ANALISIS ALGORITMA *ASSIGNMENT* PADA PROGRAM PENJADWALAN

Bernardinus Harnadi
Program Studi Teknik Elektro Unika Soegijapranata
Jl.Pawiyatan Luhur IV/1,Bendan Duwur, Semarang
e-mail: b_harnadi@yahoo.com

Abstrak

Algoritma Assignment banyak diterapkan pada program penjadwalan. Penjadwalan memiliki arti pemakaian suatu sumber daya yang terbatas untuk memperoleh suatu jadwal pemakaian sumber daya yang optimal. Sehingga penjadwalan merupakan persoalan optimasi dan banyak ditemukan pada berbagai bidang terutama yang memiliki sumber daya yang terbatas.

Dalam makalah ini dilakukan suatu analisis sebuah algoritma assignment yang digunakan dalam program penjadwalan ruang. Analisis dibagi menjadi 2 tahap sesuai dengan pentahapan dalam program penjadwalan ruang. Proses pentahapannya yaitu: proses pembobotan tiap kegiatan terhadap ruang dan proses assignment. Pembobotan yang dilakukan merupakan proses memberi nilai bobot tiap kegiatan terhadap ruang. Proses assignment yang dilakukan dibagi dalam tiga sub proses, yaitu : menemukan nilai optimum berdasarkan bobot kegiatan terhadap ruang, mengecek hasil perhitungan optimum terhadap ketersediaan fasilitas dalam ruang, dan meletakkan kegiatan optimum ke dalam tabel jadwal.

Analisis pada algritma pembobotan dilakukan pada kode sumber yang mengkonstruksi langkah-langkah pembobotan. Model analisis algortimam dilakukan per bagian penggalan algritma dengan bantuan kode sumber yang bersangkutan.

Kata kunci: *assignment, pembobotan, penjadwalan, analisis, kode sumber*

PENDAHULUAN

Optimisasi ialah suatu proses untuk mencapai hasil yang ideal atau optimal (nilai efektif yang dapat dicapai). Optimisasi merujuk pada studi permasalahan yang mencoba untuk mencari nilai minimal atau maximal dari suatu fungsi riil. Untuk dapat mencapai nilai optimal baik minimal atau maximal tersebut, secara sistimatis dilakukan pemilihan nilai variabel bilangan bulat atau riil yang akan memberikan solusi optimal.

Dalam program penjadwalan ruang dibagi dalam 2 langkah. Pertama dibentuk suatu bipartite graph yang sisi-sisi mewakili kegiatan dan ruang. Penerapan komputasi oleh komputer dengan cara membentuk matriks kegiatan terhadap ruang. Kegiatan/event yang dimaksud mengandung data jumlah peserta dan lama waktu kegiatan. Sedangkan data ruang terdiri dari nama ruang, kapasitas dan fasilitas ruang. Matriks kegiatan terhadap ruang yang dibuat terkait dengan waktu dalam arti bahwa kegiatan dan pemakaian ruang memiliki batasan waktu tertentu yang telah ditetapkan.

Hal kedua yang dilakukan sesuai dengan metode assignment adalah memberi bobot setiap kegiatan yang dijadwalkan pada ruang. Ketiga adalah mencari bobot maksimum yang sesuai atau nilai optimum berdasar bobot kegiatan terhadap ruang. Kegiatan yang optimum dimasukkan dalam matrik jadwal. Urutan langkah-langkah tersebut direpresentasikan dalam bentuk rancangan algoritma assignment untuk keperluan penjadwalan.

METODOLOGI

Penelitian disini dilakukan dengan mengadopsi langkah-langkah penjadwalan yang dilakukan dalam bentuk analisis algoritma. Analisis dibagi menjadi 2 tahap sesuai dengan pentahapan dalam program penjadwalan ruang. Proses pentahapannya yaitu: proses pembobotan tiap kegiatan terhadap ruang dan proses assignment. Pembobotan merupakan proses memberi nilai bobot tiap kegiatan terhadap ruang. Proses assignment dibagi dalam tiga sub proses, yaitu : menemukan nilai optimum berdasarkan bobot kegiatan terhadap ruang, mengecek hasil perhitungan optimum terhadap ketersediaan fasilitas dalam ruang dan gedung, dan meletakkan kegiatan optimum ke dalam tabel jadwal. Alat analisis yang digunakan berupa proses kerja algoritma dan analisis kode sumber yang bersesuaian.

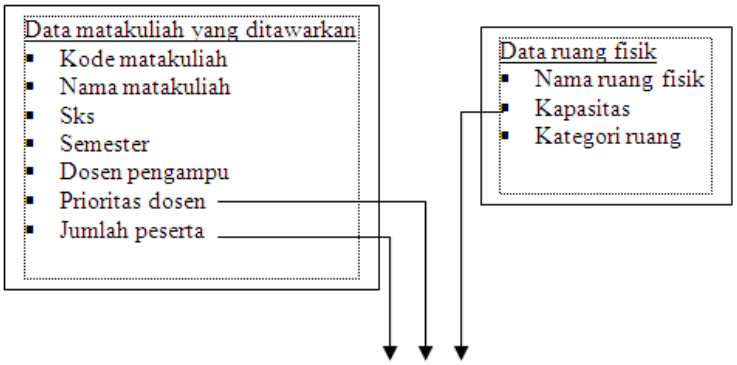
Analisis Algoritma Pembobotan

Data-data yang diperlukan untuk proses penjadwalan adalah data matakuliah yang ditawarkan, data preferensi dosen, dan data ruang fisik seperti ditunjukkan dalam gambar 1.

Data matakuliah yang ditawarkan	Data preferensi dosen	Data ruang fisik
▪ Kode matakuliah ▪ Nama matakuliah ▪ Sks ▪ Semester ▪ Dosen pengampu ▪ Prioritas dosen ▪ Jumlah peserta	▪ Kode dosen ▪ Nama dosen ▪ Waktu ketidaksanggupan mengajar	▪ Nama ruang fisik ▪ Kapasitas ▪ Kategori ruang

Gambar 1. Data yang digunakan untuk proses penjadwalan

Proses pembobotan memerlukan data prioritas dosen dan jumlah peserta dari data matakuliah yang ditawarkan, dan data kapasitas dari data ruang fisik seperti terlihat dalam gambar 2. Bobot dihitung untuk setiap kegiatan yang dijawalkan pada setiap ruang. Seperti ditunjukkan dalam gambar 3.



$$Bobot = \frac{jumlah\_peserta}{kapasitas\_ruangfisik} \times 100 \times prioritas\_dosen$$

Gambar 2. Proses pembobotan

Pembobotan dilakukan dengan memberi bobot pada setiap rusuk graf bipartite dengan menggunakan rumus : *Bobot = bobot_v x prioritas*

$$bobot\_v = \begin{cases} 0 & ; \quad 0\% = v > 120\% \\ 50 & ; \quad 0\% < v \le 50\% \\ 70 & ; \quad 50\% < v \le 70\% \\ 90 & ; \quad 70\% < v \le 90\% \\ 100 & ; \quad 90\% < v \le 100\% \\ 90 & ; \quad 100\% < v \le 110\% \\ 50 & ; \quad 110\% < v \le 120\% \end{cases}$$

$$v = \frac{jumlah\_peserta}{kapasitas\_ruangfisik} \times 100\%$$

Gambar 3. Penghitungan bobot_v

Hasil perhitungan akan maksimum / mencapai 100 % bila jumlah peserta matakuliah = kapasitas ruang fisik yang ditempati. Angka 100 % dihargai dengan bobot 100. Dibawah dan diatas 100 % bobot dibuat menurun karena penggunaan ruang tidak maksimal.

Algoritma pembobotan :

```
if arrPref [hr, jm] then
```

ISBN. 978-602-99334-0-6

```
begin
  { Perhitungan bobot menggunakan bil. integer }
  temp := round (qyJdwk1Peserta.Value * 100/Kaps);
  case temp of
    0..1 : MtxMK [maxMtx].Bobot := 0;
    2..20 : MtxMK [maxMtx].Bobot := 50 * qyJdwk1Bobot.Value;
    21..70 : MtxMK [maxMtx].Bobot := 70 * qyJdwk1Bobot.Value;
    71..90 : MtxMK [maxMtx].Bobot := 90 * qyJdwk1Bobot.Value;
    91..100 : MtxMK [maxMtx].Bobot := 100 * qyJdwk1Bobot.Value;
    101..110 : MtxMK [maxMtx].Bobot := 90 * qyJdwk1Bobot.Value;
    111..120 : MtxMK [maxMtx].Bobot := 50 * qyJdwk1Bobot.Value;
  else
    MtxMK [maxMtx].Bobot := 0;
  end;
end
else if not arrPref [hr, jm] then
     MtxMK [maxMtx].Bobot := 0;
```

Analisis Algoritma Assignment

Dalam penjadwalan assignment didefinisikan sebagai proses mencari bobot maksimum yang sesuai (optimum) di dalam matiks kegiatan Vs ruang (matrik A-matriks matakuliah) untuk kemudian menempatkannya dalam matriks kegiatan Vs ruang (matrik B - matriks jadwal). Matrik A dan B merupakan matrik yang identik, tetapi dengan isi yang berbeda.

Rancangan algoritma assignment:

1. Temukan nilai bobot maksimum (n = 1) dari matrik A dan catat data kegiatan dan koordinatnya.
2. Lakukan jenis cek a, b dan c pada data kegiatan dari langkah 1.
3. Jika lolos cek dari langkah 2, lakukan cek pada alamat matrik B dengan menggunakan alamat koordinat langkah 1.
4. Jika pada alamat matrik B tersebut telah terisi, bandingkan isi tersebut dengan nilai bobot yang diambil pada langkah 1.
5. Jika hasil perbandingan langkah ke 4 isi sebelumnya lebih kecil, tukar isinya.
6. Jika isi sebelumnya lebih besar, masuk ke langkah 8.
7. Jika hasil dari langkah ke 3 pada alamat matrik B masih kosong, letakkan nilai bobot dari langkah ke 1 ke dalamnya.
8. Jika tidak lolos cek dari langkah 2, ulangi langkah 1s/d 7 dengan menambah n dengan 1 (setiap pertambahan n dipandang sebagai mencari nilai bobot yang kurang maksimum dibandingkan pencarian nilai bobot langkah sebelumnya).
9. Jika masih ada kegiatan yang belum terjadwal (tidak lolos semua cek) set bobotnya dengan –2 (ditandai sebagai kegiatan yang tidak berhasil dijadwalkan).
10. Tampilkan isi matrik B dengan format tampilan seperti matrik jadwal.

Proses pencarian nilai maksimum dilakukan den
gan melakukan searching pada matrik matakuliah untuk menemukan nilai terbesar dan kemudian diletakkan dalam matrik jadwal. Apabila pada ruang dimaksud sudah ada isinya maka akan dibandingkan. Bila lebih besar tempat tersebut akan ditukar dan bila lebih kecil akan dicari ruang lain.

Langkah persiapan/inisialisasi

Setiap matakuliah diberi tanda dengan nilai :

- 0 = Belum dijadwalkan/telah diset lagi ke-0 karena telah digeser MK lain.
- 1 = Telah dijadwalkan.
- -1 = Tidak dapat dijadwalkan dalam suatu loop.
- -2 = Tidak dapat dijadwalkan sama sekali.

```
InitMtxRg; // Inisialisasi Matrix Ruang
```

```
bypass := false;
repeat
  qyLoop.Close;
  qyLoop.Open;
  frmProc.pbAll.Progress := JmlMk-qyLoop['Count'];
  { qyLoopCount menghitung cek = 0 dan cek = -1 }
  byPass := qyLoop['Count'] = 0;
  counter := 0;

  qyJdwk1.First;

  while not qyJdwk1.EOF and not bypass do
  begin
    inc (counter);
    idxRg := 0;
    idxMK := 0;
    maks  := 0;
    Sets  := false;
    frmProc.pbAss1.Progress := counter;

    // nilai  0 = Belum dijadwalkan/telah diset lagi ke-0 karena telah digeser MK lain.
    // nilai  1 = Telah dijadwalkan.
    // nilai -1 = Tidak dapat dijadwalkan dalam suatu loop.
    // nilai -2 = Tidak dapat dijadwalkan sama sekali.
    Nmr := qyJdwk1NoJdwk.Value;
    // Proses assignment hanya pada MK dg cek 0 & -1, bukan 1 atau -2
```

Tabel 1. Kode sumber langkah-langkah algoritma assignment

Langkah 1dan 2
if (qyJdwk1Cek.Value = 0) then
maks := SearchMKMax (Nmr) // cari nilai bobot maksimal
else if (qyJdwk1Cek.Value = -1) then
maks := SearchMKLower (Nmr, qyJdwk1NoPrio.value);
 	
if maks = -1 then
setJdwk (Nmr, -2, 1)
else
if (qyJdwk1Cek.Value =0) or (qyJdwk1Cek.Value =-1) then
repeat
// Awal proses penjadwalan
// Cari indeks nilai maksimal
if (SearchIdxMKMax (idxMK, nmr, maks) and
SearchIdxRg (idxRg, MtxMK [idxMK].NoRg, MtxMK [idxMK].Nohari,
MtxMK [idxMK].NoJam)) then
begin
sets := true;
// Cek SKS pada ruang & hari
if Sets then sets := not CekHrRg (idxMK, idxRg);
	langkah 4, 5 dan 7:
	// Jika sudah ada assignment sebelumnya
	else if sets and (MtxRg [idxRg].NoJdwk <> 0) then
	begin
	// Jika MK sebelumnya lebih kecil bobotnya
	if MtxRg [idxRg].Bobot < MtxMK [idxMK].Bobot then
	begin
	// Penghapusan data ruang yang lama
	SetJdwk (MtxRg [idxRg].NoJdwk, 0, idxMk);
	SetMtx (idxMk, idxRg, reset);
	SetJdwk (Nmr, 1, idxMk);
	SetMtx (idxMk, idxRg, Change);
	end // end if i := 0
	langkah 6
	// Jika MK sebelumnya lebih besar bobotnya
	else if MtxRg [idxRg].Bobot >= MtxMK [idxMK].Bobot then
	sets := false;
	langkah 8 :
	// !!! Jika IdxMK = MaxMtx, maka cek = -1 !!!

ISBN. 978-602-99334-0-6

```
        if sets then sets := not CekBtsHr
    (idxMk);
        if sets then sets := not CekPref (idxMk);
        if sets then sets := not CekSmt (idxMK);
        if sets then sets := not CekDsn (idxMK,
    idxRg);
        if sets then sets := not CekRuntun
    (idxMK, idxRg);
langkah 3 :
        // !!! Jika belum ada assignment !!!
        if sets and (MtxRg [idxRg].NoJdwk = 0)
         then
        begin
          SetJdwk (Nmr, 1, idxMk);
          SetMtx (idxMk, idxRg, change);
        end   // end if belum ada assignment
```

```
        // !!! Ini berarti tidak dapat dijadwalkan
           !!!
        begin
          SetJdwk (Nmr, -1, idxMk);
          sets := true; // bypass to next schedule
        end;
langkah 9 :
    if maks = -1 then
      setJdwk (Nmr, -2, 1)
    else
      if      (qyJdwk1Cek.Value      =0)      or
          (qyJdwk1Cek.Value =-1) then
      repeat
        // Awal proses penjadwalan
            ……..
            ……..
Langkah 10
begin
              RptJdwk := TRptJdwk.Create (self);
              RptJdwk.qyRptJdwk.Close;
              RptJdwk.qyRptJdwk.Open;
              RptJdwk.Preview;
              RptJdwk.Free;
end;
```

Pada penjadwalan sebelum meletakkan nilai maksimum pada ruang diperlukan beberapa cek, ada beberapa cek yang harus dilalui, yaitu :

a. cek tumbukan semester

Untuk sistem perkuliahan di perguruan tinggi, matakuliah-matakuliah yang diajarkan pada semester yang sama tidak dapat dijadwalkan pada waktu (hari dan jam) yang bersamaan.

b. cek tumbukan dosen pengampu

Cek tumbukan dosen diperlukan karena dosen dimungkinkan mengajar lebih dari satu matakuliah. Maka waktu mengajar matakuliah-matakuliah dari dosen tidak dapat dijadwalkan pada waktu (hari dan jam) yang bersamaan.

c. cek preferensi dosen

Didalam menjadwalkan suatu matakuliah harus memperhatikan preferensi dosen, yaitu waktu dimana dosen tidak dapat mengajar. Sehingga suatu matakuliah tidak dapat dijadwalkan pada waktu(hari dan jam) dimana dosen pengampunya tidak dapat mengajar.

KESIMPULAN

1. Pembobotan dilakukan dengan memberi bobot pada setiap rusuk graf *bipartite* dengan menggunakan rumus : $\frac{peserta}{kapasitas} x \ 100\%$. Hasil perhitungan akan maksimum / mencapai 100 % bila jumlah peserta matakuliah = kapasitas ruang fisik yang ditempati. Angka 100 % dihargai dengan bobot 100. Dibawah dan diatas 100 % bobot dibuat menurun karena penggunaan ruang tidak maksimal.
2. Proses assignment pada dasarnya merupakan proses mencari bobot maksimum yang sesuai (optimum) di dalam matiks kegiatan Vs ruang untuk kemudian menempatkannya dalam matriks jadwal.
3. Tidak semua data penjadwalan berhasil diletakkan ke dalam matrik jadwal, sehingga diberikan tanda pada data matakuliah dengan : (-2) sebagai tidak dapat dijadwalkan sama sekali, (-1) sebagai tidak dapat dijadwalkan dalam satu putaran, (1) sebagai telah berhasil dijadwalkan dan (0) sebagai belum dijadwalkan/telah digeser oleh matakuliah lain.

4. Dalam algoritma *assignment*, sebelum meletakkan nilai maksimum pada ruang diperlukan beberapa cek seperti cek tumbukan semester, cek tumbukan dosen pengampu, cek preferensi dosen.

DAFTAR PUSTAKA

Harnadi, B., 2010, " *Perancangan Algoritma Assignment pada Proses Penjadwalan Berdasarkan Data Matrik dari Graph Bipartite Matching* ", Prosiding EECCIS Teknik Elektro UNIBRAW Malang, Vol.II, E.8.1-E.8.4.

Harnadi, B., Susianto, F., 2006, " *Penjadwalan Mata Kuliah di Perguruan Tinggi*", Media Teknik *FT. UGM*, Th. XXVIII, no. 1.

Purwanto, E.B., *Perancangan dan Analisis Algoritma*, Graha Ilmu, Yogyakarta, 2008.

Wikipedia, *Assignment_problem*, Waktu akses April 2010, <URL:http://en.wikipedia.org/wiki/Assignment_problem> .

Wikipedia, *Bipartite-graph*, Waktu akses April 2010, <URL:http://en.wikipedia.org/wiki/Bipartite_graph> .

Wikipedia, *Matching_(graph_theory)*, Waktu akses April 2010, <URL:http://en.wikipedia.org/wiki/Matching_(graph_theory)> .

Wikipedia, *Optimisasi*, Waktu akses April 2010, <URL: http://id.wikipedia.org/wiki/Optimisasi> .

ISBN. 978-602-99334-0-6