

## PENJADWALAN PERTANDINGAN LIGA DENGAN ALGORITMA GREEDY

William

Institut Teknologi Bandung  
Jl. Tamansari 64 Bandung 40116  
e-mail: if16085@students.if.itb.ac.id

### ABSTRAK

Makalah ini membahas mengenai salah satu penerapan algoritma *greedy*. Algoritma *greedy* ini diterapkan pada suatu cara penjadwalan sebuah pertandingan pada cabang olah raga yang berbasis liga (bukan berbasis turnamen) maupun bidang-bidang lainnya yang memiliki permasalahan yang serupa dengan penjadwalan pertandingan berbasis liga. Pertandingan berbasis liga adalah pertandingan antar  $n$  buah peserta yang akan melakukan pertandingan dengan semua peserta kecuali dirinya sendiri. Permasalahan ini adalah permasalahan yang umum dan sering ditemui dalam kehidupan kita sehari-hari. Salah satu masalah nyata sebagai contoh masalah penjadwalan ini adalah penjadwalan pertandingan sepak bola, liga inggris. Algoritma *greedy* yang akan diterapkan dalam makalah ini adalah algoritma yang sederhana dan mudah diterapkan oleh orang awam. Algoritma ini akan sangat berguna untuk suatu penjadwalan pertandingan berbasis liga yang diikuti oleh banyak peserta dan merepotkan dalam penjadwalannya.

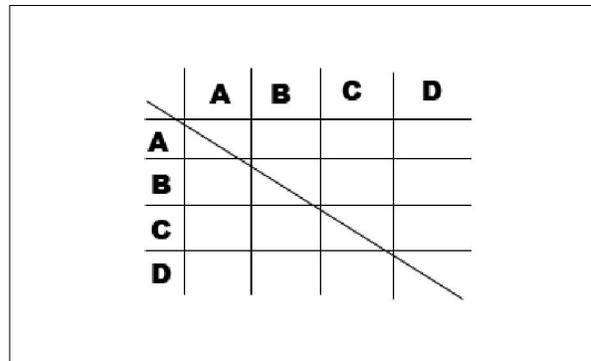
Kata kunci: algoritma *greedy*, berbasis liga.

### 1. PENDAHULUAN

Dalam kehidupan kita sering kita temui suatu kompetisi atau pertandingan dalam berbagai bidang. Salah satu kompetisi yang umum adalah kompetisi pada bidang olahraga. Bila kita menonton televisi maka dapat kita lihat berbagai pertandingan olahraga, seperti pertandingan voli, bola basket, dan sepak bola. Pertandingan sepak bola dapat dikatakan sebagai salah satu tontonan paling populer di Indonesia maupun di dunia. Pertandingan sepak bola yang ada dapat dibagi menjadi dua jenis yaitu pertandingan antar klub pada suatu turnamen, yaitu setiap tim akan bertanding dengan sebuah tim pada tahap awal bila menang maka tim tersebut akan maju ke babak berikutnya sampai babak semifinal dan kemudian babak final. Ada juga pertandingan antar klub pada sebuah liga. Pada tingkat liga setiap tim harus melawan setiap tim lawan

sebanyak dua kali dalam satu musim liga tersebut dan pemenang liga tersebut adalah tim yang meraih poin tertinggi pada akhir liga.

Untuk memperjelas sistem pertandingan berbasis liga maka sebaiknya kita melihat gambar 1 di bawah ini. Pada gambar 1, dapat dilihat bahwa ada 4 tim yang melakukan suatu pertandingan maka sebuah tim harus melawan 3 tim lainnya. Pada gambar 1, tim A harus melawan tim B, C, dan D, begitu seterusnya dengan 3 tim lainnya.



Gambar 1. Sistem Pertandingan Berbasis Liga

Penjadwalan untuk pertandingan berbasis liga, dilakukan dalam suatu rentang waktu tertentu. Dalam satu ruang waktu ini setiap tim akan melawan satu tim lawan dan tentu saja jadwal pertandingan tidak memungkinkan bahwa ada 1 tim yang akan melakukan pertandingan dengan dua tim yang berbeda. Untuk kasus pada gambar 1, penjadwalannya dapat dibuat dan ditampilkan dalam tabel 1. Tabel 1 merupakan salah satu kombinasi penjadwalan yang dapat dibuat.

Hari ke-	Pertandingan I	Pertandingan II
1	A vs B	C vs D
2	A vs C	B vs D
3	A vs D	B vs C

Tabel 1. Jadwal Pertandingan Pada Gambar 1

Persoalan penjadwalan pertandingan yang sudah dijelaskan sebelumnya sebenarnya merupakan suatu persoalan yang sederhana. Tetapi terkadang persoalan ini akan sangat menyulitkan bagi kebanyakan orang bila suatu liga memiliki banyak tim atau peserta. Bila pada gambar 1 terdapat 4 tim yang berkompetisi maka dalam satu satuan waktu terdapat 2 pertandingan yang terjadi, sehingga penjadwalan dengan cepat dapat dibuat. Kita akan mendapat kesulitan bila tim yang ada cukup banyak. Kita ambil contoh saja pada liga sepak bola di Inggris, tim yang ikut serta pada liga tersebut berjumlah 20 tim. Penjadwalan menjadi sedikit sulit bila dilakukan dengan cara coba-coba, mungkin bisa memakan waktu 15-60 menit untuk membuat penjadwalannya tanpa kalkulasi.

Dengan melakukan perhitungan dan kalkulasi pada komputer, Kesulitan pada penjadwalan dengan banyak tim akan dengan cepat dapat terselesaikan, bahkan dalam satu klik tombol saja. Dengan program yang dapat mengkalkulasi dan melakukan penjadwalan maka kita tidak akan mendapat permasalahan pada liga yang memiliki 50 peserta sekalipun. Tentu saja diperlukan suatu algoritma yang baik dalam melakukan penjadwalan.

## 2. METODE

Algoritma *greedy* merupakan salah satu metode yang paling populer dalam memecahkan persoalan. Algoritma ini sangat sederhana dan *fleksible*. Secara harafiah *greedy* berarti rakus atau tamak, yang sebenarnya memiliki konotasi negatif.

Prinsip algoritma *greedy* adalah “ambil sesuatu yang diinginkan sekarang”. Prinsip ini sering diadopsi untuk pemecahan masalah, tentu dalam konteks yang positif. Adapun beberapa contoh penggunaan algoritma *greedy* ini adalah :

1. Persoalan pencarian jalur tersingkat dari Jakarta ke Surabaya.
2. Persoalan *knapsack*.

### 2.1 Definisi Algoritma *Greedy*

Dalam membentuk solusi, algoritma *greedy* pada setiap langkahnya akan mengambil pilihan yang merupakan optimum lokal atau pilihan yang sesuai dengan spesifikasi pembuat algoritma. Dengan pengambilan pilihan yang sesuai pada setiap langkah ini diharapkan solusi yang didapat optimum global atau sesuai keinginan pembuat algoritma.

Pada setiap langkah algoritma *greedy*, kita akan mendapat optimum lokal. Bila algoritma berakhir maka diharapkan optimum lokal ini akan menjadi optimum global. Sehingga sebenarnya algoritma *greedy*

mengasumsikan bahwa optimum lokal ini merupakan bagian dari optimum global.

## 2.2 Skema Algoritma *Greedy*

Seperti algoritma atau metode pemecahan masalah lainnya, algoritma *greedy* juga memiliki sebuah pola atau skema umum yang berlaku untuk berbagai persoalan. Adapun skema algoritma *greedy* adalah sebagai berikut :

1. Inisialisasi S (himpunan solusi) yang kosong.
2. Pilih sebuah pilihan dari C (himpunan kandidat pilihan) dengan sebuah fungsi pemilih.
3. Periksa apakah pilihan yang dipilih layak dengan sebuah fungsi uji kelayakan. Jika layak maka pilihan dimasukkan ke S dan pilihan tersebut dihapus dari C. Jika tidak, maka tidak dipilih.
4. Periksa apakah S sudah memberikan solusi yang lengkap atau C sudah kosong. Jika ya, maka algoritma dihentikan dan jika tidak maka ulangi lagi mulai dari langkah ke-2.

## 3. PENERAPAN METODE

Untuk memakai algoritma *greedy* dalam permasalahan penjadwalan yang sudah dijelaskan pada bab pertama, kita membutuhkan beberapa struktur data. Struktur data yang dibutuhkan adalah :

1. PointString (Char x, Char y). Point digunakan untuk memasukkan 1 pasang simbol tim yang akan bertanding
2. List of PointofChar ( S ). List of PointString ini merupakan struktur untuk jadwal pertandingan dan merupakan himpunan solusi.
3. Array of Char ( C ). Array ini sebagai himpunan string yang berisi semua peserta dan merupakan himpunan simbol kandidat.

Selain struktur di atas juga diperlukan fungsi pemilihan dan fungsi uji kelayakan :

1. Fungsi pemilihan ( Pilih(C) ). Fungsi ini memilih kandidat dari C sesuai dengan spesifikasi yang diinginkan pembuat algoritma. Pada makalah ini fungsi pemilihan, memilih kandidat yang memiliki simbol pada C yang terkecil.
2. Fungsi pengisian ulang (IsiUlang(C)). Fungsi ini akan menginisialisasi ulang himpunan C bila himpunan ini kosong, sehingga C akan terisi dengan String-string awal.
3. Fungsi pembandingan dan uji kelayakan. Fungsi ini melihat apakah kandidat yang diambil dari C apabila dimasukkan ke S membentuk PointofChar yang sudah ada atau tidak. Bila belum maka

masuk ke elemen ke-i dari S sekarang bila tidak masukkan ke elemen ke-(i+1) dari S.

### 3.1 Skema Metode

Skema dalam penerapan metode dengan algoritma *greedy* adalah sebagai berikut :

1. Inisialisasi S yang kosong. S memiliki indeks sebesar  $(n/2) * (n-1)$ , n peserta.
2. Pilih pilihan pada C dengan fungsi pemilihan.
3. Masukkan pada elemen x pada PointString di indeks yang belum penuh (x atau y masih belum terisi) terkecil pada L. Bila x sudah terisi maka isi elemen y-nya.
4. Hapus kandidat dari C bila layak dan posisikan sebagai elemen berikutnya pada S bila tidak layak.
5. Setelah C kosong dan S belum penuh maka IsiUlang(C) dan ulangi tahap 2.

### 3.2 Penggunaan Metode

Untuk memperjelas akan skema pada subbab sebelumnya, metode akan dipakai untuk menyelesaikan masalah berikut. Ada 8 buah tim yang bertanding pada sebuah kejuaraan catur, kejuaraan ini memakai sistem liga dimana setiap tim harus melawan tim lain sebanyak satu kali. Tim-tim tersebut diberi simbol tim A, B, C, D. Kejuaraan akan berlangsung selama 3 hari dimana pada setiap harinya, semua tim melakukan pertandingan sebanyak satu kali. Sekarang kita akan memakai metode algoritma *greedy* pada subbab sebelumnya untuk menyelesaikan masalah ini.

Inisialisasi himpunan C,  $C = \{ A, B, C, D \}$ .

Inisialisasi himpunan S,  $S[ (4/2) * (4-1) ] = S[6]$ .

Tahap 1 :

$C = \{ B, C, D \}$

$S = \{ (A, ) \}$

$C = \{ C, D \}$

$S = \{ (A,B) \}$

$C = \{ D \}$

$S = \{ (A,B), (C, ) \}$

$C = \{ \}$

$S = \{ (A,B), (C,D) \}$

Tahap 2 :

$C = \{ A, B, C, D \}$

$S = \{ (A,B), (C,D) \}$

$C = \{ B, C, D \}$

$S = \{ (A,B), (C,D), (A, ) \}$

$C = \{ C, D, B \}$

$S = \{ (A,B), (C,D), (A, ) \}$

$C = \{ D, B \}$

$S = \{ (A,B), (C,D), (A,C) \}$

$C = \{ D \}$

$S = \{ (A,B), (C,D), (A,C), (B, ) \}$

$C = \{ \}$

$S = \{ (A,B), (C,D), (A,C), (B,D) \}$

Tahap 3 :

$C = \{ A, B, C, D \}$

$S = \{ (A,B), (C,D), (A,C), (B,D) \}$

$C = \{ B, C, D \}$

$S = \{ (A,B), (C,D), (A,C), (B,D), (A, ) \}$

$C = \{ C, D \}$

$S = \{ (A,B), (C,D), (A,C), (B,D), (A, ), (B, ) \}$

$C = \{ D \}$

$S = \{ (A,B), (C,D), (A,C), (B,D), (A, ), (B,C) \}$

$C = \{ \}$

$S = \{ (A,B), (C,D), (A,C), (B,D), (A,D), (B,C) \}$

Bila dilakukan parsing terhadap S untuk melihat jadwal setiap harinya maka cukup dibentuk sebuah list of list of point string H.

$H = \{ ((A,B),(C,D)), ((A,C),(D,B)), ((A,D),(B,C)) \}$

Dimana jadwal hari ke-n adalah  $H[ n - 1 ]$ .

Hari ke-1 ,  $H[0] = \{ (A,B),(C,D) \}$

Hari ke-2 ,  $H[1] = \{ (A,C),(B,D) \}$

Hari ke-3 ,  $H[2] = \{ (A,D),(B,C) \}$

Kombinasi jadwal pertandingan per-juga dapat dilakukan dengan mengambil secara acak elemen H untuk setiap harinya.

Contoh lainnya adalah kasus di atas dengan 8 buah tim yang ikut serta dalam kejuaraan tersebut tim tersebut diberi simbol A, B, C, D, E, F, G, H dan pertandingan dilakukan selama 1 minggu dengan setiap tim harus bertanding sebanyak satu kali setiap harinya. Maka dengan metode di atas akan didapat H sebagai berikut :

$H[0] = \{ (A,B), (C,D), (E,F), (G,H) \}$

$H[1] = \{ (A,C), (B,D), (E,G), (F,H) \}$

$H[2] = \{ (A,D), (B,C), (E,H), (F,G) \}$

$H[3] = \{ (A,E), (B,F), (C,G), (D,H) \}$

$H[4] = \{ (A,F), (B,E), (C,H), (D,G) \}$

$H[5] = \{ (A,G), (B,H), (C,E), (D,F) \}$

$H[6] = \{ (A,H), (B,G), (C,F), (D,E) \}$

Dalam menjalankan algoritma *greedy*, dipakai fungsi kelayakan yang memeriksa apakah point yang akan dibentuk sudah dibentuk apa belum, sehingga perlu melakukan perbandingan sebanyak  $n$ . Selain itu dalam fungsi pemilihan dicari kandidat dengan symbol karakter terkecil sehingga diperlukan perbandingan sebanyak  $n$  pula. Maka kompleksitas waktu algoritma pada persoalan penjadwalan berbasis liga ini adalah  $O(n^2)$ .

#### IV. KESIMPULAN

Persoalan penjadwalan ini tidak menuntut optimalitas solusi tetapi hanya menuntut terbentuknya solusi saja, sehingga sebenarnya persoalan ini dapat diselesaikan dengan baik oleh semua algoritma. Hanya saja dari segi keefesienitas waktu mungkin algoritma *greedy* lebih baik daripada algoritma *brute-force*, selain itu algoritma *greedy* adalah algoritma yang sederhana dan mudah dipahami oleh orang banyak. Selain itu pembuatan algoritma ini juga tidak memiliki kerumitan seperti algoritma lainnya, sehingga kesalahan algoritma bagi pemula dapat dihindari. Dari segi pemakaian memori, pada persoalan penjadwalan berbasis liga, metode yang dipakai pada makalah ini sudah baik dan efisien karena hanya memakai 3 buah struktur data untuk menampung semua elemen yaitu C, S, dan H.

#### REFERENSI

- [1] Introduction to Algorithms (Cormen, Leiserson, Rivest, and Stein) 2001, Chapter 16 "Greedy Algorithms".