

Algoritma Greedy untuk Optimasi Penugasan Wasit pada Pertandingan Olahraga

Fransiska Putri Wina – 13508060

Program Studi Teknik Informatika, STEI, ITB
Jln. Ganesha 10 Bandung 40132
If18060@students.if.itb.ac.id

ABSTRAK

Optimasi dalam dunia olahraga merupakan suatu bidang terus berkembang. Teknik optimasi kombinatorial telah diaplikasikan untuk menunjang sistem pertandingan, contohnya dalam penjadwalan pertandingan dan eliminasi *playoff*. Suatu masalah utama dalam manajemen olahraga adalah penugasan wasit, di mana wasit dalam jumlah terbatas dengan kualifikasi dan keterampilan berbeda harus ditugaskan pada suatu himpunan pertandingan yang telah dijadwalkan. Terdapat beberapa kondisi dan peraturan dan standar yang harus diperhitungkan ketika wasit ditugaskan dalam pertandingan. Peraturan yang dirujuk selanjutnya adalah peraturan yang telah digeneralisasikan sebagaimana yang biasa dipakai dalam banyak liga olahraga, seperti dalam sepakbola, dan bola basket. Pada makalah ini akan diulas mengenai solusi dari permasalahan wasit, dengan memanfaatkan algoritma Greedy. Pada akhir makalah ini juga akan diberikan alternatif solusi dari permasalahan Penugasan Wasit ini dengan memanfaatkan algoritma *Iterated Local Search* (ILS).

Kata kunci: penugasan wasit, algoritma greedy, *Iterated Local Search*.

1. PENDAHULUAN

Salah satu permasalahan utama yang biasa ditemukan dalam manajemen olahraga amatir adalah penugasan wasit pada pertandingan yang telah dijadwalkan. Jumlah wasit yang akan ditugaskan pada suatu pertandingan dapat bervariasi bergantung pada jenis olahraga atau liga yang bersangkutan : pertandingan sepakbola biasanya membutuhkan tiga wasit, sedangkan bola basket membutuhkan dua wasit. Terdapat beberapa peraturan dan standar yang harus diperhitungkan ketika wasit ditugaskan pada suatu pertandingan. Pertandingan pada divisi yang lebih tinggi akan membutuhkan wasit dengan kemampuan dan lisensi yang lebih tinggi. Seorang wasit dapat

memiliki keberhalangan pada suatu waktu tertentu. Selain ia juga memiliki jumlah maksimal pertandingan yang dapat diisinya, dan target pertandingan yang ingin diisinya.

Pada makalah ini peraturan yang dirujuk merupakan peraturan yang telah digeneralisasikan pada banyak liga amatir olahraga, seperti sepakbola, bola basket, dan olahraga lainnya.

Pada bab selanjutnya, akan diulas secara singkat mengenai algoritma greedy. Bab 3 akan mengulas singkat mengenai ILS (*Iterated Local Search*). Dalam bab 4 akan dibahas mengenai deskripsi masalah dari topik ini, termasuk juga batasan masalah yang diambil. Selanjutnya bab 5 akan memaparkan mengenai solusi dari permasalahan penugasan wasit ini, dengan memanfaatkan pengimplementasian algoritma greedy dan ILS. Bab 6 berisi rangkuman dari pemaparan dari bab-bab sebelumnya.

2. ALGORITMA GREEDY

Algoritma Greedy merupakan metode yang paling populer untuk memecahkan persoalan optimasi. Terdapat dua jenis persoalan optimasi, yaitu maksimasi dan minimasi.

Algoritma greedy adalah algoritma yang memecahkan masalah langkah per langkah dengan pada setiap langkah dilakukan hal berikut :

1. Mengambil pilihan yang terbaik yang dapat diperoleh pada saat itu tanpa memperhatikan konsekuensi ke depan (prinsip “*take what you can get now!*”)
2. Berharap bahwa dengan memilih optimum lokal pada setiap langkah akan berakhir dengan optimum global.

Elemen-elemen algoritma Greedy :

1. Himpunan kandidat, C
2. Himpunan solusi, S
3. Fungsi seleksi (*selection function*)
4. Fungsi kelayakan (*feasible*)
5. Fungsi obyektif

Dengan kata lain, algoritma Greedy melibatkan himpunan bagian, S , dari himpunan kandidat, C ; yang dalam hal ini S harus memenuhi beberapa kriteria yang ditentukan, yaitu menyatakan suatu solusi dan S dioptimasi oleh fungsi obyektif.

Berikut adalah skema umum algoritma Greedy :

```
function greedy(input C: himpunan_kandidat) → himpunan_kandidat
{ Mengembalikan solusi dari persoalan optimasi dengan algoritma greedy
Masukan: himpunan_kandidat C
Keluaran: himpunan solusi yang bertipe himpunan_kandidat
}
Deklarasi
x : kandidat
S : himpunan_kandidat
Algoritma:
S ← {} ( inisialisasi S dengan kosong )
while (not SOLUSI(S) and (C ≠ {})) do
x ← SELEKSI(C) ( pilih sebuah kandidat dari C )
C ← C - {x} ( elemen himpunan_kandidat berkurang satu )
if LAYAK(S U {x}) then
S ← S U {x}
endif
endwhile
(SOLUSI(S) or C = {} )
if SOLUSI(S) then
return S
else
write('tidak ada solusi')
endif
```

Namun solusi optimum global yang diperoleh pada algoritma Greedy ini belum tentu merupakan solusi optimum (terbaik), tetapi merupakan *sub-optimum* atau *pseudo-optimum*.

Alasan :

1. Algoritma Greedy tidak beroperasi secara menyeluruh terhadap semua alternatif solusi yang ada.
2. Terdapat beberapa fungsi seleksi yang berbeda sehingga kita harus memilih fungsi yang tepat jika kita ingin algoritma menghasilkan solusi optimal.

3. DESKRIPSI MASALAH

Pada makalah ini diambil penyederhanaan masalah secara umum, di mana setiap pertandingan memiliki beberapa jumlah posisi wasit untuk ditugaskan pada wasit. Pertandingan telah dijadwalkan sebelumnya serta fasilitas dan alokasi waktu untuk setiap pertandingan telah diketahui sebelumnya. Pada pendekatan ini, wasit ditugaskan pada posisi wasit yang kosong, bukan pada pertandingan. Hal ini memungkinkan untuk tidak hanya mengatasi permasalahan penugasan wasit pada cabang olahraga yang membutuhkan jumlah wasit yang berbeda, namun juga pada turnamen di mana pada pertandingan yang berbeda dari cabang olahraga yang sama dapat dibutuhkan jumlah wasit yang berbeda, bergantung pada divisi atau tingkat kepentingan pertandingan tersebut. Pertandingan dengan wasit yang telah ditugaskan terlebih dahulu untuk beberapa posisi wasit juga dapat diatasi oleh pendekatan ini. Setiap posisi wasit yang dapat diisi oleh seorang wasit disebut slot wasit.

Ambil $S = \{1, \dots, m\}$ sebagai himpunan dari slot wasit. Setiap slot wasit $j \in S$ harus diisi oleh seorang wasit dengan penentuan level **kemampuan minimum**, q_j . Ambil $R = \{1, \dots, n\}$ sebagai himpunan wasit, direpresentasikan oleh tingkatan masing-masing. Setiap wasit $i \in R$ memiliki tingkat **kemampuan** tersendiri, dilambangkan dengan p_i , yang juga menentukan slot wasit di mana ia dapat bertugas. Wasit dapat mengutarakan **keberhalangannya** untuk bertugas pada suatu slot waktu tertentu. Selanjutnya, setiap wasit $i \in R$ menetapkan M_i sebagai **jumlah pertandingan maksimum** di mana ia dapat bertugas dan T_i sebagai jumlah target pertandingan yang ingin ia kerjakan. Wasit hanya bertugas pada satu pertandingan pada satu hari.

Permasalahan Penugasan Wasit (PPW) dalam penugasan wasit untuk semua slot wasit berkaitan dengan jadwal pertandingan untuk suatu waktu interval yang telah ditentukan, **meminimalkan jumlah total dari absolut selisih antara target dan jumlah pertandingan aktual** yang ditugaskan pada wasit tersebut, dan memenuhi himpunan batasan yang dijabarkan sebagai berikut :

- a. semua slot wasit harus diisi untuk semua pertandingan
- b. wasit tidak dapat ditugaskan pada slot wasit pada hari yang sama
- c. wasit tidak dapat ditugaskan pada pertandingan dalam waktu di mana ia menyatakan keberhalangannya
- d. wasit harus memenuhi level minimum kemampuan yang telah ditentukan untuk masing-masing slot wasit
- e. wasit tidak dapat bertugas lebih dari angka maksimum jumlah pertandingan yang diberikan

4. IMPLEMENTASI

4.1 Pendefinisian Elemen Algoritma Greedy

Permasalahan penugasan wasit ini akan dicari solusi optimumnya menggunakan algoritma Greedy. Maka dari itu, akan didefinisikan elemen-elemen algoritma Greedy dalam permasalahan ini.

1. Himpunan kandidat
Berisi himpunan wasit, $R = \{1, \dots, n\}$
2. Himpunan solusi
Merupakan himpunan dari slot wasit $S = \{1, \dots, m\}$ diisi dengan himpunan wasit yang berseuaian
3. Fungsi seleksi (*selection function*)
Untuk setiap slot wasit yang kosong, dipilih wasit dengan nilai absolut selisih antara target T_i dan jumlah pertandingan aktual N_i yang paling besar.
4. Fungsi kelayakan (*feasible*)

- Kemampuan wasit $p_i \geq$ kemampuan minimum pertandingan q_j .
 - Wasit tidak dapat ditugaskan dua kali pada hari yang sama
 - Wasit tidak dapat bertugas pada hari di mana ia menyatakan keberhalangannya
 - Wasit tidak dapat bertugas lebih dari angka maksimum jumlah pertandingan yang diberikan M_i .
5. Fungsi obyektif
Nilai absolut selisih antara target T_i dan jumlah pertandingan aktual N_i minimum.

4.2 Struktur Data

```
{Struktur data Slot Wasit}
type SlotWasit
<
  HariP : string
  {Hari pertandingan dari slot wasit}
  KemampuanMin : integer
  {Minimum kemampuan wasit yang diperlukan untuk slot wasit}
  NamaWasitS : string
  {Nama wasit yang mengisi slot yang berkaitan}
>

{Struktur data Wasit}
type Wasit
<
  NamaWasit : string
  {Nama wasit yang bernilai unik untuk masing-masing wasit}
  MaksP : integer
  {Jumlah maksimum pertandingan yang ditangani oleh wasit}
  TargetP : integer
  {Jumlah target pertandingan yang ditangani oleh wasit}
  AktualP : integer
  {Jumlah aktual pertandingan yang ditangani oleh wasit}
  KemampuanW : integer
  {Kemampuan wasit}
  HBerhalangan : array[] of string
  {Hari di mana wasit berhalangan}
  HBertugas : array[] of string
  {Hari di mana wasit bertugas}
>

NSlotWasit : integer
NWasit : integer
{NWasit  $\geq$  NSlotWasit}
TblSlotWasit : array[1..NSlotWasit] of SlotWasit
TblWasit : array[1..NWasit] of Wasit
```

4.3 Algoritma Solusi

```
procedure SortWasit (input NWasit : integer, input/output TblWasit1 : TblWasit)
{fungsi untuk mensorting wasit terurut mengecil berdasarkan nilai absolut selisih TargetP dan AktualP paling besar}
```

KAMUS LOKAL
i, j : integer
temp : Wasit

```
ALGORITMA
i traversal [1..NWasit-1]
  j traversal [i..NWasit]
    if abs(TblWasit[i].TargetP - TblWasit[i].AktualP) < abs(TblWasit[j].TargetP - TblWasit[j].AktualP) then
      temp  $\leftarrow$  TblWasit[i]
      TblWasit[i]  $\leftarrow$  TblWasit[j]
      TblWasit[j]  $\leftarrow$  temp
```

```
function GreedyWasit (input NWasit, TblWasit1: TblWasit, SW1 : SlotWasit)  $\rightarrow$  string
{Fungsi untuk memilih wasit yang memenuhi kriteria SlotWasit dengan nilai absolut selisih TargetP dan MaksP paling besar}
```

KAMUS LOKAL
i, j, k : integer
solusi : bool

```
ALGORITMA
solusi  $\leftarrow$  false
i  $\leftarrow$  1
while (NOT solusi AND i  $\leq$  NWasit) do
  if TblWasit[i].AktualP+1 < TblWasit[i].MaksP AND TblWasit[i].KemampuanW  $\geq$  SW1.KemampuanMin AND SW1.HariP NOT IN TblWasit[i].HBerhalangan AND SW1.HariP NOT IN TblWasit[i].HBertugas then
    solusi  $\leftarrow$  true
  else
    i++
if solusi then
   $\rightarrow$  TblWasit[i].NamaWasit
else
   $\rightarrow$  NULL
```

```

Program OptimasiPenugasanWasit
{Program utama}

KAMUS
TblWasit1 : TblWasit
TblSW1 : TblSlotWasit
gagal : bool
j : integer

ALGORITMA
gagal ← false
SortWasit(NWasit, TblWasit1)
j traversal [1..NSlotWasit]
    if GreedyWasit(NWasit, TblWasit1,
        TblSW1[j]) ≠ NULL then
        TblSW1[j].NamaWasitS ←
            GreedyWasit(NWasit,
                TblWasit1, TblSW1[j])
    else
        gagal ← true
if gagal then
    write ('Solusi lengkap tidak
        ditemukan')

```

5. ANALISIS

5.1 Analisis Algoritma Solusi

Dalam algoritma solusi yang dijabarkan di atas, terdapat 3 bagian utama, yaitu prosedur SortWasit, fungsi GreedyWasit, dan program OptimasiPenugasanWasit.

Prosedur SortWasit digunakan untuk melakukan sorting terhadap Tabel TblWasit mengurut mengecil berdasarkan nilai absolut dari selis target jadwal wasit pada 1 minggu yang terkait dengan jumlah aktual penugasa wasit pada 1 minggu tersebut. Prosedur ini bertujuan untuk menyederhanakan algoritma Greedy selanjutnya.

Fungsi GreedyWasit berguna untuk melakukan pemilihan wasit yang sesuai untuk satu buah slot wasit. Algoritma ini akan melakukan pengecekan pertama kali pada wasit dengan nilai absolu dari selis target jadwal dan aktual yang paling besar terlebih dahulu. Hal ini sudah dilakukan oleh prosedur SortWasit. Selanjutnya algortima GreedyWasit akan melakukan pengecekan apakah wasit yang berkaitan memenuhi semua fungsi kelayakan. Jika ya, maka wasit terebut akan ditugaskan untuk slot waktu tersebut. Jika ada fungsi kelayakan yang tidak dipenuhi, maka akan wasit dengan nilai absolut selisih terbesar kedua akan dicek apakah memenuhi fungsi kelayakan. Demikian seterusnya sampai ditemukan wasit yang memenuhi fungsi persyaratan.

Bagian utama dari program, yaitu program Optimasi Penugasan Wasit melakukan pemanggilan terhadap prosedur SortWasit terlebih dahulu untuk mendapatkan urutan wasit sesuai dengan yang diinginkan. Kemudian untuk setiap slot wasit akan dijalankan fungsi GreedyWasit untuk menentukan wasit mana yang akan mengisi slot wasit tersebut.

5.2 Analisis Kompleksitas Algoritma Solusi

Analisis kompleksitas algoritma ini akan dilakukan per bagian program.

Ambil $n = NWasit$ dan $m = NSlotWasit$.

Untuk prosedur SortWasit, algoritma sorting yang digunakan adalah maksimum sort. Algoritma ini memiliki kompleksitas $O(n^2)$.

Untuk fungsi GreedyWasit, karena fungsi ini merupakan algoritma Greedy terhadap suatu larik terurut, maka kompleksitasnya maksimumnya adalah $O(n)$. Sedangkan untuk kompleksitas minimumnya = 1. Sehingga kompleksitas rata-ratanya $O((n+1)/2)$.

Sedangkan untuk program utama, karena pengecekan dilakukan terhadap semua data SlotWasit, maka kompleksitas algoritmanya adalah $O(m)$.

Maka dengan melihat strukut program utama (Program OptimasiPenugasanWasit), untuk keseluruhan, kompleksitas algoritmanya adalah :
 $O(n^2) + O(m) * O((n+1)/2)$ sehingga menjadi
 $O(n^2 + m * (n+1)/2)$.

Komplekstitas algoritma ini termasuk ke dalam algoritma polinomial, di mana performansinya masih terbilang cukup baik.

6. ALTERNATIF

Seperitnya yang telah disebutkan sebelumnya, optimasi dalam dunia olahraga, termasuk juga mengenai penugasan wasit merupakan bidang yang terus berkembang. Telah ada penelitian-penelitian sebelumnya yang mengupas mengenai permasalahan-permasalahan ini. Di sini, penulis akan memberikan alternatif solusi untuk permasalahan penugasan wasit ini.

Untuk alternatif ini, ada permasalahan yang ditambah, yaitu wasit dapat menjalankan tugas (atau merupakan seorang pemain juga dari suatu pertandingan) lebih dari sekali dalam sehari, asalkan tempatnya tidak berpindah. Alternatif ini menggunakan pendekatan tiga tingkat, dengan pada awalnya menggunakan algoritam Greedy,

kemudian dilakukan perbaikan dengan algoritma *Iterated Local Search* (ILS), dan kemudian dilakukan proses optimasi dengan ILS lagi. Pendekatan heuristik tiga fasa ini merupakan hasil penelitian dari Celso C. Ribeiro dkk di Reyjavik, Iceland, pada July 2006^[4].

Berikut terlebih dahulu penulis akan berikan sedikit penjelasan mengenai algoritma ILS

6.1 Iterated Local Search

Iterated Local Search (ILS) merupakan suatu metode SLS yang menghasilkan runtutan solusi yang dihasilkan oleh sebuah algoritma lain, di mana hasil yang diperoleh jauh lebih baik daripada jika kita dilakukan percobaan berulang yang random terhadap algoritma tersebut.

Notasi pada ILS :

- S : himpunan dari (kandidat) solusi
- s : solusi di S
- f : *cost* fungsi
- $f(s)$: *cost* fungsi dari solusi s
- s^* : solusi optimum lokal
- S^* : himpunan dari solusi optimum lokal
- LocalSearch memetakan dari $S \rightarrow S^*$

Prinsip Kerja ILS

Pencarian di S^*

- LocalSearch membatasi ruang lingkup dari S yang besar menjadi S^* yang relatif lebih kecil
- Berikut langkah-langkah dalam ILS :
 - Diberikan s^* , lakukan perturbasi : $s^* \rightarrow s'$
 - Lakukan LocalSearch : $s' \rightarrow s^{**}$
 - Lakukan tes keberterimaan : $s^*, s^{**} \rightarrow s^{*new}$

Berikut algoritma proseduralnya :

```

Procedure IteratedLocalSearch
s0 ← GenerateInitialSolution
s* ← LocalSearch(s0)
repeat
  s' ← Perturbasi (s*, history)
  s** ← LocalSearch(s')
  s* ← AcceptanceCriterion
    (s*, s**, history)
until termination condition met
end
  
```

Performansi dari ILS bergantung pada interaksi dari semua modul

Versi dasar dari ILS

- GenerateInitialSolution : random atau heruristic pembangunan
- LocalSearch : seringkali telah dapat digunakan secara siap
- Perturbation : perpindahan random pada tetangga dalam orde yang lebih tinggi

- AcceptanceCriterion : mendorong menurunkan *cost*

Versi dasar ini seringkali memberikan performansi baik, hasil yang maksimal dikembangkan dengan penambahan optimasi lanjutan.

Selanjutnya akan dijelaskan mengenai algoritma alternatif yang digunakan Celso C. Ribeiro dkk untuk permasalahan penugasan wasit ini.

6.2 Heuristic Tiga Fasa

Sesuai dengan namanya, algoritma ini terdiri atas 3 fasa, yaitu :

1. Pembangunan herusitic Greedy
2. Perbaikan dengan dasar ILS untuk membuat solusi awal feasible (jika diperlukan)
3. Peningkatan solusi feasible dengan prosedur berbasis ILS

Atau dengan algoritma dapat dijabarkan sebagai berikut :

```

Algoritma HeuristikTgsWasit(MaxIter)
S* ← BangunSolusiRandomGreedy();
if Not isFeasible(S*) then
  S* ← RepairHeuristic(S*,MaxIter);
if isFeasible(S*) then
  S* ← ImprovementHeuristic(S*);
else "infeasible"
return S*
  
```

6.2.1 Pembangunan Heuristic Greedy

Berikut urutan langkah yang dilakukan :

- Berikan slot wasit pada wasit yang juga merupakan pemain
- Selama terdapat slot wasi yang kosong dan ada wasit yang masih kosong juga lakukan :
 - Pilih wasit dengan tingkatan kemampuan tertinggi
 - Secara Greedy pilih fasilitas dengan slot wasit yang masih kosong yang paling membutuhkan wasit dengan level tertinggi tersebut
 - Lakkan penugasan wasit dengan level tertinggi ke fasilitas ini
- Lengkapi solusi dengan penugasan yang tidak feasible

6.2.2 Local Search dan Tetangga

Solusi yang dibangun dari algoritma Greedy tersebut tidak harus optimal. Algoritma Local Search akan dengan sukses mengganti solusi sekarang dengan solusi yang lebih baik dari tetangga solusi pertama, dan berhenti pada optimum lokal.

Strategi peningkatan yang pertama : Solusi saat ini diganti dengan tetangga pertama yang nilai *cost* fungsinya lebih baik dari solusi saat ini

Swap moves : wasit yang ditugaskan pada dua slot wasi ditukar (jumlah pertandingan pada masing-masing wasait tidak berubah)

Exchange move : wasit ditugaskan ke suatu slot wasit diganti dengan wasit lain (jumlah pertandingan yang ditugaskan pada masing-masing wasit akan bertambah dan berkurang satu)

Hanya perpindahan yang melibatkan wasit pada fasilitas yang sama (atau yang sedang tidak bertugas sama sekali) yang diperbolehkan

6.2.3 Heuristic Perbaikan

Dari herusitik pembangunan maka akan dihasilkan solusi di mana wasit tidak melakukan perpindahan.

Batasan yang mungkin dilanggar adalah mengenai jadwal, keberhalangan wasit, skill level, jumlah maksimal pertandingan.

Pendekatan yang dilakuakn adalah : meminimalkan jumlah pelanggaran (tidak ada pelanggaran pada solusi feasible).

Proses tahapan dalam herusitik perbaikan ini termasuk ke dalam perturbasi :

- Pilih fasilitas f_k dengan penugasan yang infeasible
- Pilih slot wasit dengan maksimum minimum skil level yang ditugaskan pada wasit dengan minimal satu pelanggaran
- Pilih wasit r_i yang bertugas di fasilitas lain (atau tidak bertugas sama sekali) yang skillnya paling tidak sama dengan pada fasilitas tersebut
- Secara acak pilih wasit yang tidak bertugas pada fasilitas yang sama untuk ditugaskan pada slost wasit yang sedang diberikan pada r_i
- Akhirnya, tugaskan wasit r_i ke slot wasit pada fsilitas f_k yang sedang diberikan pada wasit dengan paling sedikit satu pelanggaran

6.2.4 Herusitik Peningkatan Performa

Beberapa target yang ingin dicapai pada Heuristik Peningkatan Performa adalah sebagai berikut :

- Minimalkan total jumlah wasit yang berbeda dari jumlah target penugasan dengan jumlah penugasan aktual
- Hanya eschange move dan perturbasi yang tetap menjaga feasibilitas yang akan diperhitungkan
- Swap move tidak diaplikasikan, karena swap move tidak mengurangi jumlah fungsi objektif.

Selanjutnya dilakukan proses perturbasi sebagai berikut :

- Pilih dua wasit yang bertugas pada fasilitas berbeda sedemikianhingga penukaran semua tugas mereka feasible
- Perhatikan : cek apakah ada pertandingan lain yang dapat ditugasi oleh wasit-wasit ini setelah penukaran
 - Hanya slot wasit yang ditugaskan pada wasit yang melebihi jumlah targetnya yang diperhitungan
- Pasangan pertam dari wasit yang penukarannya dapat mengurangi fungsi objektif dipilih dan semua tugas mereka ditukar

Setelah melalui ketiga tahapan tersebut, maka diharapkan kan didapat solusi untuk Permasalahan Penugasan Wasit yang optimal.

7. KESIMPULAN

Algoritma Greedy merupakan algoritma yang mangkus untuk permasalahan optimasi. Begitu halnya dengan algoritma Greedy pada permasalahan Penugasaan Wasit ini. Dengan total kompleksitas waktu dari solusi adalah $O(n^2 + m * (n+1)/2)$ yang merupakan kompleksitas waktu polinomial, performansi dari solusi permasalahan penugasan wasit pada makalah ini cukuplah baik.

Namun perlu diingat di sini, bahwa suatu permasalahan tidak hanya dapat diselesaikan oleh suat kakas saja. Untuk permasalahan penugasan wasit misalnya, terdapat algoritma penunjang lain yang dapat digunakan untuk semkain mengomptimalkan hasil, yaitu misalnya dengan pemanfaatan algoritma *Iterated Local Search*.

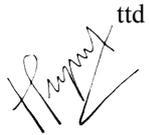
REFERENSI

- [1]Munir, Rinaldi. "Diktat Kuliah IF3051 Strategi Algoritma". Institut Teknologi Bandung. 2009
- [2]www.cis.upenn.edu/~matuszek/cit594-2005/Lectures/36-greedy.ppt
Tanggal akses : 6 Desember 2010
- [3]http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.132.2138
Tanggal akses : 6 Desember 2010
- [4]http://www-di.inf.puc-rio.br/~celso/talks/RefereeAssignment-EURO21.pdf
Tanggal akses : 6 Desember 2010
- [5]http://www.econ.upf.edu/docs/papers/downloads/513.pdf.copia
Tanggal akses : 6 Desember 2010
- [6]http://www2.imm.dtu.dk/courses/02723/RefereeAssignment_HM.pdf
Tanggal akses : 6 Desmber 2010
- [7]http://www.sls-book.net/Slides/sls-ils+vns.pdf
Tanggal akses : 6 Desmber 2010

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, 7 Desember 2010

 ttd

Fransiska Putri Wina