

Optimasi Penjadwalan Sumberdaya dengan Metode Algoritma Genetik dan Algoritma Momen Minimum

Biemo W. Soemardi¹⁾
Dede Sumirto²⁾

Abstrak

Penjadwalan sumberdaya merupakan salah satu aspek penting dalam pengendalian dan penjadwalan proyek. Metoda-metoda penjadwalan yang dikembangkan saat ini, seperti pendekatan coba-coba, algoritma heuristic, atau algoritma momen minimum, telah mampu menjawab problema penjadwalan sehingga fluktuasi penggunaan sumberdaya dapat diminimalkan. Selain penjadwalan sumberdaya dan problem perimbangan biaya-waktu, optimisasi dalam pencarian solusi terhadap dua fungsi objektif juga sulit dipecahkan melalui suatu program matematis sederhana. Tulisan ini membahas upaya menerapkan pendekatan algoritma genetik guna mencari solusi optimal dari problema penjadwalan sumberdaya dengan menggunakan pendekatan perimbangan biaya-waktu. Upaya mencari solusi optimal dilakukan dengan pendekatan algoritma momen minimum melalui penghitungan iteratif faktor peningkatan. Pada penelitian ini ditunjukkan bahwa solusi mendekati optimal terhadap deviasi sumberdaya dan biaya total minimal dapat dicapai secara bersamaan.

Kata-kata kunci: Penjadwalan sumberdaya, keseimbangan biaya-waktu, algoritma genetik, algoritma momen minimum.

Abstract

Resource scheduling is one of the most important aspects of project control and scheduling. Existing methods such as Trial And Error Approach, Heuristics Algorithms, and Minimum Moment Algorithms, have the ability to solve resource scheduling problems, by means of minimizing fluctuations of resource utilization. In addition to resource scheduling and time-cost trade-off problems, the optimization for searching the optimal solution of two objective functions can hardly be solved by using simple mathematical programming. This paper presents an attempt to implement Genetic Algorithms approach in pursuit of finding optimal solution for resource scheduling problem by also considering the time-cost trade-off problems. Effort to find the optimal solution has been developed using minimum moment algorithm approach through the iterative calculation of improvement factor. The result shows that a near optimal solution for both resource schedule deviation and minimum cost can be achieved simultaneously.

Keywords: Resource scheduling, time-cost trade-off, genetic algorithm, minimum moment algorithm.

1. Pendahuluan

Penjadwalan merupakan salah satu aspek pengelolaan proyek yang memegang peran penting dalam upaya mencapai keberhasilan penyelesaian proyek. Seperti halnya dengan kegiatan-kegiatan di industri lain, selain pembiayaan, penjadwalan pada proyek konstruksi merupakan hal penting yang selalu menjadi perhatian utama dari pengelola proyek. Selain bertujuan untuk mengoptimalkan biaya dan waktu, penjadwalan juga dimanfaatkan untuk mengatur pemanfaatan sumberdaya. Masalah penjadwalan sumberdaya (*resource scheduling*) mencakup pengalokasian sumberdaya sehingga tidak terjadi fluktuasi yang besar antara sumberdaya yang

dibutuhkan pada suatu waktu dengan waktu-waktu berikutnya. Masalah ini dikenal sebagai masalah perataan sumberdaya atau *resource leveling*.

Fluktuasi yang berlebihan antara penggunaan sumberdaya pada suatu waktu dengan waktu-waktu lainnya akan mengakibatkan peningkatan biaya hiring and firing. Guna memperoleh solusi optimal terhadap penjadwalan sumberdaya, berbagai metoda telah dikembangkan, baik melalui pendekatan coba-coba (*trial and error approach*), hingga penerapan algoritma-algoritma terstruktur, seperti algoritma momen minimum (*minimum moments algorithms*), yang kesemuanya itu bertujuan untuk meminimumkan fluktuasi penggunaan sumberdaya.

-
1. Anggota KK Manajemen & Rekayasa Konstruksi, FTSL-ITB, Jl. Ganesha No.10 Bandung 40132.
 2. Mahasiswa Program Magister MRK, FTSL-ITB, Jl. Ganesha No.10 Bandung 40132.

2. Metoda Penjadwalan Sumberdaya

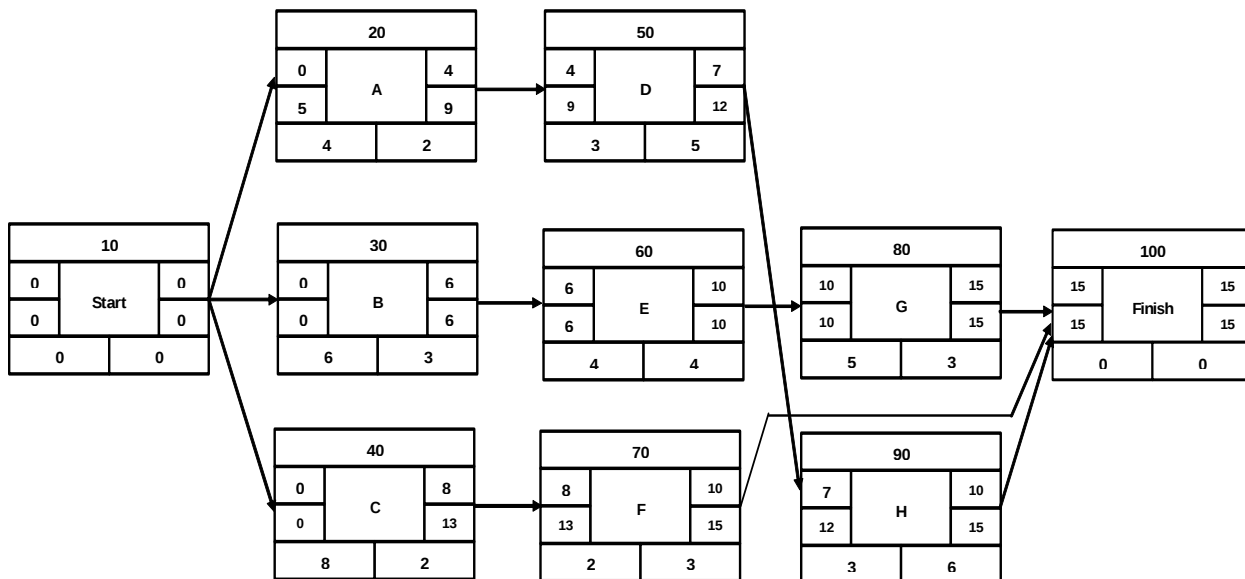
Salah satu pioneer dalam studi penjadwalan sumberdaya adalah Harris (1978), yang membuat algoritma untuk melakukan perataan sumberdaya dengan cara mencari distribusi yang seragam atas penggunaan sumberdaya setiap hari pada suatu proyek dalam waktu proyek yang telah ditentukan, dengan menggunakan indikator yang disebut *Improvement Factor*. Metode ini dikenal sebagai metode algoritma minimum momen (*Minimum Moment Algorithm*). Namun, pada prosedur ini tidak melibatkan biaya dan dalam melakukan perataan sumberdayanya, tetapi hanya melakukan penggeseran kegiatan-kegiatan pada jalur non-kritis saja, tanpa melakukan kemungkinan pemendekan dan pemanjangan waktu suatu kegiatan. Selanjutnya penelitian yang berkaitan dengan penjadwalan sumberdaya antara lain dilakukan oleh Chan, et.al. (1996), yaitu dengan membuat prosedur untuk meminimumkan deviasi antara sumberdaya yang tersedia dengan sumberdaya yang dibutuhkan. Pada penelitian ini prosedur yang dibuat belum melibatkan biaya sebagai parameter yang ikut diperhitungkan. Selanjutnya Feng, et.al. (1997), melakukan penelitian berkaitan dengan masalah *time-cost trade-off*, yaitu dengan mencari hubungan antara waktu dan biaya

yang optimal di antara beberapa alternatif yang mungkin untuk menyelesaikan suatu proyek dalam waktu yang telah ditetapkan. Upaya untuk mencari solusi optimum dari persoalan alokasi sumberdaya juga dilakukan oleh Hegazy (1999) yang meneliti masalah alokasi dan perataan sumberdaya, yaitu dengan mencoba membuat prosedur pencarian sekumpulan prioritas kegiatan (misalnya dari *Highest* sampai *Lowest*) yang menghasilkan waktu dan fluktuasi antara kebutuhan sumberdaya proyek yang optimum secara bersamaan. Namun demikian, penelitian ini tidak menyertakan masalah dilema antara waktu dan biaya (*Time-Cost Trade-Off Problem*).

Dengan merujuk pada penelitian-penelitian yang telah dilakukan, salah satu permasalahan yang masih memerlukan solusi lebih baik adalah yang berkaitan dengan pengalokasian sumberdaya yang terbatas secara efisien sehingga menghasilkan fluktuasi sumberdaya dan biaya yang minimum dalam waktu penyelesaian proyek yang telah ditentukan.

2.1 Problema penjadwalan sumberdaya

Contoh dari jaringan kerja penjadwalan sederhana ditunjukkan pada **Gambar 1** berikut:



Keterangan :

No. Kegiatan			
ES	Kegiatan	EF	
LS		LF	
Durasi		SD	

ES : Early Start Time
 EF : Early Finish Time
 LS : Late Start Time
 LF : Late Finish Time
 SD : Sumber Daya

Gambar 1. Contoh jaringan kerja penjadwalan sederhana

Secara tradisional penjadwalan sumberdaya dimulai dengan menggambar diagram batang (*bar-chart*) yang mencantumkan urutan kegiatan sesuai posisinya dalam rentang waktu tertentu. Pada diagram batang tersebut dapat dicantumkan jumlah sumberdaya yang terdistribusi sepanjang rentang waktu pada masing-masing kegiatan. Dengan menjumlahkan seluruh sumberdaya pada rentang waktu tertentu maka dapat diperoleh akumulasi sumberdaya yang terjadwal ada rentang waktu tersebut; seperti yang tampak pada contoh penjadwalan pada **Gambar 2**.

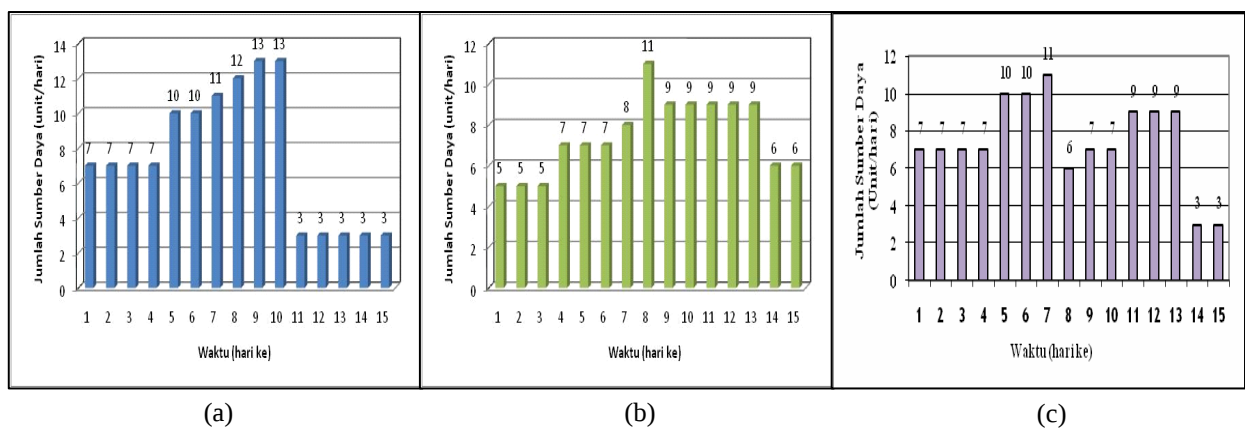
Pada **Gambar 2** tersebut terlihat bahwa jumlah distribusi sumberdaya yang terjadwalkan pada masing-masing berfluktuasi. Fluktuasi sumberdaya ini tentunya akan membebani proyek, baik dari sisi proses pengadaan maupun biayanya. Untuk itu diperlukan penjadwalan kembali yang mampu mengurangi fluktuasi tersebut. Langkah pertama dalam penjadwalan penggunaan sumberdaya adalah mengidentifikasi waktu luang (*float*) yang tersedia

dari kegiatan yang paling akhir dan kegiatan yang paling awal, untuk mengetahui sejauh mana kegiatan-kegiatan tersebut dapat dipindahkan/digeser jadwalnya. Langkah ini sangat penting agar ketika kegiatan-kegiatan tersebut dipindahkan, perubahan *float* yang diakibatkan oleh pemindahan kegiatan-kegiatan tersebut dapat diketahui. Begitu pun jumlah sumberdaya direvisi akibat perpindahan kegiatan-kegiatan. Begitu seterusnya, sampai didapatkan distribusi sumberdaya yang lebih baik.

Untuk melihat perubahan distribusi sumberdaya sebelum dilakukan perataan dan sesudah dilakukan perataan dengan pendekatan manual dapat dilihat pada **Gambar 3 (a), (b) dan (c)**. Walaupun pendekatan secara manual memungkinkan untuk mendapatkan distribusi penggunaan sumberdaya yang relatif baik, pendekatan ini berpotensi menemui kesulitan jika kegiatan yang terlibat memiliki jumlah yang sangat banyak.

Kegiatan	Sumberdaya (unit/hari)	Waktu														
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
A	2															
B	3															
C	2															
D	5															
E	4															
F	3															
G	3															
H	6															
Total Sumberdaya		7	7	7	7	10	10	11	12	13	13	3	3	3	3	3

Gambar 2. Penjadwalan sumberdaya pada kondisi *early start*



Gambar 3. Distribusi sumberdaya

2.2 Algoritma momen minimum

Algoritma minimum momen merupakan suatu proses sistematis yang digunakan untuk melakukan perataan sumberdaya dengan cara mengukur tingkat perbaikan dalam usaha meratakan sumberdaya. Algoritma momen minimum merupakan suatu rangkaian proses yang mencoba seluruh kemungkinan perbaikan jadwal suatu kegiatan dan pengaruh perbaikan tersebut terhadap distribusi sumberdaya proyek. Tingkat perbaikan ini dikenal sebagai *Improvement Factor* (IF). Tujuan dari algoritma momen minimum ini adalah mendapatkan suatu distribusi sumberdaya harian yang seragam dari suatu proyek.

Improvement Factor (IF) merupakan suatu nilai yang menentukan baik atau buruknya distribusi sumberdaya yang diakibatkan oleh penggeseran suatu kegiatan pada suatu rangkaian proyek. Semakin besar nilai IF suatu kegiatan akibat penggeseran kegiatan tersebut, maka semakin baik atau semakin seragam distribusi sumberdaya yang diakibatkan oleh penggeseran kegiatan tersebut untuk perbaikan jadwal. Indikator yang dapat digunakan dalam menentukan kelayakan suatu jadwal adalah *Improvement Factor* - IF (Harris, 1978), yang dinyatakan sebagai:

$$IF_{A,D} = r \left(\sum_1^m x_i - \sum_1^m w_i - mr \right) \quad (1)$$

dimana:

- IF_{A,D} : *Improvement Factor* untuk menggeser kegiatan A sebanyak D hari
 r : Tingkat sumberdaya harian kegiatan (unit/hari)
 m : Jumlah hari minimum yang digeser (hari)
 x_i : Jumlah sumberdaya harian yang akan berkurang akibat penggeseran (unit)
 w_i : Jumlah sumberdaya harian yang bertambah akibat penggeseran tersebut (unit)

Kegiatan pada suatu proyek yang berada pada jalur non-kritis, memiliki nilai *free float*. Kegiatan-kegiatan tersebut dapat digeser sejauh jumlah hari ketersediaan *free float*-nya. Selama nilai IF-nya memiliki nilai positif, maka histogram sumberdaya dapat diperbaiki. Ini artinya, penggeseran kegiatan-kegiatan dalam rangkaian kegiatan tersebut dapat dilakukan. Sebelum melakukan penggeseran kegiatan, terlebih dahulu ditentukan tahap-tahap kegiatan sesuai dengan urutan kegiatannya, yang akan menentukan kegiatan mana yang akan digeser pada langkah pertama, kedua dan seterusnya. Hasil perataan sumberdaya dengan pendekatan algoritma momen minimum dapat dilihat pada **Gambar 3(c)**. Perataan dengan pendekatan ini menghasilkan IF sebesar +66.

Meskipun berhasil mendapatkan jadwal sumberdaya yang relative cukup merata, pendekatan ini belum sepenuhnya memanfaatkan peluang *float* yang ada. Dalam pendekatan ini tidak dilakukan pemendekan maupun pemanjangan durasi kegiatan dengan memanfaatkan float yang ada. Pendekatan ini hanya melakukan kemungkinan penggeseran kegiatan dengan memanfaatkan *float* yang ada.

2.3 Keseimbangan biaya-waktu

Masalah klasik yang berkaitan dengan penjadwalan sumberdaya ini dikenal sebagai masalah *time-cost trade-off*. Dalam hubungan antara biaya dan waktu, biaya tidak langsung pada suatu proyek akan selalu meningkat seiring dengan perjalanan waktu, sementara biaya langsungnya akan cenderung berkurang. Kombinasi keduanya akan memberikan suatu titik optimal dimana pada suatu waktu (durasi) tertentu kegiatan atau proyek akan menghasilkan biaya total yang paling rendah.

Biaya total proyek meliputi biaya langsung (*direct cost*) dan biaya tak langsung (*indirect cost*). Biaya langsung sangat tergantung pada jumlah pemakaian sumberdaya untuk menyelesaikan suatu kegiatan, sedangkan biaya tak langsung berbanding lurus dengan lama waktu penyelesaian suatu kegiatan. Jika Rr_{jk} adalah sumberdaya ke-j yang dibutuhkan pada hari ke-k dan P_{dj} adalah biaya langsung sumberdaya ke-j, maka biaya total langsung (C_d) dapat diformulasikan sebagai:

$$C_d = \sum_{j=1}^m \sum_{k=1}^n Rr_{jk} * P_{dj} \quad (2)$$

Jika d adalah waktu penyelesaian suatu kegiatan dan P_{ind} menyatakan biaya tak langsung per hari, maka biaya tak langsung (C_{ind}) dapat dinyatakan sebagai:

$$C_{ind} = \sum d * P_{ind} \quad (3)$$

sehingga biaya total proyek (C) dinyatakan sebagai:

$$C = C_d + C_{ind} \quad (4)$$

Jika Rr_{jk} menyatakan jumlah sumberdaya ke-j yang dibutuhkan pada hari ke-k dan Ra_j adalah jumlah sumberdaya ke-j yang tersedia per hari, dan $\sum d$ adalah durasi total proyek, maka jumlah deviasi sumberdaya (ΣR_d) dinyatakan sebagai:

dimana:

$$\sum R_d = \sum_{j=1}^m \left(\sum_{k=1}^n Rr_{jk} - Ra_j \right)^2 \quad (5)$$

$$Ra_j = \frac{\sum_{k=1}^n Rr_{jk}}{\sum d} \quad (6)$$

Dengan demikian maka objektif penjadwalan sumberdaya dalam suatu rangkaian jaringan jadwal proyek dapat dirumuskan menjadi:

(a) meminimumkan biaya total proyek:

$$C = C_d + C_{ind} \quad (7)$$

dan (b) meminimalkan deviasi sumberdaya:

$$\sum R_d = \sum_{l=1}^n \left(\sum_{j=1}^m Rr_{jl} - Ra_k \right)^2 \quad (8)$$

3. Optimasi Multiobyektif dengan Prosedur Algoritma Genetik

Untuk memecahkan masalah ini, suatu teknik pencarian yang berbasis pada *artificial intelligence*, algoritma genetik (*Genetic Algorithms*), telah digunakan. Algoritma genetik, yang dipopulerkan oleh Holland, (1989), merupakan suatu pendekatan yang meniru proses seleksi alamiah dan genetika dalam proses reproduksi, telah berhasil diadopsi untuk memecahkan masalah-masalah sains dan bidang rekayasa dan telah terbukti dapat memecahkan masalah-masalah diskrit yang mungkin sulit dilakukan dengan pendekatan matematik. Dalam problem penjadwalan sumberdaya, algoritma genetik telah banyak memperoleh perhatian dan diaplikasikan oleh berbagai peneliti seperti Toklu (2002), Senouci et.al. (2004), Kim dan Ellis Jr (2005), serta Liu, at.al. (2005).

Metode optimasi tradisional pada umumnya memakai sebuah titik solusi (satu individu) dalam mengeksplorasi ruang desain, dan meloncat dari sebuah titik solusi ke titik solusi tunggal berikutnya pada proses iterasinya. Algoritma genetik memakai banyak titik solusi (banyak individu) dalam ruang desain pada setiap iterasinya, masing-masing individu tidak saling bergantung, sehingga dapat diproses secara paralel dan memungkinkan algoritma genetik terhindar dari jebakan optimum lokal. Meskipun tidak menjamin hasil yang diperolehnya optimum global algoritma genetik mampu mengidentifikasi daerah optimum global secara cepat. Konvergensi algoritma genetik pada hasil optimum tertentu tidak dapat diperlihatkan secara matematik, karena algoritma genetik bukan prosedur yang berdasarkan pada pendekatan matematis. Meskipun demikian karena

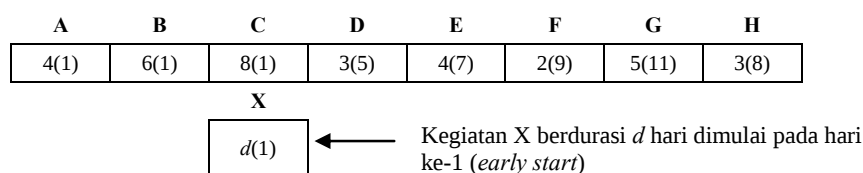
algoritma genetik memberikan hasil yang lebih baik dibanding metode lain, algoritma genetik banyak dipakai pada masalah optimasi penjadwalan sumberdaya, yang merupakan sesuatu yang mungkin sulit dilakukan dengan metode lainnya, seperti program linier, integer programming ataupun *dynamic programming*, karena algoritma genetik mampu menyelesaikan masalah optimasi fungsi diskrit.

3.1 Prosedur penjadwalan sumberdaya dengan algoritma genetik

Prosedur algoritma genetik dimulai dengan merumuskan masalah dan mengkodekannya, kemudian membangkitkan secara random kode string sejumlah individu generasi pertama sebuah populasi. Masing-masing individu dihitung nilai kebugarannya, yang bergantung pada nilai fungsi sasarannya. Individu dengan nilai kebugaran lebih besar mempunyai peluang yang lebih besar untuk tetap tinggal di populasi, bahkan mungkin mempunyai salinan. Sedangkan individu dengan nilai fitness terkecil mungkin mempunyai peluang terbesar untuk mati atau musnah dari populasi karena kalah bersaing. Proses ini dilakukan oleh operator seleksi alamiah. Proses berikutnya dilakukan oleh operator kawin silang yang mengkombinasikan gen-gen dua individu untuk menghasilkan keturunan. Bilangan acak dipakai pada proses seleksi alamiah dan kawin silang. Anak hasil kawin silang diharapkan mempunyai nilai kebugaran lebih besar dibanding orang tuanya. Anak hasil kawin silang ini disebut generasi kedua. Prosedur ini berlangsung terus untuk menghasilkan generasi ketiga, keempat dan seterusnya. Proses dihentikan bila semua individu atau sebagian besar individu dalam populasi telah mempunyai string yang sama, maka proses dihentikan. Umumnya batasan konvergen tercapai bila 95% atau lebih individu dalam populasi telah mempunyai string yang sama.

3.1.1 Penyusunan struktur gen

Dalam pekerjaan menyusun struktur gen, termasuk didalamnya melakukan pengkodean. Pengkodean suatu masalah ke dalam algoritma genetik merupakan problem utama. Kode yang optimum sangat sulit dibuat, karena untuk menghasilkannya identik dengan memecahkan masalah utamanya sendiri (Holland, 1992). Berikut ini contoh penyusunan struktur gen serta pengkodeannya untuk permasalahan penjadwalan sumberdaya:



Gambar 5. Struktur gen satu individu dalam penjadwalan sumberdaya

3.1.2 Menentukan fungsi objektif

Selain pengkodean, fungsi obyektif atau fungsi kebugaran (*fitness function*) merupakan aspek yang sangat penting dalam algoritma genetik. Individu akan mati atau berkembang biak bergantung pada nilai kebugarannya. Nilai kebugaran sangat bergantung pada nilai obyektifnya. Pada masalah minimasi, individu dengan nilai fungsi obyektif makin kecil akan mempunyai nilai kebugaran makin besar. Masalah optimasi penjadwalan sumberdaya ini pun termasuk masalah minimasi, yaitu jika nilai sasarannya makin kecil maka akan mempunyai nilai kebugaran yang makin besar, sehingga kemungkinan individu ini bertahan dalam populasi semakin besar. Masalah penjadwalan sumberdaya pada penelitian ini melibatkan dua parameter penting, yaitu distribusi sumberdaya dan biaya total proyek. Pada penelitian ini akan dicoba mengakomodasi dua keinginan dalam melakukan penjadwalan sumberdaya, yaitu keinginan memperoleh fluktuasi sumberdaya yang minimum dan biaya yang minimum sehingga proyek dapat diselesaikan dalam waktu yang telah ditentukan.

Dalam genetika algoritma ini fungsi obyektif pada optimasi masalah penjadwalan sumberdaya dirumuskan berdasarkan gabungan dari **Persamaan (7)** dan **Persamaan (8)**. Namun, karena prosedur dalam algoritma genetik dilakukan melalui proses iteratif untuk mencari individu terbaik dengan membandingkan individu generasi pertama (generasi ke-0) dan individu-individu dari generasi berikutnya (generasi ke-i), maka fungsi obyektifnya menjadi:

$$FS = W_d \cdot \frac{\sum R d_i}{\sum R d_0} + W_c \frac{C_i}{C_0} \quad (9)$$

dimana $\sum R d_i$ adalah jumlah deviasi sumberdaya generasi ke-i, dan C_i adalah biaya pada generasi ke-i, dan $W_d + W_c = 1$, W_d = bobot prioritas optimasi

terhadap deviasi sumberdaya, sedangkan W_c = bobot prioritas optimasi terhadap biaya. Penetapan bobot prioritas terhadap deviasi sumberdaya dan biaya total memberikan keluasaan analisis berbagai alternatif solusi, sesuai dengan prioritas yang ditetapkan dalam pengelolaan proyek.

3.1.3 Pembangkitan generasi orang tua

Generasi orang tua dapat dipilih dengan memilih dua individu dengan gen-gen berbeda. Atau, generasi orang tua dapat dipilih dari dua rangkaian kegiatan dengan karakteristik kegiatan-kegiatan yang berbeda. Untuk contoh kasus di atas dapat dipilih dua rangkaian kegiatan yang mungkin terjadi dalam jaringan penjadwalan yang dikaji.

Berbeda dengan struktur gen orang tua 1 yang dengan aktivitas A yang dimulai pada hari pertama, gen orang tua 2 tersusun dengan aktivitas A dimulai pada hari kedua, dengan memanfaatkan satu hari float (lihat **Gambar 6**).

3.1.4 Mekanisme reproduksi untuk menghasilkan generasi turunan

Setelah memilih dua individu orang tua, maka dilakukan kawin silang (*crossover*) atau mutasi (*mutation*) untuk menghasilkan generasi turunan (*offspring*). Anak hasil kawin silang dua individu yang berbeda diharapkan mempunyai nilai *fitness* yang lebih besar dibanding orang tuanya. Teknik kawin silang yang lazim digunakan pada algoritma genetik adalah kawin silang satu lokasi, kawin silang dua lokasi, dan kawin silang merata. Pada kawin silang satu lokasi, pertukaran gen hanya ditentukan oleh satu lokasi saja. Pada kawin silang dua lokasi, perlu ditentukan secara random dua buah lokasi gen. Teknik kawin satu lokasi akan diterapkan pada penelitian ini. Sedangkan pada kawin silang merata, pertukaran gen dapat terjadi di sepanjang gen.

	A	B	C	D	E	F	G	H
Orang Tua 1	4(1)	6(1)	8(1)	3(5)	4(7)	2(9)	5(11)	3(8)
Orang Tua 2	4(2)	6(1)	8(3)	3(5)	4(7)	2(9)	5(11)	3(12)

Gambar 6. Struktur gen individu orangtua

	A	B	C	D	E	F	G	H
Orang Tua 1	4(1)	6(1)	8(1)	3(5)	4(7)	2(9)	5(11)	3(8)
					↑			
Orang Tua 2	4(2)	6(1)	8(3)	3(5)	4(7)	2(9)	5(11)	3(12)
Turunan 1	4(1)	6(1)	8(3)	3(5)	4(7)	2(9)	5(11)	3(12)
Turunan 2	4(1)	6(1)	8(3)	3(5)	4(7)	2(9)	5(11)	3(8)

Gambar 7. Proses reproduksi gen

Mutasi dilakukan pada individu turunan pada setiap generasi setelah kawin silang. Artinya penggeseran *early start* suatu kegiatan dilakukan setiap pertukaran bagian rangkaian kegiatan. Mutasi dilakukan dengan cara mengganti *early start* suatu kegiatan. Seperti tampak pada **Gambar 7**, pada individu orang tua 1, gen A memiliki *early start* pada hari ke-1, sedangkan pada individu orang tua 2, gen A memiliki *early start* pada hari ke-2. Kedua individu tersebut sebenarnya berasal dari individu yang sama, namun telah terjadi mutasi pada salah satu individu orang tua tersebut.

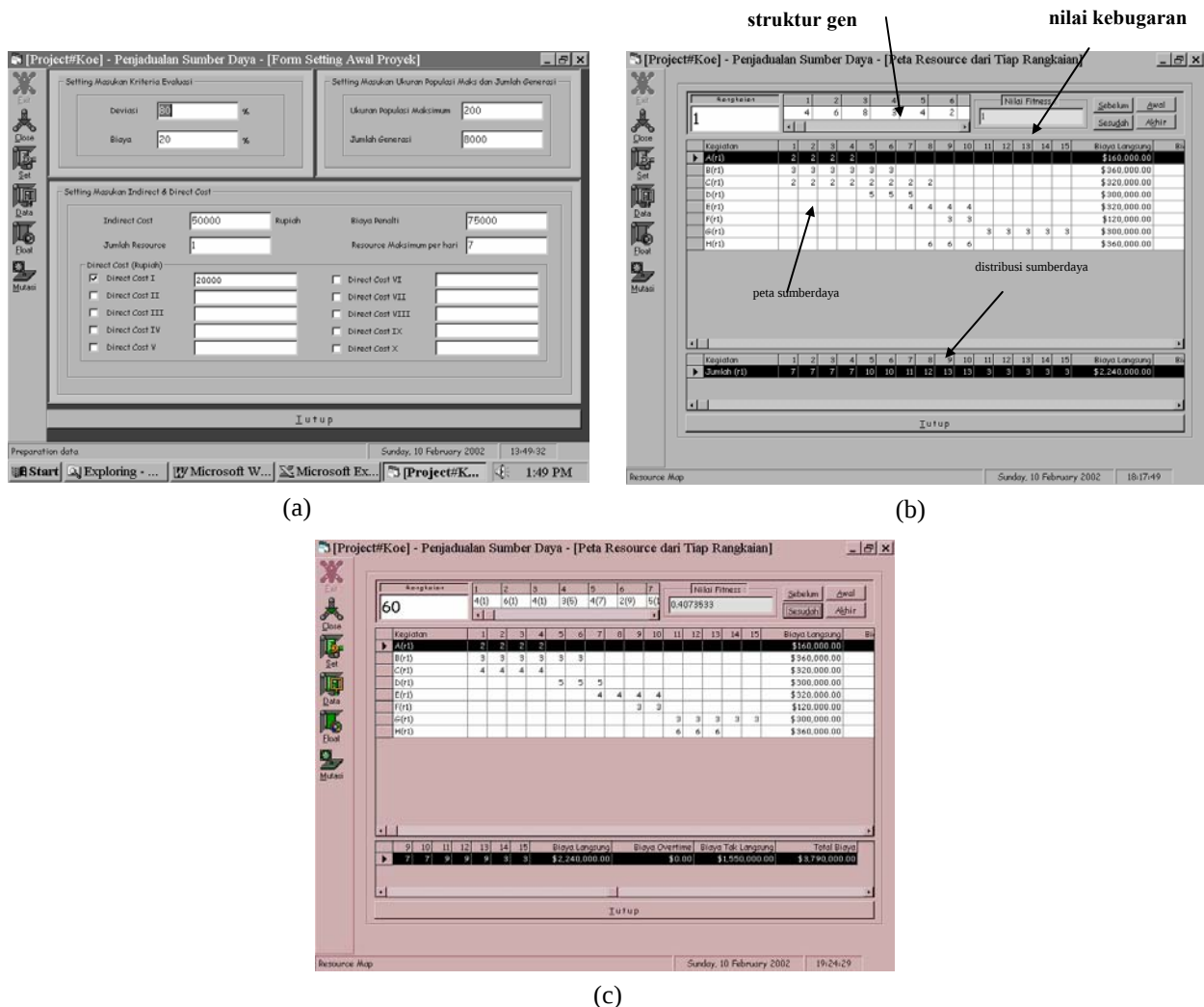
Angka mutasi umumnya kecil karena pada kenyataannya mutasi gen tidak diharapkan, karena secara biologis hal ini merugikan, sementara kawin silang dilakukan karena suatu individu ingin mendapatkan keturunan yang lebih baik.

3.2 Program algoritma genetik

Karena masalah penjadwalan tersebut akan mempunyai jumlah kemungkinan solusi yang sangat banyak, maka suatu program komputer algoritma genetik telah

disusun dalam bahasa Visual Basic. Dalam program tersebut selain menyediakan masukan berupa data penjadwalan (kegiatan, urutan kegiatan, dan durasi kegiatan), juga masukan berupa jumlah jenis sumberdaya, satuan biaya langsung P_d , satuan biaya tidak langsung P_{ind} , batasan sumberdaya maksimum $\max \Sigma R_{rjk}$, biaya penalti P_{ob} , serta masukan data bobot prioritas optimasi W_d & W_c . Untuk proses bangkitan dan reproduksi generasi turunan, program ini juga menyediakan masukan berupa ukuran populasi maksimum, jumlah generasi, probabilitas kawin silang (*Probability of Crossover* P_c), dan probabilitas mutasi (*Probability of Mutation* P_m).

Program algoritma genetik untuk penjadwalan sumberdaya (PSD) pada dasarnya terbagi menjadi dua kelompok pemrosesan. Kelompok pertama adalah program penjadwalan yang menghasilkan peta sumberdaya dan *float*, sementara program kedua memuat proses algoritma genetik untuk mencari solusi mendekati optimal untuk meminimalkan total biaya dan deviasi sumberdaya. Penjelasan lengkap mengenai program ini dapat dilihat pada laporan Sumirto (2002).



Gambar 8. Contoh layar program: (a) masukan (b) peta sumberdaya (c) hasil akhir

3.3 Contoh kasus

Sebagai ilustrasi aplikasi algoritma genetik pada penjadwalan sumberdaya, disajikan contoh kasus yang mengacu pada penjadwalan seperti yang tersaji pada **Gambar 1** dan **2**. Sebagai pelengkap penjadwalan tersebut ditetapkan bahwa biaya langsung, $P_d = \text{Rp } 20.000,- / \text{unit}$, sedangkan biaya tidak langsung $P_{ind} = \text{Rp } 50.000,- / \text{hari}$. Selain itu ditetapkan pula bahwa dalam kondisi normal batasan kapasitas pengolahan dan ketersediaan sumberdaya perhari, $\max \Sigma R_{rjk} = 7$ unit/hari sebesar, Jika ternyata dalam penjadwalan ternyata sumberdaya yang dibutuhkan melebihi 7 unit/hari, maka terpaksa dilakukan lembur dengan biaya $P_{ot} = \text{Rp } 75.000,- / \text{hari}$.

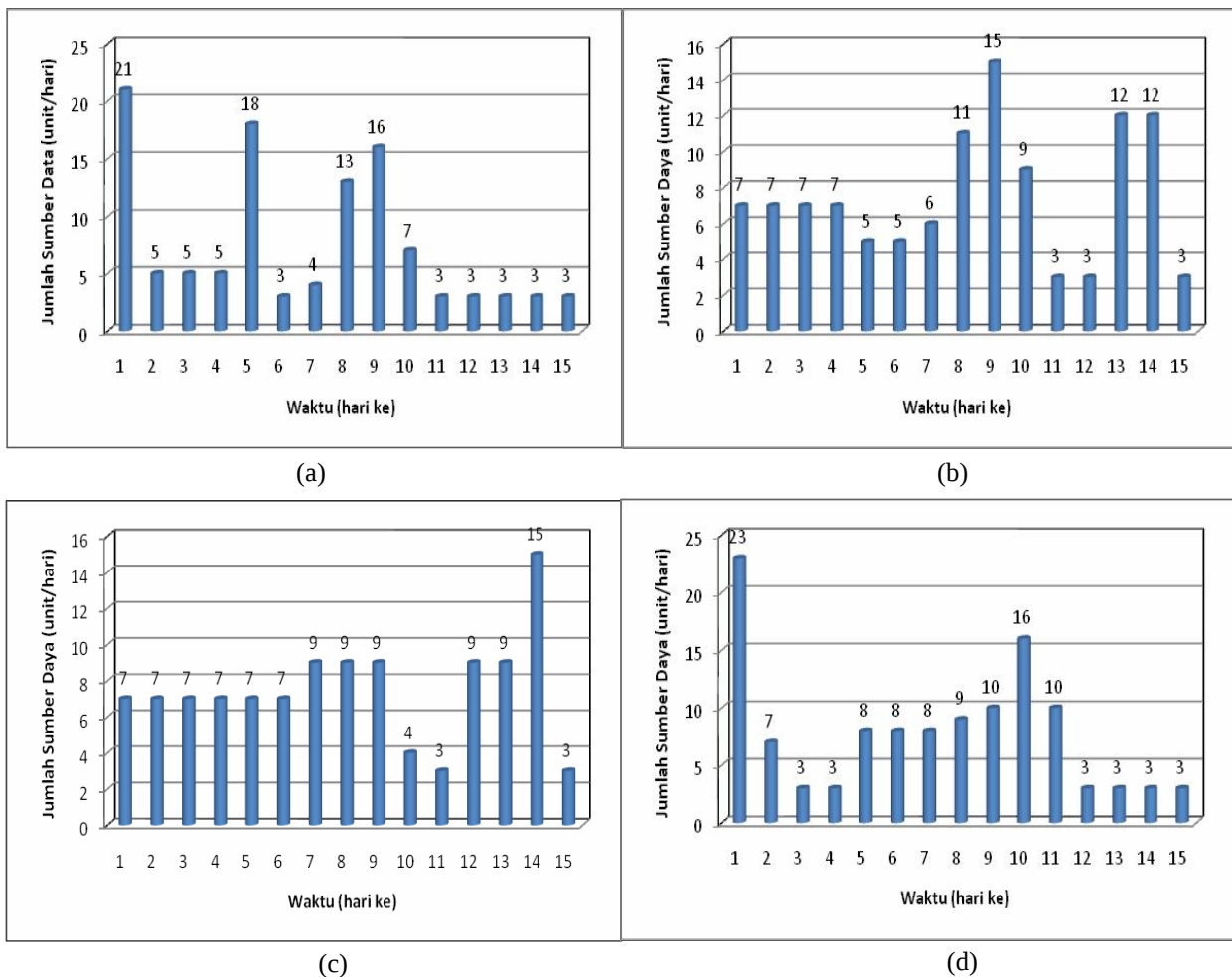
Untuk melengkapi karakteristik algoritma genetik, ditetapkan pula $P_c = 1$ dan $P_m = 0,001$ yang berarti setiap individu akan mengalami kawin silang setiap bertemu individu lainnya. Setiap gen pada contoh merupakan rangkaian 8 kegiatan (A, B, C, D, E, F, G, H) yang membentuk satu rantai jadwal kegiatan. Jika ditetapkan bahwa populasi maksimum ditetapkan sebanyak 114, maka dengan $P_m = 0,001$ maka akan terjadi mutasi pada gen ke 115 ($144 \times 8 \times 0,001$).

Dengan perbandingan bobot prioritas yang sama ($W_d = W_c$) dan melakukan generasi sejumlah 100, 1000, 2000, dan 8000 selanjutnya diperoleh hasil seperti yang tersaji pada **Tabel 1**.

Pada **Tabel 1** terlihat bahwa dengan pembobotan yang sama yaitu 50% untuk deviasi sumberdaya dan 50% untuk biaya total, semakin banyak jumlah generasi maka semakin kecil nilai *fitness* yang dihasilkan. Hal ini berarti semakin dekat solusi tersebut pada solusi global optimal. **Gambar 9 (a), (b), (c), dan (d)** menunjukkan distribusi sumberdaya hasil optimasi dari **Tabel 1** di atas untuk setiap jumlah generasi masing-masing.

Tabel 1. Hasil optimasi

Jumlah Generasi	Deviasi Sumberdaya, ΣR_d	Biaya, C	Nilai <i>Fitness</i> , F_s
100	484	4.645.000	0.7915
1000	218	3.985.000	0.6452
2000	191	4.110.000	0.5792
8000	125	3.640.000	0.5100



Gambar 9. Grafik komposisi pembobotan terhadap deviasi dan biaya total

Dengan komposisi prioritas yang berimbang antara deviasi sumberdaya dan biaya total, pada kondisi ini kita belum mengetahui apakah pada generasi ke 8000 tersebut, deviasi sumberdaya maupun biaya total, keduanya telah betul-betul mencapai nilai paling minimum; atau dengan kata lain keduanya telah mencapai nilai optimum global.

Selanjutnya untuk mencari solusi optimal global lainnya dapat dilakukan variasi komposisi pembobotan terhadap deviasi sumberdaya dan biaya total untuk jumlah generasi 1000.

Dari **Tabel 2** di bawah dapat dilihat bahwa komposisi pembobotan antara deviasi sumberdaya dan biaya total yang menghasilkan mendekati optimum global (*near optimal*) secara bersamaan adalah 90% untuk deviasi sumberdaya dan 10% untuk biaya total, dengan deviasi sumberdaya 75 dan biaya total Rp. 3.790.000, dengan nilai kebugaran terkecil yakni 0,3856. Hasil ini dapat dikatakan mendekati optimum global (*near optimal*) karena jika dibandingkan dengan jumlah generasi 8000 dengan pembobotan masing-masing untuk deviasi sumberdaya dan biaya total masing-masing 50% dapat menghasilkan biaya total yang lebih kecil yaitu Rp. 3.640.000, walaupun menghasilkan deviasi yang lebih besar yaitu 125.

Jika dibandingkan, hasil optimasi beberapa pendekatan dilihat dari besarnya deviasi sumberdaya dapat dilihat pada **Tabel 3**. Dari hasil ini terlihat bahwa pendekatan algoritma genetik merupakan salah satu pendekatan yang menghasilkan hasil yang lebih baik dibandingkan algoritma momen minimum walaupun pendekatan coba-coba tampak lebih baik. Meski demikian tentunya hal ini tidak berlaku jika optimasi terhadap dua fungsi objektif diperhitungkan secara bersamaan.

Tabel 3. Perbandingan hasil optimasi beberapa pendekatan

Pendekatan Optimasi	Deviasi Sumberdaya, ΣR_d
Coba-Coba	51
Algoritma Momen Minimum	79
Algoritma Genetik	75

Tabel 2. Hasil optimasi berbagai komposisi pembobotan dengan jumlah generasi 1000

Pembobotan $W_d : W_c$	Deviasi Sumberdaya, ΣR_d	Biaya, C (Rp.)	Nilai <i>Fitness</i> , F_s
50% : 50%	218	3.985.000	0.6452
60% : 40%	190	4.110.000	0.5712
70% : 30%	97	3.940.000	0.5182
80% : 20%	75	3.990.000	0.4657
90% : 10%	75	3.790.000	0.3856
100% : 0%	129	3.810.000	0.3301
40% : 60%	93	3.740.000	0.6516
30% : 70%	113	3.840.000	0.6770
20% : 80%	157	3.940.000	0.7431
10% : 90%	227	4.110.000	0.7767
0% : 100%	1085	4.490.000	0.7318

4. Kesimpulan

Metode optimasi dengan menggunakan algoritma genetik dapat digunakan untuk memecahkan masalah-masalah diskrit, yang mungkin tidak bisa dipecahkan dengan pendekatan matematik sederhana. Prosedur algoritma genetik yang digunakan pada penelitian Sumitro (2002) berhasil menggabungkan dua pendekatan yakni algoritma momen minimum dan pertimbangan biaya-waktu, sehingga prosedur ini memungkinkan komponen deviasi sumberdaya dan biaya dijadikan pertimbangan dalam pengambilan keputusan secara bersamaan.

Pengaruh pembobotan yang menghasilkan solusi yang mendekati optimum (*near optimal*) untuk deviasi sumberdaya dan biaya total proyek secara bersamaan dapat ditentukan dengan cara melakukan perubahan pembobotan kedua fungsi obyektif tersebut secara bergantian. Pendekatan ini memungkinkan untuk mengeksplorasi konsekuensi dari berbagai skenario prioritas penjadwalan sebagai landasan pengambilan keputusan dalam penjadwalan sumberdaya yang lebih tepat.

Daftar Pustaka

- Chan, W.T., Chua, D.K.H., & Kannan, G., 1996, "Construction Resource Scheduling with Genetic Algorithm", ASCE Journal of Construction Engineering and Management, 122(2),
- Feng, C.W, Liu, L., & Burn, S.A., 1995, "Using Genetic Algorithms to Solve Construction Time-Cost Trade-Off Problems", ASCE Journal of Computing in Civil Engineering, 111(3).
- Goldberg, D.E., 1989, "Genetics Algorithm in Search, Optimization and Machine Learning", Addison-Wesley, Reading, Massachusetts.
- Harris, R., 1978, "Precedence and Arrow Networking Techniques for Construction", John Wiley.
- Hegazy, T., 1999, "Optimization of Resource Allocation and Leveling Using Genetic Algorithms", Journal of Construction Engineering and Management, 125(3), 167-175.
- Holland, J., 1992, "Adaptation in Natural and Artificial System", MIT Press, Cambridge.
- Kim, Jin-Lee, & R.D. Ellis, Jr., 2005, "A Framework for Integrating Model of Resource-Constrained Scheduling Using Genetic Algorithm", Proceeding of the 2005 Winter Simulation Conference, 2119-2126.
- Liu, Y., Zhao S., Du, Z, & Li, S., 2005, "Optimization of Resource Allocation in Construction Using Generic Algorithms", Machine Learning and Cybernetics, Proceeding of 2005 International, 3428-3432.
- Moselhi, O & Lorterapong, P., 1993, "Near Optimal Solution for Resource-constrained Scheduling Problems", Construction Management and Economics, 11(4), 293-303.
- Senouci, A.B., & Eldin, N.N., 2004, "Use of Genetic Algorithms in Resource Scheduling of Construction Projects", ASCE Journal of Construction Engineering and Management, 130(6), 869-877.
- Sumirto, D., 2002, "Optimasi Penjadwalan Sumberdaya dengan Metoda Algoritma Genetik", Thesis Magister Teknik Sipil, Institut Teknologi Bandung.
- Toklu, Y.C., 2002, "Application of Genetic Algorithms to Construction Scheduling with or Without Resource Constraints", Canadian Journal of Civil Engineering, 29(3), 421-429.