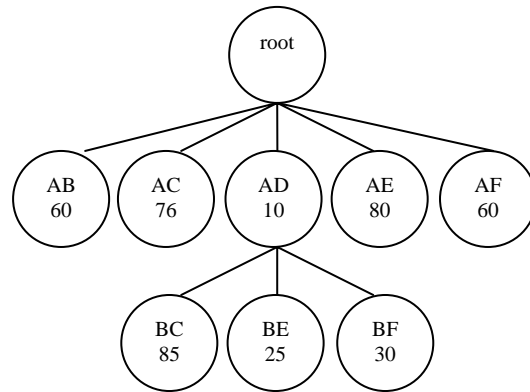
Buat sebuah pohon solusi yang memiliki node berupa pasangan antara kota pertama dengan kota lainnya yang belum ditandai.



Pilih node dengan jarak terpendek untuk dihidupkan pertama kali, yaitu AD. Set baris dan kolom A serta baris dan kolom D dengan nilai tak hingga agar tidak akan terpilih lagi pada langkah berikutnya sehingga matriknya menjadi :

	A	B	C	D	E	F
A	-	60	76	10	80	60
B	60	-	75	18	15	20
C	76	75	-	20	25	20
D	10	18	20	-	100	80
E	80	15	25	100	-	5
F	60	20	20	80	5	-

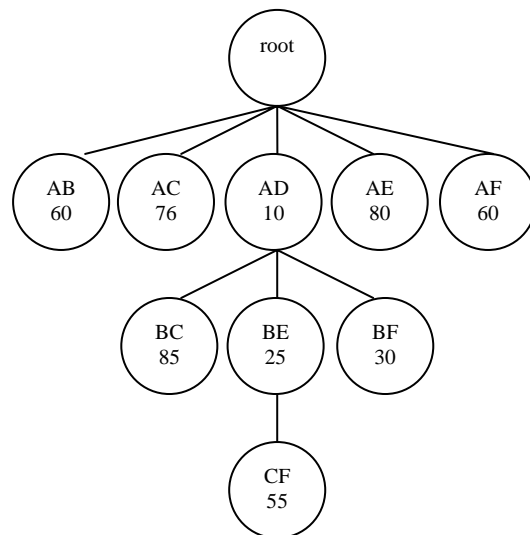
Lanjutkan dengan membangkitkan pasangan antara kota kedua dengan kota lainnya yang belum ditandai hingga terbentuk pohon sebagai berikut :



Pilih node dengan jarak terpendek untuk dihidupkan yaitu BE. Tandai baris dan kolom B serta baris dan kolom E agar tidak terpilih lagi pada langkah selanjutnya. Matrik yang didapat adalah sebagai berikut :

	A	B	C	D	E	F
A	-	60	76	10	80	60
B	60	-	75	18	15	20
C	76	75	-	20	25	20
D	10	18	20	-	100	80
E	80	15	25	100	-	5
F	60	20	20	80	5	-

Lanjutkan dengan membangkitkan pasangan antara kota ketiga dengan kota lainnya yang belum ditandai hingga terbentuk pohon sebagai berikut :



Pilih node dengan jarak terpendek untuk dihidupkan yaitu CF. Tandai baris dan kolom C serta baris dan kolom F agar tidak terpilih lagi pada langkah selanjutnya. Matrik yang didapat adalah sebagai berikut :

	A	B	C	D	E	F
A	-	50	76	10	80	60
B	50	-	75	18	15	20
C	76	75	-	20	25	20
D	10	18	20	-	100	80
E	80	15	25	100	-	5
F	60	20	20	80	5	-

Karena tidak ada lagi elemen matrik yang belum ditandai, maka telah didapat solusi terbaik sejauh ini yaitu $\{(AD),(BE),(CF)\}$ dengan jarak total 55.

Langkah selanjutnya adalah men-traversal simpul yang masih hidup. Jika nilai simpul hidup yang dicek saat ini lebih jelek dari solusi terbaik yang sudah didapat, matikan simpul tersebut dan lanjutkan ke simpul hidup selanjutnya. Jika simpul hidup yang dicek nilainya masih lebih baik dari solusi terbaik, maka lanjutkan pencarian solusi dari simpul hidup tersebut.

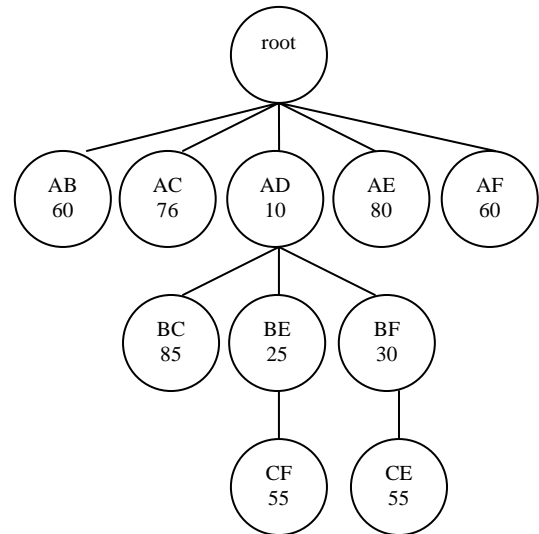
Untuk contoh kasus di atas, langkah selanjutnya adalah sebagai berikut :

Diketahui simpul hidup sampai saat ini adalah AB, AC, AE, AF, BC, BF. Simpul AB, AC, AE, AF, BC dimatikan karena nilainya lebih besar dari solusi terbaik yang telah didapat. Simpul hidup yang tersisa adalah BF. Karena nilainya masih belum melebihi solusi terbaik, maka pencarian solusi dilanjutkan dari simpul BF.

Langkah selanjutnya adalah kembalikan matrik seperti semula. Telusuri semua simpul dari BF sampai ke *root*. Tandai elemen matrik yang mengandung kota-kota hasil penelusuran tadi (semua baris dan kolom A, D, B, F).

	A	B	C	D	E	F
A	-	60	76	10	80	60
B	60	-	75	18	15	20
C	76	75	-	20	25	20
D	10	18	20	-	100	80
E	80	15	25	100	-	5
F	60	20	20	80	5	-

Bangkitkan pasangan antara kota ketiga dengan kota lainnya yang belum ditandai hingga terbentuk pohon sebagai berikut :



Pilih node dengan jarak terpendek untuk dihidupkan yaitu CE. Tandai baris dan kolom C serta baris dan kolom E agar tidak terpilih lagi pada langkah selanjutnya. Matrik yang didapat adalah sebagai berikut :

	A	B	C	D	E	F
A	-	50	76	10	80	60
B	50	-	75	18	15	20
C	76	75	-	20	25	20
D	10	18	20	-	100	80
E	80	15	25	100	-	5
F	60	20	20	80	5	-

Karena semua matrik sudah ditandai, maka didapat satu lagi solusi yaitu $\{(AD), (BF), (CE)\}$. Bandingkan dengan solusi sebelumnya, solusi yang lebih baik menjadi solusi terbaik sampai saat ini.

Karena sudah tidak ada lagi simpul hidup, maka solusi terbaik sejauh ini menjadi solusi akhir persoalan ini.

5. Kesimpulan

Algoritma Branch and Bound yang digunakan dalam contoh kasus di atas memiliki kompleksitas algoritma untuk kasus terburuk sebesar n^2 . Sedangkan untuk kasus terbaik, kompleksitas algoritmanya sebesar n .

Untuk jumlah total kota dan jumlah kota dalam satu paket yang banyak, algoritma ini memberikan hasil yang kurang mangkus.

Dalam prakteknya, banyak batasan yang harus dipenuhi dalam pembuatan jadwal suatu liga sepakbola sehingga penyelesaian dengan cara di atas tidak bisa diterapkan secara langsung dalam kehidupan nyata. Misalnya, jarak antara paket yang satu dengan yang lain juga harus diperhitungkan,

jadwal antara klub yang melakukan tur dengan klub yang didatangi harus diselaraskan agar tidak bentrok dan lain – lain.

6. Daftar Pustaka

1. Munir, Rinaldi. 2005. Strategi Algoritmik. Teknik Informatika ITB : Bandung
2. Clausen, J., Branch and Bound Algorithms - Principles and Examples, <http://www.imada.sdu.dk/Courses/DM85/TSP/text.pdf>, diakses tanggal 19 Mei 2006 pukul 11.00 WIB