

**OPTIMASI PENJADWALAN MATA KULIAH PADA
PERGURUAN TINGGI DENGAN MENGGUNAKAN
ALGORITMA GENETIKA**

SKRIPSI

Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik

**EVI FERAWATY
0606077094**



**UNIVERSITAS INDONESIA
FAKULTAS TEKNIK
PROGRAM STUDI TEKNIK INDUSTRI
DEPOK
JUNI 2010**

HALAMAN PERNYATAAN ORISINALITAS

**Skripsi ini adalah hasil karya saya sendiri,
dan semua sumber baik yang dikutip maupun yang dirujuk
telah saya nyatakan dengan benar.**

Nama : Evi Ferawaty

NPM : 0606077094

Tanda tangan :

Tanggal : Juli 2010

HALAMAN PENGESAHAN

Skripsi ini diajukan oleh

Nama : Evi Ferawaty
NPM : 0606077094
Program Studi : Teknik Industri
Judul Skripsi : Optimasi Penjadwalan Mata Kuliah pada
Perguruan Tinggi dengan Menggunakan
Algoritma Genetika

Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Teknik pada Program Studi Teknik Industri, Fakultas Teknik, Universitas Indonesia.

DEWAN PENGUJI

Pembimbing : Ir. Amar Rachman, MEIM. ()

Penguji : Ir. Boy Nurtjahyo, MSIE. ()

Penguji : Ir. M. Dachyar, MSc. ()

Penguji : Ir. Sri Bintang P., MSISE., PhD. ()

Ditetapkan di : Depok

Tanggal : 5 Juli 2010

KATA PENGANTAR

Puji syukur kepada Allah Bapa yang Mahakuasa karena berkat kasih, anugerah, dan penyertaan-Nya sehingga skripsi ini dapat selesai dengan baik. Penulis juga ingin menyampaikan terima kasih kepada Papa, Mama, Kakak, dan Adik saya atas kasih sayang dan dukungan yang diberikan selama penulis mengerjakan skripsi ini. Penulisan skripsi ini dilakukan dalam rangka melengkapi salah satu persyaratan untuk menyelesaikan Program Pendidikan Sarjana Teknik Industri, Fakultas Teknik Universitas Indonesia. Penulis menyadari bahwa tanpa bantuan dan bimbingan dari berbagai pihak, sangatlah sulit bagi penulis untuk menyelesaikan skripsi ini. Oleh karena itu, pada kesempatan ini, saya menyampaikan ucapan terima kasih kepada:

1. Bapak Ir. Amar Rachman, MEIM. selaku dosen pembimbing skripsi atas pembelajaran, saran, arahan, waktu dan bantuan yang telah diberikan selama pengerjaan skripsi.
2. Bapak Armand Omar Moeis, ST., M.Sc., Bapak Komarudin, Bapak Ir. Yadrifil, M.Sc., Bapak Ir. Farizal, M.Sc., Bapak Ahkmad Hidayatno atas masukan yang telah diberikan kepada penulis saat seminar.
3. Bapak Ir. Boy Nurtjahyo, MSIE., Bapak Ir. M. Dachyar, MSc., dan Bapak Ir. Sri Bintang P., MSISE., PhD. Yang telah memberikan penilaian serta perbaikan yang diperlukan pada saat sidang untuk membuat skripsi ini menjadi lebih baik.
4. Yanti Nitra, rekan seperjuangan dalam mengerjakan skripsi sejak awal sampai akhir serta sahabat yang telah berbagi pemikiran, cerita, hiburan, dan dukungan sejak awal kuliah hingga saat ini.
5. Rekan-rekan bimbingan penulis: Rizki Amalia, Tuty, Asa, Leo, Nicholas, Dira yang telah memberikan dukungan dan berbagi keluh kesah selama penulis mengerjakan skripsi. Dan untuk seluruh teman-teman angkatan 2006 Teknik Industri atas kebersamaan dan keceriaan bersama.

6. Teman-temanku yang tercinta: Sarah, Sisca, Naul yang telah menemani dalam suka dan duka selama menjalani 4 tahun kuliah bersama. Persahabatan kalian sangat berarti dan takkan kulupakan.
7. Teman-teman seperjuangan pulang-pergi Jakarta-Depok: Fitria, Dame, Rika, dan lainnya. Terutama untuk Fitria yang telah menemani pulang-pergi Jakarta-Depok selama 4 tahun bahkan sampai saat masa-masa sulit di semester 5 dan 6. Berbagi cerita bersamanya selama di perjalanan membuat perjalanan yang panjang jadi tidak terasa.
8. Teman-teman kelompok kecilku: Kak Frisca, Nana, Rika, Yanti, Shanti atas persekutuan bersama yang indah dalam Tuhan.
9. Kak Lismanto, Kak Nola, dan Kak Reza (lulusan FMIPA tahun 2008), atas bimbingan belajar yang diberikan sehingga penulis dapat memahami algoritma genetika dan bahasa pemograman MATLAB yang digunakan untuk mengerjakan skripsi ini.
10. Bu Har, Mbak Triana, Mbak Willy, Mas Mursyid, Mas Latief, dan seluruh karyawan Departemen Teknik atas bantuannya selama ini.
11. Semua pihak yang terlibat dan telah membantu penulis sehingga skripsi ini dapat terselesaikan dengan baik.

Akhirnya, penulis berharap agar skripsi ini bisa memberikan inspirasi dan manfaat bagi semua pihak yang membacanya dan bagi pengembangan ilmu pengetahuan.

Depok, 23 Juni 2010

Penulis

**HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI
TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS**

Sebagai sivitas akademik Universitas Indonesia, saya yang bertanda tangan di bawah ini:

Nama : Evi Ferawaty

NPM : 0606077094

Program Studi : Teknik Industri

Departemen : Teknik Industri

Fakultas : Teknik

Jenis Karya : Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Indonesia **Hak Bebas Royalti Noneksklusif** (*Non-exclusive Royalty-Free Right*) atas karya ilmiah saya yang berjudul:

**Optimasi Penjadwalan Mata Kuliah pada Perguruan Tinggi dengan
Menggunakan Algoritma Genetika**

beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Universitas Indonesia berhak menyimpan, mengalih media/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat dan mempublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok

Pada tanggal : Juni 2010

Yang Menyatakan

(Evi Ferawaty)

ABSTRAK

Nama : Evi Ferawaty
Program Studi : Teknik Industri
Judul : Optimasi Penjadwalan Mata Kuliah di Perguruan Tinggi dengan Menggunakan Algoritma Genetika

Penelitian ini dilakukan untuk memperoleh jadwal mata kuliah pada perguruan tinggi yang optimal, sehingga jumlah mata kuliah yang bentrok dapat dikurangi, beban mahasiswa lebih merata, serta penggunaan ruang kelas lebih optimal. Hal ini dilakukan dengan optimasi yaitu melalui penentuan jadwal mata kuliah. Optimasi jadwal dilakukan dengan menggunakan algoritma genetika. Algoritma genetika adalah salah satu algoritma yang tepat digunakan untuk memecahkan permasalahan dalam skala besar dan memiliki tingkat kompleksitas yang tinggi sehingga cocok untuk digunakan memecahkan masalah penjadwalan mata kuliah pada perguruan tinggi yang terkenal rumit dan memiliki banyak sekali variabel kendala yang harus dipenuhi dalam pembuatan jadwal yang baik. Hasil yang diharapkan dari penelitian ini adalah sebuah jadwal mata kuliah yang optimal berdasarkan jumlah mata kuliah yang bentrok, beban mahasiswa, dan penggunaan ruang kelas. Setelah penelitian dilakukan diperoleh jadwal baru yang optimal.

Kata Kunci:

Optimasi, penjadwalan mata kuliah, algoritma genetika

ABSTRACT

Name : Evi Ferawaty
Study Program : Industrial Engineering
Title : Optimization of University Course Timetabling Problem
Using Genetic Algorithm

The aim of this research is to gain an optimum schedule for university course timetabling problem, in order to decrease the number of clashed courses, even distribute student's study burden, and optimize class utilization. Schedule optimization were achieved using genetic algorithm. Genetic algorithm is one kind of algorithms that appropriate to be used for solving large scale problem with high complexity so it suitable to be used for solving university course timetabling problem which is an *NP-hard* problem and many constraints variable that has to be fulfilled in order to make a good schedule. The result of this research is an optimum schedule based on number of clashed courses, student's study burden, and class utilization. After the research is completely done, the result is a new optimum schedule.

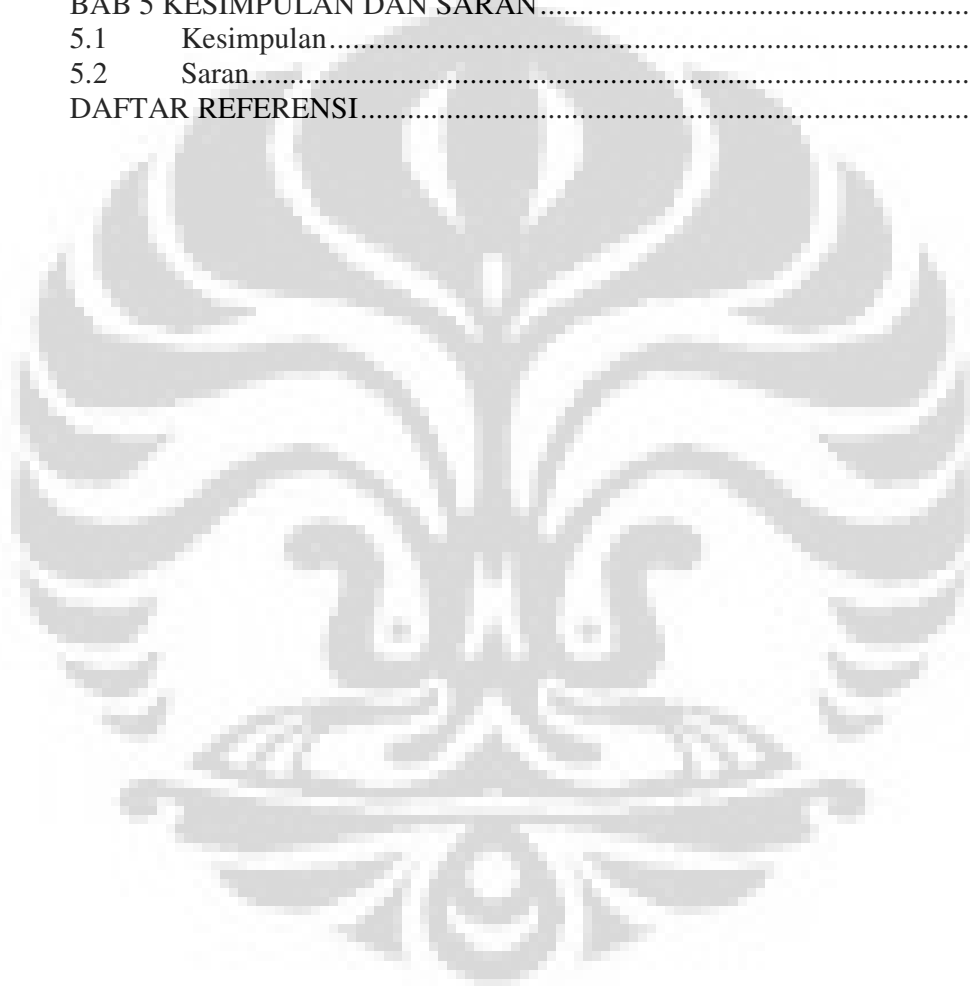
Keywords:

Optimization, university course timetabling problem, genetic algorithm

DAFTAR ISI

HALAMAN JUDUL.....	i
HALAMAN PERNYATAAN ORISINALITAS	ii
HALAMAN PENGESAHAN	iii
KATA PENGANTAR.....	iv
HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI	vi
ABSTRAK	vii
ABSTRACT.....	viii
DAFTAR ISI.....	ix
DAFTAR TABEL.....	xi
DAFTAR GAMBAR	xi
DAFTAR LAMPIRAN	xiii
1. PENDAHULUAN.....	1
1.1 Latar Belakang Masalah.....	1
1.2 Diagram Keterkaitan Masalah.....	4
1.3 Rumusan Masalah.....	4
1.4 Tujuan Penelitian	4
1.5 Ruang Lingkup Penelitian.....	6
1.6 Metodologi Penelitian	6
1.7 Sistematika Penulisan.....	8
2. LANDASAN TEORI	11
2.1 Programa Linier Integer	11
2.1.1 Definisi.....	11
2.1.2 Sejarah.....	12
2.1.3 Terminologi.....	12
2.2 Penjadwalan Kuliah Standar Internasional.....	13
2.2.1 Deskripsi Masalah Penjadwalan Kuliah Standar Internasional.....	13
2.2.2 Mesin Penjadwalan Kuliah	14
2.3 Algoritma Genetika (AG).....	15
2.3.1 Algoritma Genetika Untuk Optimasi	15
2.3.2 Komponen-komponen Algoritma Genetika	16
3. PENGUMPULAN DATA	26
3.1 Pengumpulan Data	26
3.1.1 Data Mata Kuliah	26
3.1.2 Data Ruangan Kuliah.....	28
3.1.3 Data Mahasiswa.....	31
3.1.4 Data Dosen	31
3.2 Pengalokasian Kelas Tiap Departemen.....	33
3.3 Matrik Hubungan Antara Dosen, Mahasiswa, dan Mata Kuliah.....	38
4. PENGOLAHAN DATA DAN ANALISIS	39
4.1 Model Penjadwalan Mata Kuliah di Departemen Teknik Industri UI.....	39
4.2 Penyusunan Algoritma Genetika	42

4.2.1 Langkah-Langkah Penyusunan Algoritma Genetika.....	42
4.2.2 Verifikasi dan Validasi Program	47
4.2.2.1 Hasil Piranti Lunak.....	50
4.2.2.2 Hasil Perhitungan Manual.....	50
4.2.3 Input Data.....	57
4.3 Hasil Pengolahan Data	61
4.4 Analisis.....	61
4.4.1 Analisis Metode.....	63
4.4.2 Analisis Skenario Parameter	65
4.4.3 Analisis Hasil	65
BAB 5 KESIMPULAN DAN SARAN.....	69
5.1 Kesimpulan.....	69
5.2 Saran.....	70
DAFTAR REFERENSI.....	71



DAFTAR TABEL

Tabel 3.1 Struktur Mata Ajaran Program Studi Teknik Industri	27
Tabel 3.2 Struktur Mata Ajaran Pilihan Program Studi Teknik Industri	28
Tabel 3.3 Data Ruangan Kuliah di Fakultas Teknik UI.....	28
Tabel 3.4 Data Mahasiswa Departemen Teknik Industri UI.....	31
Tabel 3.5 Data Dosen Departemen Teknik Industri UI Beserta Mata Kuliah yang Diajarkan pada Semester Ganjil.....	32
Tabel 3.6 Slot-Waktu yang Disediakan.....	34
Tabel 3.7 Total Kelas yang Tersedia untuk Perkuliahan di Fakultas Teknik UI.....	34
Tabel 3.8 Estimasi Jumlah Kelas yang Dibutuhkan untuk PDPT	35
Tabel 3.9 Perbandingan Kebutuhan Kelas Tiap Departemen	35
Tabel 3.10 Ringkasan Kebutuhan Kelas Tiap Departemen.....	37
Tabel 3.11 Jumlah Ruangan Kelas yang Dialokasikan untuk Departemen Teknik Industri	37
Tabel 3.12 Data Ruangan Kelas yang Dialokasikan untuk Departemen Teknik Industri	38
Tabel 4.1 Data <i>Dummy</i> untuk Mata Kuliah.....	48
Tabel 4.2 Data <i>Dummy</i> untuk Kapasitas Ruangan.....	48
Tabel 4.3 Data <i>Dummy</i> untuk Dosen	48
Tabel 4.4 Slot-Waktu yang Disediakan.....	49
Tabel 4.5 Matrik KK <i>Dummy</i>	49
Tabel 4.6 Matrik KT <i>Dummy</i>	49
Tabel 4.7 Parameter yang Digunakan Dalam Validasi	50
Tabel 4.8 Jadwal Terbaik Hasil <i>Run</i> Program untuk Data <i>Dummy</i>	50
Tabel 4.9 Populasi Awal (dari program).....	51
Tabel 4.10 Kromosom ke-1	51
Tabel 4.11 Penalti Kendala <i>Hard</i> dan <i>Soft</i> Setiap Gen pada Kromosom ke-1	52
Tabel 4.12 Kromosom ke-2	52
Tabel 4.13 Penalti Kendala <i>Hard</i> dan <i>Soft</i> Setiap Gen pada Kromosom ke-2	52
Tabel 4.14 Kromosom ke-3	52
Tabel 4.15 Penalti Kendala <i>Hard</i> dan <i>Soft</i> Setiap Gen pada Kromosom ke-3	53
Tabel 4.16 Kromosom ke-4	53
Tabel 4.17 Penalti Kendala <i>Hard</i> dan <i>Soft</i> Setiap Gen pada Kromosom ke-4	53
Tabel 4.18 Perhitungan Fitness Value pada Keempat Kromosom	53
Tabel 4.19 Kromosom Anak Hasil Mutasi	55
Tabel 4.20 Perhitungan Total Penalti Kendala <i>Hard</i> dan <i>Soft</i> Setiap Gen pada Kromosom Anak Hasil Mutasi	55
Tabel 4.21 Jadwal Terbaik Hasil Perhitungan Manual	56
Tabel 4.22 Parameter yang Digunakan dalam Pembuatan Model.....	59
Tabel 4.23 Kromosom Jadwal Hasil 10000 Iterasi	60
Tabel 4.24 Jumlah Nilai Penalti <i>Hard</i> dan <i>Soft</i>	60
Tabel 4.25 Kromosom Jadwal dengan Input Matrik $KT_{ K \times T }$ Baru	61
Tabel 4.26 Jumlah Nilai Penalti <i>Hard</i> dan <i>Soft</i>	61
Tabel 4.27 Kromosom Jadwal dengan Input 6 Ruang Kelas.....	62
Tabel 4.28 Jumlah Nilai Penalti <i>Hard</i> dan <i>Soft</i>	62
Tabel 4.29 Hasil Kombinasi Parameter.....	65

DAFTAR GAMBAR

Gambar 1.1 Diagram Keterkaitan Masalah	5
Gambar 1.2 Diagram Alir Metodologi Penelitian.....	10
Gambar 2.1 Representasi Kromosom Menggunakan <i>Binary</i> Encoding. Kromosom tersebut mengkodekan dua variabel, x_1 & x_2 , masing-masing 10 <i>bit</i> (<i>binary digit</i>)	15
Gambar 2.2 Pseudocode Algoritma Genetika.....	16
Gambar 2.3 Tiga Skema Pengkodean	17
Gambar 2.4 Contoh Penggunaan Metode Roulette-Wheel Selection	22
Gambar 2.5 Contoh Proses Pindah Silang.....	23
Gambar 2.6. Contoh Proses Mutasi.....	24
Gambar 4.1 Contoh Uniform Crossover pada Pengkodean Biner dengan Titik Persilangannya yang Acak adalah 2, 3, 5, dan 7	45
Gambar 4.2 <i>Exchange Mutation</i>	45
Gambar 4.3 Diagram Alur Pengerjaan Algoritma Genetika.....	46
Gambar 4.4 Kawin Silang dengan <i>Uniform Crossover</i>	54
Gambar 4.5 Mutasi	54
Gambar 4.6 Representasi Jadwal	57
Gambar 4.7 Kromosom Jadwal sebagai Kandidat Solusi.....	58
Gambar 4.8 Hasil <i>Running</i> Program Setelah 15000 Iterasi.....	66

DAFTAR LAMPIRAN

Lampiran 1. Matrik KK (Matrik Hubungan antar Mata Kuliah)	73
Lampiran 2. Matrik KT (Hubungan antara Mata Kuliah dengan Slot Waktu)	76
Lampiran 3. Matrik KM (Hubungan antara Mata Kuliah dengan Mahasiswa)	80
Lampiran 4. <i>Source Code</i> Program MATLAB R2007a Program Penjadwalan Mata Kuliah.....	81
Lampiran 5. Kromosom Terbaik Hasil 10000 Iterasi.....	86
Lampiran 6. Jadwal Optimal Hasil 10000 Iterasi.....	87



BAB 1 PENDAHULUAN

1.1 Latar Belakang Masalah

Ada beberapa hal yang menjadi latar belakang dilakukannya penelitian ini. Salah satunya adalah sering terjadinya bentrok antara beberapa mata kuliah sehingga menyulitkan baik bagi mahasiswa maupun dosen. Hal ini telah terjadi hampir setiap tahun ajaran. Dengan mempertimbangkan hal ini maka sudah sebaiknya dilakukan optimasi penjadwalan mata kuliah pada perguruan tinggi.

Sebelum dijelaskan lebih lanjut mengenai latar belakang masalah, lebih baik memahami terlebih dahulu arti dari penjadwalan secara umum. Pemahaman masalah penjadwalan secara umum adalah kegiatan pengalokasian sejumlah sumber daya dalam jangka waktu tertentu ke dalam berbagai operasi yang mungkin dengan batasan atau syarat tertentu yang harus dipenuhi sehingga fungsi objektif sedekat mungkin dapat terpenuhi. Salah satu masalah penjadwalan yang terkenal sulit baik secara teoritis maupun praktis adalah masalah penjadwalan mata kuliah pada perguruan tinggi¹, atau sering dikenal dengan istilah *University Course Timetabling Problem (UCTP)*.

Dua masalah penjadwalan yang umum dijumpai di perguruan tinggi adalah masalah penjadwalan mata kuliah dan ujian. Pada penjadwalan kuliah sejumlah mata kuliah harus dijadwalkan ke dalam ruang dan slot-slot waktu tertentu, dimana beberapa kendala harus dipenuhi agar mata kuliah yang telah dijadwalkan tidak akan bentrok. Sering kali terjadi penempatan kapasitas kelas yang tidak tepat untuk suatu mata kuliah tertentu. Mata kuliah wajib yang akan diikuti oleh ratusan mahasiswa pada satu semester tertentu hanya memiliki kapasitas kelas yang sedikit. Hal ini menimbulkan masalah bagi para mahasiswa tersebut. Banyak dari mereka yang kemungkinan tidak dapat terdaftar pada mata kuliah tersebut dikarenakan kapasitas kelas yang dibuka telah penuh. Sedangkan mereka tidak mungkin tidak mendaftar pada suatu mata kuliah wajib untuk semester tersebut karena itu berarti mereka harus menunggu satu atau dua semester untuk dapat mengambilnya. Dan hal itu membuat beban perkuliahan mereka bertambah.

¹A. Schaerf, A survey of automated timetabling, *Artificial Intelligence Review*,13(2): 87-127,1999

Selain itu, seringkali ada jadwal dua mata kuliah pilihan yang bentrok atau bahkan antar mata kuliah wajib yang diambil pada semester yang sama juga bentrok. Jadwal mata kuliah yang bentrok biasanya disebabkan oleh penjadwalan yang masih belum optimal sehingga jadwal yang disusun saling bertumpuk. Sistem penjadwalan yang ada saat ini juga mempunyai kekurangan lain, yaitu tidak meratanya beban mahasiswa setiap harinya. Sering kali mahasiswa dibebani dengan mata kuliah yang banyak pada satu hari tertentu saja. Hal ini membuat mahasiswa kesulitan mengatur waktu studi.

Dari penjelasan di atas, maka dilakukanlah penelitian agar dapat disusun jadwal yang terbaik sehingga masalah jadwal yang bentrok dapat dihindari. Untuk menyusun jadwal kuliah diperlukan penyesuaian antara waktu dan ruangan kelas yang tersedia agar kapasitas mata kuliah yang bersangkutan tidak melebihi kapasitas ruang kelas tempat diselenggarakannya kuliah tersebut.

Untuk mendapatkan penjadwalan mata kuliah yang tepat tersebut, maka perlu dilakukan metode optimasi. Optimasi merupakan suatu metode dengan pendekatan matematis yang bertujuan untuk memperoleh tujuan yang maksimal dengan *cost* (biaya) yang minimal. Pengertian *cost* dalam hal ini dapat berupa uang, sumber daya, tenaga kerja, atau berbagai bentuk pengorbanan lain yang perlu dikeluarkan untuk mencapai suatu tujuan. Hingga saat ini ada berbagai jenis teknik-teknik optimasi. Namun, untuk kasus-kasus yang berskala besar dibutuhkan teknik-teknik optimasi yang lebih modern seperti algoritma optimasi.

Algoritma Genetika (AG) merupakan algoritma yang meniru proses seleksi alam dan dapat digunakan untuk memecahkan masalah optimasi yang rumit yang memiliki ruang lingkup pencarian yang sangat luas^{2,3,4,5,6}. Algoritma

² Goldberg, D. E. "Genetic Algorithms : In Search, Optimization and Machine Learning". 1989, Addison-Wesly Publishing Co.

³ Wilson, S. W. "Hierarchical Credit Allocation in a Classifier System", Research Notes on Artificial Intelligence, Genetic Algorithms and Simulated Annealing, Pittman, 1987, London.

⁴ Holland, J. H, Holyonk, K. J., Nisbett, R. E., Thayard, P. R. "Classifier Systems, Q-Morphisms and Induction". Research Notes on Artificial Intelligence, Genetic Algorithms and Simulated Annealing, Pittman, 1987, London.

⁵ Robertson, G. "Parallel Implementation of Genetic Algorithms in a Classifier Design". Research Notes on Artificial Intelligence, Genetic Algorithms and Simulated Annealing, Pittman, 1987, London.

⁶ Glover, D. E. "Solving a Complex Keyboard Configuration Problem Through Generalized Adaptive Search". Research Notes on Artificial Intelligence, Genetic Algorithms and Simulated Annealing, Pittman, 1987, London.

genetika diinspirasi dari proses evolusi Darwin, yaitu suatu prosedur pencarian solusi yang alami berdasarkan pada metode optimasi⁷. Tidak seperti metode meta-heuristik lainnya yang hanya memiliki satu solusi sub-optimal pada satu waktu, AG memelihara banyak solusi individual dalam bentuk sebuah populasi. Individu-individu atau orang tua dipilih dari populasi dan kemudian dikawinkan untuk membentuk satu individu baru (anak). Anak ini akan dimutasi lebih jauh untuk memasukkan perbedaan ke dalam populasi.

Dalam seleksi alam, beberapa individu dilahirkan dengan sifat-sifat yang berasal dari orang tua mereka, dan dengan demikian mewarisi sifat orang tua mereka. Pada umumnya, sifat-sifat tersebut ditandai dengan gen-gen khusus, yang digabungkan ke dalam kromosom. Sifat-sifat dari anak tersebut dapat dimodifikasi lebih lanjut melalui mutasi, yang mengubah kromosom-kromosom tadi untuk memberikan sifat-sifat baru yang tidak ditemukan pada orang tua yang manapun. Jika sifat-sifat gabungan dari seorang anak dapat membuatnya lebih sesuai dengan lingkungan, maka kemungkinan anak tersebut akan bertahan hidup dan bereproduksi meningkat. Sebagai akibatnya, sifat-sifat yang diinginkan dari anak tersebut akan diwariskan kepada generasi selanjutnya. Akan tetapi, jika seorang anak lahir dengan sifat-sifat yang tidak diinginkan, yang membuat anak tersebut kurang dapat menyesuaikan diri dengan lingkungannya, maka kemungkinan anak tersebut akan mati sebelum bereproduksi, dan sifat-sifat yang kurang diinginkan tadi akan dikeluarkan dari generasi selanjutnya. Efek yang ditimbulkan oleh individu-individu yang bertahan adalah meningkatnya rata-rata kesesuaian (*fitness*) dari populasi meningkat tiap generasi. Adanya populasi menyebabkan terjadinya keanekaragaman dalam populasi, yang memperbolehkan sifat-sifat yang baru dan lebih baik untuk dapat diciptakan.

Evolusi tiruan memperkenankan prinsip individu yang dapat bertahan adalah individu yang paling dapat menyesuaikan diri dengan lingkungan dapat diaplikasikan pada masalah-masalah optimasi lainnya, yang memiliki tujuan untuk mencari satu individu yang benar-benar sesuai. Sebagai contoh, jika masalah

⁷ F. Sivrikaya S. , Erifog˘lu and G Ulusoy, “Multiprocessor Task Scheduling In Multistage Hybrid Flow-Shops: A Genetic Algorithm Approach”, in Journal of the Operational Research Society, Vol. 55, No. 5, 2004, hal.505.

optimisasi memerlukan sebuah jadwal untuk dibuat di mana seorang pengajar tidak dapat hadir lebih dari satu kali dalam satu waktu tertentu yang terdapat dalam daftar jadwal yang disediakan, lalu solusi yang didapatkan akan diperlihatkan oleh satu daftar jadwal di mana tidak ada dosen yang mengalami bentrok. Populasi terdiri dari sekumpulan daftar jadwal yang berbeda, yang banyak di antaranya tidak dapat mencapai hasil yang optimal. Hasil yang sedemikian akan dilahirkan dari beberapa orang tua (*parents*) di mana dalam setiap daftar jadwal memiliki pengajar yang mengalami bentrok, tetapi kombinasi khusus dari beberapa gen orang tua memperbolehkan daftar jadwal yang baru untuk dapat dibuat tanpa adanya bentrok.

1.2 Diagram Keterkaitan Masalah

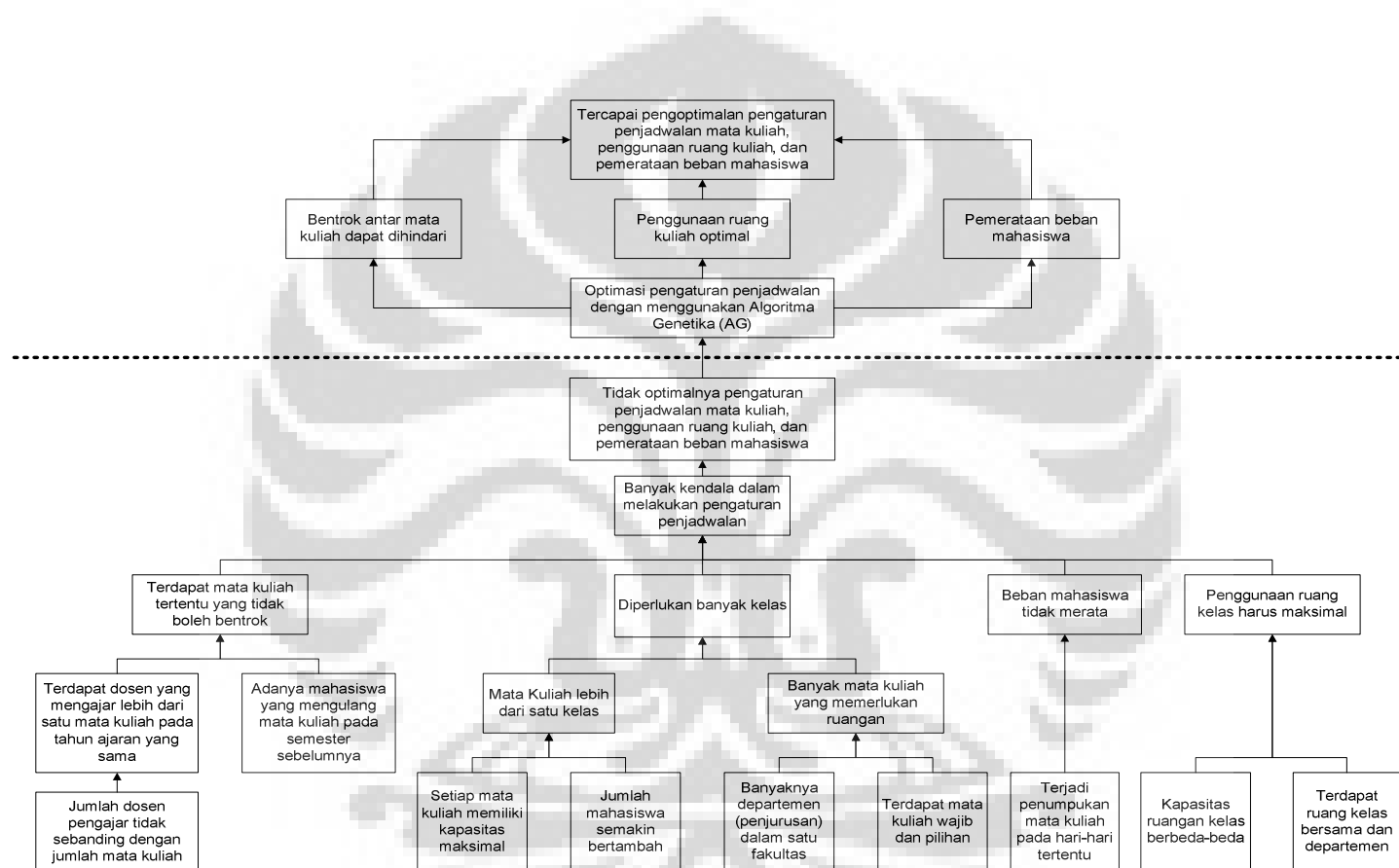
Berdasarkan latar belakang masalah di atas, maka dapat dibuat diagram keterkaitan masalah yang menampilkan permasalahan secara visual dan sistematis. Diagram keterkaitan masalah untuk penelitian ini ditunjukkan oleh Gambar 1.1.

1.3 Rumusan Masalah

Permasalahan yang dijadikan fokus dari penelitian ini adalah perlunya perancangan suatu sistem penjadwalan kuliah pada perguruan tinggi yang efisien dengan menggunakan Algoritma Genetika. Solusi yang dihasilkan adalah solusi yang paling mendekati optimal dimana jumlah jadwal yang bentrok dapat dikurangi dan beban mahasiswa lebih merata.

1.4 Tujuan Penelitian

Berdasarkan latar belakang dan diagram keterkaitan masalah yang telah dijelaskan sebelumnya, maka tujuan dari penelitian ini adalah memperoleh penjadwalan kuliah yang optimal pada perguruan tinggi, yang memenuhi semua kendala *hard* dan kendala *soft*, sehingga dapat meminimalkan jumlah jadwal yang bentrok dan pemerataan beban mahasiswa dengan menggunakan metode Algoritma Genetika (AG).



Gambar 1.1 Diagram Keterkaitan Masalah

1.5 Ruang Lingkup Penelitian

Dalam penelitian ini dilakukan pembatasan masalah agar pelaksanaan serta hasil yang akan diperoleh sesuai dengan tujuan pelaksanaannya. Adapun batasan masalahnya adalah:

1. Penelitian ini hanya meliputi wilayah kampus Universitas Indonesia. Meskipun demikian, hasil dari penelitian ini akan dapat diterapkan pula pada perguruan tinggi lainnya.
2. Optimasi penjadwalan hanya dilakukan untuk perkuliahan yang diadakan di Fakultas Teknik.
3. Dari 7 departemen yang ada di Fakultas Teknik, hanya akan dipilih penjadwalan pada departemen Teknik Industri untuk memperkecil model penjadwalan.
4. Data yang digunakan diperoleh dari Departemen Teknik Industri UI semester ganjil tahun 2009.

1.6 Metodologi Penelitian

Metode penelitian yang digunakan, dalam penelitian ini secara sistematis dapat dilihat dari Gambar 1.2 dengan uraian sebagai berikut:

1. Perumusan ide-ide topik penelitian dan mengidentifikasi permasalahan
Pada tahap pertama, dilakukan pencarian tema-tema yang menarik untuk diangkat, baik dari pencarian pada situs-situs internet, jurnal, maupun buku.
2. Studi literatur dasar teori penelitian
Dilakukan studi literatur untuk menentukan landasan teori yang berhubungan dengan topik penelitian ini sebagai dasar dalam pelaksanaan penelitian. Landasan teori ini kemudian akan dijadikan acuan dalam pelaksanaan tugas akhir. Adapun landasan teori yang terkait dengan penelitian ini diantaranya adalah Program Linier Integer, Teori Penjadwalan, dan Algoritma Genetika.
3. Perumusan masalah
Berdasarkan identifikasi masalah dan studi literatur teori, dapat dirumuskan masalah dalam penelitian ini, yaitu perlu dilakukannya

optimasi jadwal mata kuliah pada perguruan tinggi yang dapat diaplikasikan pada perguruan tinggi manapun.

4. Penentuan topik penelitian

Adapun topik dari penelitian ini adalah Optimasi Penjadwalan Mata Kuliah Pada Perguruan Tinggi Menggunakan Algoritma Genetika.

5. Penentuan tujuan penelitian

Adapun tujuan penelitian ini adalah memperoleh penjadwalan mata kuliah yang optimal sehingga dapat mengurangi jumlah mata kuliah yang bentrok dan pemeratakan beban mahasiswa.

6. Pengumpulan data

Dalam melakukan penelitian ini, data yang dibutuhkan adalah data ruangan-ruangan kelas yang tersedia serta kapasitas masing-masing ruangan untuk tempat kegiatan perkuliahan berlangsung. Dari ruangan-ruangan yang tersedia tersebut akan dicari mata kuliah yang kapasitasnya sesuai sehingga tidak terjadi kelebihan kapasitas. Selain itu, untuk dapat menentukan jadwal mata kuliah agar tidak bentrok, maka dibutuhkan juga nama dosen beserta nama mata kuliah yang diajarnya dan ditujukan untuk semester berapa mata kuliah tersebut diajarkan. Jumlah mahasiswa tiap angkatan juga diperlukan untuk menentukan jumlah mahasiswa pengambil suatu mata kuliah tertentu.

7. Pengolahan data

Pada tahapan ini, dilakukan pengolahan data dari data yang sudah diperoleh. Pengolahan data ini dilakukan dengan membuat matrik-matrik yang memperlihatkan hubungan antar variabel yang ingin diteliti. Matrik-matrik tersebut akan digunakan sebagai input bagi pembuatan model penjadwalan.

8. Pembuatan model optimasi penjadwalan mata kuliah pada perguruan tinggi

Setelah pemantapan studi terhadap dasar teori dan pengolahan data, maka dapat dibuat model optimasi menggunakan perangkat lunak MATLAB R2007a, yaitu merangkai kode-kode logika pemrograman yang dapat mengaplikasikan algoritma genetika pada masalah penjadwalan.

9. Analisis hasil model optimasi jadwal

Dalam tahap ini dilakukan analisis terhadap jadwal yang telah dibuat dengan bantuan perangkat lunak MATLAB untuk memperoleh tujuan penelitian.

10. Kesimpulan

Dalam tahapan ini akan dihasilkan kesimpulan mengenai keseluruhan penelitian. Kesimpulan dari penelitian ini merupakan ringkasan dari pengolahan data dan analisis yang telah dilakukan sebelumnya.

1.7 Sistematika Penulisan

Untuk mempermudah pemahaman alur penelitian ini, maka penulisan penelitian mengenai optimasi penjadwalan mata kuliah pada perguruan tinggi ini disajikan dalam beberapa bab. Bab pertama adalah Pendahuluan, yang menjelaskan mengenai latar belakang permasalahan yang menyebabkan dilakukannya penelitian ini. Selain itu, tujuan penelitian dan metodologi penelitian juga dipaparkan dalam bab ini. Penjelasan dalam bab Pendahuluan dilengkapi pula dengan diagram-diagram yang dapat menggambarkan alur permasalahan dan alur penelitian secara sistematis, yaitu diagram keterkaitan masalah dan diagram alir metodologi penelitian.

Pada bab kedua, dipaparkan dasar teori yang digunakan dalam mengerjakan penelitian ini. Landasan teori ini diperoleh dari tinjauan pustaka baik dari buku, jurnal, artikel, maupun informasi dari situs-situs di internet. Teori-teori yang dipakai meliputi teori Program Linier Integer, Teori Penjadwalan, dan Algoritma Genetika, yaitu metode yang digunakan untuk melakukan optimasi.

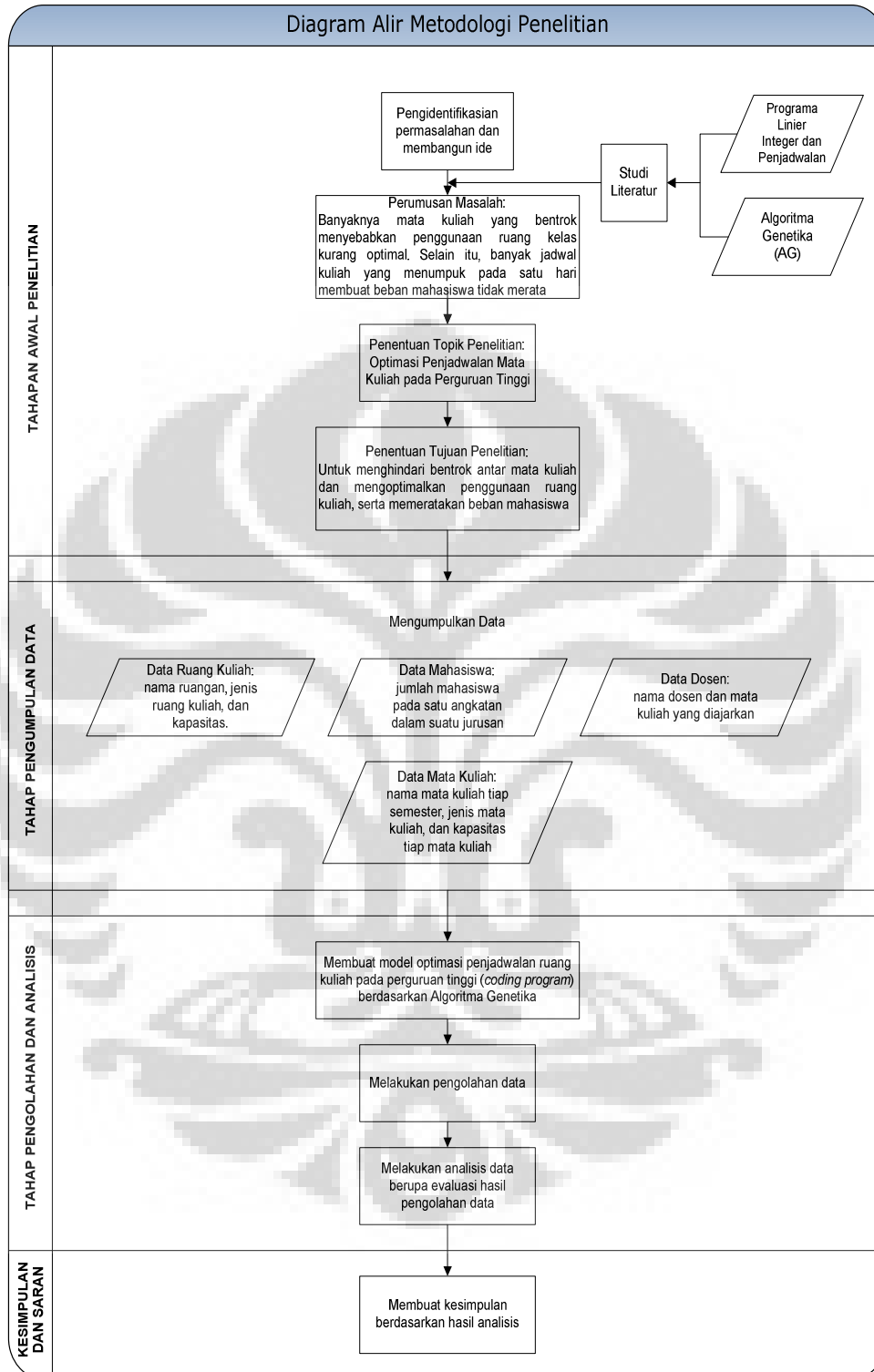
Bab ketiga berisi pengumpulan data yang dibutuhkan dalam melakukan penelitian ini. Data-data tersebut adalah data mata kuliah yang berupa nama, jumlah sks dan semester diperuntukkannya, data dosen yang berupa nama dan mata kuliah yang diajarkannya, data ruangan berupa nama dan kapasitasnya, dan data jumlah mahasiswa tiap angkatan.

Bab keempat merupakan bab yang berisi pengolahan data dan analisis. Pada bab ini dijelaskan secara terperinci langkah-langkah yang digunakan dalam pengolahan data sampai diperoleh hasil yang diharapkan. Pengolahan data pada

penelitian ini dilakukan dengan menggunakan algoritma genetika. Setelah melakukan pengolahan data, dilakukan analisa hasil pengolahan data.

Setelah pengolahan data dan analisa dilakukan, maka dapat ditarik suatu kesimpulan dari pengerjaan penelitian secara keseluruhan. Kesimpulan ini merupakan jawaban dari tujuan penelitian. Kesimpulan ini ditulis pada bab yang kelima. Selain itu, pada bab lima juga akan dituliskan saran yang berguna untuk memperbaiki kondisi penjadwalan mata kuliah yang ada di lingkungan Fakultas Teknik UI di masa mendatang.





Gambar 1.2 Diagram Alir Metodologi Penelitian

BAB 2

LANDASAN TEORI

Landasan teori yang digunakan dalam penelitian ini dipilih dari berbagai literatur yang berhubungan dengan masalah penjadwalan mata kuliah pada perguruan tinggi yang terdiri dari Program Linier Integer, Teori Penjadwalan, dan Algoritma Genetika. Masing-masing teori memiliki kegunaan yang berbeda-beda dalam proses pembuatan jadwal. Program linier integer menjadi landasan bagi pembuatan dan penulisan model matematis masalah penjadwalan mata kuliah pada perguruan tinggi. Teori penjadwalan menjadi dasar untuk menentukan kendala-kendala yang harus dipenuhi agar jadwal yang dibentuk tidak bentrok. Sedangkan algoritma genetika menjadi landasan bagi langkah-langkah pembuatan bahasa program dengan perangkat lunak MATLAB.

2.1 Program Linier Integer

Sebagian besar masalah yang dapat diselesaikan menggunakan algoritma adalah masalah-masalah di bidang optimasi; misalnya penentuan rute terpendek, biaya paling murah, dan lain-lain. Dalam kasus-kasus seperti itu, kita mencari sebuah solusi yang:

- (1) Memenuhi beberapa kendala
- (2) Merupakan kemungkinan yang terbaik, yang berdasarkan pada beberapa kriteria yang didefinisikan dengan baik, di antara semua solusi yang memenuhi kriteria-kriteria tersebut.

Program linier menjelaskan masalah-masalah optimasi secara luas dimana semua kendala dan standar optimasi dinyatakan dalam bentuk fungsi linier. Sebagian besar masalah-masalah lain juga dapat dinyatakan dalam bentuk fungsi linier.

2.1.1 Definisi

Program linier integer merupakan salah satu cabang ilmu matematika terapan yang dapat digunakan untuk memecahkan masalah optimasi dalam bentuk tertentu dimana semua variabel yang terdapat harus berbentuk integer.

Permasalahan program linier dapat berupa fungsi linier dari biaya (terdiri dari sejumlah variabel) yang harus diminimalkan atau memaksimalkan tujuan berdasarkan sejumlah kendala. Kendala-kendala tersebut dinyatakan dalam bentuk pertidaksamaan sejumlah variabel yang digunakan dalam fungsi biaya. Fungsi biaya ini seringkali disebut sebagai fungsi objektif. Program linier erat hubungannya dengan aljabar linier; perbedaan yang paling jelas terlihat yaitu program linier sering menggunakan pertidaksamaan daripada persamaan dalam pernyataan masalah.

2.1.2 Sejarah

Program linier merupakan cabang ilmu matematika yang usianya relatif muda, berawal dari penemuan metode simpleks oleh G.B. Dantzig pada tahun 1947. Menurut sejarah, perkembangan program linier dipicu oleh kegunaannya dalam bidang ekonomi dan manajemen. Awalnya Dantzig mengembangkan metode simpleks untuk memecahkan masalah perencanaan U.S. Air Force, dan masalah perencanaan dan penjadwalan masih mendominasi penggunaan dari program linier. Salah satu alasan yang menyebabkan program linier merupakan bidang yang relatif baru adalah hanya model program linier terkecil saja yang dapat diselesaikan tanpa bantuan komputer.

2.1.3 Terminologi

Masalah program linier dituliskan dalam bentuk standar sebagai:

Maksimalkan atau minimalkan:

$$Z = \sum_{j=1}^n c_j x_j \quad (2.1)$$

Dengan kendala:

$$\sum_{j=1}^n a_{ij} x_j (\leq, =, \geq) b_i \quad i = 1 \dots m \quad (2.2)$$

$$x_j \begin{cases} 1 \\ 0 \end{cases} \quad j = 1 \dots n \quad (2.3)$$

Dimana:

x_j = banyaknya kegiatan j , dimana $j = 1, 2, \dots, n$, yang berarti terdapat n variabel keputusan

Z = nilai fungsi tujuan

c_j = sumbangan per unit kegiatan j , untuk masalah maksimasi c_j menunjukkan atau penerimaan per unit, sementara dalam kasus minimasi ia menunjukkan biaya per unit

b_i = jumlah sumber daya ke i ($i = 1, 2, \dots, m$), berarti terdapat m jenis sumber daya

x_{ij} = banyaknya sumber daya i yang dikonsumsi oleh sumber daya j

2.2 Penjadwalan Kuliah Standar Internasional

Masalah penjadwalan kuliah bermula dari kompetisi membuat jadwal kuliah di 20 instansi di Universitas Napier Edinburg Skotlandia. Model permasalahan telah ditetapkan, tetapi partisipan dapat menggunakan berbagai metode untuk menyelesaikannya. Berikut akan dibahas model masalah penjadwalan kuliah standar internasional dan mesin penjadwalan kuliah.

2.2.1 Deskripsi Masalah Penjadwalan Kuliah Standar Internasional

Masalah penjadwalan mata kuliah standar internasional, dikenal dengan nama *University Course Timetabling Problem (UCTP)*, akan menjadwalkan sejumlah kuliah dengan 45 slot-waktu yakni 9 slot-waktu per hari dari Senin sampai dengan Jumat. Komponen yang mempengaruhi jadwal terdiri dari kapasitas ruangan yang tersedia, waktu kehadiran dosen, mahasiswa pengikut kuliah, serta batasan-batasan yang ditentukan.

Batasan atau kendala yang terdapat dalam *UCTP* terbagi dalam dua kategori, yaitu kendala *hard* dan kendala *soft*. Kendala *hard* merupakan kendala penting yang harus terpenuhi dalam pembuatan jadwal, sedangkan kendala *soft* merupakan kendala yang menentukan kualitas dari jadwal yang dibuat.

Kendala *hard* dan *soft* terjadi akibat interaksi antar komponen jadwal, seperti kapasitas ruang dan jumlah mahasiswa pengikut kuliah, selain itu interaksi juga dapat terjadi antar kuliah-kuliah yang tidak boleh dijadwalkan pada waktu yang bersamaan. Klasifikasi kedua kendala tersebut telah ditentukan oleh

Universitas Napier karena disesuaikan dengan kondisi penjadwalan kuliah di Universitas tersebut.

Berikut adalah kendala *hard* dalam *UCTP*, untuk membentuk solusi jadwal kuliah yang layak kendala-kendala ini harus terpenuhi:

1. H1: Dua atau lebih kuliah tidak dapat diberikan pada suatu waktu dalam ruangan yang sama.
2. H2: Setiap kuliah harus diselenggarakan pada ruangan yang memenuhi fasilitas dan kapasitas untuk kuliah tersebut.
3. H3: Tidak ada mahasiswa yang mendapat dua atau lebih kuliah pada waktu yang bersamaan.

Setelah seluruh kendala *hard* terpenuhi dalam pembentukan jadwal, maka akan dilanjutkan dengan meminimumkan fungsi penalti kendala *soft*, yakni pelanggaran jadwal terhadap kendala *soft* sedapat mungkin diminimalisasi. Berikut adalah kendala *soft* yang telah didefinisikan dalam *UCTP*:

1. S1: Mahasiswa tidak memperoleh kuliah pada slot-waktu terakhir setiap hari.
2. S2: Mahasiswa tidak mendapat kuliah lebih dari 6 jam berturut-turut.
3. S3: Mahasiswa tidak memperoleh hanya satu mata kuliah dalam satu hari.

2.2.2 Mesin Penjadwalan Kuliah

Sejarah mesin penjadwalan kuliah bermula dari kompetisi menyelesaikan masalah penjadwalan kuliah standar internasional. Pengujian dan penilaian kompetisi ini menggunakan data 20 instansi di Universitas Napier Edinburg Skotlandia dimana penalti *soft* terbaik dan terburuk setiap mata kuliah telah diketahui. Kompetisi tersebut mengundang perhatian partisipan dari seluruh penjuru dunia dengan berbagai metode pendekatan.

Penilaian untuk masing-masing partisipan berdasarkan pada penalti kendala *soft* yang dihitung sebagai berikut:

$$p^j = \sum \frac{f_i(j_i^l) - f_i(j_i^p)}{f_i(j_i^w) - f_i(j_i^p)} \quad (2.4)$$

dimana:

j = Partisipan ke- j

i = himpunan 20 instansi

f_i = penalti *soft* instansi ke- i

j_i^j = Jadwal yang disusun oleh partisipan ke- j untuk instansi ke- i

j_i^b = Jadwal terbaik pada instansi ke- i yang telah diketahui

j_i^w = Jadwal terburuk pada instansi ke- i yang telah diketahui

Penjumlahan suku-suku tersebut menghasilkan penalti untuk partisipan ke- j , sehingga pemenang kompetisi adalah partisipan dengan penalti minimum.

2.3 Algoritma Genetika (AG)

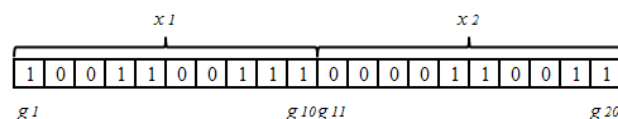
Kemunculan AG diinspirasi dari teori-teori dalam ilmu biologi, sehingga banyak istilah dan konsep biologi yang digunakan dalam AG. Sesuai dengan namanya, proses-proses yang terjadi dalam AG sama dengan apa yang terjadi dalam evolusi biologi.

2.3.1 Algoritma Genetika Untuk Optimasi

Dalam bukunya, DE Goldberg mendefinisikan algoritma genetika sebagai algoritma pencarian yang didasarkan pada mekanisme seleksi alamiah dan genetika alamiah. Untuk lebih memahami definisi ini, marilah kita perhatikan suatu masalah sederhana berikut ini. Berapa nilai x_1 dan x_2 yang membuat fungsi

$$h(x_1, x_2) = 4x_1 - 5x_2 \quad (2.5)$$

menjadi maksimum? Jika diketahui bahwa $x_1, x_2 \in [-4, 3]$. Pada interval tersebut, fungsi h mencapai maksimum, yakni bernilai 32, pada saat $x_1=3$ dan $x_2=-4$. Bagaimana menyelesaikan masalah tersebut dengan menggunakan AG? Pertama, kedua variabel, x_1 dan x_2 , dikodekan ke dalam kromosom. Masing-masing kromosom berisi sejumlah gen, yang mengkodekan informasi yang disimpan di dalam kromosom. Untuk menyelesaikan masalah di atas, misal digunakan *binary encoding* dengan panjang kromosom 20 gen, masing-masing variabel dikodekan dengan 10 gen. Gambar 2.1 berikut ini mengilustrasikan skema pengkodean tersebut.



Gambar 2.1 Representasi Kromosom Menggunakan *Binary Encoding*.

Kromosom tersebut mengkodekan dua variabel, x_1 & x_2 , masing-masing 10 *bit* (*binary digit*)

Setelah skema pengkodean ditentukan, AG diinisialisasi untuk sebuah populasi dengan N kromosom. Gen-gen yang mengisi masing-masing kromosom dibangkitkan secara random, biasanya menggunakan distribusi seragam (*uniform distribution*). Masing-masing kromosom akan dikodekan menjadi individu dengan nilai *fitness* tertentu. Sebuah populasi baru dihasilkan dengan menggunakan mekanisme seleksi alamiah, yaitu memilih individu-individu secara proporsional dengan nilai *fitness*-nya, dan genetika alamiah, yakni pindah silang dan mutasi. Sebuah *pseudocode* pada Gambar 2.2 di bawah ini memperlihatkan langkah-langkah AG.

```

Inisialisasi populasi, N kromosom
Loop
  Loop untuk N kromosom
    Dekodekan kromosom
    Evaluasi kromosom
  End
  Buat satu atau dua kopi kromosom terbaik (elitisme)
  Loop sampai didapatkan N kromosom baru
    Pilih dua kromosom
    Pindah silang
    Mutasi
  End
End

```

Gambar 2.2 *Pseudocode* Algoritma Genetika

Pada algoritma genetika di atas digunakan skema penggantian populasi disebut *generational replacement*. Artinya N kromosom dari suatu populasi digantikan sekaligus oleh N kromosom baru hasil pindah silang dan mutasi.

2.3.2 Komponen-komponen Algoritma Genetika

Pada dasarnya AG memiliki tujuh komponen. Tetapi banyak metode yang bervariasi yang diusulkan pada masing-masing komponen tersebut. Masing-masing metode mempunyai kelebihan dan kekurangan. Suatu metode yang bagus

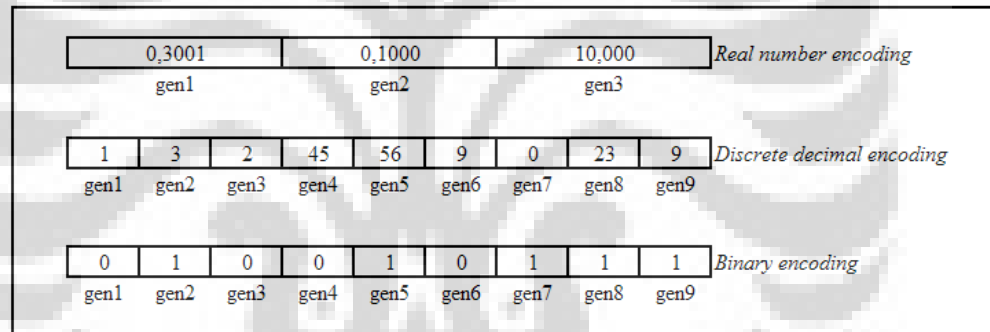
Universitas Indonesia

untuk penyelesaian masalah A belum tentu bagus untuk masalah B, atau bahkan tidak bisa digunakan untuk masalah C. Tujuh komponen AG adalah:

1. Skema Pengkodean

Terdapat tiga skema yang paling umum digunakan dalam pengkodean yang dapat dilihat pada Gambar 2.3, yaitu:

- *Real-number encoding*. Pada skema ini, gen yang berada dalam interval $[0,R]$, dimana R adalah bilangan real positif dan biasanya $R=1$.
- *Discrete decimal encoding*. Setiap gen bisa bernilai salah satu bilangan bulat dalam interval $[0,9]$.
- *Binary encoding*. Setiap gen hanya bisa bernilai 0 atau 1.



Gambar 2.3 Tiga jenis skema pengkodean

Pada contoh di atas terdapat tiga variabel, yaitu x_1 , x_2 , x_3 yang dikodekan ke dalam sebuah kromosom yang terdiri dari 3 gen (untuk *real-number encoding*). Sedangkan pada *discrete decimal encoding* maupun *binary encoding*, ketiga variabel dikodekan ke dalam kromosom yang terdiri dari 9 gen (masing-masing variabel dikodekan ke dalam 3 gen).

Berdasarkan contoh di atas, pada *real-number encoding*, sebuah gen g digunakan untuk merepresentasikan suatu bilangan *real* antara 0 sampai 1 (dalam contoh ini $R = 1$). Dengan menggunakan suatu interval tertentu, batas bawah r_b dan batas atas r_a , pengkodean dalam dilakukan dengan cara sebagai berikut:

$$x = r_b + (r_a - r_b)g \quad (2.6)$$

Sedangkan pengkodean untuk *discrete decimal encoding* adalah sebagai berikut:

$$x = r_b + (r_a - r_b)(g_1 \times 10^{-1} + g_2 \times 10^{-2} + \dots + g_N \times 10^{-N}) \quad (2.7)$$

Dan prosedur pengkodean untuk *binary encoding* adalah sebagai berikut:

$$x = r_b + (r_a - r_b)(g_1 \times 2^{-1} + g_2 \times 2^{-2} + \dots + g_N \times 2^{-N}) \quad (2.8)$$

Jika ketiga variabel dibatasi pada interval nilai tertentu, misalkan $[-1,2]$, maka hasil pengkodean dari contoh di atas adalah sebagai berikut:

Pengkodean pada skema real-number encoding

$$x_1 = -1 + (2 - (-1)) \times g_1 = -1 + 3 \times 0,2390 = -0,2830$$

$$x_2 = -1 + (2 - (-1)) \times g_2 = -1 + 3 \times 1,0000 = 2,0000$$

$$x_3 = -1 + (2 - (-1)) \times g_3 = -1 + 3 \times 0,0131 = -0,9607$$

Pengkodean pada skema discrete decimal encoding

$$x_1 = -1 + (2 - (-1))(g_1 \times 10^{-1} + g_2 \times 10^{-2} + g_3 \times 10^{-3}) = -1 + 3(0,2 + 0,03 + 0,009) = -0,2830$$

$$x_2 = -1 + (2 - (-1))(g_4 \times 10^{-1} + g_5 \times 10^{-2} + g_6 \times 10^{-3}) = -1 + 3(0,9 + 0,09 + 0,009) = 1,9970$$

$$x_3 = -1 + (2 - (-1))(g_7 \times 10^{-1} + g_8 \times 10^{-2} + g_9 \times 10^{-3}) = -1 + 3(0 + 0,01 + 0,003) = -0,9610$$

Pengkodean pada skema binary encoding

$$x_1 = -1 + (2 - (-1))(g_1 \times 2^{-1} + g_2 \times 2^{-2} + g_3 \times 2^{-3}) = -1 + 3(0 + 0,25 + 0) = -0,2500$$

$$x_2 = -1 + (2 - (-1))(g_4 \times 2^{-1} + g_5 \times 2^{-2} + g_6 \times 2^{-3}) = -1 + 3(0,5 + 0,25 + 0,125) = 1,6250$$

$$x_3 = -1 + (2 - (-1))(g_7 \times 2^{-1} + g_8 \times 2^{-2} + g_9 \times 2^{-3}) = -1 + 3(0 + 0 + 0) = -1,0000$$

Dengan skema pengkodean *discrete decimal* di atas, nilai x_2 yang dihasilkan adalah 1,9970, tidak maksimum seperti pada *real-number encoding* (yaitu 2,0000). Hal ini disebabkan nilai maksimum dari *discrete decimal encoding* adalah kurang dari 1, yaitu:

$$\sum_{i=1}^3 10^{-i} = \frac{10^3 - 1}{10^3} = \frac{999}{1000} \quad (2.9)$$

Untuk permasalahan di mana solusi yang dicari ternyata berada pada batas atas interval yang diberikan, maka formula pengkodean tersebut di atas perlu diubah menjadi:

$$x = r_b + \frac{(r_a - r_b)}{\sum_{i=1}^N 10^{-i}} (g_1 \times 10^{-1} + g_2 \times 10^{-2} + \dots + g_N \times 10^{-N}) \quad (2.10)$$

di mana N adalah jumlah gen dalam kromosom (panjang kromosom). Dengan rumus pendekatan yang baru ini didapatkan:

$$x_1 = -1 + \frac{3}{999/1000} (0,2 + 0,03 + 0,009) = -0,2822 \quad (2.11)$$

$$x_2 = -1 + \frac{3}{999/1000} (0,9 + 0,09 + 0,009) = 2,0000 \quad (2.12)$$

$$x_3 = -1 + \frac{3}{999/1000} (0 + 0,01 + 0,003) = -0,9609 \quad (2.13)$$

Hal yang sama juga terjadi pada skema pengkodean *binary*, di mana nilai x_2 yang dihasilkan adalah 1,650. Hal ini disebabkan nilai maksimum dari *binary encoding* adalah kurang dari 1 (bahkan jauh lebih kecil daripada 1), yaitu:

$$\sum_{i=1}^3 2^{-i} = \frac{2^3 - 1}{2^3} = 0,875 \quad (2.14)$$

Dengan cara yang sama seperti pada discrete decimal, maka formula pendekodean untuk *binary* adalah:

$$x = r_b + \frac{(r_a - r_b)}{\sum_{i=1}^N 2^{-i}} (g_1 \times 2^{-1} + g_2 \times 2^{-2} + \dots + g_N \times 2^{-N}) \quad (2.15)$$

di mana N adalah jumlah gen dalam kromosom (panjang kromosom). Dengan rumus pendekatan yang baru ini didapatkan:

$$x_1 = -1 + \frac{3}{7/8} (0 + 0,25 + 0) = -0,1428 \quad (2.16)$$

$$x_2 = -1 + \frac{3}{7/8} (0,5 + 0,25 + 0,125) = 2,0000 \quad (2.17)$$

$$x_3 = -1 + \frac{3}{7/8} (0 + 0 + 0) = -1,0000 \quad (2.18)$$

Pada pendekatan *binary* dihasilkan nilai x_1 (-0,1428) yang sangat jauh berbeda dengan nilai x_1 pada skema *real-number encoding* (yakni -0,2830). Hal ini disebabkan terlalu sedikitnya jumlah gen yang digunakan pada *binary encoding* (hanya 3). Untuk permasalahan yang membutuhkan ketelitian tinggi

diperlukan jumlah gen yang lebih banyak pada skema pengkodean *binary*. Begitu juga dengan *discrete decimal encoding*. Tetapi perlu disadari bahwa jumlah gen yang terlalu banyak akan mempengaruhi kecepatan proses dari algoritma genetika secara signifikan. Untuk itu perlu dipertimbangkan jumlah gen yang sesuai. Beberapa aplikasi sederhana menggunakan *binary encoding* dengan 10 gen untuk satu variabel. Dengan 10 gen, maka nilai maksimum yang bisa dikodekan adalah $(2^{10} - 1)/2^{10} \approx 0.9990234375$, yang berarti sudah cukup mendekati 1.

2. Nilai *fitness*

Suatu individu dievaluasi berdasarkan fungsi tertentu sebagai suatu ukuran performansinya. Di dalam evolusi alam, sebuah individu yang bernilai *fitness* tinggi yang akan bertahan hidup. Sedangkan individu yang bernilai *fitness* rendah akan mati. Pada masalah optimasi, jika solusi yang dicari adalah memaksimalkan sebuah fungsi h (dikenal dengan nama maksimasi), maka nilai *fitness* yang digunakan adalah nilai dari fungsi h tersebut, yakni $f=h$ (dimana f adalah nilai *fitness*). Tetapi jika masalahnya adalah meminimalkan fungsi h (masalah minimasi), maka fungsi h tidak bisa digunakan secara langsung. Hal ini disebabkan adanya aturan bahwa individu yang memiliki nilai *fitness* lebih tinggi lebih mampu bertahan hidup pada generasi berikutnya. Oleh karena itu nilai *fitness* yang digunakan adalah $f = 1/h$ yang artinya semakin kecil nilai h , semakin besar nilai f . Tetapi hal ini akan menjadi masalah jika h bisa bernilai 0, yang mengakibatkan f bisa bernilai tak hingga. Untuk mengatasinya, h perlu ditambah sebuah bilangan yang sangat kecil sehingga nilai *fitness*-nya menjadi:

di mana a adalah bilangan yang dianggap sangat kecil dan bervariasi sesuai dengan masalah yang akan diselesaikan. Sebagai contoh, perhatikan masalah berikut ini.

Jika diketahui bahwa $x_1, x_2 \in [-2,5]$, bagaimana nilai *fitness* yang bisa digunakan untuk mencari nilai maksimal dari fungsi pada persamaan 2.19 di bawah ini? Dan bagaimana nilai *fitness* untuk mencari nilai minimalnya?

$$h(x_1, x_2) = x_1^2 + x_2^2 \quad (2.19)$$

Fungsi h di atas mencapai nilai ekstrem maksimum 50, pada saat $x_1 = 5$ dan $x_2 = 5$. Nilai *fitness* untuk masalah maksimasi adalah h itu sendiri. Fungsi h mencapai ekstrem minimum 0, pada saat $x_1 = 0$ dan $x_2 = 0$. Karena h bisa bernilai 0, maka nilai *fitness* yang bisa digunakan untuk masalah ini adalah $f = 1/(h+a)$.

Untuk suatu fungsi h yang memiliki variansi yang kecil, seperti pada persamaan 2.20 di bawah ini, bagaimana nilai *fitness* yang bisa digunakan?

$$h(x_1, x_2) = 1000 + x_1^2 + x_2^2 \quad (2.20)$$

Pada fungsi 2.20 di atas, nilai-nilai h berada dalam kisaran 1000 sehingga semua individu memiliki nilai *fitness* yang hampir sama. Hal ini berakibat buruk pada proses seleksi yang memilih orang tua secara proporsional sesuai dengan nilai *fitness*-nya. Untuk itu diperlukan suatu mekanisme yang disebut *linier fitness ranking*. Mekanisme ini bertujuan untuk melakukan penskalaan nilai-nilai *fitness*. Individu bernilai *fitness* tertinggi diberi nilai *fitness* N (jumlah individu dalam populasi). Individu bernilai *fitness* tertinggi kedua diberi nilai *fitness* $N-1$, dan seterusnya sehingga individu bernilai *fitness* terendah diberi nilai *fitness* 1. Misalkan $R(i)$ menyatakan ranking individu ke i , $R(i) = 1$ jika i adalah individu bernilai *fitness* tertinggi dan $R(i) = N$ jika i adalah individu bernilai *fitness* terendah, maka nilai *fitness* yang baru adalah:

$$f(i) = (N + 1 - R(i)) \quad (2.21)$$

Tetapi penggunaan nilai *fitness* 2.21 di atas bisa berakibat evolusi akan mencapai optimum lokal karena kecilnya perbedaan nilai-nilai *fitness* pada semua individu dalam populasi. Kecenderungan untuk konvergen pada optimum lokal dapat dikurangi dengan menggunakan persamaan 2.22 di bawah ini. Dengan demikian diperoleh nilai *fitness* yang berada dalam interval $[f_{min}, f_{max}]$.

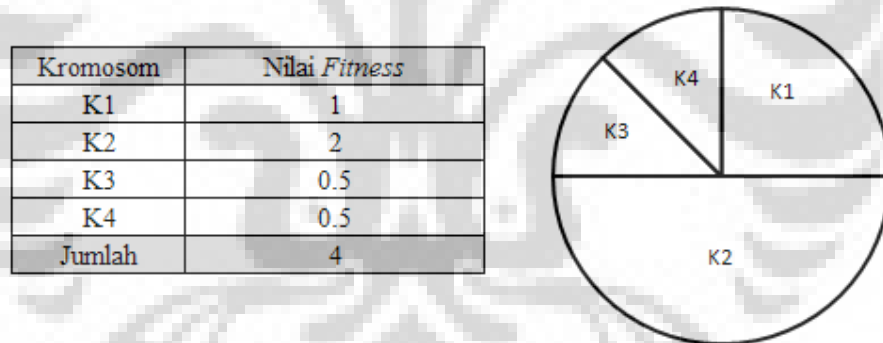
$$f(i) = f_{max} - (f_{max} - f_{min}) \left(\frac{R(i) - 1}{N - 1} \right) \quad (2.22)$$

Penentuan nilai *fitness* sangat berpengaruh pada performansi AG secara keseluruhan. Dalam beberapa kasus, nilai *fitness* yang sangat sederhana

bisa ditemukan dengan mudah. Tetapi dalam beberapa kasus lain diperlukan nilai *fitness* yang sangat kompleks dan sulit ditemukan.

3. Seleksi Orang Tua

Pemilihan dua buah kromosom sebagai orang tua, yang akan dipindahsilangkan, biasanya dilakukan secara proporsional sesuai dengan nilai *fitness*-nya. Suatu metode seleksi yang umum digunakan adalah *roulette-wheel* (roda *roulette*). Sesuai dengan namanya, metode ini menirukan permainan *roulette-wheel* di mana masing-masing kromosom menempati potongan lingkaran pada roda *roulette* secara proporsional sesuai dengan nilai *fitness*-nya. Kromosom yang memiliki nilai *fitness* lebih besar menempati potongan lingkaran yang lebih besar dibandingkan dengan kromosom bernilai *fitness* rendah. Gambar 2.4 di bawah ini mengilustrasikan sebuah contoh penggunaan metode *roulette-wheel*.



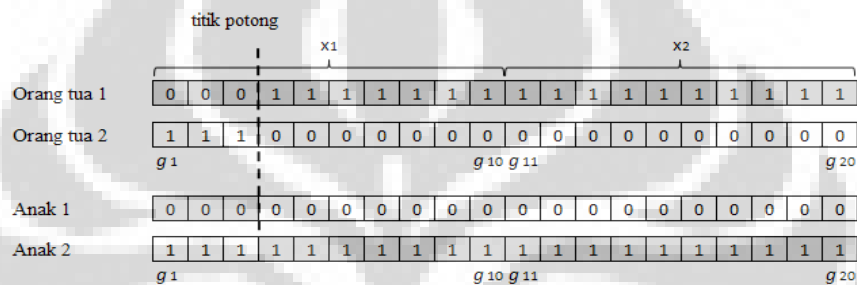
Gambar 2.4 Contoh Penggunaan Metode *Roulette-Wheel Selection*

Metode *roulette-wheel selection* sangat mudah diimplementasikan dalam pemrograman. Pertama, dibuat interval nilai kumulatif (dalam interval $[0,1]$) dari nilai *fitness* masing-masing kromosom dibagi total nilai *fitness* dari semua kromosom. Sebuah kromosom akan terpilih jika bilangan random yang dibangkitkan berada dalam interval akumulatifnya. Pada Gambar 2.4 di atas, K1 menempati nilai kumulatif $[0;0,25]$, K2 berada dalam interval $[0,25;0,75]$, K3 dalam interval $[0,75;0,875]$ dan K4 dalam interval $[0,875;1]$. Misalkan, jika bilangan random yang dibangkitkan adalah 0,6 maka kromosom K2

terpilih sebagai orang tua. Tetapi jika bilangan random yang dibangkitkan adalah 0,99 maka kromosom K4 yang terpilih.

4. Pindah Silang

Salah satu komponen paling penting dalam algoritma genetika adalah *crossover* atau pindah silang. Sebuah kromosom yang mengarah pada solusi yang bagus bisa diperoleh dari proses memindah-silangkan dua buah kromosom. Perhatikan contoh pada Gambar 2.5.



Gambar 2.5 Contoh Proses Pindah Silang

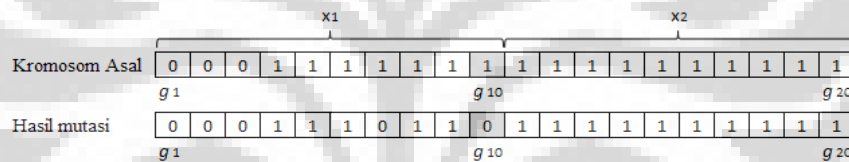
Pindah silang juga berakibat buruk jika ukuran populasi sangat kecil, suatu kromosom dengan gen-gen yang mengarah ke solusi akan sangat dengan cepat menyebar ke kromosom-kromosom lainnya. Untuk mengatasi masalah ini digunakan suatu aturan bahwa pindah silang hanya bisa dilakukan dengan suatu probabilitas tertentu p_c . Artinya, pindah silang bisa dilakukan hanya jika suatu bilangan random $[0,1)$ yang dibangkitkan kurang dari p_c yang ditentukan. Pada umumnya, p_c diset mendekati 1, misalnya 0,8.

Pindah silang bisa dilakukan dalam beberapa cara berbeda. Yang paling sederhana adalah pindah silang satu titik potong (*one-point crossover*). Suatu titik potong dipilih secara random, kemudian bagian pertama dari orang tua 1 digabungkan dengan bagian kedua dari orang tua 2 (lihat Gambar 2.5). Untuk kromosom yang sangat panjang, misalkan 1000 gen, mungkin saja diperlukan beberapa titik potong. Pindah silang lebih dari satu titik potong disebut *n-point crossover*, dimana n titik potong dipilih secara random dan bagian-bagian kromosom dipilih dengan probabilitas 0,5 dari salah satu orang tuanya. Satu skema pindah silang yang lain adalah *uniform crossover*, yang

merupakan kasus khusus dari *n-point crossover* di mana n sama dengan jumlah gen dikurangi satu.

5. Mutasi

Prosedur mutasi sangatlah sederhana. Untuk semua gen yang ada, jika bilangan random yang dibangkitkan kurang dari probabilitas mutasi p_{mut} yang ditentukan maka ubah gen tersebut menjadi nilai kebalikannya (dalam *binary encoding*, 0 diubah 1, dan 1 diubah 0). Biasanya p_{mut} diset sebagai $1/n$, di mana n adalah jumlah gen dalam kromosom. Dengan p_{mut} sebesar ini berarti mutasi hanya terjadi pada sekitar satu gen saja. Pada AG sederhana, nilai p_{mut} adalah tetap selama evolusi. Gambar 2.6 menggambarkan proses mutasi yang terjadi pada gen g_{10} .



Gambar 2.6 Contoh Proses Mutasi

6. Elitisme

Karena seleksi dilakukan secara random, maka tidak ada jaminan bahwa suatu individu bernilai *fitness* tertinggi akan selalu terpilih. Kalaupun individu bernilai *fitness* tertinggi terpilih, mungkin saja individu tersebut akan rusak (nilai *fitness*-nya menurun) karena proses pindah silang. Untuk menjaga agar individu bernilai *fitness* tertinggi tersebut tidak hilang selama evolusi, maka perlu dibuat satu atau beberapa kopinya. Prosedur ini disebut sebagai elitisme.

7. Pergantian Populasi

Dalam algoritma genetika dikenal skema penggantian populasi yang disebut *general replacement*, yang berarti semua individu (misal N individu dalam satu populasi). Skema penggantian ini tidak realistis dari sudut pandang biologi. Di dunia nyata, individu-individu dari generasi berbeda bisa berada

dalam waktu yang bersamaan. Fakta lainnya adalah individu-individu muncul dan hilang secara konstan, tidak pada generasi tertentu. Secara umum skema penggantian populasi dapat dirumuskan berdasarkan suatu ukuran yang disebut *generational gap* G . Ukuran ini menunjukkan persentase populasi yang digantikan dalam setiap generasi. Pada skema *generational replacement*, $G = 1$.

Skema penggantian yang paling ekstrem adalah hanya mengganti satu individu dalam setiap generasi, yaitu $G = 1/N$, di mana N adalah jumlah individu dalam populasi. Skema penggantian ini disebut sebagai *steady-state reproduction*. Pada skema tersebut, G biasanya sama dengan $1/N$ atau $2/N$. Dalam setiap generasi sejumlah NG individu harus dihapus untuk menjaga ukuran populasi tetap N . Terdapat beberapa prosedur penghapusan individu, yaitu penghapusan individu yang bernilai *fitness* paling rendah atau penghapusan individu yang paling tua. Penghapusan bisa berlaku hanya pada individu orang tua saja atau bisa juga berlaku pada semua individu dalam populasi.

BAB 3 PENGUMPULAN DATA

3.1 Pengumpulan Data

Data yang digunakan dalam penelitian ini merupakan data sekunder yang diperoleh dari data-data perkuliahan yang diselenggarakan di Fakultas Teknik UI untuk jenjang pendidikan S1 Reguler pada tahun ajaran 2009/2010 yang diperoleh dari Pusat Administrasi Fakultas (PAF) Teknik UI. Data yang digunakan untuk penelitian penjadwalan kuliah pada perkuliahan di Fakultas Teknik UI adalah:

1. Data mata kuliah, yaitu berupa nama, jumlah sks, jenis dan kapasitas tiap mata kuliah yang diajarkan oleh seluruh departemen yang terdapat di Fakultas Teknik UI.
2. Data ruangan kuliah yang tersedia untuk menyelenggarakan perkuliahan di Fakultas Teknik UI, berupa nama ruangan dan daya tampung ruangan.
3. Data mahasiswa, yaitu berupa jumlah mahasiswa tiap angkatan dalam suatu jurusan.
4. Data dosen, yaitu berupa nama dosen dan mata kuliah yang diajarkan.

3.1.1 Data Mata Kuliah

Data mata kuliah yang digunakan dalam penelitian ini berdasarkan pada buku kurikulum Fakultas Teknik UI yang berlaku untuk tahun ajaran 2009/2010. Semua mata kuliah yang diajarkan dikelompokkan terlebih dahulu menurut departemen, semester ajar, jenis mata kuliah (wajib/pilihan), dan jumlah sks mata kuliah yang bersangkutan. Struktur mata ajaran program studi departemen Teknik Industri UI dapat dilihat pada Tabel 3.1. Sedangkan struktur mata ajaran pilihan program studi Teknik Industri dapat dilihat pada Tabel 3.2.

Tabel 3.1 Struktur Mata Ajaran Program Studi Teknik Industri

KODE	MATA AJARAN	SUBJECT	SKS
Semester 1		1st Semester	
UUI11001	MPK Terintegrasi	Integrated Characteristic Building Subject	6
UUI11010	Bahasa Inggris	English	3
UUI12030	Olah Raga / Seni	Sport / Art	1
ENG100801	Kalkulus	Calculus	4
ENG200801	Statistik & Probabilitas	Statistics and Probability	3
IES110801	Pengantar Teknik Industri	Introduction to Industrial Engineering	2
Sub Total			19
Semester 2		2nd Semester	
UUI12020-24	Agama	Religion	2
ENG100804	Aljabar Linear	Linear Algebra	4
ENG100805	Fisika Dasar 1	Basic Physics 1	4
ENG100807	Dasar Komputer	Basic Computer	3
IES120801	Pengetahuan Bahan	Material Science	2
ENG100808	Menggambar Teknik	Engineering Drawing	2
IES120802	Pengantar Ilmu Ekonomi	Introduction to Economics	2
Sub Total			19
Semester 3		3rd Semester	
ENG200802	Fisika Dasar 2	Basic Physics 2	4
ENG100802	Kimia Dasar	Basic Chemistry	2
IES210801	Akuntansi dan Biaya	Cost Accounting	2
IES210802	Facilitas Industri	Industrial Facilities	2
IES210803	Psikologi Industri	Industrial Psychology	2
ENG300801	Ekonomi Teknik	Engineering Economics	2
IES210804	Programa Linier	Linear Programming	3
IES210805	Mekanika Teknik	Engineering Mechanics	2
Sub Total			19
Semester 4		4th Semester	
IES220801	Fasilitas Pabrik	Plant Facilities	2
IES220802	Proses Produksi - Praktikum	Production Process - Lab	2
IES220803	Statistik Industri	Industrial Statistics	3
IES220804	Penelitian Operasional	Operation Research	3
IES220805	Sistem Penjaminan dan Pengendalian Kualitas	Quality Control and Assurance Systems	3
IES220806	Perancangan Organisasi	Organizational Design	2
IES220807	Perancangan Kerja, Metode dan Standar Kerja	Work Design, Methods and Standards	2
IES220808	Elemen Mesin	Machine Elements	2
Sub Total			20
Semester 5		5th Semester	
IES310801	Faktor Manusia Dalam Rekyasa Disain	Human Factors in Engineering Design	3
IES310802	Perencanaan & Pengendalian Produksi	Production Planning and Control	3
IES310803	Pemodelan Sistem - Praktikum	Systems Modeling- Lab	3
IES310804	Pemasaran Industri	Industrial Marketing	3
IES310805	Sistem Informasi	Information System	3
	Pilihan 1	Elective 1	3
	Pilihan 2	Elective 2	3
Sub Total			21
Semester 6		6th Semester	
IES320801	Perancangan Produk	Product Design	2
IES320802	Tata Letak Pabrik	Plant Lay Out	3
ENG300803	Etika Enjiniring	Engineering Ethics	2
ENG300802	Kesehatan, Keselamatan Kerja & Lindung Lingkungan	Ooocupational, Health, Safety & Environment	2
IES320803	Sistem Produksi	Production System	3
IES320804	Simulasi Industri	Industrial Simulation	3
	Pilihan 3	Elective 3	3
	Pilihan 4	Elective 4	3
Sub Total			21
Semester 7		7th Semester	
IES410801	Perancangan Proses	Process Design	2
IES410802	Manajemen Proyek	Project Management	3
IES410803	Kapita Selekta Industri	Special Topics in Industrial Engineering	2
IES410804	Sistem Pemeliharaan	Maintenance System	3
IES410805	Manajemen Teknologi	Management of Tehnology	3
IES400801	Kerja Praktek	Internship	2
	Pilihan 5	Elective 5	3
	Pilihan 6	Elective 6	3
Sub Total			21
Semester 8		8th Semester	
IES400802	Skripsi dan Seminar	Final Project in Ind.Eng.	5
Sub Total			5

(Sumber: Panduan Akademik Program Pendidikan Sarjana Teknik 2008-2011 & Profesi Arsitek 2009-2011, Edisi 2009)

Tabel 3.2 Struktur Mata Ajaran Pilihan Program Studi Teknik Industri

MATA AJARAN PILIHAN			
KODE	MATA AJARAN	SUBJECT	SKS
IEF300801	Total Quality Management (TQM)	Total Quality Management	3
IEF300802	Teknik Proses dalam industri	Process Technique in Industry	3
IEF300803	Keputusan, Ketidakpastian dan Risiko	Decisions, Uncertainties and Risk	3
IEF300804	Teori Keputusan	Decision Theory	3
IEF300805	Manajemen Strategi Industri	Industrial Strategio Management	3
IEF300806	Sistem Distribusi dan Logistik	Logistio and Distribution System	3
IEF300807	Manajemen Inovasi	Inovatio Management	3
IEF300808	Prinsip-prinsip Rekayasa Sistem Industri	System Engineering Principles and Practice	3
IEF300809	Analisa Multivariat	Multivariate Analysis	3
IEF300810	Keterampilan Interpersonal	Interpersonal Skills	3
IEF300811	Perancangan Permainan Simulasi	Simulation gaming Design	3
IEF300812	Manajemen Pengetahuan	Knowledge Management	3
IEF300813	Berfikir Sistem	System Thinking	3
IEF300814	Analisa Kelayakan Industri	Industrial Feasibility Study	3
IEF300815	Manajemen Investasi	Investment Management	3
IEF300816	Sistem Keuangan Perusahaan	Corporate Finance	3
IEF300817	Kebijakan Industri	Industrial Policy	3
IEF300818	Manajemen Energi	Energy Management	3
IEF300819	Ekonomi Industri	Industrial Economics	3

(Sumber: Panduan Akademik Program Pendidikan Sarjana Teknik 2008-2011 & Profesi Arsitek 2009-2011, Edisi 2009)

3.1.2 Data Ruangan Kuliah

Data ruangan kuliah pada Tabel 3.3 merupakan nama-nama ruangan yang tersedia untuk menyelenggarakan kegiatan perkuliahan di Fakultas Teknik UI. Data ruangan yang ada disusun menurut lokasi ruangan tersebut berada. Dari seluruh ruangan yang tersedia di Fakultas Teknik UI, kapasitas ruangan terbesar adalah 140 dan kapasitas ruangan terkecil adalah 15.

Tabel 3.3 Data Ruangan Kuliah di Fakultas Teknik UI

No	Nama ruangan	Kapasitas	Lokasi
1	A.101	40	Gedung Pasca Sarjana
2	A.102	40	Gedung Pasca Sarjana
3	A.103	40	Gedung Pasca Sarjana
4	A.104	40	Gedung Pasca Sarjana
5	A.201	30	Gedung Pasca Sarjana
6	A.202	30	Gedung Pasca Sarjana
7	A.203	30	Gedung Pasca Sarjana
8	A.204	20	Gedung Pasca Sarjana
9	A.205	20	Gedung Pasca Sarjana
10	A.601	40	Gedung Pasca Sarjana
11	A.602	40	Gedung Pasca Sarjana
12	A.603	40	Gedung Pasca Sarjana
13	A.604	40	Gedung Pasca Sarjana
14	A.605	40	Gedung Pasca Sarjana
15	A.606	40	Gedung Pasca Sarjana
16	MM PASCA	20	Gedung Pasca Sarjana

(Sumber: Pusat Administrasi Fakultas Teknik UI, telah diolah kembali)

Tabel 3.3 Data Ruangan Kuliah di Fakultas Teknik UI (sambungan)

No	Nama ruangan	Kapasitas	Lokasi
17	GK.201	40	Gedung PAF
18	GK.301	100	Gedung PAF
19	GK.302	20	Gedung PAF
20	GK.303	20	Gedung PAF
21	GK.304	20	Gedung PAF
22	GK.305	20	Gedung PAF
23	GK.306	100	Gedung PAF
24	LAB PUSGIWA	70	Gedung PAF
25	K.101	40	Gedung Kuliah Bersama
26	K.102	70	Gedung Kuliah Bersama
27	K.103	70	Gedung Kuliah Bersama
28	K.104	40	Gedung Kuliah Bersama
29	K.105	40	Gedung Kuliah Bersama
30	K.106	70	Gedung Kuliah Bersama
31	K.107	70	Gedung Kuliah Bersama
32	K.108	40	Gedung Kuliah Bersama
33	K.201	30	Gedung Kuliah Bersama
34	K.202	80	Gedung Kuliah Bersama
35	K.203	30	Gedung Kuliah Bersama
36	K.204	70	Gedung Kuliah Bersama
37	K.205	30	Gedung Kuliah Bersama
38	K.206	80	Gedung Kuliah Bersama
39	K.207	80	Gedung Kuliah Bersama
40	K.208	30	Gedung Kuliah Bersama
41	K.209	50	Gedung Kuliah Bersama
42	K.210	70	Gedung Kuliah Bersama
43	K.211	50	Gedung Kuliah Bersama
44	K.301	120	Gedung Kuliah Bersama
45	EC 101	20	Gedung EC
46	EC 102	20	Gedung EC
47	EC 103	20	Gedung EC
48	EC 104	20	Gedung EC
49	EC 105	20	Gedung EC
50	EC 106	20	Gedung EC
51	EC 107	20	Gedung EC
52	EC 108	20	Gedung EC
53	EC 109	20	Gedung EC
54	EC 110	20	Gedung EC
55	EC 202	20	Gedung EC
56	EC 203	120	Gedung EC
57	EC 302	20	Gedung EC
58	EC 303	120	Gedung EC
59	EC STUDIO	20	Gedung EC (depan ATM)
60	RS CHEVRON		Gedung Dekanat

(Sumber: Pusat Administrasi Fakultas Teknik UI, telah diolah kembali)

Tabel 3.3 Data Ruang Kuliah di Fakultas Teknik UI (sambungan)

No	Nama ruangan	Kapasitas	Lokasi
61	GB 101	70	Gedung Baru
62	GB 102	40	Gedung Baru
63	GB 201	70	Gedung Baru
64	GB 202	70	Gedung Baru
65	GB 203	70	Gedung Baru
66	GB 204	40	Gedung Baru
67	GB 205	40	Gedung Baru
68	GB 301	70	Gedung Baru
69	GB 302	40	Gedung Baru
70	GB 303	40	Gedung Baru
71	GB 304	140	Gedung Baru
72	GB 401	70	Gedung Baru
73	GB 402	40	Gedung Baru
74	GB 403	40	Gedung Baru
75	GB 404	140	Gedung Baru
76	GB 501	70	Gedung Baru
77	GB 502	140	Gedung Baru
78	GB 503	140	Gedung Baru
79	S 201	40	Gedung Baru
80	S 202	40	Gedung Baru
81	S 203	40	Gedung Baru
82	S 204	40	Gedung Baru
83	S 301	70	Gedung Baru
84	S 303	40	Gedung Baru
85	RDM_S	10	
86	R SEM_S	15	
87	BP3 101		
88	BP3 102		
89	C. 102		
90	C. 102		
91	C. 301		
92	C. 302		
93	C. 303		
94	C. 304		
95	C. 401		
96	C. 402		
97	STUDIO DA		
98	MULTIMEDIA DA		
99	R. SEMINAT DA		
100	DA 101		
101	DA 501		

(Sumber: Pusat Administrasi Fakultas Teknik UI, telah diolah kembali)

3.1.3 Data Mahasiswa

Data mahasiswa yang digunakan dalam penelitian ini merupakan data jumlah mahasiswa tiap angkatan, mulai dari angkatan 2006-2009, untuk tiap departemen. Pada Tabel 3.4 menunjukkan jumlah mahasiswa tiap angkatan pada departemen Teknik Industri UI. Data ini akan menjadi dasar bagi penentuan banyaknya jumlah peserta tiap mata kuliah pada suatu jurusan. Kapasitas suatu mata kuliah wajib akan dianggap sama dengan jumlah mahasiswa yang terdapat dalam suatu kelompok mahasiswa dimana mata kuliah tersebut diperuntukkan. Selain itu, kapasitas maksimal mata kuliah wajib diperoleh dengan mempertimbangkan kemungkinan adanya mahasiswa yang akan mengulang mata kuliah tersebut. Jumlah mahasiswa yang mengulang mata kuliah ditentukan berkisar antara 10-20 mahasiswa untuk tiap mata kuliah wajib.

Tabel 3.4 Data Mahasiswa Departemen Teknik Industri UI

NO	Angkatan	Jumlah Mahasiswa
1	2006	85
2	2007	84
3	2008	104
4	2009	80

(Sumber: Sekretariat Departemen Teknik Industri UI, telah diolah kembali)

3.1.4 Data Dosen

Data dosen yang digunakan dalam penelitian ini yaitu berupa nama-nama dosen yang mengajar di Departemen Teknik Industri UI beserta mata kuliah yang diajarkannya pada perkuliahan yang diadakan di semester ganjil. Data dosen ini dapat dilihat pada Tabel 3.5.

Data dosen ini akan digunakan untuk melihat hubungan antar mata kuliah yang satu dengan mata kuliah yang lain. Mata kuliah yang diajarkan oleh dosen yang sama tidak boleh dijadwalkan pada saat yang bersamaan untuk menghindari terjadinya bentrok. Jika bentrok, maka mata kuliah tersebut harus dijadwalkan pada waktu yang lain.

Tabel 3.5 Data Dosen Departemen Teknik Industri UI Beserta Mata Kuliah yang Diajarkan pada Semester Ganjil

No.	Dosen	Mata Kuliah
1	Ir. Amar Rachman,MEIM	Programa Linier
		Sistem Distribusi dan Logistik
2	Ir. Akhmad Hidayatno,MBT	Manajemen Teknologi
		Berpikir Sistem
		Keterampilan Interpersonal
		Manajemen Pengetahuan
3	Armand Omar Moeis, ST, MSc	Pemodelan Sistim + Praktikum
		Manajemen Energi
		Berpikir Sistem
4	Arian Dhini,ST.,MT.	Statistik dan Probabilitas
		Akuntansi Biaya
5	Ir. Boy Nurtjahyo M.,MSIE	Faktor Manusia Dalam rekayasa Industri Sistim Pemeliharaan
6	Ir. Djoko Sihono Gabriel,MT	Pengantar Teknik Industri
		Total quality Management
		Analisa Kelayakan Industri
7	Ir. Hj. Erlinda Muslim,MEE	Pemasaran Industri
		Manajemen Energi
		Manajemen Strategi Industri
8	Farizal,PhD	Statistik dan Probabilitas
		Programa Linier
9	Ir. Fauzia Dianawati,Msi	Perancangan Proses
		Sistem Pemeliharaan
10	Ir. Isti Surjandari,MT.,MA.,PhD	Kapita Selekt Industri
		Analisa Multivariat
11	Dr.Ir. Sri Bintang Pamungkas	Fasilitas Industri
		Sistem Keuangan Perusahaan
12	Komarudin,ST.,M.Eng	Pemodelan Sestim + Praktikum
		Keputusan, Ketidakpastian dan Risiko
13	Ir.M. Dachyar,MSc	Ekonomi Teknik
		Sistem Informasi
		Manajemen Proyek
		Teori Keputusan
14	Ir. Rahmat Nurcahyo,M.Eng.Sc	Psikologi Industri
		Total Quality Management
		Ekonomi Industri
15	Dr. Ir. Teuku Yuri MZ.,M.Eng.Sc.	Pengantar Teknik Industri
		Total Quality Management
16	Ir. Yadrifil,MSc	Perencanaan & Pengendalian Industri
17	Dosen MIPA1	Fisika Dasar 2 (Gelombang)
18	Dosen MIPA2	Fisika Dasar 2 (Optik)
19	Ir. Dendy P.,MSc	Fisika Dasar 2
		Sistem Informasi
20	Dr.Ir. Dewi Tristantini,MT.	Kimia Dasar
21	Ir. Yulianto S.N.,MSc.,PhD	Mekanika Teknik
	Ir. Gatot Prayogo,M.Eng	Mekanika Teknik
22	Dosen MIPA3	Kalkulus A
		Kalkulus B

(Sumber: Sekretariat Departemen Teknik Industri UI, telah diolah kembali)

3.2 Pengalokasian Kelas Tiap Departemen

Setelah pengumpulan data selesai dilakukan, maka data-data tersebut kemudian dikelompokkan untuk membuat hubungan antar data yang diperlukan dalam penelitian ini. Data yang pertama kali dikelompokkan adalah data mata kuliah dan data ruangan kelas untuk melihat jumlah kelas yang dibutuhkan oleh tiap departemen dalam menyelenggarakan perkuliahan pada semester ganjil dan genap.

Berikut ini adalah asumsi-asumsi yang digunakan dalam pengalokasian kelas ke tiap departemen:

- Semua mata kuliah wajib akan diparalelkan menjadi 2 kelas, kecuali mata kuliah wajib yang termasuk ke dalam kuliah umum tidak akan diparalelkan
- Mata kuliah berbobot empat sks atau lebih ditransformasikan ke dalam dua kali pertemuan dalam seminggu
- Ruangan dikelompokkan menjadi tiga kelompok berdasarkan kapasitas, yaitu kelompok ruangan berkapasitas 40-50, 70-80, 100-140.
- Pengalokasian mata kuliah ke ruangan didasarkan pada jenis mata kuliahnya, yaitu:
 - Mata kuliah wajib yang tidak diparalelkan akan dialokasikan ke kelompok ruangan berkapasitas 100-140
 - Mata kuliah wajib yang diparalelkan akan dialokasikan ke kelompok ruangan berkapasitas 70-80
 - Mata kuliah pilihan dan peminatan akan dialokasikan ke kelompok ruangan berkapasitas 40-50
- Mata kuliah PDPT di alokasikan langsung terhadap ruangan dan waktu, karena dianggap merupakan jadwal yang tetap setiap tahunnya.

Berdasarkan asumsi-asumsi di atas, dibuatlah pengalokasian kelas untuk tiap departemen ke dalam 19 slot-waktu yang terdapat dalam satu minggu yang dapat dilihat pada Tabel 3.6:

Tabel 3.6 Slot-waktu yang Disediakan

Senin	08.00-09.50	Kamis	08.00-09.50
	10.00-12.30		10.00-12.30
	13.30-16.00		13.30-16.00
	16.00-17.50		16.00-17.50
Selasa	08.00-09.50	Jumat	07.30-09.20
	10.00-12.30		09.30-11.20
	13.30-16.00		13.30-16.00
	16.00-17.50		
Rabu	08.00-09.50		
	10.00-12.30		
	13.30-16.00		
	16.00-17.50		

Slot-waktu pada Tabel 3.6 dapat disesuaikan dengan kebutuhan masing-masing perkuliahan pada semua perguruan tinggi. Selain itu, slot-waktu tersebut juga dapat dibuat berbeda untuk masing-masing hari serta masing-masing ruang. Berdasarkan slot-waktu yang disediakan, perhitungan total kelas yang dapat digunakan untuk perkuliahan dapat dilihat pada Tabel 3.7.

Tabel 3.7 Total Kelas yang Tersedia untuk Perkuliahan Di Fakultas Teknik UI

Alokasi Penggunaan	Kapasitas Kelas	Jumlah Kelas	Penggunaan kelas dalam	Total penggunaan	TOTAL
Mata Kuliah Pilihan dan PDPT	10	1	19	0	551
	15	1	19	0	
	20	20	19	0	
	30	7	19	0	
	40	27	19	513	
Mata Kuliah Wajib Gabungan	50	2	19	38	323
	70	14	19	266	
	80	3	19	57	
Mata Kuliah Wajib Per Jurusan	100	2	19	38	171
	120	2	19	38	
	130	1	19	19	
	140	4	19	76	
TOTAL		84	228	1045	1240

Data estimasi jumlah kelas yang dibutuhkan untuk PDPT didapatkan dari Pusat Administrasi Fakultas (PAF). Total kelas yang dibutuhkan untuk PDPT akan dikeluarkan dari perhitungan karena dianggap sudah merupakan jadwal yang tetap. Estimasi jumlah kelas yang dibutuhkan untuk PDPT dapat dilihat pada Tabel 3.8.

Tabel 3.8 Estimasi Jumlah Kelas yang Dibutuhkan untuk PDPT

Estimasi Jumlah Kelas (dari data PAF)	
Budha=	1
Hindu=	1
Islam=	16
Khatolik=	2
Protestan=	3
PDPT Inggris =	48
PDPT =	48

(Sumber: Pusat Administrasi Fakultas Teknik UI, telah diolah kembali)

Perhitungan alokasi kelas yang dibutuhkan oleh tiap departemen dilakukan dengan mengurangi jumlah kelas yang dibutuhkan untuk PDPT pada tiap departemen. Pengalokasian kelas tiap departemen dapat dilihat pada Tabel 3.9.

Tabel 3.9 Perhitungan Kebutuhan Kelas Tiap Departemen

1. TEKNIK SIPIL/LINGKUNGAN													
Semester	Sipil (Sp)			Lingkungan (Lgk)			Mata Kuliah Gabungan Sp-Lgk			Total Kelas yang Dibutuhkan NON-PDPT			TOTAL KEBUTUHAN KELAS
							dengan PDPT	Non - PDPT < 4sks	Non - PDPT ≥ 4sks	40-50	70-80	100-140	
Ganjil	Wajib	PDPT	2	Wajib	PDPT	2	13	7	4	38	30	10	78
		Non PDPTs3sks	11		Non PDPTs3sks	11							
	Non PDPT24sks	4	Non PDPT24sks	5									
Genap	Wajib	PDPT	2	Wajib	PDPT	2	11	6	3	25	24	22	71
		Non PDPTs3sks	12		Non PDPTs3sks	14							
	Non PDPT24sks	5	Non PDPT24sks	5									
	Pilihan		28	Pilihan		10							

2. TEKNIK MESIN													
Semester	Mesin (Ms)			Perkapalan (Kpl)			Mata Kuliah Gabungan Ms-Kpl			Total Kelas yang Dibutuhkan NON-PDPT			TOTAL KEBUTUHAN KELAS
							dengan PDPT	Non - PDPT < 4sks	Non - PDPT ≥ 4sks	40-50	70-80	100-140	
Ganjil	Wajib	PDPT	3	Wajib	PDPT	3	13	7	3	23	26	16	65
		Non PDPTs3sks	8		Non PDPTs3sks	14							
	Non PDPT24sks	7	Non PDPT24sks	3									
Genap	Wajib	PDPT	1	Wajib	PDPT	1	9	3	5	24	26	27	77
		Non PDPTs3sks	6		Non PDPTs3sks	7							
	Non PDPT24sks	10	Non PDPT24sks	10									
	Pilihan		16	Pilihan		7							

3. TEKNIK ELEKTRO/KOMPUTER													
Semester	Elektro (Elktr)			Komputer (Kmptr)			Mata Kuliah Gabungan Elktr-			Total Kelas yang Dibutuhkan NON-PDPT			TOTAL KEBUTUHAN KELAS
							dengan PDPT	Non - PDPT < 4sks	Non - PDPT ≥ 4sks	40-50	70-80	100-140	
Ganjil	Wajib	PDPT	3	Wajib	PDPT	3	9	2	4	20	20	27	67
		Non PDPTs3sks	9		Non PDPTs3sks	12							
	Non PDPT24sks	7	Non PDPT24sks	6									
Genap	Wajib	PDPT	1	Wajib	PDPT	1	12	6	5	17	32	18	67
		Non PDPTs3sks	10		Non PDPTs3sks	16							
	Non PDPT24sks	6	Non PDPT24sks	6									
	Pilihan		19	Pilihan		1							

Tabel 3.9 Perhitungan Kebutuhan Kelas Tiap Departemen (sambungan)

4. TEKNIK METALURGI											
Semester	Metalurgi			Total Kelas yang Dibutuhkan NON-PDPT			TOTAL KEBUTUHAN				
				40-50	70-80	100-140					
Ganjil	Wajib	PDPT	3	6	0	27	33				
		Non PDPT≤3sks	21								
		Non PDPT≥4sks	3								
	Pilihan		6								
Genap	Wajib	PDPT	1	6	0	25	31				
		Non PDPT≤3sks	19								
		Non PDPT≥4sks	3								
	Pilihan		6								

5. TEKNIK ARSITEKTUR													
Semester	Arsitektur (Ars)			Desain Interior (D.Int)			Mata Kuliah Gabungan Ars-D.Int.			Total Kelas yang Dibutuhkan NON-PDPT			TOTAL KEBUTUHAN KELAS
							dengan PDPT	Non - PDPT < 4sks	Non - PDPT ≥ 4sks	40-50	70-80	100-140	
Ganjil	Wajib	PDPT	3	Wajib	PDPT	3	7	3	1	49	10	4	63
		Non PDPT≤3sks	3		Non PDPT≤3sks	3							
		Non PDPT≥4sks	1		Non PDPT≥4sks	3							
	Pilihan		28	Pilihan		21							
Genap	Wajib	PDPT	1	Wajib	PDPT	1	7	3	3	49	18	0	67
		Non PDPT≤3sks	3		Non PDPT≤3sks	3							
		Non PDPT≥4sks	3		Non PDPT≥4sks	3							
	Pilihan		28	Pilihan		21							

6. TEKNIK KIMIA													
Semester	Kimia (Kim)			Bioproses (Bio)			Mata Kuliah Gabungan Kim-Bio			Total Kelas yang Dibutuhkan NON-PDPT			TOTAL KEBUTUHAN KELAS
							dengan PDPT	Non - PDPT < 4sks	Non - PDPT ≥ 4sks	40-50	70-80	100-140	
Ganjil	Wajib	PDPT	3	Wajib	PDPT	3	15	10	2	16	28	27	71
		Non PDPT≤3sks	16		Non PDPT	17							
		Non PDPT≥4sks	6		Non PDPT≥4sks	5							
	Pilihan		11	Pilihan		5							
Genap	Wajib	PDPT	1	Wajib	PDPT	1	9	6	2	16	20	26	62
		Non PDPT≤3sks	19		Non PDPT	19							
		Non PDPT≥4sks	2		Non PDPT≥4sks	2							
	Pilihan		11	Pilihan		5							

Semester	TI			Total Kelas yang Dibutuhkan NON-MPK			TOTAL KEBUTUHAN
				10-50	70-80	100-140	
Ganjil	Wajib	MPK	3	11	44	1	56
		Non MPK≤3sks	36				
		Non MPK (paralel)≥4sks	2				
		Non-MPK Paralel≤3sks	1				
	Pilihan		11				
Genap	Wajib	MPK	1	9	44	0	53
		Non MPK≤3sks	8				
		Non MPK (paralel)≥4sks	2				
		Non-MPK Paralel≤3sks	14				
	Pilihan		9				

Perhitungan alokasi kebutuhan kelas tiap departemen pada Tabel 3.9 tersebut dapat diringkas menjadi seperti yang ditunjukkan pada tabel 3.10:

Tabel 3.10 Ringkasan Kebutuhan Kelas Tiap Departemen

Ringkasan Kebutuhan Kelas									
No	Jurusan	Semester							
		Ganjil				Genap			
		40-50	70-80	100-140	Total	40-50	70-80	100-140	Total
1	Sipil/Lingkungan	38	30	10	78	25	24	22	71
2	Mesin	23	26	16	65	24	26	27	77
3	Elektro	20	20	27	67	17	32	18	67
4	Metalurgi	6	0	27	33	6	0	25	31
5	Arsitektur/Desain Interior	49	10	4	63	49	18	0	67
6	Kimia	16	28	27	71	16	20	26	62
7	Industri	11	44	1	56	9	44	0	53
Total Kelas yang Dibutuhkan		163	158	112		146	164	118	
Total Kelas yang Tersedia		551	323	171		551	323	171	

Dari perhitungan alokasi jumlah kelas yang dibutuhkan untuk tiap departemen, dapat dihitung total kelas sebenarnya yang dibutuhkan oleh departemen Teknik Industri. Perhitungan total kelas yang dibutuhkan departemen Teknik Industri dapat dilihat pada Tabel 3.11:

Tabel 3.11 Jumlah Ruangan Kelas yang Dialokasikan untuk Departemen Teknik Industri

Jumlah ruangan yang tersedia	Jumlah kelas yang tersedia dalam		Kebutuhan kelas di TI		Ruangan yang dibutuhkan TI		Ruangan buat TI berdasarkan banyak Dept.		Ruangan yang dialokasikan buat TI	
	1jam50min	2jam30min	1jam50min	2jam30min						
	R*12	R*7		in						
40-50	29	348	203	0	11	1.571429	2	4.142857	4	2
70-80	17	204	119	17	27	3.130952	3	2.428571	3	2
100-140	9	108	63	1	0	0.22619	1	1.285714	1	2

Berdasarkan Tabel 3.11, maka dapat ditentukan kelas yang akan dialokasikan untuk perkuliahan di Departemen Teknik Industri. Pengalokasian kelas yang digunakan didasarkan pada pertimbangan frekuensi penggunaan suatu kelas oleh departemen Teknik Industri. Ruangan-ruangan yang dialokasikan untuk perkuliahan di Departemen Teknik Industri adalah ruangan-ruangan yang biasa digunakan dalam perkuliahan sehari-hari. Daftar ruangan serta kapasitas dari ruangan kelas yang dialokasikan untuk Departemen Teknik Industri dapat dilihat pada Tabel 3.12:

Tabel 3.12 Data Ruangan Kelas yang Dialokasikan untuk Departemen Teknik Industri

DATA KAPASITAS RUANG KULIAH

NO	RUANG	KAPASITAS
1	K.108	40
2	K.209	50
3	K.211	50
4	K.206	80
5	K.207	80
6	GK.306	100
7	GK.301	100

3.3 Matrik Hubungan Antara Dosen, Mahasiswa, dan Mata Kuliah

Setiap mata kuliah yang diselenggarakan pada suatu departemen diperuntukkan bagi kelompok mahasiswa tertentu. Pada Lampiran diperlihatkan alokasi jumlah mahasiswa yang mengambil suatu mata kuliah tertentu yang diselenggarakan di Departemen Teknik Industri UI pada semester ganjil menurut kelompok mahasiswa yang mengambil kuliah tersebut. Hal ini membuat mata kuliah yang diselenggarakan pada semester yang beriringan tidak dapat dijadwalkan pada slot-waktu yang bersamaan untuk memberikan kesempatan bagi para mahasiswa yang akan mengulang mata kuliah tersebut. Selain itu, penjadwalan mata kuliah di Departemen Teknik Industri UI juga harus memperhatikan dosen pengajar mata kuliah yang bersangkutan. Hal ini untuk menghindari adanya jadwal yang bentrok diakibatkan oleh dosen. Alokasi dosen terhadap mata kuliah dapat dilihat pada Lampiran.

BAB 4 PENGOLAHAN DATA DAN ANALISIS

4.1 Model Penjadwalan Mata Kuliah di Departemen Teknik Industri UI

Penjadwalan kuliah di Fakultas Teknik UI melibatkan beberapa komponen yakni ruang kuliah, dosen serta mahasiswa. Seorang dosen dapat mengajar lebih dari satu mata kuliah dan satu mata kuliah dapat diajar oleh satu atau dua dosen, tetapi seorang dosen tidak dapat memberikan dua atau lebih kuliah pada waktu yang bersamaan. Tersedia 6 ruang kuliah dengan kapasitas beragam untuk menampung sejumlah mahasiswa yang terbagi atas 4 kelompok yakni mahasiswa tingkat 1, tingkat 2, tingkat 3, dan tingkat 4. Dari keempat kelompok mahasiswa tadi, kelompok mahasiswa tingkat 1 dan tingkat 2 masing-masing dibagi menjadi 2 yaitu kelompok mahasiswa tingkat 1A, 1B, 2A, dan 2B sehingga kelompok mahasiswa yang ada sekarang menjadi 6. Pembagian ini dikarenakan adanya pemisahan kelas bagi beberapa mata kuliah wajib tertentu dengan bobot lebih dari atau sama dengan 4 sks yang menyebabkan mahasiswa pada tingkat 1 dan 2 harus dibagi menjadi 2. Selain itu, pembagian ini juga bertujuan untuk mempermudah pembentukan matrik hubungan antara mata kuliah dengan mahasiswa pada tahap input data.

Dalam membuat model penjadwalan mata kuliah terlebih dahulu harus didefinisikan kendala *hard* dan *soft* yang berpengaruh dalam proses penjadwalan. Kendala *hard* dan *soft* yang akan didefinisikan pada model penjadwalan di Departemen Teknik Industri disesuaikan dengan kondisi yang ada dalam keadaan sebenarnya. Berikut asumsi-asumsi yang digunakan untuk membentuk model penjadwalan di Departemen Teknik Industri UI:

1. Kelompok mahasiswa terbagi atas mahasiswa tingkat 1, tingkat 2, tingkat 3, dan tingkat 4.
2. Perkuliahan hanya diselenggarakan pada slot-waktu tertentu yang sesuai dengan jumlah sks masing-masing mata kuliah.
3. Mata kuliah yang lebih dari 3 sks akan ditransformasikan menjadi 2 kelas.

Variabel-variabel yang terdapat dalam penjadwalan kuliah:

Universitas Indonesia

- $K = \text{Mata Kuliah} = 56$
- $R = \text{Ruangan} = 7 \text{ ruang}$
- $T = \text{Slot waktu} = 19$
- $D = \text{Dosen} = 22 \text{ orang}$
- $M = \text{Kelompok Mahasiswa} = 6 \text{ kelompok}$

Berikut didefinisikan sejumlah komponen yang digunakan dalam formulasi model:

- $J = \text{Himpunan semua jadwal yang mungkin}$
- $j = \text{Anggota himpunan } J$
- $H = \{H1, H2, H3, H4, \text{ dan } H5\}$, yakni himpunan kendala *hard*
- $S = \{S1, S2, \text{ dan } S3\}$, yakni himpunan kendala *soft*
- $M = \text{Himpunan kelompok mahasiswa}$
- $K = \text{Himpunan Kuliah yang akan dijadwalkan}$

Kendala *hard* yang didefinisikan dalam formulasi model yang berfungsi untuk menentukan layak atau tidaknya suatu jadwal sebagai berikut:

- H1: Tidak ada mahasiswa yang menghadiri lebih dari satu mata kuliah pada waktu yang bersamaan
- H2: Kapasitas ruangan kelas cukup untuk menyelenggarakan mata kuliah yang bersangkutan
- H3: Hanya satu mata kuliah yang diselenggarakan pada periode yang sama. (periode = tiap kombinasi ruangan dan slot)
- H4: Tidak ada dosen yang mengajar lebih dari satu mata kuliah pada waktu yang bersamaan.
- H5: Mata Kuliah harus dilangsungkan pada slot waktu yang telah ditentukan (karena terdapat perbedaan lama jam pelajaran untuk beberapa mata kuliah tertentu)

Kendala *soft* yang didefinisikan dalam formulasi model yang berfungsi untuk menentukan suatu jadwal optimal atau tidak sebagai berikut:

- S1: Tidak ada mahasiswa yang mendapat hanya satu mata kuliah pada satu hari
- S2: Tidak ada mahasiswa yang mendapat kuliah pada slot terakhir setiap harinya

Fungsi kendala untuk kedua kendala didefinisikan sebagai berikut:

- Kendala *hard*

$$\forall j \in J \text{ dan } k \in K$$

$f_1(k) = 3$, jika kuliah k melanggar kendala *hard* ke-1,3,4

$f_2(k) = 2$, jika kuliah k tidak melanggar kendala *hard* ke-2

$f_3(k) = 1$, jika kuliah k tidak melanggar kendala *hard* ke-5

$f_1(k)$ adalah fungsi penalti kendala *hard* yang memenuhi kendala H1, H3, dan H4

$f_2(k)$ adalah fungsi penalti kendala *hard* yang memenuhi kendala H2

$f_3(k)$ adalah fungsi penalti kendala *hard* yang memenuhi kendala H5

$$\text{Jadi } f(k) = f_1(k) + f_2(k) + f_3(k)$$

Total kendala *hard* yang dilanggar pada jadwal ke $j = F(j)$

$$F(j) = \sum f(k)$$

Jadwal yang layak apabila, $F(j) = 0$

- Kendala *soft*

$$\forall j \in J \text{ dan } k \in K$$

$g_i(k) = 1$, jika kuliah k melanggar kendala *soft* ke- i

$g_i(k) = 0$, jika kuliah k tidak melanggar kendala *soft* ke- i

$i = 1, 2$ dimana $g(k)$ adalah penalti kendala *soft* untuk kuliah ke- k dan

$G(j)$ adalah penalti kendala *soft* untuk jadwal j . Perhitungan kendala

soft setelah jadwal layak terbentuk, dan jadwal yang optimal merupakan jadwal yang mempunyai penalti kendala *soft* terkecil.

$$\text{Jadi } \sum g(k) = g_1(k) + g_2(k)$$

Total kendala *soft* yang dilanggar pada jadwal ke $j = G(j)$

$$G(j) = \sum g(k)$$

- Fungsi tujuan: $g = \min_{j \in J} F(j) + G(j)$

4.2 Penyusunan Algoritma Genetika

Dalam algoritma genetika terdapat beberapa komponen yang berhubungan dengan penjadwalan kuliah di Fakultas Teknik UI. Skema pengkodean yang digunakan dalam penelitian ini adalah *discrete decimal encoding*, proses seleksi orang tua dilakukan dengan sistem turnamen ukuran 2, proses pindah silang dengan *uniform crossover* yang menghasilkan satu kromosom anak, serta mutasi hanya dilakukan untuk kuliah-kuliah yang tidak terkena penalti.

Penjadwalan kuliah menggunakan metode algoritma genetika dibuat dengan bantuan perangkat lunak MATLAB R2007a. MATLAB adalah bahasa komputasi teknis tingkat tinggi dan merupakan lingkungan yang interaktif untuk pengembangan algoritma, visualisasi data, analisis data, dan komputasi numerik. Konstruksi penyelesaian komputasi teknis dengan MATLAB dapat dilakukan lebih cepat dibandingkan dengan bahasa pemrograman tradisional seperti C, C++, dan Fortran. MATLAB menyediakan fungsi-fungsi matematis untuk aljabar linear, statistik, optimasi, dan lainnya. Selain itu, MATLAB juga menyediakan fitur-fitur dokumentasi dan integrasi algoritma berbasis MATLAB dengan bahasa dan aplikasi lain, seperti C, C++, Fortran, Java, COM, dan Microsoft Excel. Bahasa MATLAB memudahkan operasi-operasi vektor dan matriks yang merupakan dasar bagi permasalahan di bidang teknik dan ilmiah⁸.

4.2.1 Langkah-langkah penyusunan algoritma genetika

Algoritma genetika yang dirancang untuk menyelesaikan penjadwalan kuliah pada perguruan tinggi dapat dilihat pada Gambar 4.3 dan *Script M-file* dapat dilihat pada Lampiran. Berikut adalah prosedur penyelesaian yang terdiri dari beberapa tahapan, yaitu:

1. Menentukan parameter/variabel yang digunakan
 - Ukuran populasi

Seperti yang telah dijelaskan sebelumnya, algoritma genetika bekerja dengan membentuk suatu populasi solusi bukan solusi tunggal, sehingga perlu ditetapkan ukuran populasi permasalahan. Karena

⁸ www.mathworks.com

ukuran populasi yang lebih kecil memberikan hasil yang lebih baik^{9,10} sehingga dipilih 16 menjadi ukuran populasi terkecil.

- Probabilitas kawin silang

Merupakan parameter penting dalam algoritma genetika karena sebuah kromosom yang mengarah pada kualitas solusi yang baik bisa diperoleh dari proses kawin silang.¹¹

- Probabilitas mutasi

Biasanya probabilitas mutasi diset sebagai $1/n$ di mana n adalah jumlah gen dalam kromosom¹².

- dan maksimum iterasi

2. Menentukan kromosom/populasi awal secara acak.

Populasi awal inilah yang nantinya akan diperbaiki dengan menggunakan operator genetika yang ada.

3. Melakukan evaluasi terhadap setiap kromosom dalam populasi awal.

Tahapan-tahapan dalam melakukan evaluasi individu ialah:

- Menghitung total penalti kendala *hard* dan *soft* untuk tiap kromosom yang ada dalam populasi awal. Penalti ini dihitung dengan melihat kembali hubungan antar mata kuliah mana yang boleh bentrok atau tidak dan syarat-syarat lainnya berdasarkan tabel yang dijadikan input dalam pembuatan model penjadwalan ini.

- Kemudian, menghitung *fitness value*. Karena fungsi tujuan permasalahan ini adalah meminimumkan total penalti kendala *hard* dan *soft*, maka $fitness\ value = 1/total\ penalti + 1$ untuk mencegah nilai *fitness* tidak terdefinisi.

4. Setelah itu, kromosom terbaik akan dilestarikan dengan jalan mengkopi kromosom terbaik (*elitisme*).

Jika ukuran populasi genap, maka ada 2 kromosom yang disimpan, dan jika ganjil maka ada 1 kromosom yang disimpan. *Elitisme* ini

⁹ O. Ender, A memetic algorithm for solving a timetabling problem: an incremental strategy

¹⁰ A. Vivas Andrade, Analysis of Selection and Crossover Methods used by Genetic Algorithm-based Heuristic to solve the LSP Allocation Problem in MPLS Networks under Capacity Constraints, International Conference on Engineering Optimization, Rio de Janeiro, June 2008

¹¹ Suyanto, *Op. Cit.* hal. 13.

¹² *Ibid.* hal. 14.

menyebabkan kromosom terbaik yang tidak terpilih dalam proses seleksi menjadi tidak hilang. Kromosom terbaik selalu dimiliki oleh kromosom 1, sehingga kromosom ini tidak akan mengalami proses kawin silang dan mutasi.

5. Untuk mencegah terjadinya *local optima*, maka perlu dilakukan penskalaan nilai *fitness* dengan menggunakan metode *Linear Fitness ranking* (LFR)¹³. Nilai *fitness* yang telah dilinierkan/dinormalisasi disebut *linear fitness*.
6. Setelah didapatkan *linear fitness*, maka dilakukan proses seleksi terhadap kromosom untuk mendapatkan 2 kromosom yang disebut orangtua (*parent*). Pada seleksi alami yang terjadi di dunia nyata, beberapa individu (biasanya individu jantan) berkompetisi dalam sebuah kelompok kecil sampai tersisa hanya satu individu pemenang. Individu pemenang inilah yang bisa kawin (pindah silang). Metode *roulette-wheel selection* tidak bisa mengakomodasi masalah ini. Sebuah metode lain yang disebut *tournament selection* mencoba mengadopsi karakteristik alami ini. Dalam bentuk paling sederhana, metode ini mengambil dua kromosom secara acak dan kemudian menyeleksi salah satu yang bernilai *fitness* paling tinggi untuk diambil menjadi orang tua pertama. Cara yang sama dilakukan lagi untuk mendapatkan orang tua yang kedua. Pada *tournament selection*, variabel m adalah *tournament size* dan p adalah *tournament probability*. Biasanya m diset sebagai suatu nilai yang sangat kecil, dalam kasus ini nilai m adalah 2. Sedangkan p biasanya diset sekitar 0,75.
7. Selanjutnya, dilakukan operasi genetika yang pertama yaitu kawin silang/*crossover*.

Tipe kawin silang yang digunakan adalah *uniform crossover*. Kawin silang ini bertujuan untuk meminimalkan penalti kendala *hard* dan *soft*. Kawin silang dilakukan pada dua kromosom *parent* yang dihasilkan dari proses seleksi. Awalnya, kromosom jadwal diubah menjadi gen-gen jadwal yang merupakan matrik kombinasi antara mata kuliah dengan ruang dan slot waktu. Dari tiap gen jadwal dihitung penalti kendala *hard* dan *soft*

¹³ *Ibid.* hal.12.

terhadap tiap mata kuliah. Setelah nilai penalti kendala *hard* dan *soft* tiap gen diketahui, kemudian dihitung nilai *fitness* masing-masing gen dengan rumus $1/(\text{total penalti}+1)$. Hasilnya adalah matrik ukuran $1 \times \text{jumlah kuliah}$ yang berisi nilai *fitness* masing-masing gen. setelah itu dilakukan operasi kawin silang dengan *uniform crossover*. *Uniform crossover* dilakukan dengan memilih gen-gen mana dari tiap orang tua yang akan diturunkan ke kromosom anak berdasarkan nilai *fitness* masing-masing gen. Gen-gen yang memiliki nilai *fitness* lebih tinggi yang akan diturunkan ke kromosom anak. Satu kali proses kawin silang hanya akan menghasilkan satu anak.

Kromosom Orangtua 1 :	0	0	0	0	0	0	0	0
Kromosom Orangtua 2 :	1	1	1	1	1	1	1	1
Keturunan 1 :	0	0	1	0	0	1	1	0
Keturunan 2 :	1	1	0	1	1	0	0	1

Gambar 4.1 Contoh Uniform Crossover pada Pengkodean Biner dengan Titik Persilangannya yang Acak adalah 2,3,5, dan 7

8. Operasi genetika yang kedua adalah mutasi. Metode mutasi yang digunakan yaitu *exchange mutation*¹⁴ dengan probabilitas mutasi sebesar 0,5. Metode ini menukarkan 2 buah gen, dalam satu kromosom yang dipilih secara acak. Kromosom yang akan mengalami mutasi adalah kromosom anak. Proses mutasi dengan metode *exchange mutation* ditunjukkan oleh Gambar 4.2

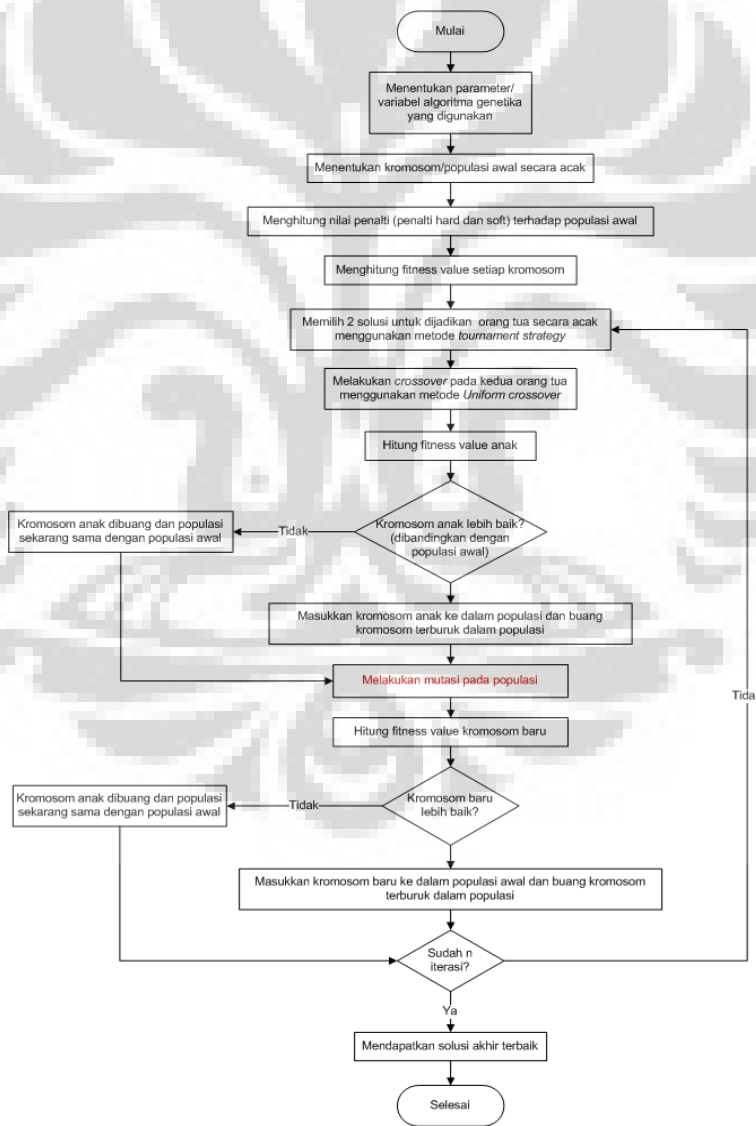
	TM1		TM2			
Sebelum Mutasi	2	3	5	1	4	6
Sesudah Mutasi	2	4	5	1	3	6

Gambar 4.2 *Exchange Mutation*. Mutasi dilakukan pada kromosom anak hasil kawin silang

¹⁴ *Ibid.*

9. Tahap terakhir yaitu penggantian populasi dengan sistem *steady state replacement*. Populasi terburuk dari 16 jadwal akan diganti dengan populasi baru yang terbaik hasil dari sebelumnya. Oleh karena proses iterasi hanya menghasilkan satu anak (*child*) sehingga yang diganti hanya satu populasi.

Prosedur ini akan berulang terus hingga tercapainya *stopping criteria* yaitu maksimum iterasi sesuai dengan jumlah yang ditentukan. Lalu, akan diambil kromosom terbaik akan menjadi solusi penjadwalan mata kuliah ini (kromosom terbaik selalu merupakan kromosom pertama dan kedua dalam populasi).



Gambar 4.3 Diagram Alur Pengerjaan Algoritma Genetika

Universitas Indonesia

4.2.2 Verifikasi dan Validasi Program

Sebelum dilakukan penyelesaian terhadap permasalahan penjadwalan mata kuliah pada perguruan tinggi, maka terlebih dahulu dilakukan verifikasi dan validasi program. Tahap verifikasi merupakan tahap melihat kesesuaian antara model program yang didapat dengan konseptual model yang kita buat/inginkan. Parameter model program dikatakan telah terverifikasi apabila telah berjalan sesuai konseptual model, dimana ada perubahan total penalti. Ketika program dijalankan dengan mengubah-ubah parameter maka akan didapatkan output yang berbeda-beda. Berarti program telah terverifikasi.

Selanjutnya, dilakukan validasi program. Validasi terhadap program dilakukan dengan memasukkan data *dummy*. Tujuannya adalah agar dapat divalidasi bahwa program berjalan sesuai dengan fungsinya. Hasil *run* program dengan data *dummy* kemudian dibandingkan dengan perhitungan manual untuk memperoleh validasi tersebut.

Data *dummy* yang digunakan terdiri dari 3 yaitu data mata kuliah, data kapasitas ruangan, dan data dosen. Data mata kuliah terdiri dari 10 mata kuliah yang merupakan gabungan mata kuliah yang diperuntukkan untuk mahasiswa tiap tingkatan. Mata kuliah dilambangkan dengan g_i , dimana $i = 1, 2, 3, 4, \dots, 10$. Kelompok mahasiswa dilambangkan dengan j , dimana $j = 1, 2, 3, 4, 5, \text{ dan } 6$. Data kapasitas ruang kuliah terdiri dari 2 ruangan dengan kapasitas 50 dan 80 yang dinotasikan sebagai ruangan A dan B. Data dosen merupakan data nama-nama dosen yang mengajarkan 10 mata kuliah pada data *dummy*. Slot waktu yang disediakan pada tabel 4.4 yang digunakan dalam validasi sama dengan slot waktu yang disediakan sebelumnya. Ketiga data *dummy* tersebut kemudian diolah sehingga menghasilkan matrik hubungan KK pada tabel 4.5. Hubungan antara kuliah dan slot waktu dinyatakan dengan matrik KT yang dapat dilihat pada tabel 4.6. Parameter algoritma genetika untuk validasi ini juga ditentukan dan dapat dilihat pada tabel 4.7.

Tabel 4.1 Data *Dummy* untuk Mata Kuliah

MK	SMTR	SKS	Notasi	Tingkat 1		Tingkat 2		Tingkat 3	Tingkat 4
				1	2	3	4	5	6
Kalkulus A	1	4	1	35	0	20	0	0	0
Statistik & Probabilitas A	1	3	5	35	0	20	0	0	0
Kimia Dasar B	3	2	14	0	0	0	55	0	0
Fasilitas Industri B	3	2	18	0	0	0	55	0	0
Perencanaan & Pengendalian Produksi A	5	3	29	0	0	0	0	35	20
Pemasaran Industri A	5	3	33	0	0	0	0	35	20
Total Quality Management (TQM)		3	37	0	0	0	0	40	0
Manajemen Proyek B	7	3	45	0	0	0	0	0	45
Kapita Selektta Industri	7	2	46	0	0	0	0	0	90
Sistem Distribusi dan Logistik		3	52	0	0	0	0	0	40

Tabel 4.2 Data *Dummy* untuk Kapasitas Ruang Kuliah

Ruang	Kapasitas
A	50
B	100

Tabel 4.3 Data *Dummy* untuk Dosen

KD	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	55	0
5	0	0	0	55	0	0	0	0	0	55	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	55	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	55	0	0	0	0	0
29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	55	0	0	0	0	0	0	0
33	0	0	0	0	0	0	0	55	0	0	0	0	0	0	0	0	0	0	0	0	0	0
37	0	0	0	0	0	0	40	0	0	0	0	0	0	40	40	0	0	0	0	0	0	0
45	0	0	0	0	0	0	0	0	0	0	0	45	0	0	0	0	0	0	0	0	0	0
46	0	0	0	0	0	0	0	0	0	0	90	0	0	0	0	0	0	0	0	0	0	0
52	0	40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Tabel 4.4 Slot Waktu yang Disediakan

Senin	08.00-09.50	Kamis	08.00-09.50
	10.00-12.30		10.00-12.30
	13.30-16.00		13.30-16.00
	16.00-17.50		16.00-17.50
Selasa	08.00-09.50	Jumat	07.30-09.20
	10.00-12.30		09.30-11.20
	13.30-16.00		13.30-16.00
	16.00-17.50		
Rabu	08.00-09.50		
	10.00-12.30		
	13.30-16.00		
	16.00-17.50		

Tabel 4.5 Matrik KK *Dummy*

KK	1	5	14	18	29	33	37	45	46	52
1	0	1	0	0	0	0	0	0	0	0
5	1	0	0	0	0	0	0	0	0	0
14	0	0	0	1	0	0	0	0	0	0
18	0	0	1	0	0	0	0	0	0	0
29	0	0	0	0	0	1	1	0	0	0
33	0	0	0	0	1	0	1	0	0	0
37	0	0	0	0	1	1	0	0	0	0
45	0	0	0	0	0	0	0	0	1	1
46	0	0	0	0	0	0	0	1	0	1
52	0	0	0	0	0	0	0	1	1	0

Tabel 4.6 Matrik KT *Dummy*

KT	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1	1	1	0	1	1	0	0	1	1	1	0	1	1	0	0	1	1	1	0
5	0	0	1	0	0	1	1	0	0	0	1	0	0	1	1	0	0	0	1
14	1	1	0	1	1	0	0	1	1	1	0	1	1	0	0	1	1	1	0
18	1	1	0	1	1	0	0	1	1	1	0	1	1	0	0	1	1	1	0
29	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
33	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
37	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
45	0	0	1	0	0	1	1	0	0	0	1	0	0	1	1	0	0	0	1
46	1	1	0	1	1	0	0	1	1	1	0	1	1	0	0	1	1	1	0
52	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Tabel 4.7 Parameter yang Digunakan dalam Validasi

Parameter	Nilai
Ukuran Populasi	4
Maksimum iterasi	1
Probabilitas kawin silang	1
Probabilitas Mutasi	0.5

4.2.2.1 Hasil piranti lunak

Hasil *run* program menghasilkan jadwal terbaik dengan nilai *fitness* sebesar 0.091 yang dapat dilihat pada tabel di bawah ini:

Tabel 4.8 Jadwal Terbaik Hasil *Run* Program untuk Data *Dummy*

Kromosom Terbaik

Mata Kuliah	1	2	3	4	5	6	7	8	9	10
Ruang	2	2	2	2	2	2	1	1	2	1
Slot	9	6	10	18	1	5	13	6	13	5

Dengan nilai penalti *hard* dan *soft* untuk masing-masing gen mata kuliah sebagai berikut:

fk1	0	0	0	0	0	0	0	0	0	0
fk2	0	0	0	0	0	0	0	0	0	0
fk3	0	0	0	0	0	0	0	0	0	0
gk1	0	0	0	0	0	0	0	0	0	0
gk2	0	0	0	0	0	0	0	0	0	0

4.2.2.2 Hasil perhitungan manual

Untuk memvalidasi program, maka dilakukan perhitungan manual dimana pemilihan acak terhadap populasi awal (inisialisasi populasi) diperoleh dari program.

Iterasi yang dilakukan adalah sebagai berikut:

1. Pembentukan solusi awal (inisialisasi populasi)

Tabel 4.9 Populasi Awal (dari Program)

Kromosom 1	2	2	2	2	2	2	2	1	2	1
	9	6	10	18	1	5	17	6	13	9
Kromosom 2	2	2	2	2	2	2	1	1	2	1
	9	6	10	18	1	5	13	6	13	5
Kromosom 3	2	2	2	2	2	2	1	1	2	1
	9	6	18	4	1	5	1	6	13	9
Kromosom 4	2	2	2	2	2	2	1	1	2	1
	9	6	10	18	1	5	5	6	13	1

2. Evaluasi Individu

Proses evaluasi individu dilakukan dengan langkah-langkah sebagai berikut:

- Menghitung jumlah penalti kendala *hard* (fk1, fk2, fk3) dan *soft* (gk1 dan gk2)

Untuk menghitung penalti kendala *hard* dan *soft* diperlukan matrik hubungan KK, KT, KM, dan KR sebagai acuan untuk menentukan nilai penalti. Jika kombinasi mata kuliah, ruangan, dan slot-waktu yang terdapat pada kromosom jadwal melanggar hubungan yang terdapat pada matrik KK, KT, KM, dan KR, maka penalti untuk fk1 akan diberikan 3, fk2 akan diberikan 2, dan sisanya diberikan nilai 1. Penilaian secara bertingkat menggunakan pembobotan ini untuk memeberikan prioritas kendala yang tidak boleh dilanggar.

Tabel 4.10 Kromosom ke-1

KJ1	1	2	3	4	5	6	7	8	9	10
R	2	2	2	2	2	2	2	1	2	1
T	9	6	10	18	1	5	17	6	13	9

Tabel 4.11 Perhitungan Total Penalti *Hard* dan *Soft* Setiap Gen pada Kromosom ke-1

KJ1	1	2	3	4	5	6	7	8	9	10
R	2	2	2	2	2	2	2	1	2	1
T	9	6	10	18	1	5	17	6	13	9
fk1	0	0	3	0	3	3	3	0	3	0
fk2	0	0	0	0	0	0	0	0	0	0
fk3	0	0	0	0	0	0	0	0	0	0
gk1	1	1	0	1	0	0	1	0	0	0
gk2	0	0	0	0	0	0	0	0	0	0

Tabel 4.12 Kromosom ke-2

KJ2	1	2	3	4	5	6	7	8	9	10
R	2	2	2	2	2	2	1	1	2	1
T	9	6	10	18	1	5	13	6	13	5

Tabel 4.13 Perhitungan Total Penalti Kendala *Hard* dan *Soft* Setiap Gen pada Kromosom ke-2

KJ2	1	2	3	4	5	6	7	8	9	10
R	2	2	2	2	2	2	1	1	2	1
T	9	6	10	18	1	5	13	6	13	5
fk1	3	0	0	0	3	0	0	0	0	0
fk2	0	0	0	0	0	0	0	0	0	0
fk3	0	0	0	0	0	0	0	0	0	0
gk1	1	1	0	1	1	0	0	0	0	0
gk2	0	0	0	0	0	0	0	0	0	0

Tabel 4.14 Kromosom ke-3

KJ3	1	2	3	4	5	6	7	8	9	10
R	2	2	2	2	2	2	1	1	2	1
T	9	6	18	4	1	5	1	6	13	9

Tabel 4.15 Perhitungan Total Penalti Kendala *Hard* dan *Soft* Setiap Gen pada Kromosom ke-3

KJ3	1	2	3	4	5	6	7	8	9	10
R	2	2	2	2	2	2	1	1	2	1
T	9	6	18	4	1	5	1	6	13	9
fk1	0	0	0	0	3	3	3	0	3	0
fk2	0	0	0	0	0	0	0	0	0	0
fk3	0	0	0	0	0	0	0	0	0	0
gk1	0	1	1	0	1	0	0	0	0	0
gk2	0	0	0	1	0	0	0	0	0	0

Tabel 4.16 Kromosom ke-4

KJ4	1	2	3	4	5	6	7	8	9	10
R	2	2	2	2	2	2	1	1	2	1
T	9	6	10	18	1	5	5	6	13	1

Tabel 4.17 Perhitungan Total Penalti Kendala *Hard* dan *Soft* setiap Gen pada Kromosom ke-4

KJ4	1	2	3	4	5	6	7	8	9	10
R	2	2	2	2	2	2	1	1	2	1
T	9	6	10	18	1	5	5	6	13	1
fk1	3	0	3	0	0	3	3	0	3	0
fk2	0	0	0	0	0	0	0	0	0	0
fk3	0	0	0	0	0	0	0	0	0	0
gk1	1	1	0	0	0	0	0	0	0	0
gk2	0	0	0	0	0	0	0	0	0	0

Tabel 4.18 Perhitungan *Fitness Value* pada keempat Kromosom

Kromosom	Fitness Value
Kromosom 1	0.050
Kromosom 2	0.090909091
Kromosom 3	0.058823529
Kromosom 4	0.055555556

Jadi, diperoleh *fitness value* dari populasi awal adalah sebagai berikut: 0.050, 0.0909, 0.05882, 0.05556. Adapun kromosom yang memiliki *fitness* tertinggi ialah kromosom 2 dengan $\text{Max } F = 0.0909$, dan kromosom yang menempati *fitness* dengan nilai terkecil ialah kromosom 1 dengan $\text{Min } F = 0.050$

3. Pemilihan orang tua

Secara acak, diperoleh dua buah kromosom yang menjadi orang tua yaitu kromosom 3 dan kromosom 4.

4. Kawin Silang

Selanjutnya, dua buah kromosom ini akan mengalami proses kawin silang dengan metode *uniform crossover* (Gambar 4.4) sehingga didapatkan satu buah kromosom baru (anak). Titik potong ditentukan berdasarkan nilai penalti kendala *hard* dan *soft* tiap gen jadwal. Gen dengan nilai *fitness* lebih baik dari salah satu kromosom orang tua yang akan diturunkan ke kromosom anak.

KJ3	1	2	3	4	5	6	7	8	9	10
R	2	2	2	2	2	2	1	1	2	1
T	9	6	18	4	1	5	1	6	13	9

KJ4	1	2	3	4	5	6	7	8	9	10
R	2	2	2	2	2	2	1	1	2	1
T	9	6	10	18	1	5	5	6	13	1

Anak	1	2	3	4	5	6	7	8	9	10
R	2	2	2	2	2	2	1	1	2	1
T	9	6	18	4	1	5	1	6	13	9

Gambar 4.4 Kawin Silang dengan *Uniform Crossover*

5. Mutasi

Pada setiap anak akan dilakukan proses mutasi, yaitu menukarkan dua buah gen dalam satu kromosom dimana titik mutasi juga dipilih secara acak. Proses mutasi dapat dilihat pada Gambar 4.5.

Anak	1	2	3	4	5	6	7	8	9	10
R	2	2	2	2	2	2	2	1	2	2
T	17	10	5	3	13	1	9	3	13	2

Gambar 4.5 Mutasi

6. Perhitungan *fitness value* anak

Langkah-langkah perhitungan *fitness value* anak sama dengan langkah-langkah mendapatkan *fitness value* pada populasi awal. Tabel 4.19 dan Tabel 4.20 di bawah ini menunjukkan proses perhitungan *fitness value* pada anak hasil mutasi.

Tabel 4.19 Kromosom Anak Hasil Mutasi

Anak	1	2	3	4	5	6	7	8	9	10
R	2	2	2	2	2	2	2	1	2	2
T	17	10	5	3	13	1	9	3	13	2

Tabel 4.20 Perhitungan Total Penalti Kendala *Hard* dan *Soft* Setiap Gen pada Kromosom Anak Hasil Mutasi

Anak	1	2	3	4	5	6	7	8	9	10
R	2	2	2	2	2	2	1	1	2	1
T	5	6	9	18	4	1	1	6	13	9
fk1	0	0	0	0	0	3	3	0	0	0
fk2	0	0	0	0	0	0	0	0	0	0
fk3	0	0	0	0	0	0	0	0	0	0
gk1	0	0	1	1	0	1	1	1	1	0
gk2	0	0	0	0	0	0	0	0	0	0

Kromosom anak hasil mutasi menghasilkan nilai *fitness* sebesar 0.07692307.

7. Dilakukan penggantian kromosom terbaik

Kromosom baru dibandingkan dengan populasi awal. Apabila *fitness value* kromosom baru lebih baik dibandingkan kromosom terburuk dalam populasi awal, maka kromosom baru akan menggantikan kromosom terburuk tersebut. Hal ini dilakukan agar ukuran populasi tetap/konstan. Jadi, didapatkan bahwa kromosom anak hasil mutasi akan menggantikan kromosom ke-1 dalam populasi awal, sedangkan kromosom ke-1 akan dibuang karena memiliki nilai *fitness* yang terburuk.

8. Mendapatkan solusi akhir terbaik

Setelah didapatkan komposisi populasi baru, dipilih satu solusi terbaik dari populasi baru sehingga solusi akhir berubah menjadi kromosom ke-2 dengan Max F = 0.0909 dengan susunan jadwal terbaik seperti pada Tabel 4.21.

Tabel 4.21 Jadwal Terbaik Hasil Perhitungan Manual

KJ2	1	2	3	4	5	6	7	8	9	10
R	2	2	2	2	2	2	1	1	2	1
T	9	6	10	18	1	5	13	6	13	5

Berdasarkan hasil perhitungan manual di atas, diperoleh jawaban yang sama dengan hasil *run* program. Dengan demikian program telah tervalidasi.

4.2.3 Input Data

Data mentah dari masalah penjadwalan kuliah merupakan interaksi antara komponen-komponen masalah yang berupa dosen, kuliah, mahasiswa, dan sejumlah ruangan. Hubungan dapat terjadi antara dosen dengan kuliah yang diajarkan, mahasiswa mengambil beberapa kuliah, ruangan harus dapat menampung sejumlah peserta kuliah, beberapa kuliah tidak boleh bentrok dan sebagainya. Oleh karena itu, perlu manajemen data untuk mengolah data mentah ke dalam implementasi program. Berikut adalah manajemen data yang digunakan:

1. Matrik $KK_{|K| \times |K|}$ yang mengindikasikan hubungan antar kuliah mana yang boleh atau tidak dilangsungkan pada slot-waktu yang bersamaan.
 $KK(i,j) = 0$ jika kuliah ke- i dan ke- j dapat dilangsungkan pada slot-waktu yang bersamaan
 $KK(i,j) = n$ jika kuliah ke- i dan ke- j tidak dapat dilangsungkan pada slot-waktu yang bersamaan, dan n adalah jumlah maksimum mahasiswa kelompok ke- i dan ke- j
2. Vektor kapasitas ruangan yang mengindikasikan kapasitas dari ruangan-ruangan yang akan digunakan untuk melangsungkan kuliah
3. Matrik $KM_{|K| \times |M|}$ yang mengindikasikan kuliah yang akan diambil oleh grup mahasiswa serta jumlah mahasiswanya, dimana $|K|$ dan $|M|$ masing-masing jumlah kuliah dan jumlah grup mahasiswa
 $KM(i,j) = 0$ jika kelompok mahasiswa ke- j tidak mengambil kuliah ke- i

$KM(i,j) = n$ jika kelompok mahasiswa ke- j mengambil kuliah ke- i , dan n adalah jumlah mahasiswa kelompok ke- j

4. Matrik $KT_{|K| \times |T|}$ yang mengindikasikan slot-waktu yang sesuai untuk menyelenggarakan suatu mata kuliah tertentu, dimana $|K|$ dan $|T|$ masing-masing jumlah kuliah dan jumlah slot-waktu yang disediakan
- $KT(i,j) = 0$ jika kelompok mata kuliah ke- i tidak dapat diselenggarakan pada slot-waktu ke- j

$KT(i,j) = n$ jika kelompok mata kuliah ke- i dapat diselenggarakan pada slot-waktu ke- j

5. Matrik $KD_{|K| \times |D|}$ yang mengindikasikan dosen pengajar dari masing-masing kuliah, dimana $|D|$ adalah jumlah dosen yang mengajar mata kuliah yang bersangkutan

$KD(i,j) = 0$ jika dosen ke- j tidak mengajar kuliah ke- i

$KD(i,j) = n$ jika dosen ke- j mengajar mata kuliah ke- i , dan n adalah jumlah mahasiswa pengambil mata kuliah ke- j

Manajemen data yang telah disimpan dalam bentuk vektor/matrik dalam perangkat lunak MATLAB akan digunakan untuk membuat jadwal dimana kumpulan semua jadwal yang layak berada pada ruang solusi. Representasi jadwal ditulis dalam matrik jadwal $J_{|R| \times |W|}$ dengan $|R|$ dan $|W|$ masing-masing menyatakan jumlah ruangan dan slot-waktu dalam seminggu. Elemen dari matrik tersebut adalah label kuliah dan bilangan -1, jika dalam ruangan ke- i dan slot-waktu ke- j terdapat kuliah maka elemen $J(i,j)$ adalah label kuliahnya, sedangkan jika dalam ruangan ke- i dan slot-waktu ke- j tidak terdapat kuliah maka elemen $J(i,j)$ adalah -1. Bentuk representasi jadwal dapat dilihat pada Gambar 4.6

Ruang/Slot	1	2	3	4	5	6	7	8	9	10	11	12	13
1	3	-1	-1	-1	13	-1	-1	-1	18	-1	-1	-1	52
2	42	-1	-1	-1	7	-1	-1	-1	44	-1	-1	-1	-1
3	50	56	-1	-1	54	55	-1	-1	-1	-1	-1	-1	2
4	34	35	40	-1	36	9	-1	-1	15	21	-1	-1	16
5	22	17	-1	-1	26	4	49	-1	43	5	-1	-1	11
6	29	14	51	-1	6	12	-1	-1	28	39	-1	-1	27
7	19	32	25	-1	33	31	-1	-1	1	46	-1	-1	30

Gambar 4.6 Representasi Jadwal

Matrik J akan ditransformasikan menjadi kromosom dengan setiap gen membawa informasi mengenai ruang dan slot-waktu masing-masing kuliah. Dalam hal ini satu jadwal diwakili dengan satu kromosom, dan tiap satu kuliah diwakili dengan satu gen, akibatnya dalam kromosom yang terbentuk memiliki ukuran sama dengan sejumlah kuliah yang ada. Kromosom ini merupakan kandidat solusi jadwal kuliah serta memegang peranan penting dalam algoritma genetika.

		Mata Kuliah											
		1	2	3	4	5	6	7	8	9	...	39	
Ruang						Ruang ke-i							
Slot-waktu						Slot-waktu ke-j							

Gambar 4.7 Kromosom Jadwal Sebagai Kandidat Solusi

4.2.4 Pembuatan Populasi Awal

Sejumlah 16 kromosom sebagai kandidat awal solusi dibuat secara random. Pembuatan random cukup dengan menempatkan semua kuliah dalam matrik jadwal J dan elemen matrik J yang masih kosong diberikan nilai -1. Langkah-langkah yang dilakukan dalam proses inisialisasi populasi awal dapat dijelaskan sebagai berikut:

1. Membuat matrik hubungan KR, yang menyatakan hubungan antara mata kuliah (K) dengan ruang kelas yang tersedia untuk menyelenggarakan perkuliahan. Pertama-tama, mendefinisikan terlebih dahulu ukuran matrik yang diinginkan ke dalam program. Selanjutnya, inisialisasikan nilai 0 untuk matrik hubungan KR tersebut. Setelah itu, baru dilihat kesesuaian antara kapasitas mata kuliah dengan kapasitas ruang kelas yang tersedia. Jika kapasitas mata kuliah tertentu lebih kecil daripada kapasitas ruang kelas, maka diberi nilai 1 untuk hubungan antara mata kuliah tersebut dengan ruangan yang bersangkutan pada matrik.

2. Membuat matrik irisan bentrok berdasarkan matrik hubungan KK dan KM. Matrik irisan bentrok dibuat dengan melihat irisan jumlah mahasiswa yang berpotensi mengalami bentrok dengan memperhatikan hubungan antar mata kuliah.
3. Membuat matrik total bentrok yang berukuran $1 \times$ jumlah mata kuliah yang berisi jumlah mahasiswa yang berpotensi mengalami bentrok untuk tiap-tiap mata kuliah.
4. Mata kuliah yang memiliki total bentrok terbesar akan dijadwalkan terlebih dahulu ke dalam ruang dan slot waktu yang tepat menurut matrik hubungan KT dan KR

Proses inisialisasi populasi awal ini akan menghasilkan 16 populasi dalam bentuk matrik jadwal dan kromosom jadwal yang memiliki nilai *fitness* terbaik dari hasil *running* dengan jumlah generasi tertentu yang dipilih secara random. Setelah jadwal awal terbentuk, akan dilakukan pembentukan populasi kromosom yang memenuhi seluruh kendala *hard*.

4.2.5 Parameter

Sebelum menentukan kombinasi parameter yang dipakai, terlebih dahulu dilakukan pengujian terhadap ke-4 parameter tersebut untuk melihat kombinasi parameter terbaik untuk permasalahan penjadwalan mata kuliah pada perguruan tinggi. Hal ini dilakukan untuk menentukan parameter yang dapat memberikan hasil yang terbaik dalam masalah ini.

Keempat parameter tersebut diuji satu persatu terhadap model yang telah dibuat. Parameter-parameter tersebut diubah-ubah untuk melihat pengaruhnya terhadap jadwal yang dihasilkan oleh model. Kombinasi parameter yang menghasilkan jadwal dengan nilai *fitness* terbaik yang akan digunakan. Parameter yang digunakan dalam pembuatan model jadwal dapat dilihat pada Tabel 4.22:

Tabel 4.22 Parameter yang Digunakan Dalam Pembuatan Model

Parameter	Nilai
Ukuran Populasi	16
Maksimum Iterasi	10000
Probabilitas kawin silang	1
Probabilitas mutasi	0.5

4.3 Hasil Pengolahan Data

Setelah memasukkan data dan menjalankan model yang telah dibuat dengan program, maka keluaran yang dihasilkan oleh program yang telah dibuat adalah jadwal kuliah optimal yang memiliki penalti kendala *hard* dan *soft* yang terkecil sehingga dapat dianggap sebagai solusi yang layak.

Data yang dijadikan input dalam pembuatan model memenuhi seluruh asumsi awal tentang model penjadwalan yang layak. Setelah program dijalankan selama 10000 kali iterasi didapatkan hasil pada Tabel 4.23 sebagai berikut:

Tabel 4.23 Kromosom Jadwal Hasil 10000 Iterasi

Matrix Jadwal

Ruang/Slot	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1	-1	-1	-1	-1	41	-1	-1	-1	-1	-1	-1	-1	39	-1	-1	-1	54	-1	-1
2	49	-1	-1	-1	-1	45	-1	-1	38	-1	-1	-1	52	37	-1	-1	51	-1	-1
3	40	-1	-1	-1	56	-1	-1	-1	-1	-1	-1	-1	43	44	-1	-1	-1	-1	-1
4	16	36	-1	13	14	-1	-1	18	12	27	-1	-1	28	-1	-1	-1	4	29	-1
5	11	20	35	33	10	24	-1	-1	7	55	-1	-1	6	31	-1	-1	30	-1	-1
6	3	5	-1	42	8	2	-1	-1	26	22	-1	-1	19	-1	-1	-1	15	25	-1
7	50	53	48	17	34	32	-1	-1	47	1	-1	-1	21	23	-1	-1	9	46	-1

Dengan jumlah nilai penalti *hard* dan *soft* untuk kromosom jadwal tersebut seperti yang dapat dilihat pada Tabel 4.24:

Tabel 4.24 Jumlah Nilai Penalti *Hard* dan *Soft*

Penalti	Total
fk1	72
fk2	0
fk3	6
gk1	2
gk2	5

Sehingga, kromosom jadwal tersebut memiliki nilai *fitness* sebesar 0.011627907 dengan waktu komputasi selama 18 menit 13 detik.

Jadwal yang dihasilkan pada Tabel 4.23 belum dapat dikatakan layak karena total penalti kendala *hard* yang masih sangat besar berjumlah 78 dan

penalti kendala *soft* berjumlah 7. Alasan lain yang menyebabkan jadwal tersebut tidak layak untuk diambil sebagai solusi karena penalti terbesar ada pada fungsi kendala fk_1 yang seharusnya bernilai 0 agar jadwal dapat dikatakan layak. Sehingga dilakukan iterasi ulang dengan mengubah input matrik $KT_{|K| \times |T|}$ yang semula bernilai 0 dan 1, diubah menjadi bernilai 1 untuk semua mata kuliah pada semua slot-waktu. Dengan mengubah matrik $KT_{|K| \times |T|}$ seperti ini berarti semua mata kuliah dianggap dapat diselenggarakan pada slot-waktu manapun. Setelah program dijalankan dengan menggunakan input matrik $KT_{|K| \times |T|}$ yang baru didapatkan jadwal optimal setelah melakukan *running* program sebanyak 43 kali yang dapat dilihat pada Tabel 4.25 sebagai berikut:

Tabel 4.25 Kromosom Jadwal dengan Input Matrik $KT_{|K| \times |T|}$ Baru

Matrix Jadwal																			
Ruang/Slot	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	52	-1	-1	-1	53	-1	-1
2	42	-1	-1	-1	-1	-1	-1	-1	44	-1	-1	-1	-1	-1	-1	-1	41	-1	-1
3	50	56	-1	-1	54	55	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
4	34	35	40	3	36	9	-1	7	15	21	-1	-1	16	47	18	-1	48	-1	-1
5	22	17	-1	-1	26	4	49	-1	43	5	-1	-1	11	10	2	-1	20	24	-1
6	29	14	51	-1	6	12	-1	13	28	39	-1	-1	27	38	-1	-1	8	37	-1
7	19	32	25	-1	33	31	-1	-1	1	46	-1	-1	30	45	-1	-1	23	-1	-1

Dengan jumlah nilai penalti *hard* dan *soft* untuk kromosom jadwal tersebut seperti yang dapat dilihat pada Tabel 4.26:

Tabel 4.26 Jumlah Nilai Penalti *Hard* dan *Soft*

Penalti	Total
fk_1	0
fk_2	0
fk_3	0
gk_1	5
gk_2	3

Sehingga, jadwal tersebut memiliki nilai *fitness* sebesar 0.11111 dengan waktu komputasi selama 18 menit 39 detik.

Kromosom jadwal yang baru pada Tabel 4.25 dapat dikatakan layak karena telah memenuhi fungsi kendala $fk1$ dengan nilai penalti sebesar 0. Selain itu, jumlah penalti *hard* dan *soft* yang minimal sehingga jadwal yang dihasilkan dapat dikatakan optimal. Jadwal yang terbentuk dari kromosom jadwal pada Tabel 4.25 dapat dilihat pada Lampiran.

Setelah dihasilkan jadwal yang layak dari program dengan menggunakan data yang telah diberikan sebelumnya, dicoba running dengan mengganti data ruangan menjadi 6 ruangan. Jumlah ruangan dibuat menjadi lebih sedikit untuk melihat pengaruhnya terhadap jadwal yang dihasilkan oleh program. Jadwal yang dihasilkan dengan input 6 ruang kelas dapat dilihat pada Tabel 4.27 dan jumlah nilai penalti *hard* dan *soft* dapat dilihat pada Tabel 4.28 di bawah ini:

Tabel 4.27 Kromosom Jadwal dengan Input 6 Ruang Kelas

Matrix Jadwal

Ruang/Slot	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1	55	-1	-1	-1	44	45	-1	-1	42	52	-1	-1	-1	48	-1	-1	-1	-1	-1
2	40	-1	-1	-1	-1	39	-1	-1	-1	-1	-1	-1	38	15	-1	-1	54	-1	-1
3	31	21	-1	34	29	47	-1	-1	25	28	-1	20	19	-1	-1	-1	36	35	-1
4	12	27	-1	51	33	7	-1	-1	2	18	-1	-1	9	-1	-1	-1	26	-1	-1
5	16	37	49	8	11	23	-1	-1	13	10	-1	-1	32	41	-1	56	46	50	-1
6	4	53	-1	17	5	24	-1	22	30	6	-1	-1	43	-1	-1	-1	-1	14	-1

Tabel 4.28 Jumlah Nilai Penalti *Hard* dan *Soft*

Penalti	Total
fk1	66
fk2	2
fk3	6
gk1	0
gk2	7

Sehingga, jadwal tersebut memiliki nilai *fitness* sebesar 0.012195 dengan waktu komputasi selama 11 menit 8 detik. Jadi, dapat disimpulkan bahwa dengan menggunakan parameter yang sama, ternyata penjadwalan dengan input 6 ruangan memberikan hasil yang sangat buruk bahkan jadwal tidak layak, namun memberikan waktu komputasi yang lebih cepat

4.4 Analisis

Permasalahan penjadwalan mata kuliah pada perguruan tinggi adalah kurang optimalnya jadwal yang dihasilkan dilihat dari banyaknya mata kuliah yang bentrok, kurang optimalnya penggunaan ruang kelas, serta tidak meratanya beban mahasiswa.

Usulan penyelesaiannya adalah dengan menggunakan algoritma metaheuristik 'Algoritma Genetika'. Dalam penelitian ini, algoritma tersebut diadopsi ke dalam bahasa pemrograman MATLAB. Fungsi tujuannya adalah meminimalisasi total penalti kendala *hard* dan *soft* seluruh mata kuliah. Analisis terhadap hasil pengolahan data di atas dibagi menjadi 3, yakni analisis metode, analisis skenario parameter, dan analisis hasil.

4.4.1 Analisis Metode

Prinsip dasar dari algoritma genetika ini ialah menggunakan evolusi Darwin, yaitu individu yang kuat yang akan bertahan, sedangkan individu yang lemah akan musnah. Untuk mendapatkan populasi awal dilakukan permutasi secara acak. Lalu individu yang akan mengalami proses kawin silang dan mutasi juga dipilih secara acak dengan membandingkan antara bilangan random yang dihasilkan dengan probabilitas kawin silang atau mutasi. Apabila bilangan random yang dihasilkan lebih kecil dari probabilitas silang atau probabilitas mutasi maka individu tersebut berhak untuk mengalami proses kawin silang atau mutasi. Ditambah lagi titik potong dan titik mutasi juga ditentukan secara acak. Inilah yang membuat adanya perbedaan urutan jadwal yang dihasilkan pada setiap *run* yang dilakukan. Kombinasi parameter yang sama belum tentu menghasilkan urutan jadwal yang sama pula. Oleh karenanya skenario parameter merupakan bagian yang terpenting dalam pengujian metode ini, karena tiap kasus membutuhkan konfigurasi yang unik untuk memperoleh urutan jadwal yang stabil dan sesuai dengan fungsi tujuan (total penalti kendala *hard* dan *soft* yang minimum), serta waktu *run* yang kurang dari 30 menit.

Salah satu metode yang dipakai pada algoritma genetika ini ialah penggantian populasi dengan skema *steady state replacement*, yaitu populasi terburuk dari 16 jadwal akan diganti dengan populasi baru yang terbaik hasil dari

sebelumnya. Oleh karena proses iterasi hanya menghasilkan satu anak (*child*) sehingga yang diganti hanya satu populasi. Alasan yang menyebabkan metode ini dipilih yaitu:

- a. *Steady-state replacement* melakukan dua kali evaluasi individu yaitu pertama pada individu pada populasi awal, kedua pada individu baru yang dihasilkan sehingga kromosom yang terpilih benar-benar merupakan kromosom yang memiliki nilai *fitness* yang terbaik.
- b. *Steady-state replacement* mengganti satu individu terburuk dalam setiap generasi sehingga jika individu baru yang dihasilkan ternyata memiliki kualitas solusi yang lebih baik dari populasi. Sebaliknya jika individu baru yang dihasilkan ternyata memiliki kualitas solusi yang lebih buruk dari populasi maka individu baru tersebut tidak dapat masuk ke dalam populasi, dan hal ini menyebabkan komposisi individu dalam populasi tidak berubah dari keadaan awal yaitu populasi awal. Ini menyebabkan populasi akan terdiri dari individu-individu yang memiliki nilai *fitness* terbaik.
- c. *Fitness* rata-rata metode *generational replacement* lebih buruk dibanding metode *steady-state replacement* karena individu baru selalu menggantikan individu awal tanpa memperhatikan apakah *fitness value* individu baru lebih baik ketimbang individu awal yang digantikannya. Hal ini menyebabkan populasi baru belum tentu merupakan kumpulan individu-individu terbaik setiap generasi. Sebaliknya, metode *steady-state replacement* memiliki kumpulan individu-individu terbaik dalam populasi.
- d. Metode *steady-state replacement* dikombinasikan dengan metode elitisme yaitu pengkopian individu terbaik dalam populasi. Hal ini menyebabkan dalam setiap generasi pasti ada satu individu terbaik yang terletak pada peringkat 1 (kromosom 1) dalam populasi. Hal ini untuk mengatasi agar tidak musnahnya individu terbaik dengan proses penggantian dengan metode *steady-state replacement*.

Penggunaan algoritma genetika dalam menyelesaikan permasalahan di atas, selain dapat menghasilkan solusi yang mendekati optimal, juga memungkinkan untuk dilakukannya penambahan atau pengurangan jumlah mata

kuliah dan ruangan. Jadi, metode ini dapat digunakan untuk menentukan jadwal terbaik jika ada jumlah mata kuliah atau ruangan yang bertambah atau berkurang.

4.4.2 Analisis Skenario Parameter

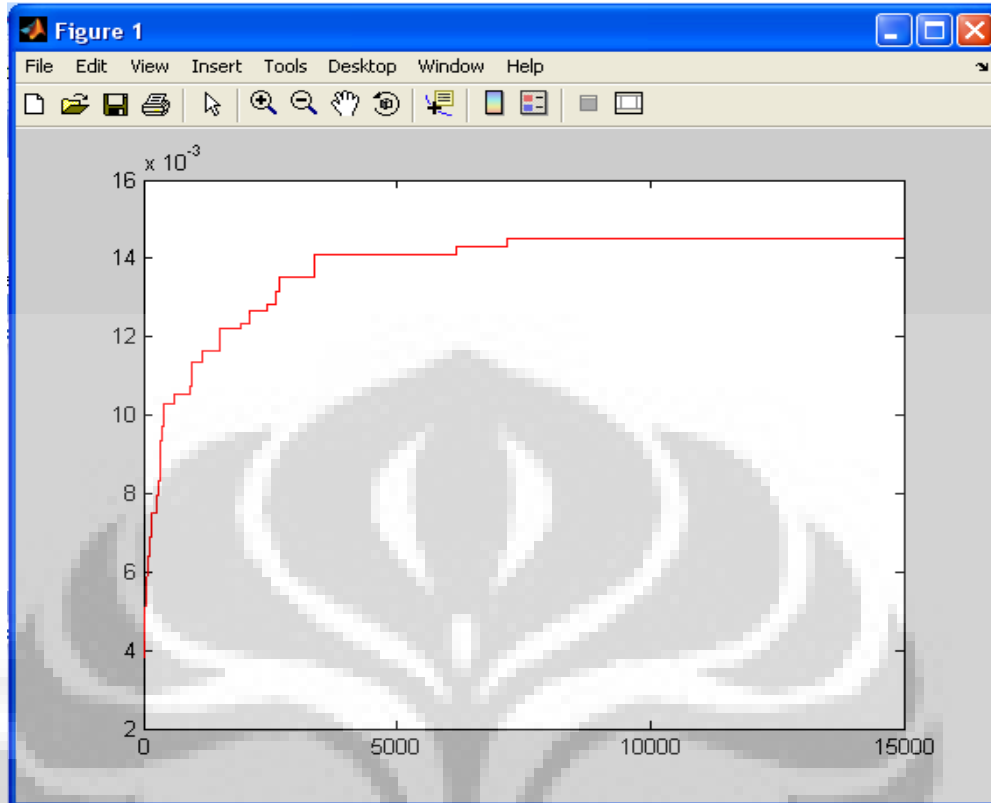
Analisis skenario parameter dapat ditentukan dengan melakukan beberapa percobaan dengan mengganti parameter-parameter yang telah kita tentukan sebelumnya untuk diuji. Parameter-parameter yang menghasilkan nilai *fitness* yang terbaik akan digunakan seterusnya dalam program yang dibuat. Parameter-parameter yang akan diuji terdiri dari ukuran populasi, maksimum iterasi, dan probabilitas mutasi. Pengujian dilakukan dengan mengganti parameter satu per satu. Setelah itu baru dikombinasikan dalam pembuatan jadwal.

Setelah menjalankan program yang telah dibuat dengan berbagai kombinasi parameter yang telah diubah-ubah, maka didapatkan hasil berupa nilai *fitness* dan waktu komputasi untuk tiap kombinasi yang dibuat dapat dilihat pada Tabel 4.29 di bawah ini:

Tabel 4.29 Hasil Kombinasi Parameter

Ukuran Populasi	Probabilitas Mutasi	Jumlah Iterasi	Fitness Value	Waktu Komputasi
16	0.2	1000	0.027027027	1m 60s
		2000	0.037037037	3m 48s
		5000	0.012195122	9m 21s
		10000	0.038461538	18m 57s
		15000	0.041666667	28m 27s
16	0.5	1000	0.008264463	1m 59s
		2000	0.023809524	3m 46s
		5000	0.016129032	9m 17s
		10000	0.0625	18m 39s
		15000	0.014492754	28m 23s
16	0.7	1000	0.008196721	1m 53s
		2000	0.040396607	3m 12s
		5000	0.009803922	9m 13s
		10000	0.035714286	18m 7s
		15000	0.038821927	28m 15s

Selain itu, nilai *fitness* untuk hasil iterasi untuk ukuran populasi 16, probabilitas mutasi 0.5, dan jumlah maksimum iterasi sebanyak 15000 kali dapat dilihat pada Gambar 4.8.



Gambar 4.8 Hasil running program setelah 15000 iterasi

Dari hasil yang didapatkan seperti yang terlihat pada Tabel 4.27 dapat disimpulkan beberapa hal, yaitu:

1. Jumlah iterasi yang makin besar cenderung memberikan hasil yang lebih baik. Hal ini dapat dilihat dari makin besarnya nilai *fitness* yang didapatkan dari tiap kombinasi setelah melakukan beberapa kali iterasi. Walaupun sebenarnya hasil *fitness* yang lebih baik belum tentu bisa diperoleh dengan memperbanyak jumlah iterasi karena terlihat adanya beberapa nilai *fitness* yang justru turun setelah jumlah iterasi yang lebih banyak dari 10000 kali. Selain itu, jumlah iterasi yang makin banyak menyebabkan waktu komputasi juga makin lama sehingga disarankan agar tidak mempergunakan jumlah iterasi yang banyak jika sudah memperoleh hasil yang diinginkan.
2. Probabilitas mutasi yang makin besar cenderung memberikan hasil yang lebih baik. Hal ini dapat dilihat dari makin besarnya nilai *fitness* yang

didapatkan dari tiap kombinasi setelah melakukan beberapa kali iterasi. Walaupun sebenarnya probabilitas mutasi yang lebih besar belum tentu bisa menjamin hasil yang lebih baik. Probabilitas mutasi tidak berpengaruh terhadap lamanya waktu komputasi.

Setelah menjalankan program dengan berbagai kombinasi parameter dapat diambil kesimpulan bahwa jadwal yang optimal dan memiliki nilai *fitness* terbaik hanya dapat diperoleh dengan kombinasi ukuran populasi 16, probabilitas mutasi 0.5, dan maksimum iterasi 10000 kali. Program ini juga telah dicoba dijalankan dengan menggunakan perangkat komputer yang memiliki spesifikasi yang berbeda-beda. Hasil yang didapatkan dari program tidak berbeda sama sekali, bahkan waktu komputasi juga tidak berbeda sehingga disimpulkan bahwa program ini dapat menghasilkan jadwal optimal yang sama dimanapun asalkan perangkat yang digunakan memenuhi spesifikasi minimum yang dibutuhkan program agar dapat dijalankan.

4.4.3 Analisis Hasil

Pada tahap pengolahan data dengan cara menjalankan program yang telah dibuat sebelumnya, input data yang digunakan mengalami perubahan karena input data mula-mula tidak dapat menghasilkan jadwal yang layak dan optimal. Input data yang diubah adalah matrik $KT_{|K| \times |T|}$ yang semula bernilai 0 dan 1, diubah menjadi bernilai 1 untuk semua mata kuliah pada semua slot-waktu. Hal ini dikarenakan semua mata kuliah yang diselenggarakan untuk mahasiswa tahun ke-3 memiliki beban studi sebesar 3 sks, yang berarti semua mata kuliah tersebut seharusnya dijadwalkan pada slot-waktu 2 jam 30 menit. Padahal slot-waktu yang disediakan selama 2 jam 30 menit hanya sedikit sehingga banyak mata kuliah yang terkena penalti untuk masalah slot-waktu.

Saat input data yang dimasukkan adalah matrik $KT_{|K| \times |T|}$ bernilai 0 dan 1, jadwal yang dihasilkan memiliki total penalti kendala *hard* sebesar 78 dan total penalti kendala *soft* sebesar 7. Dari sini dapat dilihat bahwa banyak sekali kendala *hard* yang dilanggar dibandingkan dengan kendala *soft*. Padahal jadwal yang ingin dicapai adalah jadwal ideal dimana jumlah penalti kendala *hard* dan kendala *soft* bernilai 0. Meskipun sebenarnya hal ini sulit dicapai dengan menggunakan

metode algoritma genetika sehingga solusi akan dianggap optimal jika jumlah penalti kendala *hard* dan *soft* sudah paling minimal dari jumlah iterasi yang ditentukan. Sehingga jadwal yang dihasilkan dengan input matrik $KT_{|K| \times |T|}$ bernilai 0 dan 1 tidak dianggap layak.

Akibat hal ini dilakukan iterasi ulang dengan mengubah input matrik $KT_{|K| \times |T|}$ yang semula bernilai 0 dan 1, diubah menjadi bernilai 1 untuk semua mata kuliah pada semua slot-waktu. Dengan mengubah matrik $KT_{|K| \times |T|}$ seperti ini berarti semua mata kuliah dianggap dapat diselenggarakan pada slot-waktu manapun. Setelah program dijalankan dengan menggunakan input matrik $KT_{|K| \times |T|}$ yang baru didapatkan jadwal baru dengan jumlah penalti *hard* sebesar 0 dan penalti *soft* sebesar 8. Jadwal baru yang dihasilkan dikatakan layak karena telah memenuhi fungsi kendala fk_1 , fk_2 dan fk_3 dengan nilai penalti sebesar 0.

Faktor yang menyebabkan jadwal tidak dapat menjadi ideal seperti yang diinginkan semula, dimana total kendala *hard* dan *soft* bernilai 0, adalah banyaknya jumlah mata kuliah yang diselenggarakan untuk mahasiswa tahun ke-2. Selain itu, hampir semua dari mata kuliah tersebut yang membutuhkan kelas dengan kapasitas di atas 50 sehingga menyebabkan program mengalami kesulitan mencari kombinasi yang tepat. Hal ini membuat adanya mata kuliah yang masih terkena penalti gk_1 dan gk_2 karena program terpaksa mengalokasikan mata kuliah dengan melanggar kendala *soft* agar jadwal yang terbentuk dapat merupakan solusi layak yaitu total penalti kendala *hard* bernilai 0.

Setelah program dijalankan dengan menggunakan input data 6 ruang kelas ternyata menghasilkan krosomom jadwal yang memiliki total penalti kendala *hard* 74 dan penalti kendala *soft* 7 dengan nilai *fitness* sebesar 0.012195 dan waktu komputasi selama 11 menit 8 detik. Jadi, dapat disimpulkan bahwa dengan menggunakan parameter yang sama, ternyata penjadwalan dengan input 6 ruangan memberikan hasil yang sangat buruk bahkan jadwal tidak layak, namun memberikan waktu komputasi yang lebih cepat.

BAB 5 KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil pembuatan model penjadwalan dengan menggunakan metode Algoritma Genetika dengan bantuan perangkat lunak MATLAB dapat disimpulkan beberapa hal di bawah ini:

- Penjadwalan mata kuliah pada perguruan tinggi menggunakan metode algoritma genetika ini menghasilkan jadwal baru yang layak dengan nilai *fitness* terbaik. Jadwal yang dihasilkan mendekati kondisi ideal yang diinginkan dengan jumlah penalti kendala *hard* sebesar 0 dan jumlah penalti kendala *soft* sebesar 8.
- Setelah melakukan analisis parameter bahkan menjalankan program dengan perangkat komputer yang berbeda-beda tetap memberikan hasil yang sama. Hal ini berarti bahwa program yang dibuat dapat digunakan dimanapun asalkan perangkat komputer yang digunakan memiliki spesifikasi minimum seperti yang dibutuhkan oleh perangkat lunak MATLAB untuk dapat dijalankan.
- Dalam masalah penjadwalan mata kuliah di DTI, memberikan jadwal yang tidak layak untuk jumlah ruangan 6
- Program yang dibuat dapat diimplementasikan untuk masalah penjadwalan mata kuliah pada perguruan tinggi manapun, asalkan memiliki kendala yang sama, dengan mengubah input data sesuai dengan keadaan sebenarnya.
- Program penjadwalan yang telah dibuat belum memperhatikan keterbatasan jadwal mengajar dosen. Namun, masih dapat diperbaiki lagi dengan menambahkan input matrik hubungan antara waktu dengan dosen. Selain itu, kode program untuk kendala ini juga harus dibuat dan ditambahkan ke dalam program utama yang telah dibuat.

5.2 Saran

Ada beberapa saran yang dapat diberikan kepada Fakultas Teknik UI untuk memperbaiki kondisi penjadwalan mata kuliah yang ada, yaitu:

- Jumlah ruangan dengan kapasitas 50 dan 70 dapat diperbanyak dengan menggabungkan 2 ruang kelas dengan kapasitas 20. Hal ini dengan mempertimbangkan banyaknya jumlah mata kuliah yang memerlukan kelas dengan kapasitas 50 dan 70 dibandingkan mata kuliah dengan kapasitas 20.
- Menambah jumlah tenaga pengajar agar resiko mata kuliah yang bentrok dikarenakan diajar oleh dosen yang sama dapat diminimalisasi.
- Mengalokasikan slot-waktu atau ruangan yang terpisah untuk mahasiswa S1 reguler dengan tingkat pendidikan lain seperti S2 atau Sarjana Ekstensi agar lebih memudahkan proses penjadwalan mata kuliah.
- Membuat suatu *database* yang sistematis yang dapat diubah dengan cepat sesuai dengan keadaan sebenarnya.

DAFTAR REFERENSI

- Abdullah, S., Hamza, T., & B. McCollum. (2009). *A Tabu-based Memetic Approach to the Examination Timetabling Problem*. The Second International Timetabling Competition.
- Balázs K., János, B., & László, T. K. (2009). *Comparative Analysis of Various Evolutionary and Memetic Algorithms*. 10th International Symposium of Hungarian Researchers on Computational Intelligence and Informatics, 193-205.
- Burke, K. Edmund, & Alistair, J. Smith.(n.d). *A Memetic Algorithm for the Maintenance Scheduling Problem*. In Proceedings of the International Conference on Neural Information Processing and Intelligent Information Systems Vol 1, Springer, ISBN: 981-3083-64-6, 469–473.
- Chinneck, John, W. (2001). *Practical Optimization: a Gentle Introduction*. <http://www.carleton.ca/faculty/chinneck/po.html>
- Drake, A.E. & Marks, R.E.(1998). *Genetic Algorithms in Economic and Finance Forecasting Stock Market Prices and Foreign Exchange*.
- Garg, Poonam. (2009). *A Comparison between Memetic Algorithm and Genetic Algorithm for the Cryptanalysis of Simplified Data Encryption Standard Algorithm*. International Journal of Network Security & Its Applications (IJNSA), Vol.1, No 1, April 2009.
- Herroelen, willy. (2005). *Production and Operation Management*. Winter 2005; 14, 4; ABI/INFORM Global hal. 413.
- Jat, Sadaf, N., & Shengxiang, Yang. (2008). *A Memetic Algorithm for the University Course Timetabling Problem*. 2008 20th IEEE International Conference on Tools with Artificial Intelligence, DOI 10.1109/ICTAI.2008.126.
- Karabati, Selcuk & Panos, Kouvelis. (1996). *Optimization Applications in Scheduling Theory*. Journal of Global Optimization 9:223-226.
- Lismanto. (2008). *Penjadwalan Kuliah dengan Algoritma Memetika*. Universitas Indonesia, Fakultas Matematika dan Ilmu Pengetahuan, Departemen Matematika.

- McCollum, B., et al. (2009). *Setting the Research Agenda in Automated Timetabling*. The Second International Timetabling Competition.
- Mostaco, P., & Carlos, C. (n.d). *A Gentle Introduction to Memetic Algorithms*. Computer Science Department, University of Newcastle, School of Electrical Engineering & Computer Science, Building EA, University Drive, Callghan NSW 2308, Australia. <http://www.cs.newcastle.edu.au/~nbi>
- Obitko, M. (1998). *Introduction to Genetic Algorithm*. <http://www.obitko.com/tutorials/genetic-algorithms/>
- Özcan, E. & Alpay, A. (2003). *A Memetic Algorithm for Solving a Timetabling Problem: An Incremental Strategy*. Computer Engineering Department, Istanbul, Turkey.
- Schwiegelshohn, Uwe. (2004, Summer Term). *Scheduling Problems and Solutions*. Computer Engineering Institute, University Dortmund.
- Sri, Kusumadewi, & Hari, P. (2005). *Penyelesaian Masalah Optimasi Menggunakan Teknik-teknik Heuristik*. Graha Ilmu, Yogyakarta, hal. 233.
- Suyanto. (2005). *Algoritma Genetika dalam MATLAB*. Yogyakarta
- Wook Chang Ahn. (2006). *Advances in Evolutionary Algorithms*. Student in Computational Intelligence, Vol. 18. Republic of Korea:Springer. Hal.9. www.mathworks.com

Lampiran 1. Matrik KK (Matrik Hubungan Antar Mata Kuliah)

Mata Kuliah	Sem	SKS	Notasi																											
				1								3																		
				Kalkulus A	Kalkulus B	Kalkulus A'	Kalkulus B'	Statistik & Probabilitas A	Statistik & Probabilitas B	Pengantar Teknik Industri A	Pengantar Teknik Industri B	Fisika Dasar 2A (Gelombang)	Fisika Dasar 2B (Gelombang)	Fisika Dasar 2A (Optik)	Fisika Dasar 2B' (Optik)	Kimia Dasar A	Kimia Dasar B	Akuntansi dan Biaya A	Akuntansi dan Biaya B	Fasilitas Industri A	Fasilitas Industri B	Psikologi Industri A	Psikologi Industri B	Ekonomi Teknik A	Ekonomi Teknik B	Programa Linier A	Programa Linier B	Mekanika Teknik A	Mekanika Teknik B	
				1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	
Kalkulus A	1	4	1	0	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Kalkulus B	1	4	2	1	0	1	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Kalkulus A'	1	4	3	1	1	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Kalkulus B'	1	4	4	1	1	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Statistik & Probabilitas A	1	3	5	1	0	1	0	0	1	1	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1	0	0	0
Statistik & Probabilitas B	1	3	6	0	1	0	1	1	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1	0	0	0
Pengantar Teknik Industri A	1	2	7	1	0	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Pengantar Teknik Industri B	1	2	8	0	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Fisika Dasar 2A (Gelombang)	3	4	9	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
Fisika Dasar 2B (Gelombang)	3	4	10	0	0	0	0	0	0	0	0	1	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
Fisika Dasar 2A (Optik)	3	4	11	0	0	0	0	0	0	0	0	1	0	0	1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
Fisika Dasar 2B' (Optik)	3	4	12	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
Kimia Dasar A	3	2	13	0	0	0	0	0	0	0	0	1	0	1	0	0	1	1	0	1	0	1	0	1	0	1	0	1	0	1
Kimia Dasar B	3	2	14	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0	1	0	1	0	1	0	1	0	1	0	1	0
Akuntansi dan Biaya A	3	2	15	0	0	0	0	0	1	1	0	1	0	1	0	1	0	0	1	1	0	1	0	1	0	1	0	1	0	1
Akuntansi dan Biaya B	3	2	16	0	0	0	0	0	1	1	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
Fasilitas Industri A	3	2	17	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1	0	0	1	1	0	1	0	1	0	1	0	1
Fasilitas Industri B	3	2	18	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
Psikologi Industri A	3	2	19	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
Psikologi Industri B	3	2	20	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
Ekonomi Teknik A	3	2	21	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
Ekonomi Teknik B	3	2	22	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
Programa Linier A	3	3	23	0	0	0	0	1	1	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
Programa Linier B	3	3	24	0	0	0	0	1	1	0	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
Mekanika Teknik A	3	2	25	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
Mekanika Teknik B	3	2	26	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
Faktor Manusia Dalam Rekayasa Disain A	5	3	27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Faktor Manusia Dalam Rekayasa Disain B	5	3	28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Perencanaan & Pengendalian Produksi A	5	3	29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Perencanaan & Pengendalian Produksi B	5	3	30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Pemodelan Sistem + Praktikum A	5	3	31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Pemodelan Sistem + Praktikum B	5	3	32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Pemasaran Industri A	5	3	33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Pemasaran Industri B	5	3	34	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Sistem Informasi A	5	3	35	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0
Sistem Informasi B	5	3	36	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
Total Quality Management (TQM)	5	3	37	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0
Analisa Multivariat	5	3	38	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Manajemen Pengetahuan	5	3	39	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Sistem Keuangan Perusahaan	5	3	40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Manajemen Energi	5	3	41	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Lampiran 1. Matrik KK (Matrik Hubungan Antar Mata Kuliah) (lanjutan)

Mata Kuliah	Sem	SKS	Notasi	Perancangan Proses A	Perancangan Proses B	Manajemen Proyek A	Manajemen Proyek B	Kapita Selektia Industri	Sistem Pemeliharaan A	Sistem Pemeliharaan B	Manajemen Teknologi A	Manajemen Teknologi B	Teori Keputusan	Sistem Distribusi dan Logistik	Keterampilan Interpersonal	Berfikir Sistem	Analisa Kelayakan Industri	Ekonomi Industri
				42	43	44	45	46	47	48	49	50	51	52	53	54	55	56
Kalkulus A	1	4	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Kalkulus B		4	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Kalkulus A'		4	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Kalkulus B'		4	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Statistik & Probabilitas A		3	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Statistik & Probabilitas B		3	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Pengantar Teknik Industri A		2	7	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
Pengantar Teknik Industri B		2	8	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
Fisika Dasar 2A (Gelombang)		4	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Fisika Dasar 2B (Gelombang)		4	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Fisika Dasar 2A (Optik)	4	11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Fisika Dasar 2B' (Optik)	4	12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Kimia Dasar A	3	2	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Kimia Dasar B		2	14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Akuntansi dan Biaya A		2	15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Akuntansi dan Biaya B		2	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Fasilitas Industri A		2	17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Fasilitas Industri B		2	18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Psikologi Industri A		2	19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
Psikologi Industri B		2	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
Ekonomi Teknik A		2	21	0	0	1	1	0	0	0	0	0	1	0	0	0	0	0
Ekonomi Teknik B		2	22	0	0	1	1	0	0	0	0	0	1	0	0	0	0	0
Programa LinierA	5	3	23	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
Programa LinierB		3	24	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
Mekanika Teknik A		2	25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Mekanika Teknik B		2	26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Faktor Manusia Dalam Rekayasa Disain A		3	27	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0
Faktor Manusia Dalam Rekayasa Disain B		3	28	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0
Perencanaan & Pengendalian Produksi A		3	29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Perencanaan & Pengendalian Produksi B		3	30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Pemodelan Sistem + Praktikum A		3	31	0	0	0	0	0	0	0	1	1	0	0	1	1	0	0
Pemodelan Sistem + Praktikum B		3	32	0	0	0	0	0	0	0	1	1	0	0	1	1	0	0
Pemasaran Industri A	5	3	33	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
Pemasaran Industri B		3	34	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
Sistem Informasi A		3	35	0	0	1	1	0	0	0	0	0	1	0	0	0	0	0
Sistem Informasi B		3	36	0	0	1	1	0	0	0	0	0	1	0	0	0	0	0
Total Quality Management (TQM)		3	37	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
Analisa Multivariat		3	38	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0
Manajemen Pengetahuan		3	39	0	0	0	0	0	0	0	1	1	0	0	0	1	0	0
Sistem Keuangan Perusahaan		3	40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Manajemen Energi		3	41	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0

Lampiran 2. Matrik KT (Hubungan Antara Mata Kuliah dengan Slot Waktu)

Mata Kuliah	Sem	SKS	Notasi	Senin				Selasa			
				08.00-09.50	10.00-11.50	13.00-15.30	16.00-17.50	08.00-09.50	10.00-12.30	13.30-16.00	16.00-17.50
				1	2	3	4	5	6	7	8
Kalkulus A	1	4	1	1	1	1	1	1	1	1	1
Kalkulus B		4	2	1	1	1	1	1	1	1	1
Kalkulus A'		4	3	1	1	1	1	1	1	1	1
Kalkulus B'		4	4	1	1	1	1	1	1	1	1
Statistik & Probabilitas A		3	5	1	1	1	1	1	1	1	1
Statistik & Probabilitas B		3	6	1	1	1	1	1	1	1	1
Pengantar Teknik Industri A		2	7	1	1	1	1	1	1	1	1
Pengantar Teknik Industri B		2	8	1	1	1	1	1	1	1	1
Fisika Dasar 2A (Gelombang)	3	4	9	1	1	1	1	1	1	1	1
Fisika Dasar 2B (Gelombang)		4	10	1	1	1	1	1	1	1	1
Fisika Dasar 2A (Optik)		4	11	1	1	1	1	1	1	1	1
Fisika Dasar 2B' (Optik)		4	12	1	1	1	1	1	1	1	1
Kimia Dasar A		2	13	1	1	1	1	1	1	1	1
Kimia Dasar B		2	14	1	1	1	1	1	1	1	1
Akuntansi dan Biaya A		2	15	1	1	1	1	1	1	1	1
Akuntansi dan Biaya B		2	16	1	1	1	1	1	1	1	1
Fasilitas Industri A		2	17	1	1	1	1	1	1	1	1
Fasilitas Industri B		2	18	1	1	1	1	1	1	1	1
Psikologi Industri A		2	19	1	1	1	1	1	1	1	1
Psikologi Industri B		2	20	1	1	1	1	1	1	1	1
Ekonomi Teknik A		2	21	1	1	1	1	1	1	1	1
Ekonomi Teknik B		2	22	1	1	1	1	1	1	1	1
Programa LinierA		3	23	1	1	1	1	1	1	1	1
Programa LinierB		3	24	1	1	1	1	1	1	1	1
Mekanika Teknik A	2	25	1	1	1	1	1	1	1	1	
Mekanika Teknik B	2	26	1	1	1	1	1	1	1	1	

Lampiran 2. Matrik KT (Hubungan antara Mata Kuliah dengan Slot Waktu) (lanjutan)

Mata Kuliah	Sem	SKS	Notasi	Rabu				Kamis				Jumat		
				08.00-09.50	10.00-11.50	13.00-15.30	16.00-17.50	08.00-09.50	10.00-12.30	13.30-16.00	16.00-17.50	07.30-09.20	09.30-11.20	13.30-16.00
				9	10	11	12	13	14	15	16	17	18	19
Kalkulus A	1	4	1	1	1	1	1	1	1	1	1	1	1	1
Kalkulus B		4	2	1	1	1	1	1	1	1	1	1	1	1
Kalkulus A'		4	3	1	1	1	1	1	1	1	1	1	1	1
Kalkulus B'		4	4	1	1	1	1	1	1	1	1	1	1	1
Statistik & Probabilitas A		3	5	1	1	1	1	1	1	1	1	1	1	1
Statistik & Probabilitas B		3	6	1	1	1	1	1	1	1	1	1	1	1
Pengantar Teknik Industri A		2	7	1	1	1	1	1	1	1	1	1	1	1
Pengantar Teknik Industri B		2	8	1	1	1	1	1	1	1	1	1	1	1
Fisika Dasar 2A (Gelombang)	3	4	9	1	1	1	1	1	1	1	1	1	1	1
Fisika Dasar 2B (Gelombang)		4	10	1	1	1	1	1	1	1	1	1	1	1
Fisika Dasar 2A (Optik)		4	11	1	1	1	1	1	1	1	1	1	1	1
Fisika Dasar 2B' (Optik)		4	12	1	1	1	1	1	1	1	1	1	1	1
Kimia Dasar A		2	13	1	1	1	1	1	1	1	1	1	1	1
Kimia Dasar B		2	14	1	1	1	1	1	1	1	1	1	1	1
Akuntansi dan Biaya A		2	15	1	1	1	1	1	1	1	1	1	1	1
Akuntansi dan Biaya B		2	16	1	1	1	1	1	1	1	1	1	1	1
Fasilitas Industri A		2	17	1	1	1	1	1	1	1	1	1	1	1
Fasilitas Industri B		2	18	1	1	1	1	1	1	1	1	1	1	1
Psikologi Industri A		2	19	1	1	1	1	1	1	1	1	1	1	1
Psikologi Industri B		2	20	1	1	1	1	1	1	1	1	1	1	1
Ekonomi Teknik A		2	21	1	1	1	1	1	1	1	1	1	1	1
Ekonomi Teknik B		2	22	1	1	1	1	1	1	1	1	1	1	1
Programa LinierA		3	23	1	1	1	1	1	1	1	1	1	1	1
Programa LinierB		3	24	1	1	1	1	1	1	1	1	1	1	1
Mekanika Teknik A		2	25	1	1	1	1	1	1	1	1	1	1	1
Mekanika Teknik B		2	26	1	1	1	1	1	1	1	1	1	1	1

Lampiran 2. Matrik KT (Hubungan antara Mata Kuliah dengan Slot Waktu) (lanjutan)

Mata Kuliah	Sem	SKS	Notasi	Senin				Selasa			
				08.00-09.50	10.00-11.50	13.00-15.30	16.00-17.50	08.00-09.50	10.00-12.30	13.30-16.00	16.00-17.50
				1	2	3	4	5	6	7	8
Faktor Manusia Dalam Rekayasa Disain A	5	3	27	1	1	1	1	1	1	1	1
Faktor Manusia Dalam Rekayasa Disain B		3	28	1	1	1	1	1	1	1	1
Perencanaan & Pengendalian Produksi A		3	29	1	1	1	1	1	1	1	1
Perencanaan & Pengendalian Produksi B		3	30	1	1	1	1	1	1	1	1
Pemodelan Sistem + Praktikum A		3	31	1	1	1	1	1	1	1	1
Pemodelan Sistem + Praktikum B		3	32	1	1	1	1	1	1	1	1
Pemasaran Industri A		3	33	1	1	1	1	1	1	1	1
Pemasaran Industri B		3	34	1	1	1	1	1	1	1	1
Sistem Informasi A		3	35	1	1	1	1	1	1	1	1
Sistem Informasi B		3	36	1	1	1	1	1	1	1	1
Total Quality Management (TQM)		3	37	1	1	1	1	1	1	1	1
Analisa Multivariat		3	38	1	1	1	1	1	1	1	1
Manajemen Pengetahuan		3	39	1	1	1	1	1	1	1	1
Sistem Keuangan Perusahaan		3	40	1	1	1	1	1	1	1	1
Manajemen Energi		3	41	1	1	1	1	1	1	1	1
Perancangan Proses A	7	2	42	1	1	1	1	1	1	1	1
Perancangan Proses B		2	43	1	1	1	1	1	1	1	1
Manajemen Proyek A		3	44	1	1	1	1	1	1	1	1
Manajemen Proyek B		3	45	1	1	1	1	1	1	1	1
Kapita Selekt Industri		2	46	1	1	1	1	1	1	1	1
Sistem Pemeliharaan A		3	47	1	1	1	1	1	1	1	1
Sistem Pemeliharaan B		3	48	1	1	1	1	1	1	1	1
Manajemen Teknologi A		3	49	1	1	1	1	1	1	1	1
Manajemen Teknologi B		3	50	1	1	1	1	1	1	1	1
Teori Keputusan		3	51	1	1	1	1	1	1	1	1
Sistem Distribusi dan Logistik		3	52	1	1	1	1	1	1	1	1
Keterampilan Interpersonal		3	53	1	1	1	1	1	1	1	1
Berfikir Sistem		3	54	1	1	1	1	1	1	1	1
Analisa Kelayakan Industri		3	55	1	1	1	1	1	1	1	1
Ekonomi Industri		3	56	1	1	1	1	1	1	1	1

Lampiran 2. Matrik KT (Hubungan antara Mata Kuliah dengan Slot Waktu) (lanjutan)

Mata Kuliah	Sem	SKS	Notasi	Rabu				Kamis				Jumat		
				08.00-09.50	10.00-11.50	13.00-15.30	16.00-17.50	08.00-09.50	10.00-12.30	13.30-16.00	16.00-17.50	07.30-09.20	09.30-11.20	13.30-16.00
				9	10	11	12	13	14	15	16	17	18	19
Faktor Manusia Dalam Rekeyasa Disain A		3	27	1	1	1	1	1	1	1	1	1	1	1
Faktor Manusia Dalam Rekeyasa Disain B		3	28	1	1	1	1	1	1	1	1	1	1	1
Perencanaan & Pengendalian Produksi A		3	29	1	1	1	1	1	1	1	1	1	1	1
Perencanaan & Pengendalian Produksi B		3	30	1	1	1	1	1	1	1	1	1	1	1
Pemodelan Sistem + Praktikum A		3	31	1	1	1	1	1	1	1	1	1	1	1
Pemodelan Sistem + Praktikum B		3	32	1	1	1	1	1	1	1	1	1	1	1
Pemasaran Industri A		3	33	1	1	1	1	1	1	1	1	1	1	1
Pemasaran Industri B	5	3	34	1	1	1	1	1	1	1	1	1	1	1
Sistem Informasi A		3	35	1	1	1	1	1	1	1	1	1	1	1
Sistem Informasi B		3	36	1	1	1	1	1	1	1	1	1	1	1
Total Quality Management (TQM)		3	37	1	1	1	1	1	1	1	1	1	1	1
Analisa Multivariat		3	38	1	1	1	1	1	1	1	1	1	1	1
Manajemen Pengetahuan		3	39	1	1	1	1	1	1	1	1	1	1	1
Sistem Keuangan Perusahaan		3	40	1	1	1	1	1	1	1	1	1	1	1
Manajemen Energi		3	41	1	1	1	1	1	1	1	1	1	1	1
Perancangan Proses A		2	42	1	1	1	1	1	1	1	1	1	1	1
Perancangan Proses B		2	43	1	1	1	1	1	1	1	1	1	1	1
Manajemen Proyek A		3	44	1	1	1	1	1	1	1	1	1	1	1
Manajemen Proyek B		3	45	1	1	1	1	1	1	1	1	1	1	1
Kapita Selekt Industri		2	46	1	1	1	1	1	1	1	1	1	1	1
Sistem Pemeliharaan A		3	47	1	1	1	1	1	1	1	1	1	1	1
Sistem Pemeliharaan B		3	48	1	1	1	1	1	1	1	1	1	1	1
Manajemen Teknologi A	7	3	49	1	1	1	1	1	1	1	1	1	1	1
Manajemen Teknologi B		3	50	1	1	1	1	1	1	1	1	1	1	1
Teori Keputusan		3	51	1	1	1	1	1	1	1	1	1	1	1
Sistem Distribusi dan Logistik		3	52	1	1	1	1	1	1	1	1	1	1	1
Keterampilan Interpersonal		3	53	1	1	1	1	1	1	1	1	1	1	1
Berfikir Sistem		3	54	1	1	1	1	1	1	1	1	1	1	1
Analisa Kelayakan Industri		3	55	1	1	1	1	1	1	1	1	1	1	1
Ekonomi Industri		3	56	1	1	1	1	1	1	1	1	1	1	1

Lampiran 3. Matrik KM (Hubungan antara Mata Kuliah dengan Mahasiswa)

KELOMPOK MAHASISWA											
NO	MK	SMTR	SKS	Notasi	Tingkat 1		Tingkat 2		Tingkat 3	Tingkat 4	
					1	2	3	4	5	6	
1	Kalkulus A	1	4	1	35	0	20	0	0	0	
2	Kalkulus B		4	2	0	55	0	0	0	0	
3	Kalkulus A'		4	3	35	0	20	0	0	0	
4	Kalkulus B'		4	4	0	55	0	0	0	0	
5	Statistik & Probabilitas A		3	5	35	0	20	0	0	0	
	Statistik & Probabilitas B		3	6	0	55	0	0	0	0	
6	Pengantar Teknik Industri A		2	7	35	0	20	0	0	0	
	Pengantar Teknik Industri B		2	8	0	55	0	0	0	0	
7	Fisika Dasar 2A (Gelombang)	3	4	9	0	0	35	0	20	0	
8	Fisika Dasar 2B (Gelombang)		4	10	0	0	0	55	0	0	
9	Fisika Dasar 2A (Optik)		4	11	0	0	35	0	20	0	
10	Fisika Dasar 2B' (Optik)		4	12	0	0	0	55	0	0	
11	Kimia Dasar A		2	13	0	0	35	0	20	0	
	Kimia Dasar B		2	14	0	0	0	55	0	0	
	Akuntansi dan Biaya A		2	15	0	0	35	0	20	0	
	Akuntansi dan Biaya B		2	16	0	0	0	55	0	0	
13	Fasilitas Industri A		2	17	0	0	35	0	20	0	
	Fasilitas Industri B		2	18	0	0	0	55	0	0	
14	Psikologi Industri A		2	19	0	0	35	0	20	0	
	Psikologi Industri B		2	20	0	0	0	55	0	0	
15	Ekonomi Teknik A		2	21	0	0	35	0	20	0	
	Ekonomi Teknik B		2	22	0	0	0	55	0	0	
16	Programa LinierA		3	23	0	0	35	0	20	0	
17	Programa LinierB		3	24	0	0	0	55	0	0	
18	Mekanika Teknik A	2	25	0	0	35	0	20	0		
	Mekanika Teknik B	2	26	0	0	0	55	0	0		
19	Faktor Manusia Dalam Rekayasa Disain A	5	3	27	0	0	0	0	35	20	
	Faktor Manusia Dalam Rekayasa Disain B		3	28	0	0	0	0	55	0	
20	Perencanaan & Pengendalian Produksi A		3	29	0	0	0	0	35	20	
	Perencanaan & Pengendalian Produksi B		3	30	0	0	0	0	55	0	
21	Pemodelan Sistem + Praktikum A		3	31	0	0	0	0	35	20	
	Pemodelan Sistem + Praktikum B		3	32	0	0	0	0	55	0	
22	Pemasaran Industri A		3	33	0	0	0	0	35	20	
	Pemasaran Industri B		3	34	0	0	0	0	55	0	
23	Sistem Informasi A		3	35	0	0	0	0	35	20	
	Sistem Informasi B		3	36	0	0	0	0	55	0	
29	Total Quality Management (TQM)		3	37	0	0	0	0	40	0	
32	Analisa Multivariat		3	38	0	0	0	0	20	20	
34	Manajemen Pengetahuan		3	39	0	0	0	0	40	0	
37	Sistem Keuangan Perusahaan		3	40	0	0	0	0	40	0	
38	Manajemen Energi		3	41	0	0	0	0	40	0	
24	Perancangan Proses A		7	2	42	0	0	0	0	0	45
	Perancangan Proses B	2		43	0	0	0	0	0	0	45
25	Manajemen Proyek A	3		44	0	0	0	0	0	0	45
	Manajemen Proyek B	3		45	0	0	0	0	0	0	45
26	Kapita Selekt Industri	2		46	0	0	0	0	0	0	90
27	Sistem Pemeliharaan A	3		47	0	0	0	0	0	0	45
	Sistem Pemeliharaan B	3		48	0	0	0	0	0	0	45
28	Manajemen Teknologi A	3		49	0	0	0	0	0	0	45
	Manajemen Teknologi B	3		50	0	0	0	0	0	0	45
	Teori Keputusan	3		51	0	0	0	0	0	0	40
30	Sistem Distribusi dan Logistik	3		52	0	0	0	0	0	0	40
31	Keterampilan Interpersonal	3		53	0	0	0	0	0	0	25
	Berfikir Sistem	3		54	0	0	0	0	0	0	40
33	Analisa Kelayakan Industri	3		55	0	0	0	0	0	0	40
	Ekonomi Industri	3		56	0	0	0	0	0	0	40

Lampiran 4. Source Code Program MATLAB Program Penjadwalan Mata Kuliah

```

% Ambil data
MatrixKK=xlsread('matrixKK', 'E4:BH59');
MatrixKT=xlsread('MatrixKT19', 'E4:W59');
MatrixKM=xlsread('Jadwal', 'F4:K59');
MatrixRuangan=xlsread('Kapasitas_Ruang', 'C4:C10');

% Step 1
MatrixKR=zeros(56,7);
for i=1:1:56
    mahasiswa=sum(MatrixKM(i,:));
    for j=1:1:7
        if mahasiswa<=MatrixRuangan(j)
            MatrixKR(i,j)=1;
        end
    end
end

%Step 2
MIB=zeros(56,56);
for i=1:1:56
    for j=1:1:56
        if i~=j
            mb=zeros(1,6);
            for k=1:1:6
                mb(1,k)=min(MatrixKM(i,k),MatrixKM(j,k));
            end
            MIB(i,j)=sum(mb);
            MIB(j,i)=MIB(i,j);
        end
    end
end

MTB=sum(MIB,2);
urutan=[1 MTB(1)];
for i=2:1:56
    urutan(i,:)=[i MTB(i)];
    for j=i:-1:2
        if urutan(j,2)>urutan(j-1,2)
            dummy=urutan(j-1,:);
            urutan(j-1,:)=urutan(j,:);
            urutan(j,:)=dummy;
        else
            break;
        end
    end
end

for z=1:1:16
    eval(['MJ', int2str(z), '--ones(7,19);']);
    eval(['KJ', int2str(z), '=zeros(7,56);']);
    i=1;

```

```

while i<=56
    calonruang=[0];
    for j=1:1:7
        if MatrixKR(urutan(i,1),j)==1
            calonruang=[calonruang j];
        end
    end
    calonruang=calonruang(2:end);

    calonslot=[0];
    for j=1:1:19
        if MatrixKT(urutan(i,1),j)==1
            calonslot=[calonslot j];
        end
    end
    calonslotmod=[calonslot(2:end); mod(calonslot(2:end),4)];

    for j=1:1:length(calonslotmod)
        if calonslotmod(2,j)==0
            calonslotmod(2,j)=4;
        end
    end

    CSM=calonslotmod(:,1);
    for j=2:1:length(calonslotmod);
        CSM(:,j)=calonslotmod(:,j);
        for k=j:-1:2
            if CSM(2,k)<CSM(2,k-1)
                dummy=CSM(:,k-1);
                CSM(:,k-1)=CSM(:,k);
                CSM(:,k)=dummy;
            else
                break;
            end
        end
    end

    stopper=0;
    while stopper==0;
        ruang=calonruang(ceil(rand*length(calonruang)));
        slot=CSM(1,1);
        if eval(['MJ', int2str(z), '(ruang,slot)==-1'])
            eval(['MJ', int2str(z),
                '(ruang,slot)=urutan(i);']);
            eval(['KJ', int2str(z), '(1:2,urutan(i))=[ruang;
                slot];']);
            i=i+1;
            stopper=1;
        else
            CSM=CSM(:,2:end);
        end
    end
end

```

```

    end
end

for z=1:1:16
    eval(['KJ', int2str(z), '=evifk1(MJ', int2str(z), ', KJ',
int2str(z), ', MatrixKK);'])

    eval(['KJ', int2str(z), '=evifk2(KJ', int2str(z), ',
MatrixKR);'])

    eval(['KJ', int2str(z), '=evifk3(KJ', int2str(z), ',
MatrixKT);'])

    eval(['KJ', int2str(z), '=evigk1(MJ', int2str(z), ', KJ',
int2str(z), ', MatrixKM);'])

    eval(['KJ', int2str(z), '=evigk2(KJ', int2str(z), ');'])

    fk(z)=eval(['sum([sum(KJ', int2str(z), '(3,:) sum(KJ',
int2str(z), '(4,:) sum(KJ', int2str(z), '(5,:)]);']);
    gk(z)=eval(['sum([sum(KJ', int2str(z), '(6,:) sum(KJ',
int2str(z), '(7,:)]);']);
    fitness(z)=1/(fk(z)+gk(z)+1);
end

[bank outlayer]=elitism(fitness);
% Memulai Algoritma GA
for iterasi=1:1:MaxIterasi
    for i=1:2
        kandidat1=ceil(rand*16);
        kandidat2=kandidat1;
        while kandidat2==kandidat1
            kandidat2=ceil(rand*16);
        end
        if bank(kandidat1,1)>bank(kandidat2,1)
            P(i)=bank(kandidat1,2);
        else
            P(i)=bank(kandidat2,2);
        end
    end
    end
    % crossover
    KJchild=zeros(7,56);
    for i=1:1:56
        if eval(['sum(KJ', int2str(P(1)), '(3:7,i)<sum(KJ',
int2str(P(2)), '(3:7,i)')])
            eval(['KJchild(1:2,i)=KJ' int2str(P(1)), '(1:2,i)');']);
        else
            eval(['KJchild(1:2,i)=KJ' int2str(P(2)), '(1:2,i)');']);
        end
    end
    end
    for i=2:1:56
        for j=1:1:i
            if KJchild(1:2,i)==KJchild(1:2,j)

```



```

                                eval(['KJchild(1:2,:)=KJ'           int2str(P(1)),
'(1:2,:);']);
                                break;
                                end
                                end
                                end

                                % Mutasi
                                if rand<0.5
                                    gen1=ceil(rand*56);
                                    gen2=gen1;
                                    while gen2==gen1
                                        gen2=ceil(rand*56);
                                    end
                                    dummy=KJchild(1:2,gen1);
                                    KJchild(1:2,gen1)=KJchild(1:2,gen2);
                                    KJchild(1:2,gen2)=dummy;
                                end

                                eval(['KJ', int2str(outlayer), '=KJchild;'])

                                % membuat MJ baru menggantikan outlayer
                                eval(['MJ', int2str(outlayer), '=-ones(7,19);'])
                                for i=1:1:56
                                    eval(['MJ', int2str(outlayer), '(KJ', int2str(outlayer),
'(1,i),KJ', int2str(outlayer), '(2,i))=i;'])
                                end

                                % menghitung fitness dari child
                                eval(['KJ', int2str(outlayer), '=evifk1(MJ', int2str(outlayer),
', KJ', int2str(outlayer), ', MatrixKK);'])

                                eval(['KJ', int2str(outlayer), '=evifk2(KJ', int2str(outlayer),
', MatrixKR);'])

                                eval(['KJ', int2str(outlayer), '=evifk3(KJ', int2str(outlayer),
', MatrixKT);'])

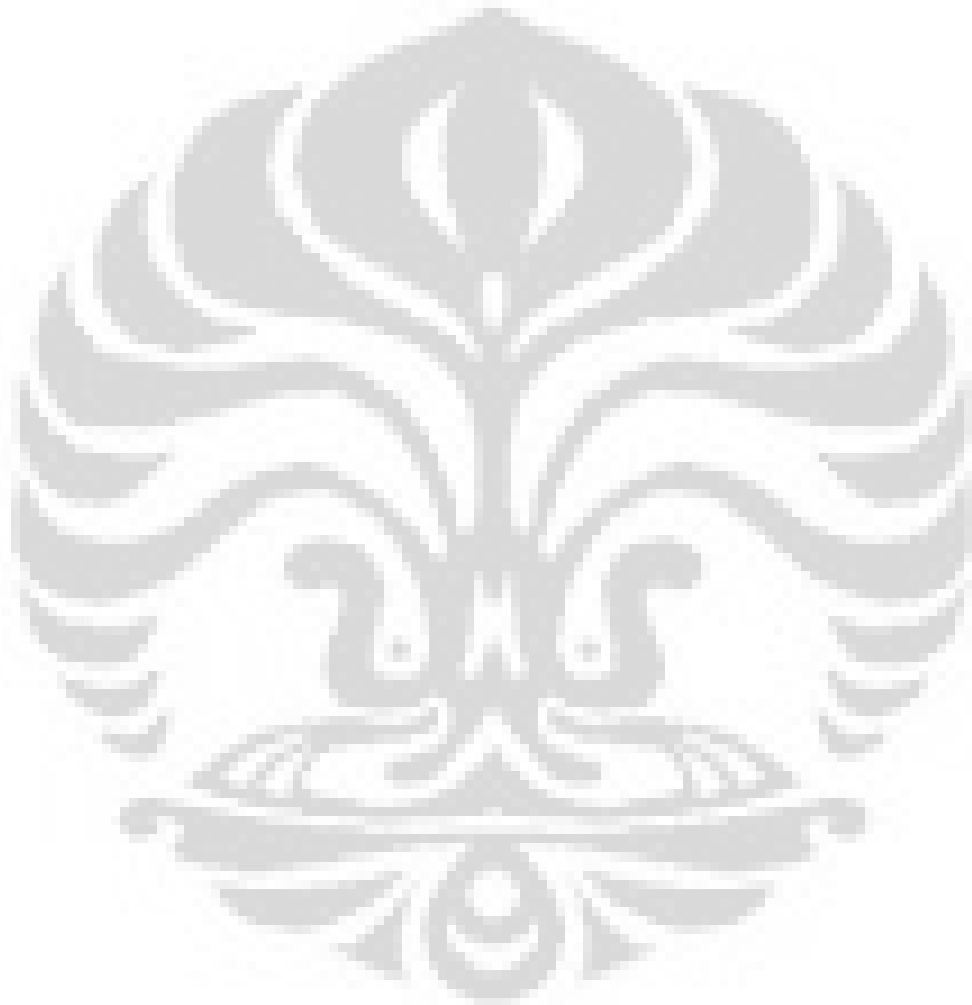
                                eval(['KJ', int2str(outlayer), '=evigk1(MJ', int2str(outlayer),
', KJ', int2str(outlayer), ', MatrixKM);'])

                                eval(['KJ', int2str(outlayer), '=evigk2(KJ', int2str(outlayer),
');'])
                                fk(outlayer)=eval(['sum([sum(KJ', int2str(outlayer), '(3,:)
sum(KJ', int2str(outlayer), '(4,:) sum(KJ', int2str(outlayer),
'(5,:)]);']);
                                gk(outlayer)=eval(['sum([sum(KJ', int2str(outlayer), '(6,:)
sum(KJ', int2str(outlayer), '(7,:)]);']);
                                fitness(outlayer)=1/(fk(outlayer)+gk(outlayer)+1);

                                [bank outlayer]=elitism(fitness);
                                F(iterasi)=bank(1,1);
                                end

```

```
JadwalOpt1=eval(['MJ', int2str(bank(1,2))]);  
JadwalOpt2=eval(['KJ', int2str(bank(1,2))]);  
PHT=fk(bank(1,2));  
PST=gk(bank(1,2));  
FitnessOpt=bank(1,1);
```



Lampiran 6. Jadwal Optimal Hasil 10000 Iterasi

Ruangan	Senin			
	08.00-09.50	10.00-11.50	13.00-15.30	16.00-17.50
K.108				
K.209	Perancangan Proses A Ir.Fauzia Dianawati,Msi (9) M4A			
K.211	Manajemen Teknologi B Akhmad Hidayatno,ST.,MBT (1) M4B	Ekonomi Industri Ir. Rahmat Nurcahyo,M.Eng.Sc. (14) M4		
K.206	Pemasaran Industri B Ir. Hj. Erlinda Muslim,MEE (8) M3B	Sistem Informasi A Ir.M.Dachyar,MSc (13) M3A	Sistem Keuangan Perusahaan Ir.Sri Bintang Pamungkas.,MSISE.,PhD (17) M3	Kalkulus A' Dosen MIPA (21) M1A
K.207	Ekonomi Teknik B Ir.M.Dachyar,MSc (13) M2B	Fasilitas Industri A Ir.Sri Bintang Pamungkas.,MSISE.,PhD (17) M2A		
GK.306	Perencanaan & Pengendalian Produksi A Ir.Yadrifil,Msc (16) M3A	Kimia Dasar B Dr.Ir. Dewi Tristantini,MT. (18) M2B	Teori Keputusan Ir. Isti Surjandari,MT.,MA.,PhD (11) M4	
GK.301	Psikologi Industri A Ir. Rahmat Nurcahyo,M.Eng.Sc. (14) M2A	Pemodelan Sistem + Praktikum B Armand Omar M.,ST.,MSc (13) M3B	Mekanika Teknik A Dosen Mesin (22) M2A	

Universitas Indonesia

Lampiran 6. Jadwal Optimal Hasil 10000 Iterasi
(lanjutan)

Ruangan	Selasa			
	08.00-09.50	10.00-12.30	13.30-16.00	16.00-17.50
K.108				
K.209				
K.211	Berfikir Sistem Akhmad Hidayatno,ST.,MBT (1) M4	Analisa Kelayakan Industri Ir. Djoko Sihono Gabriel,MT (7) M4		
K.206	Sistem Informasi B Ir.M.Dachyar,MSc (13) M3B	Fisika Dasar 2A (Gelombang) Ir. Dendi P. Ishak,MSIE (6) M2A		Pengantar Teknik Industri A Dr.Ir. T .Yuri. M.z.,MEngSc (15) M1A
K.207	Mekanika Teknik B Dosen Mesin (22) M2B	Kalkulus B' Dosen MIPA (21) M1B	Manajemen Teknologi A Akhmad Hidayatno,ST.,MBT (1) M4A	
GK.306	Statistik & Probabilitas B Arian Dhini,ST.,MT (4) M1B	Fisika Dasar 2B' (Optik) Dosen MIPA2 (Fisdas2 Optik) (20) M2B		Kimia Dasar A Dr.Ir. Dewi Tristantini,MT. (18) M2A
GK.301	Pemasaran Industri A Ir. Hj. Erlinda Muslim,MEE (8) M3A	Pemodelan Sistem + Praktikum A Armand Omar M.,ST.,MSc (13) M3A		

Universitas Indonesia

Lampiran 6. Jadwal Optimal Hasil 10000 Iterasi
(lanjutan)

Ruangan	Rabu			
	08.00-09.50	10.00-11.50	13.00-15.30	16.00-17.50
K.108				
K.209	Manajemen Proyek A Ir.M.Dachyar,MSc (13) M4A			
K.211				
K.206	Akuntansi dan Biaya A Arian Dhini,ST.,MT (4) M2A	Ekonomi Teknik A Ir.M.Dachyar,MSc (13) M2A	Fasilitas Industri B Ir.Sri Bintang Pamungkas.,MSISE.,PhD (17) M2B	
K.207	Perancangan Proses B Ir.Fauzia Dianawati,Msi (9) M4B	Statistik & Probabilitas A Arian Dhini,ST.,MT (4) M1A		
GK.306	Faktor Manusia Dalam Rekayasa Disain B Ir. Boy Nurtjahyo,MSIE (5) M3B	Manajemen Pengetahuan Armand Omar M.,ST.,MSc (3) M3		
GK.301	Kalkulus A Dosen MIPA (21) M1A	Kapita Seleka Industri Ir. Isti Surjandari,MT.,MA.,PhD (11) M4		

Universitas Indonesia

Lampiran 6. Jadwal Optimal Hasil 10000 Iterasi
(lanjutan)

Ruangan	Kamis			
	08.00-09.50	10.00-12.30	13.30-16.00	16.00-17.50
K.108	Sistem Distribusi dan Logistik Ir.Amar Rachman,MEIM (2) M4			
K.209				
K.211				
K.206	Akuntansi dan Biaya B Arian Dhini,ST.,MT (4) M2B	Sistem Pemeliharaan A Ir. Boy Nurtjahyo,MSIE (5) M4A		
K.207	Fisika Dasar 2A (Optik) Dosen MIPA2 (Fisdas2 Optik) (20) M2A	Fisika Dasar 2B (Gelombang) Ir. Dendi P. Ishak,MSIE (6) M2B	Kalkulus B Dosen MIPA (21) M1B	
GK.306	Faktor Manusia Dalam Rekayasa Disain A Ir. Boy Nurtjahyo,MSIE (5) M3A	Analisa Multivariat Ir. Isti Surjandari,MT.,MA.,PhD (11) M3		
GK.301	Perencanaan & Pengendalian Produksi B Ir.Yadrifil,Msc (16) M3B	Manajemen Proyek B Ir.M.Dachyar,MSc (13) M4B		

Universitas Indonesia

Lampiran 6. Jadwal Optimal Hasil 10000 Iterasi
(lanjutan)

Ruangan	Jumat		
	07.30-09.20	09.30-11.20	13.30-16.00
K.108	Keterampilan Interpersonal Akhmad Hidayatno,ST.,MBT (1) M4		
K.209	Manajemen Energi Ir. Hj. Erlinda Muslim,MEE (8) M3		
K.211			
K.206	Sistem Pemeliharaan B Ir.Fauzia Dianawati,Msi (9) M4B		
K.207	Psikologi Industri B Ir. Rahmat Nurcahyo,M.Eng.Sc. (14) M2B	Programa Linier B Farizal.,PhD (10) M2B	
GK.306	Pengantar Teknik Industri B Dr.Ir. T .Yuri. M.z.,MEngSc (15) M1B	Total Quality Management (TQM) Dr.Ir. T .Yuri. M.z.,MEngSc (15) M3	
GK.301	Programa LinierA Ir.Amar Rachman,MEIM (2) M2A		

Universitas Indonesia