

OPTIMASI PENJADWALAN PROYEK DENGAN PENYEIMBANGAN BIAYA MENGUNAKAN KOMBINASI CPM DAN ALGORITMA GENETIKA

Riza Arifudin

Jurusan Matematika FMIPA Universitas Negeri Semarang
email: riza_wsb@yahoo.com

Abstract

The project scheduling must be prepared systematically by using resources effectively and efficiently so that project objectives can be achieved optimally. This research aimed to examine the application of combination CPM and genetic algorithms to solve optimization problems in a project scheduling with the leveling of costs and designing a software. CPM is one of the methods to schedule the project that produce the shortest time. In this study of CPM combined with genetic algorithm to perform scheduling. Genetic algorithms are search methods that mimic the process of solution of natural selection and genetics. The allocation of activities is determined based on the earliest start time and latest start time by taking into account the cost of resources in each period of the project. Project scheduling and optimal criterion used is minimizing the cost of deviations from the average total project cost. Results obtained in this research is that the cost of scheduling with deviations smaller than the scheduling with CPM alone thus generated a shorter project schedules and project cost per day is also more equitable. This Scheduling method can be alternative decisions for the contractor in project implementation.

Keywords: project scheduling, CPM, genetic algorithm.

1. Pendahuluan

Pada manajemen proyek, sebelum proyek dikerjakan perlu adanya tahap-tahap pengelolaan proyek yang meliputi tahap perencanaan, tahap penjadwalan, dan tahap pengkoordinasian. Dari ketiga tahapan ini, tahap perencanaan dan penjadwalan adalah tahap yang paling menentukan berhasil/tidaknya suatu proyek, karena penjadwalan adalah tahap ketergantungan antar aktivitas yang membangun proyek secara keseluruhan. Penjadwalan sendiri harus disusun secara sistematis dengan penggunaan sumber daya secara efektif dan efisien agar tujuan proyek bisa tercapai secara optimal. Pemecahan masalah penjadwalan yang baik dari suatu proyek merupakan salah satu faktor keberhasilan dalam pelaksanaan proyek untuk selesai tepat pada waktunya yang merupakan tujuan pokok dan utama, baik bagi kontraktor maupun pemiliknya.

Salah satu metode penjadwalan proyek yang telah dikembangkan sejak akhir tahun 1950-an adalah PERT (*Program Evaluation and Review Technique*) dan CPM (*Critical Path Method*). Kedua metode ini berorientasi pada waktu [15]. Perbedaan yang terpenting dari kedua metode ini yaitu pada asumsi estimasi waktu, dimana pada

PERT bersifat probabilistik dan pada CPM bersifat deterministik. Penjadwalan yang dihasilkan metode-metode ini masih menggunakan batasan-batasan waktu sehingga diperlukan suatu pengembangan metode penjadwalan dengan batasan sumber daya yang lain.

Model penjadwalan proyek dengan batasan persediaan sumber daya (RCPS: *Resources Constraints Project Scheduling*) merupakan salah satu masalah riset operasi yang akhir-akhir ini dikembangkan dan diarahkan untuk lebih mendekati dunia nyata [12]. Salah satu sumber daya yang penggunaannya sangat diperhatikan dalam pelaksanaan suatu proyek yaitu sumber daya biaya. Pengaturan biaya dalam penjadwalan proyek akan memudahkan bagi kontraktor maupun pemilik proyek mengenai besarnya biaya yang harus dikeluarkan setiap harinya. Selain waktu yang cepat, jadwal dibuat agar ada penyeimbangan biaya dimana pengeluaran biaya perhari dapat diatur agar diperoleh rata-rata penyimpangan biaya perhari yang paling minimum.

Pembuatan jadwal dengan batasan waktu dan biaya dengan tujuan untuk meminimumkan rata-rata penyimpangan biaya perhari akan

mengalami kerumitan apabila diselesaikan secara matematis [15]. Salah satu teknik yang digunakan untuk membuat penjadwalan dengan mengalokasikan waktu maupun penyeimbangan biaya yaitu menggunakan Algoritma Genetika [6]. Algoritma genetika memecahkan masalah dalam cara yang evolusioner. Algoritma genetika meniru proses evolusi alam dan mencari solusi yang sangat baik [14].

Di dalam ilmu komputer algoritma digunakan untuk menemukan penyelesaian perkiraan untuk optimisasi dan masalah pencarian. Algoritma genetika berangkat dari himpunan solusi yang dihasilkan secara acak yang disebut populasi. Sedangkan setiap individu dalam populasi disebut kromosom yang merupakan representasi dari solusi. Kromosom-kromosom berevolusi dalam suatu proses iterasi yang berkelanjutan dan disebut generasi. Pada setiap generasi, kromosom dievaluasi berdasarkan suatu fungsi evaluasi [3]. Setelah beberapa generasi maka algoritma genetika akan konvergen pada kromosom terbaik, yang diharapkan merupakan solusi optimal.

Berdasarkan uraian yang telah dikemukakan di atas, tulisan ini mengkaji penerapan kombinasi CPM dan algoritma genetika sebagai suatu metode untuk menyelesaikan masalah penjadwalan proyek dengan optimasi penyeimbangan biaya.

2. Formulasi permasalahan

2.1 Proyek

Sebuah proyek mendefinisikan satu kombinasi kegiatan-kegiatan yang saling berkaitan yang harus dilakukan dalam urutan tertentu sebelum keseluruhan tugas dapat diselesaikan [15].

Kegiatan-kegiatan ini saling berkaitan dalam satu urutan yang logis dalam arti bahwa beberapa kegiatan tidak dapat dimulai sampai kegiatan-kegiatan lainnya diselesaikan. Sebuah kegiatan (aktivitas) dalam sebuah proyek biasanya dipandang sebagai sebuah tugas yang memerlukan waktu sumber daya untuk penyelesaiannya. Pada umumnya, sebuah proyek adalah satu usaha satu kali; yaitu, urutan kegiatan yang sama kemungkinan tidak diulangi di masa mendatang.

2.2 Penjadwalan Proyek

Penjadwalan proyek adalah pembuatan rencana pelaksanaan setiap kegiatan di dalam suatu proyek dengan mengoptimalkan efisiensi pemakaian waktu dan sumber daya yang tersedia, tetapi kesesuaian presedensi diantara kegiatan tetap dipenuhi [15]. Menurut Lawrence dan Pasternack (2001)[9], ada beberapa tujuan penjadwalan proyek meliputi: (1) Menentukan jadwal paling awal dan paling akhir dari waktu mulai dan berakhir untuk setiap kegiatan yang mengarah ke waktu penyelesaian paling awal untuk keseluruhan proyek; (2) Menghitung kemungkinan bahwa proyek akan selesai dalam jangka waktu tertentu; (3) Mencari biaya jadwal minimum yang akan menyelesaikan sebuah proyek dengan tanggal tertentu; (4) Meninvestigasi bagaimana keterlambatan untuk kegiatan tertentu mempengaruhi waktu penyelesaian keseluruhan proyek; (5) Monitoring sebuah proyek untuk menentukan apakah berjalan tepat waktu dan sesuai anggaran; (6) Mencari jadwal kegiatan yang akan memuluskan alokasi sumber daya selama durasi proyek.

Manajemen proyek telah berkembang sebagai satu bidang baru dengan dikembangkannya dua teknik analitis untuk perencanaan, penjadwalan, dan pengendalian proyek. Keduanya adalah metode PERT (*Project Evaluation and Review Technique*) dan CPM (*Critical Path Method*). PERT dan CPM pada dasarnya merupakan metode penjadwalan proyek berorientasi waktu, dalam arti keduanya mengarah pada penentuan sebuah jadwal. Tetapi kedua metode ini mengabaikan batasan sumber daya. Perbedaan dari kedua metode ini ada pada estimasi waktu. Metode PERT menggunakan estimasi waktu secara probabilistik sedangkan metode CPM menggunakan estimasi waktu secara deterministik [15].

2.3 CPM (*Critical Path Method*)

CPM atau disebut metode jalur kritis merupakan salah satu metode penjadwalan proyek yang dikembangkan oleh E.I. du Pont de Nemours & Company dan kemudian diperluas oleh Mauchly Associates [15]. Jalur (lintasan) kritis suatu proyek adalah jalur dalam suatu jaringan kerja sedemikian sehingga kegiatan pada lintasan ini memiliki kelambanan nol [7].

Pada penjadwalan CPM, kegiatan-kegiatan dalam proyek diklasifikasikan menjadi kegiatan kritis dan kegiatan nonkritis [15]. Sebuah kegiatan dikatakan kritis jika penundaan saat awalnya akan menyebabkan penundaan waktu penyelesaian keseluruhan proyek. Sebuah kegiatan nonkritis adalah kegiatan-kegiatan dengan jumlah waktu di antara waktu awal yang paling cepat dengan waktu penyelesaian yang paling akhir (sebagaimana diijinkan oleh proyek yang bersangkutan) adalah lebih panjang dari durasi aktualnya.

2.4 Perhitungan Jalur Kritis

Perhitungan jalur kritis mencakup dua tahap. Tahap pertama disebut perhitungan maju (*forward pass*), dimana perhitungan dimulai dari node “awal” dan bergerak ke node “akhir”. Di setiap node, sebuah angka dihitung yang mewakili waktu yang tercepat untuk suatu kejadian yang bersangkutan. Tahap kedua yang disebut perhitungan mundur (*backward pass*), memulai perhitungan dari node “akhir” dan bergerak ke node “awal” [15].

Perhitungan Maju

Pada perhitungan maju dimaksudkan untuk menghitung saat yang paling awal terjadinya dan penyelesaian kegiatan suatu proyek. Waktu mulai paling awal ES (*Earliest Start Time*) suatu kegiatan j didapatkan dari rumus:

$$ES_j = \begin{cases} 0 & , i = 0 \\ \max_i \{ ES_i + D_{ij} \} & , i \neq 0 \end{cases} \quad (1)$$

dimana:

ES_j adalah waktu mulai paling awal suatu kegiatan j ,

ES_i adalah waktu mulai paling awal kegiatan *predecessor* dari kegiatan j ,

D_{ij} adalah durasi kegiatan i yang merupakan *predecessor* dari kegiatan j

Waktu penyelesaian paling awal EF (*Earliest Finish Time*) suatu kegiatan j didapatkan dari rumus:

$$EF_j = ES_j + D_j \quad (2)$$

dimana:

EF_j adalah waktu penyelesaian paling awal suatu kegiatan j ,

ES_j adalah waktu mulai paling awal dari kegiatan j ,

D_j adalah durasi dari kegiatan j

Perhitungan Mundur

Pada perhitungan mundur dimaksudkan untuk menghitung saat yang paling akhir penyelesaian dan terjadinya dari kegiatan suatu proyek. Waktu penyelesaian paling akhir LF (*Latest Finish Time*) suatu kegiatan i didapatkan dari rumus:

$$LF_i = \begin{cases} ES_n & , j = n \\ \min_j \{ LF_j - D_{ij} \} & , j \neq n \end{cases} \quad (3)$$

dimana:

LF_i adalah waktu penyelesaian paling akhir suatu kegiatan i ,

ES_n adalah waktu mulai paling awal dari kegiatan n ,

LF_j adalah waktu penyelesaian paling akhir kegiatan *successor* dari kegiatan i ,

D_{ij} adalah durasi kegiatan j yang merupakan *successor* dari kegiatan i

Waktu mulai paling akhir LS (*Latest Start Time*) suatu kegiatan i didapatkan dari rumus:

$$LS_i = LF_i - D_i \quad (4)$$

dimana:

LS_i adalah waktu mulai paling akhir suatu kegiatan ke- i ,

LF_i adalah waktu penyelesaian paling akhir dari kegiatan i ,

D_i adalah durasi dari kegiatan i

Suatu kegiatan berada dijalur kritis bila kegiatan tersebut memenuhi ketiga kondisi berikut ini:

$$ES = LS$$

$$EF = LF$$

$$LF - ES = \text{Durasi Kegiatan}$$

Kondisi ini sebenarnya menyatakan bahwa tidak ada waktu senggang atau waktu mengambang antara awal tercepat (penyelesaian) dan awal terakhir (penyelesaian) dari kegiatan kritis yang bersangkutan.

2.5 Pengembangan Jadwal Proyek dan Penyeimbangan Biaya

Hasil perhitungan jaringan menggunakan CPM adalah bagan waktu untuk pengembangan jadwal proyek. Bagan waktu ini dapat dikonversikan dengan mudah menjadi sebuah jadwal kalender yang dapat dipergunakan dengan mudah dalam pelaksanaan proyek [15].

Jadwal yang dihasilkan oleh CPM masih berorientasi pada waktu, sehingga faktor biaya proyek perharinya tidak diperhitungkan dalam pembuatan jadwal, untuk itu pengembangan

jadwal proyek harus dilakukan dengan melibatkan faktor biaya proyek agar lebih merata, karena kemungkinan pelaksanaan beberapa kegiatan secara berbarengan akan mengakibatkan penggunaan biaya yang cukup tinggi pada waktu tertentu.

Langkah yang dilakukan untuk pengembangan jadwal dengan melibatkan faktor biaya yaitu dengan menggeser kegiatan-kegiatan nonkritis (ke depan atau ke belakang) antara batas waktu yang diijinkan untuk dapat menurunkan kebutuhan biaya maksimum. Sekalipun biaya tidak dibatasi, dengan menggunakan waktu mengambang dapat digunakan untuk menyeimbangkan biaya di sepanjang durasi keseluruhan proyek. Langkah-langkah ini merupakan salah satu proses optimasi yang dapat dilakukan dengan Algoritma Genetika.

2.6 Konsep Algoritma Genetika

Algoritma genetika adalah algoritma pencarian (search algorithm) yang menggunakan prinsip seleksi alam dalam ilmu genetika untuk mengembangkan solusi terhadap permasalahan [5]. Algoritma Genetika merupakan kelas algoritma pencarian heuristik berdasarkan evolusi biologi [11].

Ide dasar algoritma genetika adalah mengelola suatu populasi individu yang merepresentasikan kandidat solusi sebuah permasalahan. Secara umum algoritma genetika memiliki lima komponen dasar [10] yaitu: (1) Representasi genetik dari solusi-solusi masalah; (2) Cara membentuk populasi awal dari solusi-solusi; (3) Fungsi evaluasi yang *me-rate (rating)* solusi-solusi berdasarkan *fitness* mereka; (4) Operator-operator genetik yang merubah komposisi genetik dari *offspring* selama reproduksi; (5) Nilai-nilai untuk parameter algoritma genetika.

Algoritma genetika berangkat dari himpunan solusi yang dihasilkan secara acak yang disebut populasi. Sedangkan setiap individu dalam populasi disebut kromosom yang merupakan representasi dari solusi dan masing-masing dievaluasi tingkat ketangguhannya (*fitness*) oleh fungsi yang telah ditentukan. Melalui proses seleksi alam atas operator genetik, gen-gen dari dua kromosom (disebut *parent*) diharapkan akan menghasilkan kromosom baru dengan tingkat

fitness yang lebih tinggi sebagai generasi baru atau keturunan (*offspring*) berikutnya. Kromosom-kromosom tersebut akan mengalami iterasi yang disebut generasi (*generation*). Pada setiap generasi, kromosom dievaluasi berdasarkan nilai fungsi *fitness* [3]. Setelah beberapa generasi maka algoritma genetika akan konvergen dapat kromosom terbaik, yang merupakan solusi optimal [4].

2.7 Struktur Umum Algoritma Genetika

Algoritma genetika memberikan suatu pilihan bagi penentuan nilai parameter dengan meniru cara reproduksi genetik, pembentukan kromosom baru serta seleksi alami seperti terjadi pada makhluk hidup.

Inisialisasi populasi awal dilakukan untuk menghasilkan solusi awal dari suatu permasalahan algoritma genetika. Inisialisasi ini dilakukan secara acak sebanyak jumlah kromosom/populasi yang diinginkan. Selanjutnya dihitung nilai *fitness* dan seterusnya dilakukan seleksi dengan menggunakan metode roda roulette, tournament atau ranking. Kemudian dilakukan perkawinan silang (*crossover*) dan mutasi. Setelah melalui beberapa generasi maka algoritma ini akan berhenti sebanyak generasi yang diinginkan.

Sebagaimana halnya proses evolusi di alam, suatu algoritma genetika yang sederhana umumnya terdiri dari tiga operator yaitu: operator reproduksi, operator *crossover* (persilangan) dan operator mutasi.

Ada dua hal penting yang harus dilakukan pada awal proses Algoritma Genetika. Pertama, pendefinisian atau pengkodean kromosom yang merupakan solusi yang masih berbentuk simbol. Kedua, penentuan fungsi *fitness* atau fungsi obyektif. Dua hal ini berperan penting dalam algoritma genetika untuk menyelesaikan suatu masalah.

2.8 Pengkodean

Pengkodean adalah suatu teknik untuk menyatakan populasi awal sebagai calon solusi dari suatu masalah ke dalam suatu kromosom [3] sebagai suatu kunci pokok persoalan ketika menggunakan algoritma genetika.

Berdasarkan jenis simbol yang digunakan sebagai nilai suatu gen, metode pengkodean dapat diklasifikasikan sebagai berikut :

pengkodean biner, bilangan riil, bilangan bulat dan struktur data [3].

Pengkodean biner merupakan cara pengkodean yang paling umum digunakan, ini karena pengkodean ini merupakan yang pertama kali digunakan dalam algoritma genetik oleh Holland [1]. Pengkodean biner dinyatakan dalam kromosom biner.

Pengkodean bilangan bulat adalah suatu metode mengkodekan bilangan dalam bentuk bilangan bulat. Pengkodean ini baik digunakan untuk masalah optimasi kombinatorial [3]. Dengan pengkodean bilangan bulat, ukuran kromosom menjadi lebih sederhana dan tidak terlalu panjang.

Pengkodean bilangan riil adalah suatu pengkodean bilangan dalam bentuk riil. Gen dan Cheng (2000) menyatakan bahwa pengkodean bilangan riil baik digunakan untuk masalah optimasi fungsi dan optimasi kendala.

Pengkodean struktur data adalah model pengkodean yang menggunakan struktur data. Pengkodean ini digunakan untuk masalah yang lebih kompleks seperti perencanaan trajektori robot [10] dan Masalah pewarnaan Grap [2].

Inisialisasi Populasi Awal

Inisialisasi populasi awal merupakan suatu metode untuk menghasilkan kromosom-kromosom awal. Jumlah individu pada populasi awal merupakan masukan dari pengguna. Setelah jumlah individu pada populasi awal ditentukan, dilakukan inisialisasi terhadap kromosom yang terdapat pada populasi tersebut. Inisialisasi dilakukan secara acak, namun demikian tetap memperhatikan domain solusi dan kendala permasalahan yang ada.

2.9 Nilai Fitness

Secara alamiah di dunia nyata menyatakan bahwa hanya individu yang unggul (berkualitas tinggi) saja yang akan bertahan hidup. Sedangkan individu berkualitas rendah akan mati atau punah. Pada Algoritma Genetika, suatu individu dievaluasi berdasarkan suatu fungsi tertentu sebagai ukuran nilai kualitasnya. Fungsi ini dikenal sebagai fungsi *fitness*. Fungsi *fitness* menghasilkan suatu nilai *fitness*.

Nilai *fitness* adalah nilai yang menyatakan baik tidaknya suatu individu. Nilai *fitness* merupakan sebuah nilai yang terkait dengan kromosom yang memberikan suatu manfaat

relatif terhadap kromosom itu [5]. Untuk menemukan fungsi *fitness* yang tepat untuk suatu masalah, yang harus diperhatikan adalah fungsi objektif (*objective function*). Pada kasus optimasi, dikenal dua masalah yaitu: masalah maksimasi atau masalah minimasi.

2.10 Operator Genetika

Algoritma genetika merupakan proses pencarian yang heuristik dan acak, sehingga penekanan pemilihan operator yang digunakan sangat menentukan keberhasilan algoritma genetik dalam menemukan solusi optimum suatu masalah yang diberikan. Operator genetika digunakan setelah proses evaluasi tahap pertama untuk membentuk suatu populasi baru dari generasi sekarang. Operator-operator tersebut adalah operator seleksi, *crossover* (perkawinan silang), dan mutasi.

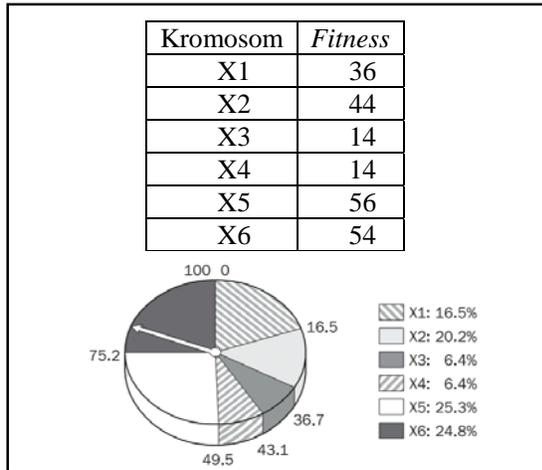
Seleksi

Seleksi merupakan proses pemilihan orang tua untuk reproduksi (biasanya didasarkan pada nilai *fitness*) [5]. Seleksi bertujuan untuk memberikan kesempatan reproduksi yang lebih besar bagi anggota populasi yang paling baik.

Ada beberapa metode untuk seleksi kromosom antara lain adalah seleksi roda *roulette* (*roulette wheel selection*), seleksi ranking (*rank selection*), dan seleksi turnamen (*tournament selection*).

Salah satu teknik kromosom yang paling umum digunakan adalah pemilihan seleksi roda *roulette* (*roulette wheel selection*) [4]. Seleksi dengan *Roulette wheel* memilih anggota populasi tertentu untuk menjadi orangtua dengan probabilitas sama dengan *fitness* dibagi dengan total *fitness* populasi [5].

Pada seleksi ini, orangtua dipilih berdasarkan *fitness* mereka. Lebih baik suatu kromosom, lebih besar kesempatan terpilih. Probabilitas suatu individu terpilih untuk perkawinan silang sebanding dengan *fitness*nya. Cara penyeleksian ini merupakan peniruan dari permainan roda *roulette*, dan dapat diilustrasikan dengan Gambar 1.

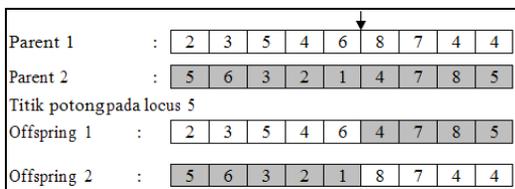


Gambar 1. Seleksi kromosom dengan roda roulette

Prosedur seleksi dimulai dengan memutar roda roulette sebanyak n , setiap waktu dipilih satu kromosom sebagai induk untuk menghasilkan kromosom baru.

Crossover

Crossover (Persilangan) adalah sebuah proses yang membentuk kromosom baru dari dua kromosom induk dengan menggabungkan bagian informasi dari masing-masing kromosom [5]. Crossover menghasilkan kromosom baru yang disebut kromosom anak (offspring). Crossover bertujuan untuk menambah keanekaragaman string dalam satu populasi dengan penyilangan antar string yang diperoleh dari reproduksi sebelumnya. Hasil crossover 2 kromosom induk akan menghasilkan 2 offspring, sehingga jumlah populasi akan bertambah 2 kali jumlah populasi awal.



Gambar 2. Crossover satu titik suatu kromosom

Metode crossover yang sering digunakan pada algoritma genetika adalah one-point crossover atau crossover satu titik (lihat Gambar 2). Pada metode ini, kita pilih sembarang bilangan secara acak untuk memilih titik crossover. Kemudian kita tukarkan bagian kanan

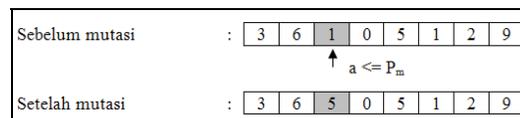
kromosom dari titik crossover dari dua kromosom induk untuk menghasilkan kromosom anak [5].

Mutasi

Mutasi merupakan proses mengubah secara acak nilai dari satu atau beberapa gen dalam suatu kromosom [5]. Mutasi ini berperan untuk menggantikan suatu nilai gen yang hilang dari proses seleksi yang memungkinkan munculnya kembali nilai gen yang tidak muncul pada saat inialisasi populasi awal.

Pada mutasi terdapat satu parameter yang sangat penting, yaitu peluang mutasi (P_m). P_m didefinisikan sebagai persentasi dari jumlah total gen dalam populasi yang akan mengalami mutasi. Misalkan offspring yang terbentuk adalah 100 dengan jumlah gen setiap kromosom adalah 5 dan peluang mutasi adalah 0.10, maka diharapkan terdapat 50 kromosom dari 500 gen yang ada pada populasi tersebut akan mengalami mutasi.

Mutasi yang digunakan dalam penelitian ini adalah mutasi dengan pengkodean bulat. Mutasi gen ini dilakukan dengan cara pemilihan nilai secara acak. Suatu gen yang terpilih untuk dimutasi nilainya diganti dengan nilai baru yang dibangkitkan secara acak dalam interval nilai-nilai gen yang diizinkan. Misalnya, jika nilai-nilai gen berada dalam interval [0,9], maka gen baru yang dibangkitkan secara acak juga berada dalam interval [0,9]. Nilai gen baru yang dihasilkan bisa saja dibatasi dengan aturan berbeda dengan nilai lama. Perhatikan ilustrasi Gambar 3.



Gambar 3 Mutasi dengan pemilihan nilai secara acak

2.11 Update

Setelah terbentuk populasi baru, populasi tersebut dipakai untuk menggantikan populasi lama. Di dalam Algoritma Genetika, suatu populasi dirancang untuk memiliki jumlah individu yang selalu tetap pada setiap generasi. Dalam proses update, kromosom dengan fitness yang terbaik selalu dipertahankan dari generasi

ke generasi [5]. Terdapat dua model update populasi yang bisa digunakan, yaitu model *generational* dan *steady state*.

Pada penelitian ini, model yang digunakan adalah model *steady state* berbasis fitness. Pada model ini, tidak semua kromosom diganti. Penggantian dilakukan hanya pada sejumlah kromosom tertentu, dimana sebanyak M kromosom yang diganti adalah kromosom-kromosom terburuk (yang paling rendah nilai *fitness*-nya).

3. Kombinasi CPM dan Algoritma Genetika

3.1 Deskripsi Sistem

Secara umum sistem yang akan dibuat dalam penelitian ini adalah sistem penjadwalan proyek dengan menggunakan Algoritma Genetika. Sistem ini merupakan kombinasi dari CPM dan Algoritma Genetika. Pada sistem ini output dari perhitungan akan di optimasi dengan Algoritma Genetika, sehingga akan dihasilkan suatu penjadwalan kegiatan-kegiatan proyek yang paling awal dan merata dalam pembiayaan setiap harinya.

Proses penyeimbangan biaya proyek tiap harinya dilakukan melalui proses pergeseran waktu pelaksanaan kegiatan nonkritis dengan menggunakan algoritma genetika. Gambaran umum dari kombinasi dari CPM dan Algoritma Genetika pada penelitian ini sebagai berikut.

- a. CPM akan menentukan waktu tercepat pelaksanaan proyek dari kegiatan-kegiatan yang diinput dan menghitung waktu ambang dari setiap kegiatan.
- b. Hasil dari perhitungan CPM digunakan sebagai batasan pada proses Algoritma Genetika untuk mengalokasikan waktu pelaksanaan setiap kegiatan non kritis dengan memperhatikan faktor biaya proyek.

Dari kombinasi ini akan dihasilkan sebuah pengembangan penjadwalan proyek dengan berorientasi pada waktu dan penyeimbangan biaya.

3.2 Analisa Sistem

Untuk menyelesaikan permasalahan penjadwalan proyek dengan penyeimbangan sumber daya biaya ini diasumsikan bahwa setiap kegiatan dalam proyek mempunyai durasi dan biaya pelaksanaan kegiatan yang tetap. Pada

sistem input data kegiatan harus dimasukkan sesuai urutan kerja, dimana sebuah kegiatan tidak boleh diinputkan sebelum input kegiatan predesornya. Input data pekerjaan yang diperlukan pada proses CPM dari sistem ini adalah; kode kegiatan, nama kegiatan, durasi kegiatan, biaya kegiatan, predesor kegiatan. Sedangkan untuk input data yang diperlukan pada proses Algoritma Genetikanya adalah; ukuran populasi, jumlah generasi, probabilitas *crossover*, probabilitas mutasi. Untuk proses pembuatan bagan penjadwalan diperlukan tambahan input data tanggal dimulainya suatu proyek.

Pada tulisan ini konfigurasi sistem yang digunakan untuk mengimplementasi program yang dibangun adalah menggunakan software Microsoft Visual Basic 6.0 dengan sistem operasi berbasis windows.

3.3 Perhitungan Waktu CPM

Terdapat dua tahap dalam perhitungan ini, yaitu perhitungan maju dan perhitungan mundur. Hasil perhitungan yang diperoleh akan digunakan sebagai inisialisasi pembentukan kromosom pada Algoritma Genetika. Pada perhitungan waktu CPM ini digunakan Contoh 1 sebagai gambaran pada sistem ini.

Contoh 1 sebuah proyek pembangunan rumah dengan hubungan ketergantungan antarkegiatan seperti dalam Tabel 1.

Tabel 1. Hubungan ketergantungan kegiatan

No	Kegiatan Pekerjaan	Predesessor	Duras (hari)	Biaya (ribu)
1	Pek. Persiapan	-	8	1000
2	Pek. galian tanah	-	12	500
3	Pek. Pondasi	2	18	1500
4	Pek. beton bertulang	1,3	10	10000
5	Pek. pasangan/plesteran	3	15	2000
6	Pek. pintu jendela	4	10	6000
7	Pek. Atap	4,5	9	7000
8	Pek. langit-langit	7	11	2000
9	Pek. Lantai	7	8	5000
10	Pek. Finishing	6,8	12	10000

Pada Tabel 1 ditambahkan kegiatan START pada awal dari kegiatan dan FINISH pada akhir kegiatan.

Berdasarkan perhitungan maju dan perhitungan mundur, dapat diketahui nilai-nilai dari ES dan LS yang akan digunakan sebagai input dan batasan pada saat menentukan nilai suatu gen dari kromosom pada proses Algoritma Genetika.

Tabel 2 Perhitungan Maju ES dan EF

No Pekerjaan	ES	Durasi	EF
0	0	-	0
1	0	8	8
2	0	12	12
3	12	18	30
4	30	10	40
5	30	15	45
6	40	10	50
7	45	9	54
8	54	11	65
9	54	8	64
10	65	12	77
11	77	-	77

Proses perhitungan maju dan perhitungan mundur untuk kasus pada Contoh 1 dapat dilihat pada Tabel 2 dan Tabel 3. Dari tabel tersebut diperoleh nilai FINISH yaitu 77, sehingga durasi keseluruhan proyek pada kasus ini adalah 77 hari. Nilai ini dipakai untuk nilai awal dari perhitungan mundur.

Tabel 3 Perhitungan Mundur LS dan LF

No Pekerjaan	LF	Durasi	LS
11	77	-	77
10	77	12	65
9	77	8	69
8	65	11	54
7	54	9	45
6	65	10	55
5	45	15	30
4	45	10	35
3	30	18	12
2	12	12	0
1	35	8	27
0	0	-	0

3.4 Algoritma Genetika dalam Sistem

Tahap-tahap algoritma genetika yang diterapkan dalam masalah penjadwalan proyek dalam penelitian ini adalah sebagai berikut.

a. Representasi dan Inisialisasi Populasi

Awal

Pada penelitian ini, sebuah kromosom menyatakan sebuah solusi penjadwalan. Kromosom berisi kumpulan dari gen yang berisi informasi saat pelaksanaan suatu kegiatan atau pekerjaan proyek. Panjang kromosom adalah n, dengan n adalah banyaknya pekerjaan. Jadi, pada sistem ini satu gen mewakili 1 pekerjaan. Nilai setiap gen adalah bilangan bulat yang diperoleh secara acak dengan interval ES sampai dengan LS sesuai dengan nomer pekerjaan. Misalkan untuk Gen ke 1 diperoleh dari bilangan bulat acak antara ES₁ sampai dengan LS₁ yaitu antara 0 sampai dengan 27. Hal ini berlaku untuk setiap gen pada kromosom.

Contoh representasi kromosom suatu populasi dengan ukuran populasi 10 dari kasus pada Contoh 1 dapat dilihat pada Gambar 5.

Dari Gambar 5 untuk kromosom 1 terdiri dari kumpulan gen x₁, x₂, ..., x₁₀ berisi bilangan bulat positif yang merupakan saat pelaksanaan pekerjaan proyek. Pada kromosom 1 dapat dijelaskan bahwa gen x₁ mempunyai nilai 3 menunjukkan bahwa pada pekerjaan 1 dilaksanakan pada hari ke 3, gen x₂ mempunyai nilai 1 menunjukkan bahwa pekerjaan 2 dilaksanakan pada hari ke 1, dan seterusnya sampai dengan gen x₁₀ yang mempunyai nilai 66 yang menunjukkan bahwa pekerjaan ke 10 dilaksanakan pada hari ke 66. Demikian juga dengan kromosom ke 2 sampai dengan kromosom ke 10.

b. Menentukan Fungsi *Fitness*

Fungsi *fitness* harus memperhatikan pada fungsi objektifnya. Pada sistem ini fungsi objektifnya adalah meminimumkan nilai rata-rata penyimpangan biaya perhari (B). Fungsi objektif dalam sistem ini dituliskan sebagai berikut:

$$Min B = \frac{1}{T} \sum_{i=1}^T |b_i - \bar{b}|$$

3.5	x ₁	x ₂	x ₃	x ₄	x ₅	x ₆	x ₇	x ₈	x ₉	x ₁₀
Kromosom 1	3	1	1	3	3	5	4	5	6	66
Kromosom 2	6	1	1	3	3	4	4	5	6	66
Kromosom 3	8	1	1	3	3	4	4	5	5	66
Kromosom 4	1	1	1	3	3	4	4	5	6	66
Kromosom 5	0	1	3	3	1	8	6	5	8	66
Kromosom 6	1	1	1	3	3	4	4	5	6	66
Kromosom 7	5	1	3	4	1	3	6	5	4	66
Kromosom 8	7	1	1	3	3	5	4	5	5	66
Kromosom 9	2	1	1	3	3	5	4	5	6	66
Kromosom 10	6	1	3	2	1	5	6	5	2	66
Kromosom 1	1	1	1	3	3	4	4	5	6	66
Kromosom 2	7	1	3	4	1	7	6	5	4	66
Kromosom 3	4	1	3	4	1	5	6	5	4	66
Kromosom 4	8	1	3	5	1	1	6	5	8	66

Gambar 5. Representasi Kromosom Awal

dimana:

T adalah waktu keseluruhan penjadwalan proyek (dalam hari),

b_i adalah penyimpangan biaya proyek perhari, dan

\bar{b} adalah rata-rata biaya proyek perhari.

Fungsi objektif pada sistem ini adalah meminimumkan fungsi B sehingga fungsi *fitness*nya dapat ditulis sebagai berikut:

$$f = \frac{1}{(B + a)}$$

dimana a merupakan suatu bilangan real yang dianggap sangat kecil untuk menghindari pembagian dengan 0.

c. Seleksi Kromosom

Pada penelitian ini, seleksi kromosom dilakukan dengan menggunakan seleksi *roulette wheel*. Prosedur seleksi adalah sebagai berikut:

- 1) Menghitung nilai *fitness* (f_k) tiap kromosom
- 2) Hitung total *fitness*
- 3) Menghitung nilai probabilitas seleksi (p_k) setiap kromosom
- 4) Menghitung nilai probabilitas kumulatif q_k

- 5) Pilih induk yang menjadi kandidat untuk populasi baru dengan cara:

- a. Bangkitkan suatu bilangan $[0,1]$
- b. Memilih kromosom dengan nilai kumulatif q_k yang lebih besar dan yang paling mendekati dari nilai random sebagai parent.

- 6) Lakukan langkah 5 sebanyak ukuran populasi.

d. Persilangan Kromosom

Prosedur persilangan satu titik sebagai berikut: (1) Tentukan jumlah populasi yang akan mengalami persilangan, berdasarkan p_c ; (2) Pilih dua kromosom sebagai induk, yaitu p_1 dan p_2 ; (3) Tentukan posisi *crossover* dengan cara membangkitkan bilangan acak dengan batasan 1 sampai $n-1$, dimana n adalah panjang kromosom. Misalkan didapat posisi *crossover* adalah 20 maka kromosom induk akan dipotong mulai gen ke 21 sampai gen ke n , kemudian potongan gen tersebut saling ditukar antar induk; (4) Posisi *cut-point crossover* dipilih menggunakan bilangan acak $1-(n-1)$ sebanyak jumlah *crossover* yang terjadi.

e. Mutasi Gen

Proses mutasi dilakukan dengan cara mengganti satu gen yang dipilih secara acak dengan suatu nilai baru yang didapat secara acak.

Langkah-langkah mutasi sebagai berikut: (1) Untuk memilih kromosom yang akan mengalami mutasi dilakukan dengan cara membangkitkan bilangan integer acak (a) antara 0 sampai 1. Jika $a < p_m$, maka kromosom akan dimutasi, misal p_m kita temukan 10% maka diharapkan ada 10% dari kromosom pada suatu populasi yang mengalami mutasi; (2) Untuk memilih posisi gen yang akan mengalami mutasi dilakukan dengan cara membangkitkan bilangan integer acak (a) antara 0 sampai 1. Jika $a < p_m$, maka gen akan dimutasi; (3) Ganti nilai gen yang akan dimutasi dengan membangkitkan bilangan acak dengan interval ES_i sampai dengan LS_i , misal posisi gen 5 akan diganti bilangan acak dengan interval ES_5 sampai dengan LS_5 ; (4) Ulangi langkah 2 – 3 sampai jumlah gen.

Parameter input adalah populasi yang akan dimutasi dan peluang mutasi (p_m). Sedangkan

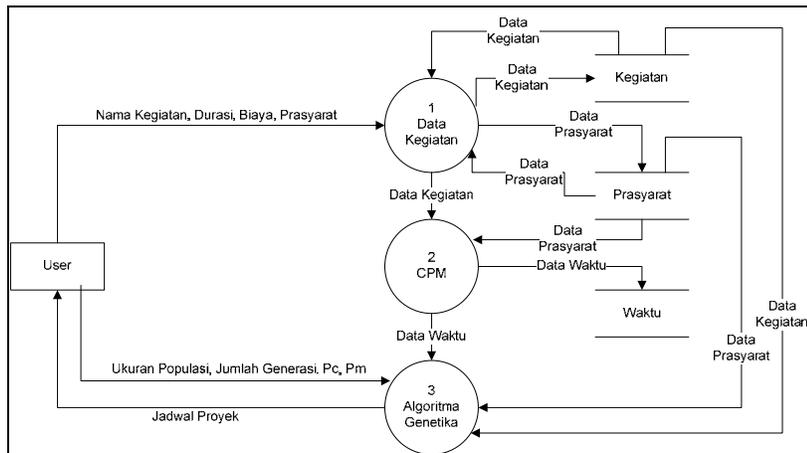
parameter outputnya adalah populasi yang sudah mengalami mutasi.

f. Update Populasi

Langkah yang dilakukan pada proses *update* ini yaitu dengan menggabungkan populasi awal dengan populasi hasil proses genetika, kemudian gabungan populasi tersebut diurutkan berdasarkan nilai *fitness*-nya. Selanjutnya dipilih lagi kromosom-kromosom sebanyak ukuran populasi awal berdasarkan nilai *fitness* tertinggi dari gabungan populasi. Dengan cara ini maka kromosom dengan nilai *fitness* tertinggi dapat tetap hidup digenerasi selanjutnya.

3.6 Rancangan Data Flow Diagram (DFD)

Data Flow Diagram (DFD) atau Diagram Aliran Data adalah sebuah teknis grafis yang menggambarkan aliran informasi dan transformasi yang diaplikasikan pada saat data bergerak dari input menjadi output [13]. Gambaran aliran data pada proses penggunaan algoritma genetika untuk penyeimbangan sumber daya pada penjadwalan proyek dapat digambarkan dalam DFD Level 1, seperti pada Gambar 6.



Gambar 6. DFD Level 1 Penjadwalan dengan Algoritma Genetika

4. Hasil

4.1 Hasil Percobaan

Sistem ini menghasilkan sebuah penjadwalan proyek dengan mengoptimasi metode lintasan kritis dengan mempertimbangkan

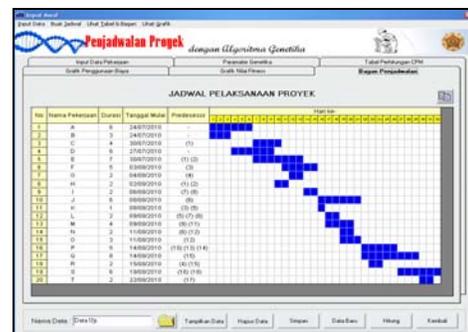
penyeimbangan biaya proyek perhari. Percobaan dari sistem yang dilakukan pada penelitian ini menggunakan data simulasi pekerjaan proyek. Data tersebut ditampilkan dalam Tabel 4 berikut ini.

Tabel 4. Data Pekerjaan Proyek

No	Nama Pekerjaan	Durasi	Predesesor	Biaya (dalam ribuan)
1	A	6	-	5,000
2	B	3	-	3,000
3	C	4	A	2,000
4	D	6	-	5,000
5	E	7	A, B	3,000
6	F	5	C	4,000
7	G	2	D	4,000
8	H	2	A, B	5,000
9	I	2	G, H	3,000
10	J	6	F	1,000
11	K	1	C, E	3,000
12	L	2	E, G, H	3,000
13	M	4	I, K	2,000
14	N	2	F, L	1,000
15	O	3	L	5,000
16	P	5	J, M, N	3,000
17	Q	8	O	4,000
18	R	2	D, O	5,000
19	S	6	P, R	2,000
20	T	2	Q	1,000

Sistem dijalankan dengan menggunakan data parameter genetika dengan ukuran populasi 100, probabilitas crossover (P_c) 0.5, probabilitas mutasi (P_m) 0.03, dan jumlah generasi 50. Parameter genetika yang dipakai dalam percobaan ini menggunakan parameter hasil penelitian sebelumnya yang dilakukan oleh Kim dan Ellis (2008)[8].

Dari data pada Tabel 4 diperoleh nilai fungsi objektif yaitu rata-rata penyimpangan biaya perhari (B) sebesar Rp. 786,310,-. Kromosom terbaik yang dihasilkan pada percobaan dengan data pada Tabel 4 yaitu 1|1|7|4|7|11|12|10|14|16|16|17|17|19|19|22|22|23|27|30 dimana gen ke-1 berisi 1 artinya bahwa pekerjaan ke-1 dilaksanakan pada hari pertama atau hari ke-1, kemudian gen ke-2 bernilai 1 artinya bahwa pekerjaan ke-2 juga dilaksanakan pada hari pertama, sedangkan gen ke-3 bernilai 7 artinya bahwa pekerjaan ke-3 dilaksanakan pada hari ke-7, dan begitu seterusnya sampai gen ke 20 yang berisi 30 yang artinya bahwa pekerjaan ke-20 dilaksanakan pada hari ke-30. Dari kromosom terbaik tersebut dibuat sebuah jadwal pelaksanaan pekerjaan proyek yang berlangsung selama 32 hari, seperti yang ditampilkan pada Gambar 7.



Gambar 7. Hasil Penjadwalan

Berdasarkan pada kromosom terbaik yang diperoleh, dapat dibuat penggunaan biaya perhari dari data pada Tabel 4 yang ditampilkan oleh sistem penjadwalan proyek dengan grafik seperti pada Gambar 8.

Hasil yang diperoleh dari penjadwalan menggunakan penggabungan CPM dengan Algoritma Genetika akan dibandingkan dengan hasil penjadwalan yang dibuat hanya dengan CPM tanpa menggunakan Algoritma Genetika.



Gambar 8. Grafik Penggunaan Biaya Perhari

Untuk menguji pengaruh parameter input pada Algoritma Genetika pada sistem penjadwalan ini, maka dilakukan pengujian dengan melakukan variasi pada ukuran populasi, nilai P_c , dan nilai P_m . Untuk parameter P_c dan P_m diambil dari petunjuk awal yang dikemukakan oleh De Jong dalam Haupt dan Haupt (2004). Pengambilan parameter tersebut diuji cobakan pada empat macam pengaruh yang akan dibahas. Nilai parameter ini apakah mendukung atau berpengaruh pada hasil yang diperoleh pada sistem penjadwalan proyek.

Dari hasil pengujian pengaruh parameter terhadap data pada Tabel 4 diperoleh bahwa nilai parameter yang direkomendasikan untuk tercapainya nilai objektif yang paling optimal yaitu dengan ukuran populasi = 50, nilai $P_c = 0.45$, nilai $P_m = 0.05$, dan jumlah generasi = 50.

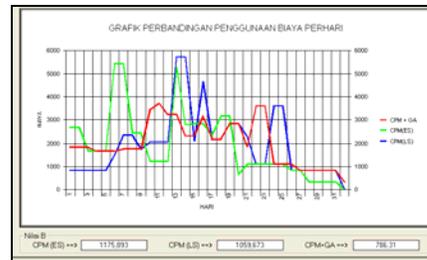
4.2 Perbandingan

Hasil akhir dari penelitian ini adalah dihasilkannya suatu penjadwalan proyek yang lebih optimal dalam penyeimbangan biaya perhari jika dibandingkan dengan metode lintasan kritis tanpa menggunakan Algoritma Genetika. Sebagai nilai perbandingan yang digunakan adalah rata-rata penyimpangan biaya perhari (B) dari penjadwalan yang dihasilkan. Untuk mengetahui besarnya rata-rata penyimpangan biaya perhari dapat dilihat pada Tabel 5.

Tabel 5. Perbandingan Nilai B

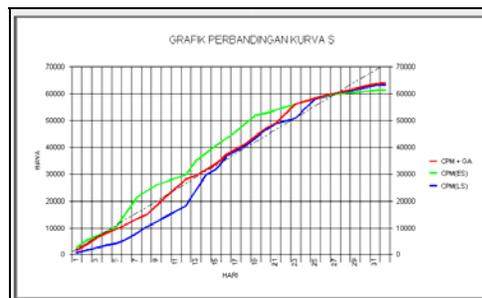
Jadwal	B
CPM (ES)	1,175,893.-
CPM (LS)	1,059,673.-
CPM+AG	786,310.-

Dari Tabel 5 ditunjukkan bahwa nilai B terendah sebesar Rp. 786,310.- yang diperoleh dari penjadwalan proyek menggunakan kombinasi CPM dan Algoritma Genetika. Nilai tersebut lebih baik dibandingkan dengan nilai B yang diperoleh dari penjadwalan menggunakan CPM dengan berdasarkan waktu ES dan waktu LS. Perbedaan ini juga dapat dilihat dari grafik penggunaan biaya perhari pada Gambar 9. Dari grafik tersebut terlihat bahwa penyimpangan biaya dari penjadwalan proyek menggunakan kombinasi CPM dan Algoritma Genetika lebih rendah dibandingkan dengan yang lainnya.



Gambar 9. Grafik Perbandingan Penggunaan Biaya Perhari

Dengan mengacu pada grafik perbandingan kurva S pada Gambar 10, dapat dilihat bahwa kurva S yang dihasilkan dari penjadwalan proyek menggunakan kombinasi CPM dan Algoritma Genetika cenderung lebih mendekati garis linier dibandingkan dengan kurva S yang dihasilkan dari penjadwalan yang hanya menggunakan CPM saja.



Gambar 10. Grafik Perbandingan Kurva S

5. Kesimpulan dan Saran

5.1 Kesimpulan

Berdasarkan pembahasan yang telah diuraikan sebelumnya, maka diambil beberapa

kesimpulan sebagai berikut: (1) Algoritma genetika dapat digunakan untuk melakukan optimasi terhadap pembuatan jadwal proyek dengan menggabungkan CPM sehingga diperoleh sebuah jadwal proyek dengan penyeimbangan biaya yang lebih merata dibandingkan dengan tanpa menggunakan Algoritma Genetika; (2) Kombinasi metode ini menghasilkan suatu penjadwalan proyek yang berorientasi pada waktu dan penyeimbangan biaya proyek; (3) Sistem ini menghasilkan penjadwalan proyek yang lebih mendekati garis linier yang ditunjukkan dengan kurva S; (4) Hasil penjadwalan menggunakan kombinasi CPM dan Algoritma Genetika merupakan salah satu alternatif bagi para Kontraktor dalam melakukan penjadwalan proyek.

5.2 Saran

Berdasarkan pada pengujian yang telah dilakukan pada sistem yang dibuat, masih banyak kekurangan dan kelemahan sehingga perlu dikembangkan lagi agar kinerjanya lebih baik, oleh karena itu saran yang diberikan adalah sebagai berikut: (1) Pada penelitian ini menggunakan kombinasi dari CPM, disarankan pada penelitian selanjutnya dapat menggunakan metode selain CPM; (2) Untuk penelitian lebih lanjut diharapkan dapat dikembangkan sistem penjadwalan proyek yang dapat menangani lebih banyak kendala; (3) Untuk penelitian lebih lanjut diharapkan sistem penjadwalan proyek hendaknya dapat memberikan lebih banyak pilihan metode seleksi dan metode *crossover*.

Referensi

- [1] Cordon, O., Herrera, F., Hoffmann, F., and Magdalena, L., *Genetic Fuzzy Systems: Evolutionary Tuning and Learning Of Fuzzy Knowledge Bases*, British, World Scientific Publishing Co. Pte. Ltd., 2001,
- [2] Davis, L., *Handbook of Genetic Algorithms*, United States of America, Van Nostrand Reinhold, 1991.
- [3] Gen, M., and Cheng, R., "*Genetic Algorithms and Engineering Optimization*", United States of America, John Wiley and Son, 2000.
- [4] Goldsberg D.E., *Genetic Algorithms in search, Optimization & Machine Learning*, New York, Addison-Wesley, 1989.
- [5] Haupt, L.R., and Haupt, E.S., "*Practical Genetic Algorithm*", United States of America, John Wiley and Son, 2004.
- [6] Hegazy, T., Optimization of Resource Allocation and Leveling Using Genetic Algorithms, *Journal of Construction Engineering and Management*, Vol.125, No.3, 1999, Hal. 167–175.
- [7] Hiller, S.F. and Lieberman G.J., Pengantar Riset Operasi, Jakarta, Erlangga, 1994.
- [8] Kim J.L. and Ellis R. D., Permutation-Based Elitist Genetic Algorithm for Optimization of Large-Sized Resource-Constrained Project Scheduling, *Journal of Construction Engineering and Management*, Vol. 134, No. 11, November 1, 2008. ©ASCE, ISSN 0733-9364/2008/11-904–913.
- [9] Lawrence, J.A. and Pasternack B.A., *Applied Management Science: Modeling, Spreadsheet Analysis, and Communication For Decision Making*, United States of America, John Wiley and Son, 2001.
- [10] Michalewicz, Z., *Genetic Algorithm + Data Structure = Evolutions Programs, 3rd Edition*, New York, Springer-Verlag, 1996.
- [11] Negnevitsky, M., *Artificial Intelligence: a guide to intelligent systems*, Essex England, Pearson Education Limited, 2004.
- [12] Noranita, B., Penjadwalan Proyek dengan Parameter Waktu Pelaksanaan yang Tidak Pasti dan Batasan Persediaan Sumber Daya, *Tesis*, Yogyakarta, Program Pasca Sarjana Ilmu Komputer, Universitas Gadjah Mada, 2003.
- [13] Pressman, R.S., *Rekayasa Perangkat Lunak*, Yogyakarta, Andi And McGraw-Hill Book Co, 1997.
- [14] Turban E., *Decision Support Systems and Intelligent Systems-7th Ed.*, New Jersey, Pearson Education Inc., 2005.

[15] Taha, H.A., *Operations Research*, Jakarta, Bina Rupa Aksara, 1997.