

**OPTIMASI PENJADWALAN MATA KULIAH DI
PERGURUAN TINGGI DENGAN MENGGUNAKAN
ALGORITMA MEMETIKA**

SKRIPSI

Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik

**YANTI NITRA SIANIPAR
0606077610**



**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 : Yanti Nitra Sianipar

NPM : 0606077610

Tanda tangan : 

Tanggal : Juni 2010

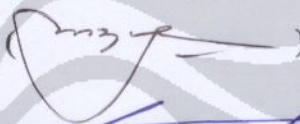
HALAMAN PENGESAHAN

Skripsi ini diajukan oleh :
Nama : Yanti Nitra Sianipar
NPM : 0606077610
Program Studi : Teknik Industri
Judul Skripsi : Optimasi Penjadwalan Mata Kuliah di Perguruan
Tinggi dengan Menggunakan Algoritma Memetika

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

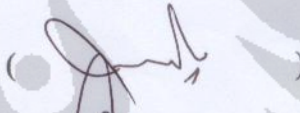
Pembimbing : Ir. Amar Rachman, MEIM

()

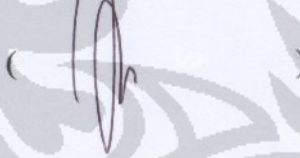
Penguji : Armand Omar Moeis, ST., M.Sc

()

Penguji : Komarudin, ST., M.Eng.

()

Penguji : Dendi P. Ishak, MSIE

()

Ditetapkan di : Depok

Tanggal : 5 Juli 2010

UCAPAN TERIMA KASIH

Puji dan syukur kepada Allah Bapa yang Mahakuasa atas segala berkat, kasih dan karunia-Nya sehingga skripsi ini dapat diselesaikan dengan baik. Skripsi ini disusun dalam rangka melengkapi salah satu persyaratan penyelesaian Program Pendidikan Sarjana di Program Studi Teknik Industri Universitas Indonesia. Penulis juga ingin mengucapkan terima kasih kepada pihak-pihak yang telah membantu dalam penyusunan skripsi ini. Adapun pihak-pihak itu adalah:

1. Orang tua dan adik-adikku tercinta, Ivan dan Irna, atas segala doa dan dukungan yang diberikan selama penyusunan skripsi ini.
2. Bapak Ir. Amar Rachman, MEIM, selaku dosen pembimbing skripsi yang telah memberikan waktu, tenaga, pikiran, serta dorongan dalam membimbing penulis.
3. Rekan-rekan bimbingan penulis; Evi, Sisi dan Nicholas atas dukungan dan kerjasamanya dalam mempelajari algoritma.
4. Bapak Armand Omar Moeis, ST., M.Sc., Bapak Komarudin, ST., M.Eng., Bapak Dendi P. Ishak, MSIE, dan Bapak Ir. Yadrifil, MSc yang memberikan ide, saran dan pengarahan dalam penyelesaian skripsi ini.
5. Kak Lismanto, Kak Nola, dan Kak Reza yang sangat besar membantu penulis dalam penyelesaian skripsi ini.
6. Sisca, Sarah, Masta, Lena, Naul, Martha, Afmieke, dan Mastora yang bersedia mendengarkan keluh kesah penulis dan berbagi kebahagiaan bersama.
7. Bapak/Ibu Staff PAF FTUI, Bu Har, Mba Willi, dan staf setiap jurusan di Fakultas Teknik atas kemudahan yang diberikan kepada penulis dalam pengumpulan data serta semua pihak yang terlibat.

Penulis menyadari bahwa skripsi ini jauh dari sempurna. Oleh karena itu, penulis sangat mengharapkan kritik dan saran yang membangun dari pembaca. Akhirnya, penulis berharap agar skripsi ini bisa memberikan manfaat bagi semua pihak yang membacanya dan bagi pengembangan ilmu pengetahuan.

Depok, 26 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 : Yanti Nitra Sianipar

NPM : 0606077610

Program Studi : Teknik Industri

Departemen : Teknik Industri

Fakultas : Teknik

Jenis Karya : Skripsi

demikian 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 di Perguruan Tinggi dengan
Menggunakan Algoritma Memetika**

berserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Universitas Indonesia berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat dan mempublikasikan tugas akhir saya tanpa meminta izin dari 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



(Yanti Nitra Sianipar)

ABSTRAK

Nama : Yanti Nitra Sianipar
Program Studi : Teknik Industri
Judul : Optimasi Penjadwalan Mata Kuliah di Perguruan Tinggi dengan Menggunakan Algoritma Memetika

Penjadwalan mata kuliah merupakan masalah yang selalu dihadapi setiap perguruan tinggi. Penjadwalan ini menjadi rumit karena banyak aspek yang harus dipertimbangkan, seperti aspek mahasiswa, dosen, mata kuliah, ruangan, dan slot. Terdapat dua jenis kendala dalam membuat model penjadwalan mata kuliah yang muncul akibat kombinasi dari kelima aspek di atas. Pertama disebut dengan kendala *hard* yang seharusnya dipenuhi agar menghasilkan jadwal yang layak. Kedua disebut dengan kendala *soft* yang sebaiknya dipenuhi untuk meningkatkan kualitas jadwal. Dalam penelitian ini diusulkan menggunakan algoritma memetika, yang mengintegrasikan teknik pencarian lokal dengan algoritma genetika yang diharapkan dapat menghasilkan jadwal yang optimal.

Kata Kunci:

Optimasi, penjadwalan mata kuliah, algoritma memetika

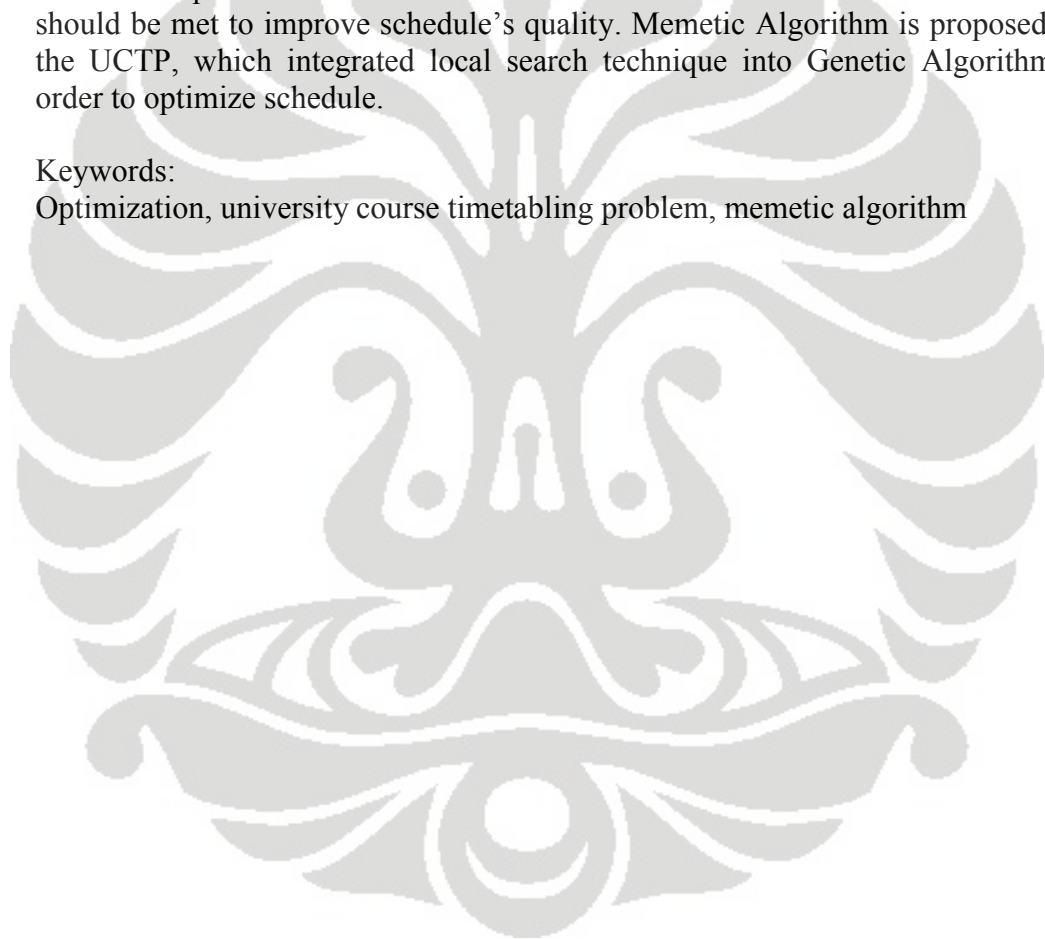
ABSTRACT

Name : Yanti Nitra Sianipar
Study Program : Industrial Engineering
Title : Optimization of University Course Timetabling Problem
Using Memetic Algorithm

University Course Timetabling Problem (UCTP) is an issue that is always faced by each college. UCTP becomes more complicated because of many aspects that must be considered, such as aspects of students, lecturers, courses, rooms, and slots. There are two type of constraints in this problem, which are emerged by combination of those aspects. First known as hard constraints that should be met in order to produce a feasible schedule. Second known as soft constraints that should be met to improve schedule's quality. Memetic Algorithm is proposed for the UCTP, which integrated local search technique into Genetic Algorithm in order to optimize schedule.

Keywords:

Optimization, university course timetabling problem, memetic algorithm



DAFTAR ISI

HALAMAN JUDUL.....	i
HALAMAN PENGESAHAN.....	iii
UCAPAN TERIMA KASIH.....	iv
ABSTRAK.....	vi
ABSTRACT.....	vii
DAFTAR ISI.....	viii
DAFTAR TABEL.....	x
DAFTAR GAMBAR.....	xi
DAFTAR LAMPIRAN.....	xii
1.PENDAHULUAN.....	1
1.1 Latar Belakang Masalah.....	1
1.2 Diagram Keterkaitan Masalah.....	3
1.3 Rumusan Permasalahan.....	5
1.4 Tujuan Penelitian.....	5
1.5 Ruang Lingkup Penelitian.....	5
1.6 Metodologi Penelitian.....	5
1.7 Sistematika Penulisan.....	6
2.LANDASAN TEORI.....	9
2.1 Teori Penjadwalan.....	9
2.1.1 Model Penjadwalan.....	9
2.1.2 Klasifikasi Penjadwalan.....	10
2.1.3 Kompleksitas dalam Penjadwalan.....	11
2.1.4 Penjadwalan Kuliah Standar Internasional.....	13
2.2 Program Integer Linier.....	15
2.3 Algoritma Memetika.....	16
2.3.1 Skema Pengkodean Kromosom.....	21
2.3.2 Pembentukan Populasi Awal.....	21
2.3.3 Pencarian Lokal (<i>Local Search</i>).....	22
2.3.4 Evaluasi Nilai <i>Fitness</i>	23
2.3.5 Seleksi.....	23
2.3.6 <i>Crossover</i>	25
2.3.7 Mutasi.....	26
2.3.8 Penggantian Populasi.....	27
2.3.9 Kriteria Berhenti.....	28
3.PENGUMPULAN DATA.....	29
3.1 Data yang Diperlukan.....	29
3.1.1 Data Ruang Kuliah.....	29
3.1.2 Data Mahasiswa.....	29
3.1.3 Data Dosen.....	30
3.1.4 Data Mata Kuliah.....	30
3.1.5 Data Slot Waktu.....	30
3.2 Model Penjadwalan Kuliah di Departemen Teknik Industri UI.....	35
3.2.1 Fungsi Penalti.....	37

4.PENGOLAHAN DATA DAN ANALISIS	39
4.1 Penyusunan Algoritma	39
4.1.1 Langkah-langkah Penyusunan Algoritma Memetika.....	39
4.1.2 Verifikasi dan Validasi Program.....	49
4.1.2.1 Hasil perhitungan manual	52
4.1.2.2 Hasil perhitungan program.....	56
4.2 INPUT DATA	56
4.2.1 Data	56
4.2.2 Parameter.....	57
4.3 Pengolahan Data dan Hasil.....	58
4.3.1 Hasil Penjadwalan dengan Algoritma Memetika.....	58
4.4 Analisis.....	61
4.4.1 Analisis Metode	61
4.4.2 Analisis Waktu <i>Running</i>	63
4.4.3 Analisis Hasil	63
5.KESIMPULAN DAN SARAN	71
5.1 Kesimpulan.....	71
5.2 Saran.....	71
DAFTAR REFERENSI	73

DAFTAR TABEL

Tabel 3.1 Jumlah Mahasiswa DTI	29
Tabel 3.2 Slot Waktu Perkuliahan	30
Tabel 3.3 Jumlah Kelas yang Tersedia	31
Tabel 3.4 Perhitungan Jumlah Kelas yang Dibutuhkan Tiap Departemen	32
Tabel 3.5 Perhitungan Kebutuhan Kelas untuk Departemen dengan Bidang Peminatan.....	34
Tabel 3.6 Perbandingan Kebutuhan Kelas dan Jumlah Kelas yang Tersedia	34
Tabel 3.7 Ruang yang Dialokasikan untuk Departemen Teknik Industri (Berdasarkan Kebutuhan Semester Ganjil).....	35
Tabel 3.8 Ruang yang Dialokasikan ke DTI.....	35
Tabel 4.1 Contoh Matrik KM.....	42
Tabel 4.2 Contoh Matrik Irisan Bentrok.....	42
Tabel 4.3 Matrik Jadwal (MJ berukuran Ruang x Slot).....	42
Tabel 4.4 Kromosom Jadwal (KJ berukuran 2 x 56).....	42
Tabel 4.5 Skenario Parameter	51
Tabel 4.6 Parameter yang Digunakan dalam Validasi.....	52
Tabel 4.7 Data <i>Dummy</i>	52
Tabel 4.8 Skenario Parameter untuk Nilai Maksimum Iterasi.....	58
Tabel 4.9 Parameter Model.....	58
Tabel 4.10 Hasil <i>Run</i> Program.....	59
Tabel 4.11 Solusi Jadwal.....	67

DAFTAR GAMBAR

Gambar 1.1 Diagram Keterkaitan Masalah.....	4
Gambar 1.2 Diagram Alir Metodologi Penelitian.....	7
Gambar 2.1 Diagram Venn Pengklasifikasian Penjadwalan.....	11
Gambar 2.2 Diagram Masalah Penjadwalan.....	11
Gambar 2.3 Perbandingan Tingkat Keakuratan antara MA dan AG.....	18
Gambar 2.4 <i>Flowchart</i> AM.....	19
Gambar 2.5 Algoritma untuk AG, AM, dan <i>Local Search</i> pada AM.....	20
Gambar 2.6 Tiga jenis skema pengkodean.....	21
Gambar 2.7 Contoh <i>One Point Crossover</i>	25
Gambar 2.8 Contoh <i>Two Point Crossover</i>	26
Gambar 4.1 Algoritma Memetika Penjadwalan Mata Kuliah.....	43
Gambar 4.2 Pencarian Lokal Satu.....	45
Gambar 4.3 Hasil Pencarian Lokal Satu.....	46
Gambar 4.4 Pencarian Lokal Dua.....	46
Gambar 4.5 Hasil Pencarian Lokal Dua.....	47
Gambar 4.6 <i>Crossover</i>	49
Gambar 4.7 Kromosom Jadwal 1-4.....	53
Gambar 4.8 Hasil Pencarian Lokal untuk Kromosom Jadwal 1-4.....	54
Gambar 4.9 Hasil Pindah Silang (<i>Crossover</i>).....	54
Gambar 4.10 Hasil Mutasi.....	55
Gambar 4.11 Hasil Pencarian Lokal Hasil Mutasi.....	55
Gambar 4.12 Solusi Jadwal Perhitungan Manual.....	56
Gambar 4.13 Solusi Jadwal Hasil <i>Running</i> Program.....	56
Gambar 4.14 Tampilan Matrix Jadwal Terbaik Hasil <i>Run</i> ke-5.....	60
Gambar 4.15 Kromosom Jadwal Terbaik Hasil <i>Run</i> ke-5 serta Nilai Penaltinya.....	60
Gambar 4.16 Grafik Peningkatan Nilai <i>Fitness</i> Jadwal.....	65
Gambar 4.17 Matrix Jadwal Terbaik Hasil <i>Run</i> dengan 6 Ruang.....	66
Gambar 4.18 Kromosom Jadwal Terbaik Hasil <i>Run</i> dengan 6 Ruang.....	66

DAFTAR LAMPIRAN

Lampiran 1. Data Ruang Kuliah	75
Lampiran 2. Data Dosen Pengajar di Departemen Teknik Industri UI.....	76
Lampiran 3. Struktur Mata Ajaran Program Studi Teknik Industri.....	78
Lampiran 4. Matrik KK (Matrik Hubungan antar Mata Kuliah)	80
Lampiran 5. Matrik KT (Hubungan antara Mata Kuliah dengan Slot Waktu)	82
Lampiran 6. Matrik KM (Hubungan antara Mata Kuliah dengan Mahasiswa)....	83
Lampiran 7. <i>Script M-File</i> Program.....	85



BAB 1

PENDAHULUAN

1.1 Latar Belakang Masalah

Penjadwalan kegiatan belajar mengajar di suatu perguruan tinggi merupakan hal yang penting demi kelancaran proses belajar mengajar. Dengan demikian perkuliahan dapat dilakukan pada waktu dan ruangan yang tepat dan dengan dosen yang tepat pula. Idealnya seperti itu, akan tetapi kenyataan di lapangan tidaklah mudah.

Penjadwalan menjadi hal yang kompleks dikarenakan banyak aspek yang perlu dipertimbangkan¹. Dari aspek mahasiswa, kita harus memperhatikan agar tidak ada mahasiswa yang menghadiri lebih dari satu mata kuliah pada waktu yang bersamaan dan dilakukan pendistribusian jadwal mata kuliah yang merata setiap harinya sehingga tidak terjadi penumpukkan jadwal dalam satu hari. Dari aspek dosen, terdapat dosen yang mengajar lebih dari satu mata kuliah sehingga tidak boleh ada dosen yang mengajar lebih dari satu mata kuliah pada waktu yang bersamaan. Dari aspek mata kuliah, terdapat mata kuliah tertentu yang tidak boleh bentrok. Hal ini disebabkan mata kuliah dalam tiap semester dan yang diajarkan oleh dosen yang sama tidak boleh dijadwalkan pada waktu yang bersamaan karena hal ini dapat membuat jadwal tidak layak. Selain itu, untuk memaksimalkan proses belajar mengajar, perguruan tinggi memberlakukan kapasitas maksimum mahasiswa untuk setiap mata kuliah yang menyebabkan jumlah kelas yang harus dialokasikan semakin bertambah.

Selain dari ketiga aspek diatas, aspek ruangan juga menjadi kendala dalam penjadwalan mata kuliah. Pekerjaan penjadwalan mata kuliah ini akan semakin berat jika melibatkan semakin banyak kelas per angkatanannya. Apalagi perguruan tinggi memberlakukan adanya pemakaian ruang kuliah bersama sehingga penjadwalan yang optimal menjadi hal yang sangat penting untuk menghindari terjadinya bentrok pemakaian ruang kuliah. Oleh karena itu terdapat dua jenis

¹ 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.

kendala dalam penjadwalan mata kuliah ini, yaitu kendala *hard* yang merupakan kendala yang harus dipenuhi untuk memberikan jadwal yang layak dan kendala *soft* kendala yang sebaiknya dipenuhi untuk meningkatkan kualitas jadwal yang dihasilkan. Jadwal dikatakan optimal jika kendala *hard* dan *soft* dapat dipenuhi.

Masalah ketidakefektifan dalam penjadwalan mata kuliah biasanya terjadi di setiap awal semester perkuliahan yang mengakibatkan perkuliahan tidak berjalan dengan baik karena ditunda oleh proses perbaikan jadwal kembali. Dalam penanganan masalah ini perlu dilakukan pengoptimalan penjadwalan dengan tujuan untuk meminimalkan jumlah bentrok yang terjadi, memaksimalkan penggunaan ruang kuliah, dan pemerataan beban mahasiswa.

Penjadwalan merupakan pengalokasian *resources* terhadap pekerjaan-pekerjaan tertentu pada waktu yang tepat². Penjadwalan berkaitan dengan pemilihan waktu operasi dalam perencanaan penggunaan fasilitas, karyawan, persediaan, dan lain-lain. Penjadwalan yang optimal membuat proses operasi berjalan dengan efektif. Untuk itu diusulkan pengaturan jadwal dengan menggunakan pendekatan algoritma memetika yang dapat menyelesaikan masalah multi-kriteria dan multi-objektif.

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 tujuan. Terdapat beberapa metode yang bisa digunakan untuk melakukan optimasi, namun untuk kasus-kasus yang berskala besar dibutuhkan teknik-teknik yang lebih modern seperti algoritma optimasi.

Algoritma merupakan detail tahapan-tahapan sebuah prosedur dalam penyelesaian masalah komputasi. Terdapat beberapa metode algoritma yang digunakan dalam penyelesaian masalah optimasi penjadwalan, diantaranya *simulated annealing*, *tabu search*, algoritma genetika, dan algoritma memetika. Algoritma memetika telah terbukti lebih efektif diantara metode-metode tersebut³.

² Herroelen, willy. (2005). *Production and Operation Management*. Winter 2005; 14, 4; ABI/INFORM Global hal. 413.

³ 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.

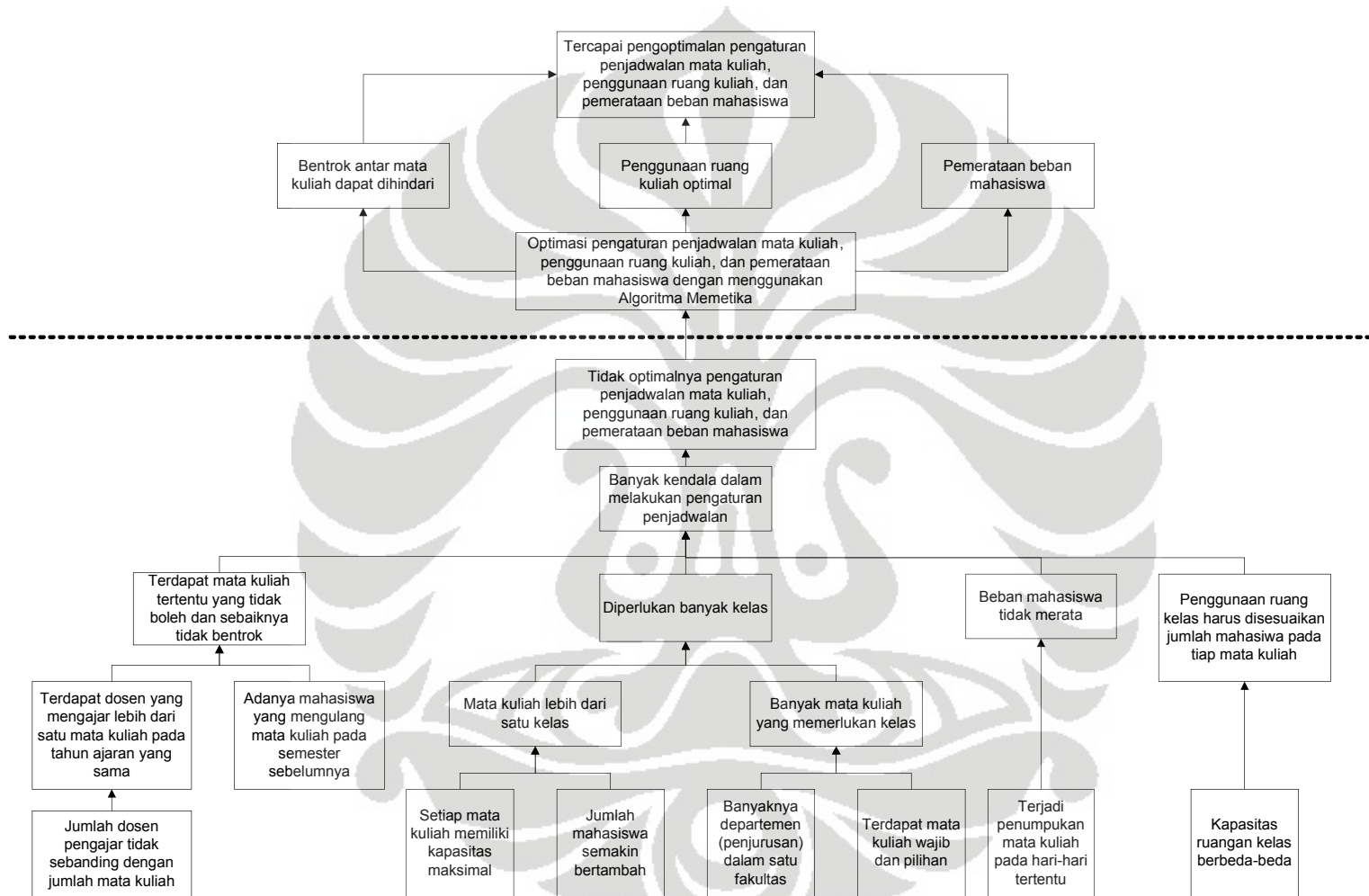
Algoritma memetika merupakan perluasan dari algoritma genetika dengan menambahkan teknik *local search* untuk mengurangi kemungkinan kesalahan dalam pencarian solusi optimal⁴. Algoritma memetika juga ampuh digunakan dalam penyelesaian masalah penjadwalan⁵. Oleh karena itu, penggunaan algoritma memetika dalam permasalahan penjadwalan mata kuliah diharapkan dapat memberikan hasil yang optimal. Metode ini sebelumnya pernah digunakan oleh mahasiswa Departemen Matematika Universitas Indonesia untuk menyelesaikan masalah penjadwalan mata kuliah, namun dalam penelitiannya lebih mengacu kepada pencarian kebutuhan ruangan dan slot yang dapat memberikan jadwal yang layak dan optimal. Tujuan penelitian yang bersangkutan adalah menemukan kombinasi ruang dan slot yang dapat melayakkan jadwal kemudian melakukan pengoptimalan dengan meminimalkan penalti kendala *soft* sehingga fungsi tujuan penelitiannya adalah meminimalkan penalti kendala *soft*. Sebaliknya penelitian ini bertujuan untuk menyelesaikan masalah penjadwalan mata kuliah dengan kendala yang disesuaikan dengan kondisi yang ada, yakni dengan jumlah ruang dan slot yang telah ditetapkan sehingga fungsi tujuan penelitian ini adalah meminimalkan total penalti kendala *hard* dan *soft* karena dengan memenuhi kedua kendala ini jadwal dapat dikatakan optimal. Selain itu dari tata alur algoritma memetika yang diterapkan dalam penelitian ini juga berbeda yang akan lebih diperjelas dalam Bab Pengolahan Data dan Analisis.

1.2 Diagram Keterkaitan Masalah

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

⁴ 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.

⁵ 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.



Gambar 1.1 Diagram Keterkaitan Masalah

1.3 Rumusan Permasalahan

Berdasarkan diagram keterkaitan masalah, fokus permasalahan adalah bagaimana menjadwalkan mata kuliah terhadap slot waktu dan ruangan dengan kendala-kendala yang telah ditentukan. Oleh karena itu, dibutuhkan penjadwalan yang mampu memberikan keoptimalan terhadap jadwal mata kuliah.

1.4 Tujuan Penelitian

Berdasarkan perumusan masalah yang telah dijelaskan sebelumnya, maka tujuan penelitian ini adalah untuk memperoleh jadwal mata kuliah yang optimal pada perguruan tinggi untuk menghindari terjadinya konflik/bentrok dengan menggunakan algoritma memetika.

1.5 Ruang Lingkup Penelitian

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

1. Pengambilan data dilakukan di Fakultas Teknik, Universitas Indonesia
2. Data perkuliahan yang diambil berdasarkan kurikulum tahun ajaran 2009/2010.
3. Model penjadwalan diimplementasikan pada perkuliahan di Departemen Teknik Industri pada semester ganjil.

1.6 Metodologi Penelitian

Metode yang digunakan dalam penelitian ini secara sistematis adalah sebagai berikut:

1. Penentuan topik penelitian

Adapun topik dalam penelitian ini adalah Optimasi Penjadwalan Mata Kuliah di Perguruan Tinggi dengan Menggunakan Algoritma Memetika.

2. Penentuan landasan teori

Tahap selanjutnya adalah menentukan landasan teori yang berhubungan dengan topik sebagai dasar dalam penelitian. Landasan teori ini kemudian akan menjadi acuan dalam pelaksanaan penelitian. Adapun landasan teori

yang terkait antara lain teori penjadwalan, program integer linier dan algoritma memetika.

3. Pengumpulan data

Dalam penelitian ini, data yang dibutuhkan adalah

- Data ruang kuliah: nama ruangan, jenis ruang kuliah (apakah merupakan ruang kuliah bersama atau tidak), dan kapasitas.
- Data mahasiswa: jumlah mahasiswa pada satu angkatan dalam suatu jurusan.
- Data dosen: nama dosen dan mata kuliah yang diajarkan.
- Data mata kuliah: nama mata kuliah tiap semester, jenis mata kuliah (mata kuliah wajib atau pilihan), dan jumlah maksimal mahasiswa pada tiap mata kuliah.

4. Pengolahan data

Pada tahapan ini, data ditelaah dan dianalisis untuk kemudian diolah dengan algoritma memetika dengan menggunakan perangkat lunak MATLAB R2008a.

5. Analisis hasil pengolahan data

Dalam tahap ini dilakukan analisis terhadap hasil pengolahan data untuk dilihat perbaikannya dibandingkan dengan keadaan yang selama ini berlangsung di perguruan tinggi.

6. Kesimpulan

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

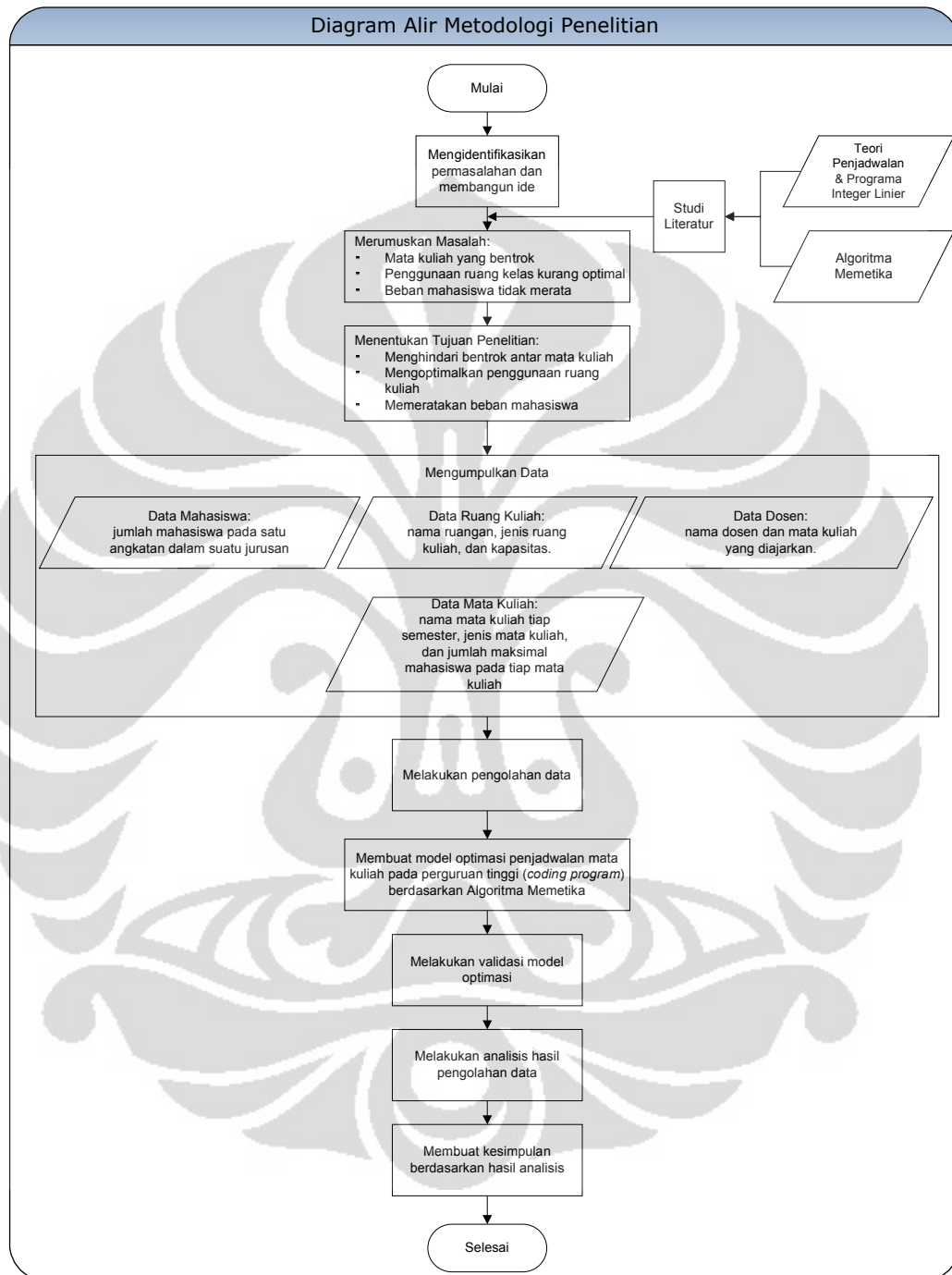
Untuk menggambarkan secara sistematis, maka dibuat diagram alir metodologi penelitian yang dapat dilihat pada Gambar 1.2.

1.7 Sistematika Penulisan

Secara umum, pembahasan penelitian terdiri dari beberapa bab dengan sistematika sebagai berikut:

Bab pertama merupakan bab pendahuluan yang menjelaskan mengenai latar belakang permasalahan dilakukannya penelitian ini, diagram keterkaitan

masalah, rumusan permasalahan, tujuan penelitian, ruang lingkup penelitian, metode penelitian, dan sistematika penulisan.



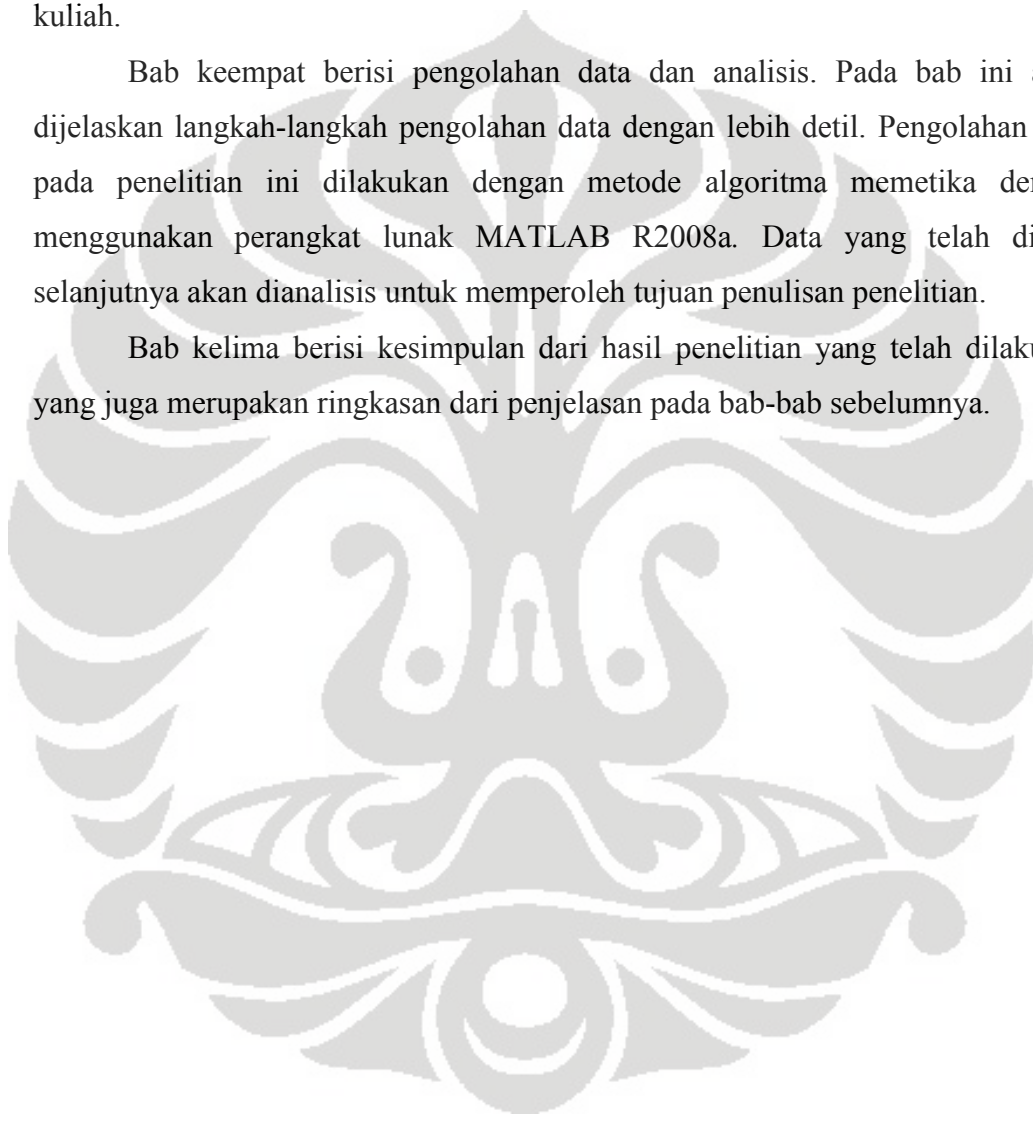
Gambar 1.2 Diagram Alir Metodologi Penelitian

Bab kedua merupakan landasan teori yang berhubungan dengan penelitian ini. Landasan teori yang dibahas meliputi teori penjadwalan, program integer linier, dan metode optimasi yang dikhususkan pada algoritma memetika.

Bab ketiga berisi tentang pelaksanaan pengumpulan data. Bab ini menjelaskan mengenai data yang dibutuhkan pada penelitian ini. Data tersebut berupa data ruang kuliah, data mahasiswa, data dosen pengajar, dan data mata kuliah.

Bab keempat berisi pengolahan data dan analisis. Pada bab ini akan dijelaskan langkah-langkah pengolahan data dengan lebih detil. Pengolahan data pada penelitian ini dilakukan dengan metode algoritma memetika dengan menggunakan perangkat lunak MATLAB R2008a. Data yang telah diolah selanjutnya akan dianalisis untuk memperoleh tujuan penulisan penelitian.

Bab kelima berisi kesimpulan dari hasil penelitian yang telah dilakukan yang juga merupakan ringkasan dari penjelasan pada bab-bab sebelumnya.



BAB 2 LANDASAN TEORI

2.1 Teori Penjadwalan

Penjadwalan berkaitan dengan pengoptimalan pengalokasian *resources* yang terbatas terhadap sejumlah aktivitas pada waktu yang tepat. Hal ini menjadi sebuah proses pengambilan keputusan yang memiliki tujuan untuk mengoptimalkan satu tujuan atau lebih⁶. Masalah penjadwalan mencakup bagaimana menjadwalkan aktivitas dapat berjalan dengan baik dengan mempertimbangkan kegiatan-kegiatan lainnya yang terkait dan juga dengan kendala *resources* yang terbatas⁷.

Masalah penjadwalan merupakan masalah yang sulit. Bahkan menjadi lebih sulit ketika harus mempertimbangkan tujuan yang banyak (*multiple objective*). Adanya kombinasi dalam masalah penjadwalan mengakitkannya sulit untuk diselesaikan walaupun dengan cara teknik matematika. Hal-hal yang menjadi faktor pertimbangan dalam penyusunan jadwal yaitu adanya batasan waktu pengerjaan (*completion time*), aliran waktu (*flow time*) yang merupakan urutan pengerjaan aktivitas termasuk waktu tunggu (*waiting time*) dan waktu proses (*processing time*). Penjadwalan yang tepat akan mencapai tujuan yang optimal dengan memenuhi semua kendala.

2.1.1 Model Penjadwalan

Perkembangan ilmu memungkinkan dalam penyelesaian masalah penjadwalan, yaitu dengan menggunakan *integer programming formulations* ataupun dengan menggunakan algoritma. Terdapat tiga notasi yang menjadi *framework* (faktor yang menentukan model penjadwalan), yaitu⁸:

- konfigurasi *resources*: dimana masing-masing *resources* memiliki kemampuan untuk mengerjakan jenis pekerjaan tertentu.

⁶ Karabati, Selcuk & Panos, Kouvelis. (1996). *Optimization Applications in Scheduling Theory*. Journal of Global Optimization 9:223-226.

⁷ Ibid, hal.2.

⁸ Schwiegelshohn, Uwe. (2004, Summer Term). *Scheduling Problems and Solutions*. Computer Engineering Institute, University Dortmund.

- kendala atau karakter proses: setiap jenis pekerjaan memiliki karakternya sendiri, dapat berupa lama waktu yang dibutuhkan untuk mengerjakannya, kebutuhan-kebutuhan yang harus dilengkapi untuk mengerjakannya.
- fungsi tujuan: fungsi yang mengarahkan jalannya model penjadwalan.

Resources yang akan dialokasikan terhadap setiap pekerjaan juga memiliki karakter seperti berikut:

- jumlah *resources* yang dapat mengerjakan tiap pekerjaan (apakah *resources* tunggal atau tidak)
- jenis-jenis pekerjaan yang dapat dikerjakan oleh masing-masing *resources*
- *resources* apa saja yang dapat dijadwalkan secara paralel

Pekerjaan yang akan dijadwalkan memiliki karakternya masing-masing, yang dapat berupa:

- waktu proses atas pekerjaan terhadap masing-masing *resources*
- tingkat kepentingan/bobot atas pekerjaan tersebut
- waktu kapan pekerjaan tersebut dapat dimulai dan diakhiri

2.1.2 Klasifikasi Penjadwalan

Ada tiga jenis pengklasifikasian sistem penjadwalan, yaitu:

1. Penjadwalan yang tidak dapat ditunda (*Nondelay/Greedy Schedule*)

Sistem penjadwalan dikatakan *greedy* apabila tidak memungkinkan terdapat *resources* yang tidak bekerja (*idle*) ketika sedang dalam tahapan menunggu untuk diproses. Penjadwalan dianggap optimal jika tidak terdapat penundaan.

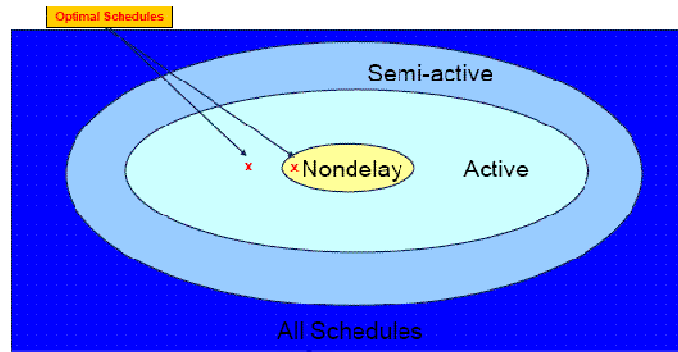
2. Penjadwalan aktif (*Active Schedule*)

Sistem penjadwalan dikatakan aktif apabila tidak memungkinkan untuk membentuk jadwal lain dengan mengganti urutan proses pada *resources* dan memiliki paling tidak satu *task* sudah selesai di awal.

3. Penjadwalan semi – aktif (*Semi – active Schedule*)

Sistem penjadwalan dikatakan semi-aktif apabila tidak memungkinkan pekerjaan dapat diselesaikan lebih dulu tanpa mengubah urutan proses dari salah satu *resources*.

Pengklasifikasian penjadwalan dapat dilihat pada diagram venn pada Gambar 2.1.

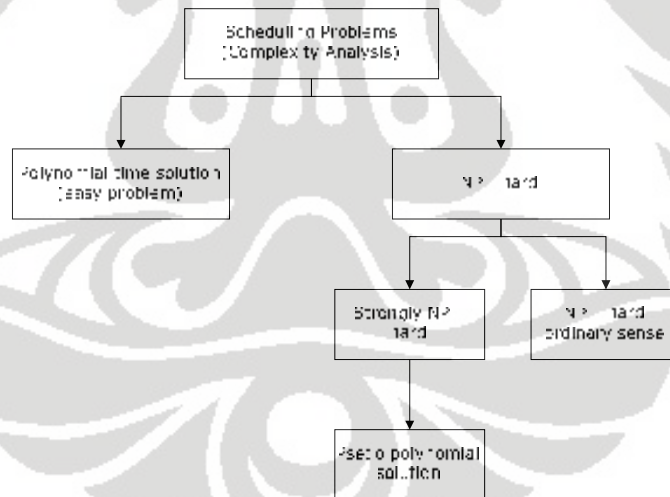


Gambar 2.1 Diagram Venn Pengklasifikasian Penjadwalan

(Sumber: Uwe Schwiegelshohn, CEI University Dortmund 2004)

2.1.3 Kompleksitas dalam Penjadwalan

Penjadwalan menjadi semakin sulit ketika memiliki kendala-kendala yang banyak dan cukup rumit. Disamping itu, apabila jumlah populasi yang harus dijadwalkan semakin banyak akan memicu kekompleksitasan masalah penjadwalan. Kekompleksitasan dalam penjadwalan dapat dilihat pada Gambar 2.2.



Gambar 2.2 Diagram Masalah Penjadwalan

(Sumber: Uwe Schwiegelshohn, CEI University Dortmund 2004)

Terdapat dua tingkat kesulitan dalam melakukan penjadwalan, yaitu:

1. Mudah (*polynomial time complexity*)

Penjadwalan dikatakan mudah apabila sebuah algoritma dapat digunakan untuk mengoptimalkan masalah dengan kompleksitas waktu $O((n \log(\max p_j))^k)$ untuk nilai k tertentu.

2. NP-hard, terdiri dari dua jenis:

a. NP-hard dalam tingkat kesulitan yang luar biasa (*pseudo polynomial time complexity*)

Masalah penjadwalan yang tidak dapat dioptimalkan dengan sebuah algoritma dengan kompleksitas *polynomial* waktu namun dengan sebuah algoritma dengan kompleksitas $O((n \max p_j)^k)$.

b. NP-hard dalam *strong sense*

Masalah tidak dapat dioptimalkan dengan sebuah *pseudo polynomial complexity*.

Oleh karena tingkat kesulitannya, maka masalah penjadwalan diselesaikan dengan menggunakan algoritma. Algoritma adalah kumpulan perintah untuk menyelesaikan suatu masalah dan dapat diterjemahkan secara bertahap dari awal hingga akhir yang bersifat diskrit dan jelas serta dapat dijalankan secara mekanik. Berikut merupakan contoh algoritma untuk penjadwalan:

Step 1 Set $J = \emptyset$, let $J^c = \{1, \dots, n\}$ dan J' and J' be the set of all jobs with no successors.

Step 2 Let $j^* \in J'$ be such that

$$h_{j^*} \left(\sum_{j \in J^c} p_j \right) = \min_{j \in J'} \left(h_j \left(\sum_{k \in J^c} p_k \right) \right)$$

Add j^* to J

Delete j^* from J^c

Modify J' to represent the new set of schedule jobs.

Step 3 If $J^c = \emptyset$ then STOP otherwise go to Step 2.

Algoritma ini memberikan sebuah penjadwalan yang optimal untuk satu kegiatan terdahulu, atau dapat ditulis dengan $1 | \text{prec} | h_{\max}$.

Penjadwalan dapat muncul dalam berbagai masalah, seperti penjadwalan pendistribusian atau logistik, penjadwalan produksi, penjadwalan perkuliahan di universitas dan lain-lain. Oleh karena penelitian ini bertujuan untuk

mengoptimalkan penjadwalan perkuliahan, maka berikut akan dibahas mengenai penjadwalan kuliah.

2.1.4 Penjadwalan Kuliah Standar Internasional

Masalah penjadwalan mata kuliah disebut juga dengan *University Course Timetabling Problem* (UCTP). Dari sejumlah masalah penjadwalan, penjadwalan mata kuliah merupakan salah satu yang paling banyak dipelajari. Pokok permasalahannya adalah perancangan jadwal mata kuliah yang harus dibuat dengan mempertimbangkan bagaimana mengalokasikan mata kuliah terhadap jumlah slot waktu dan ruang yang terbatas serta beberapa kendala lainnya⁹.

Masalah dalam pembuatan jadwal perkuliahan yang benar (valid) melibatkan penjadwalan kelas (mata kuliah), dosen, dan ruangan terhadap periode tertentu sedemikian rupa sehingga tidak ada dosen, mata kuliah, dan ruangan yang digunakan lebih dari satu kali per periode. Sebagai contoh, jika kelas harus memenuhi dua kali pertemuan dalam seminggu, maka harus ditempatkan ke dalam dua periode yang berbeda untuk menghindari terjadinya konflik.

Kendala yang terdapat pada UCTP terbagi dalam dua kategori kendala, yaitu kendala *hard* dan kendala *soft*. Kendala *hard* merupakan kendala yang harus dipenuhi dalam perancangan jadwal sedangkan kendala *soft* adalah kendala yang harus diminimalkan karena berpengaruh terhadap kualitas jadwal yang dibuat. Kendala *hard* dan *soft* terjadi akibat adanya interaksi antara beberapa entitas dalam perancangan jadwal.

Model penjadwalan ini akan terdiri dari beberapa entitas seperti akan dijelaskan berikut ini¹⁰:

1. Hari, slot waktu, dan periode

Dalam memodelkan rancangan terdapat jumlah hari yang digunakan untuk proses belajar mengajar dalam seminggu, yang biasanya lima hari yaitu Senin hingga Jumat. Setiap hari masih dibagi lagi ke dalam beberapa slot waktu yang biasanya sama untuk setiap hari. Sehingga dalam perancangan

⁹ Abdullah, S., Hamza, T., & B. McCollum. (2009). *A Tabu-based Memetic Approach to the Examination Timetabling Problem*. The Second International Timetabling Competition.

¹⁰ McCollum, B., et al. (2009). *Setting the Research Agenda in Automated Timetabling*. The Second International Timetabling Competition.

penjadwalan hari dan slot waktu merupakan pasangan. Jumlah periode penjadwalan adalah hasil dari pasangan hari dan slot waktu tiap hari.

2. Mata kuliah dan dosen

Setiap mata kuliah memiliki dosen pengajar yang tetap yang akan dijadwalkan pada periode yang berbeda, dan nantinya akan diikuti oleh mahasiswa pada jumlah tertentu dan diajar oleh seorang pengajar. Untuk setiap mata kuliah terdapat jumlah minimum hari yang dibutuhkan dosen untuk mengajarkan mata kuliah tersebut dan biasanya hal ini ditentukan dengan bobot SKS dari mata kuliah tersebut. Namun, terdapat beberapa mata kuliah yang tidak dapat dijadwalkan pada periode-periode tertentu diakibatkan oleh beberapa kendala.

3. Ruangan

Setiap ruangan memiliki kapasitas, yang ditunjukkan dengan jumlah maksimal kursi yang dapat dimuat dalam ruangan. Oleh karena itu, pengalokasian ruangan harus disesuaikan dengan jumlah mahasiswa yang mengambil setiap mata kuliah.

4. Kurikulum

Kurikulum merupakan kelompok mata kuliah yang biasanya diambil oleh sekelompok mahasiswa yang berada pada tahun ajaran tertentu. Berdasarkan kurikulum, maka akan terdapat konflik antara mata kuliah.

Dalam UTCP terdapat beberapa kendala *hard* yang timbul akibat interaksi antar entitas model, yaitu sebagai berikut:

1. H1 : Mata kuliah, semua mata kuliah harus dijadwalkan pada periode yang berbeda. Jika terdapat mata kuliah yang dijadwalkan pada periode yang sama maka dapat disebut pelanggaran.
2. H2 : Kapasitas ruangan, semua kuliah harus dijadwalkan pada ruangan yang dapat memenuhi kapasitas untuk mata kuliah yang bersangkutan.
3. H3 : Konflik, mata kuliah pada kurikulum berjalan atau yang diajarkan oleh dosen yang sama harus dijadwalkan pada periode yang berbeda.

4. H4 : Keberadaan, jika dosen mata kuliah tertentu tidak dapat mengajar pada periode yang telah ditentukan, maka dosen tersebut tidak dapat dijadwalkan pada periode tersebut.

Sedangkan yang menjadi kendala *soft* adalah sebagai berikut:

1. S1 : Mahasiswa tidak dapat mengikuti mata kuliah pada slot waktu terakhir setiap harinya.
2. S2 : Mahasiswa tidak dapat mengikuti mata kuliah sebanyak tiga kali berturut-turut atau lebih dalam satu hari.
3. S3 : Mahasiswa tidak dapat mengikuti hanya satu mata kuliah pada satu hari.

Tujuan permasalahan penjadwalan ini adalah untuk memenuhi semua kendala *hard* dan kendala *soft* yang dapat mengoptimalkan jadwal.

2.2 Program Integer Linier

Program integer (*integer programming*) adalah program linier dimana beberapa atau semua variabel terbatas pada nilai integer (diskrit). Ada beberapa jenis program integer, yaitu program integer murni dimana semua variabel adalah integer dan program integer campuran jika hanya beberapa variabel saja yang bersifat integer. Berikut adalah bentuk umum dari program integer linier:

Fungsi Tujuan:

$$\text{Min } Z = \sum$$

Dengan kendala:

$$a_{ij}, x_j (\geq = \leq) b_i$$

$$i = 1, 2, 3, \dots, m$$

$$j = 1, 2, 3, \dots, n$$

x_j = positif (*nonnegativity* dan integer)

a, b, dan c konstan

ILP terdiri dari dua jenis, yaitu¹¹:

1. *Binary Integer Programming*

¹¹ Chinneck, John, W. (2001). *Practical Optimization: a Gentle Introduction*. <http://www.carleton.ca/faculty/chinneck/po.html>

Dalam masalah biner, setiap variabel hanya dapat memiliki nilai antara 0 atau 1. Hal ini menunjukkan pemilihan atau penolakan, ya atau tidak, atau kondisi lainnya sesuai dengan masalahnya. Bentuk standard persamaan dalam penyelesaian masalah *Binary Integer Programming* adalah sebagai berikut:

- Fungsi tujuan meminimumkan $Z = \sum_{j=1}^n c_j x_j$
- m kendala memiliki bentuk pertidaksamaan $\sum_{ij} x_i \geq b_i$ for $i = 1, 2, \dots, m$
- Semua koefisien fungsi tujuan adalah *non-negative*
- Variabel disusun sesuai dengan koefisien fungsi tujuannya sehingga $0 \leq c_1 \leq c_2 \leq \dots \leq c_n$

Hal ini menunjukkan adanya pembatasan kondisi masalah, namun banyak masalah yang mudah diubah menjadi bentuk persamaan tersebut. Contohnya, koefisien fungsi tujuan yang negatif akan diatasi dengan mengubah variabel-variabel yang semula x_j menjadi $(1 - x_j')$. Kendala pada sisi sebelah kanan dapat dinegatifkan, sehingga kendala \leq dapat dengan mudah diubah menjadi \geq dengan mengalikannya dengan -1.

2. *Mixed-Integer Linier Programming*

Masalah *mixed-integer programming* (MIP) terjadi ketika terdapat beberapa variabel dalam model dengan nilai riil dan beberapa variabel lainnya bernilai integer. Ketika semua fungsi objektif dan kendala adalah berbentuk linier maka masalah ini disebut dengan *mixed-integer linier programming* (MILP). Namun biasanya MILP disamakan dengan MIP walaupun yang muncul *mixed-integer nonlinier programs* (MINLP). Teknik MILP efektif digunakan tidak hanya untuk masalah variabel campuran, tapi juga buat masalah yang murni variabel integer, masalah biner, atau pada kenyataannya untuk beberapa kombinasi atas variabel bernilai riil, integer, dan biner.

2.3 Algoritma Memetika

Algoritma memetika (AM) adalah perluasan dari Algoritma Genetika. Algoritma Genetika (AG) merupakan salah satu metode heuristik yang banyak

digunakan untuk menyelesaikan masalah optimasi kombinatorial yang sulit¹². AG pertama kali ditemukan oleh John Holland pada tahun 1960-an yang didasari oleh proses evolusi biologi. Konsep yang penting dalam AG adalah hereditas dan seleksi alam. Hereditas merupakan sebuah ide bahwa sifat-sifat individu dapat dikodekan dengan cara tertentu, sehingga sifat-sifat itu dapat diturunkan kepada generasi berikutnya. Sifat-sifat individu tersebut dikodekan kedalam sekumpulan gen-gen yang disebut kromosom, perbedaan susunan gen-gen dalam suatu kromosom ini akan membawa sifat yang berbeda-beda dan unik pada individu. Untuk itu dapat diartikan bahwa sebuah kromosom melambangkan satu individu. Sedangkan ide dari seleksi alam adalah individu-individu yang tidak bisa beradaptasi akan mati, sedangkan individu-individu yang mampu beradaptasi akan hidup dan menghasilkan keturunan. Dalam waktu yang cukup lama, individu-individu dapat melakukan perkawinan dan menghasilkan keturunan yang membawa sifat kedua orang tuanya. Dan dalam penurunan sifat ini dapat terjadi kesalahan yang menyebabkan terjadinya perubahan hereditas pada generasi berikutnya.

Pada AG, kandidat solusi (selanjutnya disebut solusi) dari suatu masalah direpresentasikan sebagai kromosom. AG diawali dengan pembentukan kumpulan kromosom yang disebut dengan populasi. Masing-masing kromosom pada populasi akan dievaluasi menggunakan fungsi *fitness*, yaitu fungsi yang mengukur secara kuantitatif kemampuan suatu kromosom untuk bertahan dalam populasi. Kemudian secara iteratif akan dibentuk populasi baru yang lebih baik dari populasi sebelumnya dengan menerapkan operator-operator genetika hingga mencapai kriteria berhenti. Operator dasar pada AG adalah seleksi, *crossover*, dan mutasi. Seleksi yaitu operator untuk memilih dua kromosom orang tua. *Crossover* adalah operator untuk mengawinkan dua kromosom orang tua sehingga menghasilkan keturunan atau anak untuk generasi berikutnya. Dan mutasi adalah operator yang mengubah urutan gen pada kromosom¹³.

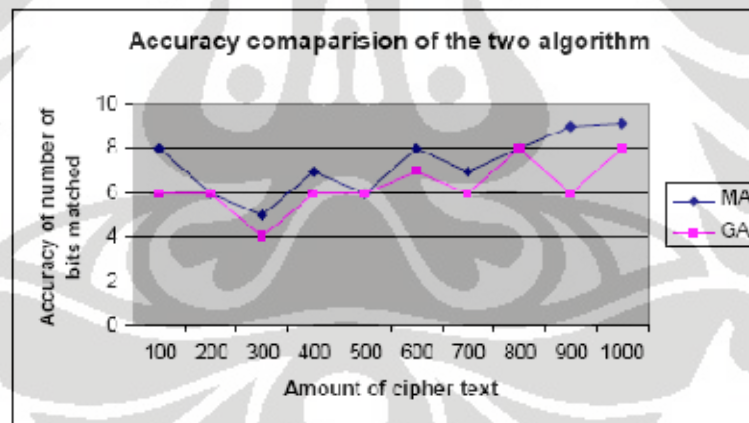
¹² Mostaco, P., & Carlos, C. (n.d). *A Gentle Introduction to Memetic Algorithms*. Computer Science Department, University of Newcastle, Australia. <http://www.cs.newcastle.edu.au/~nbi>

¹³ Wook Chang Ahn. (2006). *Advances in Evolutionary Algorithms*. Student in Computational Intelligence, Vol. 18. Republic of Korea:Springer. Hal.9.

Kelebihan AG adalah pada cara kerjanya yang paralel. AG bekerja dalam ruang pencarian yang menggunakan banyak individu sekaligus, sehingga kemungkinan AG untuk terjebak pada ekstrim lokal lebih kecil dibandingkan metode lain. AG juga mudah diimplementasikan karena jika suatu algoritma AG sudah dapat diimplementasikan, kita juga dapat menyelesaikan masalah lain hanya dengan membuat kromosom dan fungsi *fitness* yang baru sehingga AG bisa dijalankan.

Kekurangan AG adalah dalam hal waktu komputasi karena harus melakukan evaluasi *fitness* pada semua solusi di setiap iterasinya. Sehingga AG bisa lebih lambat dibandingkan dengan metode lain, namun kita dapat mengakhiri komputasi pada waktu yang diinginkan dengan maksimum iterasi yang diinginkan¹⁴.

Namun telah diketahui bahwa AG tidak cukup baik untuk mencari solusi yang sangat dekat dengan solusi optimal. Kekurangan ini dapat diakomodasi dengan cara menambahkan metode *Local Search* pada AG. Hasil penggabungan ini dikenal dengan sebutan algoritma memetika¹⁵.



Gambar 2.3 Perbandingan Tingkat Keakuratan antara MA dan AG

(Sumber: Poonam Garg , *A Comparison between Memetic algorithm and Genetic algorithm for the cryptanalysis of Simplified Data Encryption Standard algorithm*)

AM didasari oleh teori evolusi biologi *Neo-Darwinian* dan pendapat Dawkin tentang *meme* sebagai unit evolusi *cultural* yang mampu melakukan

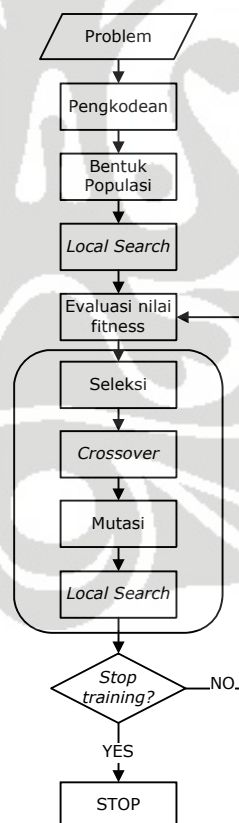
¹⁴ Obitko, M. (1998). *Introduction to Genetic Algorithm*. <http://www.obitko.com/tutorials/genetic-algorithms/>

¹⁵ Ibid, hal.3.

perbaikan terhadap dirinya sendiri. AM adalah suatu metode pencarian heuristik yang memiliki karakteristik yang sama dengan AG dikombinasikan dengan metode *Local Search*, yang secara bersama-sama dapat meningkatkan kualitas pencarian solusi (Mostaco, 1989).

Pada AM, *Local Search* bertujuan untuk melakukan perbaikan lokal yang dapat diterapkan sebelum dan atau sesudah proses seleksi, *crossover* dan mutasi. *Local Search* juga sangat berguna untuk mengontrol besarnya ruang pencarian solusi. AM dapat memberikan hasil yang lebih baik daripada AG, namun memerlukan waktu komputasi yang lebih lama. *Flowchart* dari AM standard dapat dilihat pada Gambar 2.4.

Jika dibandingkan dengan metode algoritma lainnya, AM memberikan waktu komputasi yang lebih cepat dalam jumlah iterasi yang lebih besar. Perbandingan antara AM dengan metode algoritma lainnya dalam hal waktu komputasi dapat dilihat pada Gambar 2.5.

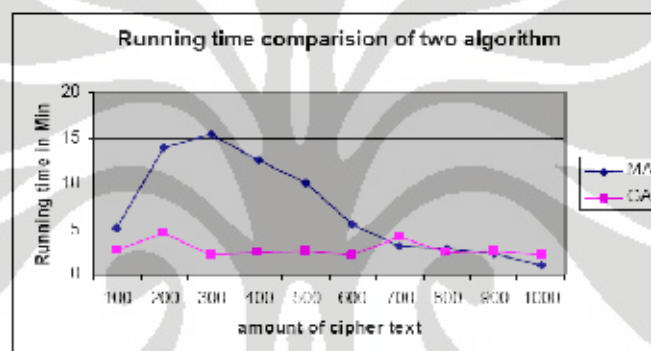


Gambar 2.4 Flowchart AM

(Sumber: Vico Duong & Percival Wong, *Memetic Algorithm*)

Berikut merupakan beberapa istilah dalam komponen AM yang diadopsi dari istilah evolusi biologi, diantaranya:

- Kromosom: individu untuk merepresentasikan solusi.
- Generasi: untuk menggantikan istilah iterasi.
- Orang tua: kromosom pada generasi sekarang yang terpilih untuk menghasilkan kromosom anak.
- Anak: kromosom hasil perkawinan kromosom orang tua pada generasi sekarang yang akan menjadi kromosom pada generasi berikutnya.
- Proses evolusi: proses yang terjadi pada tiap generasi yang meliputi proses seleksi, *crossover*, mutasi dan *local search*.



Gambar 2.5 Perbandingan waktu komputasi AM dan AG

(Sumber: Poonam Garg, *A Comparison between Memetic algorithm and Genetic algorithm for the cryptanalysis of Simplified Data Encryption Standard algorithm*)

Berikut merupakan contoh algoritma sederhana untuk AG dan AM:



Gambar 2.5 Algoritma untuk AG, AM, dan *Local Search* pada AM

(Sumber: Poonam Garg, *A Comparison between Memetic algorithm and Genetic algorithm for the cryptanalysis of Simplified Data Encryption Standard algorithm*)

2.3.1 Skema Pengkodean Kromosom

Pengkodean kromosom merupakan tahap awal yang dilakukan sebelum menggunakan AG. Dalam tahapan ini ditentukan sistem pengkodean kromosom yang cocok untuk masalah yang akan diselesaikan karena kromosom akan membawa informasi dari solusi yang ia representasikan. Secara umum, suatu kromosom berbentuk:

$$\text{Kromosom} = (gen_1, gen_2, \dots, gen_n)$$

Dimana n adalah jumlah gen dalam suatu kromosom dan setiap gen diisi oleh nilai dari gen atau disebut juga *allele*.

Terdapat tiga skema pengkodean yang paling sering digunakan, yaitu:

- Pengkodean bilangan riil (*real number encoding*), yaitu pengkodean dengan nilai gen berada dalam interval $[0, R]$, dimana R adalah bilangan positif yang biasanya bernilai 1.
- Pengkodean bilangan integer, yaitu pengkodean nilai gen bisa bernilai dari salah satu bilangan bulat dalam interval $[0, N]$.
- Pengkodean bilangan biner (*binary encoding*), yaitu setiap gen hanya bisa bernilai 0 atau 1.

1	0	1	1
Pengkodean bilangan biner			
2	3	14	9
Pengkodean bilangan integer			
1,34	0,52	7,42	0,31
Pengkodean bilangan riil			

Gambar 2.6 Tiga jenis skema pengkodean

(Sumber: Algoritma Genetika dalam Matlab, Suyanto 2005)

Sistem pengkodean yang digunakan dalam masalah ini adalah dengan pengkodean bilangan integer.

2.3.2 Pembentukan Populasi Awal

Tahap berikutnya adalah pembentukan populasi awal yang berisikan sekumpulan individu. Tahapan ini merupakan tahapan awal pada AG. Pada populasi awal, nilai dari gen-gen setiap individu ditentukan secara acak. Tidak ada

aturan yang tegas mengenai berapa seharusnya jumlah individu dalam suatu populasi, hal ini akan bergantung dengan jenis masalah yang akan diselesaikan. Ukuran populasi tergantung pada masalah yang akan diimplementasikan dan jenis operator genetika yang akan diimplementasikan. Setelah ukuran populasi ditentukan, dilakukan inisialisasi kromosom dalam populasi tersebut. Inisialisasi kromosom dilakukan secara acak, namun harus tetap memperhatikan domain solusi dan kendala permasalahan yang ada¹⁶.

Berdasarkan teorinya ukuran solusi yang besar mampu menghasilkan solusi yang lebih baik, namun hal ini akan membutuhkan sumber daya dan waktu yang lebih lama. Pada permasalahan penjadwalan mata kuliah ini ukuran populasi yang digunakan adalah 16, karena semakin besar ukuran populasi dalam masalah ini akan memberikan hasil yang kurang baik¹⁷. Hal ini disebabkan penjadwalan mata kuliah merupakan masalah kombinatorial yang cukup kompleks sehingga apabila memiliki ukuran populasi yang terlalu besar maka akan dihasilkan kombinasi-kombinasi yang tidak memenuhi kendala.

2.3.3 Pencarian Lokal (*Local Search*)

Pencarian lokal adalah metode heuristik untuk memperbaiki atau meningkatkan kualitas solusi sekarang dengan menelusuri *neighborhood*-nya hingga ditemukan solusi yang lebih baik. Jika solusi yang lebih baik ditemukan, solusi sekarang diperbaharui, kemudian dilakukan pencarian *neighborhood*-nya. Jika tidak ada peningkatan kualitas solusi, pencarian dihentikan, dan berarti solusi optimal lokal telah ditemukan. Pada AM, pencarian lokal dapat diterapkan pada dua tempat, yaitu:

- Sebelum masuk ke proses evolusi, pencarian lokal diterapkan ke seluruh kromosom pada populasi awal
- Pada proses evolusi, pencarian lokal diterapkan sebelum atau sesudah proses evolusi, *crossover*, atau mutasi. Tapi tidak diterapkan untuk semua

¹⁶ Sri, Kusumadewi, & Hari, P. (2005). *Penyelesaian Masalah Optimasi Menggunakan Teknik-teknik Heuristik*. Graha Ilmu, Yogyakarta, hal. 233.

¹⁷ Özcan, E. & Alpay, A. (2003). *A Memetic Algorithm for Solving a Timetabling Problem: An Incremental Strategy*. Computer Engineering Department, İstanbul, Turkey.

kromosom dalam populasi, melainkan dilakukan dengan probabilitas tertentu yang disebut dengan *LS_rate*.

Setelah dilakukan pencarian lokal terhadap semua kromosom, populasi awal digantikan oleh populasi hasil pencarian lokal. Kemudian, AM masuk ke proses evolusi yang dimulai dari generasi pertama hingga generasi maksimum. Proses ini dimulai dengan evaluasi individu atau kromosom dengan menggunakan nilai *fitness*.

Dalam masalah penjadwalan mata kuliah, pencarian lokal dilakukan dua kali, yaitu sebelum dan sesudah proses evolusi yang bertujuan untuk memperoleh jadwal dengan penalti kendala *hard* yang paling minimal yang kemudian akan diteruskan ke tahapan berikutnya.

2.3.4 Evaluasi Nilai *Fitness*

Di setiap generasi, kromosom dievaluasi dengan menggunakan fungsi *fitness*. Fungsi *fitness* merupakan sebuah fungsi yang mengukur secara kuantitatif kemampuan dari suatu individu untuk bertahan dalam populasi. Semakin tinggi nilai *fitness*-nya, maka semakin besar kemungkinan suatu individu akan bertahan. Nilai *fitness* sangat penting untuk mengarahkan AG dalam pencarian solusi. Dalam masalah penjadwalan mata kuliah nilai *fitness* dihitung dengan menghitung terlebih dahulu nilai penalti atas setiap kromosom dalam populasi.

2.3.5 Seleksi

Seleksi merupakan cara pemilihan dua kromosom sebagai orang tua. Dalam AM, seleksi orang tua meruokan bagian dari evolusi yakni proses mendekati kandidat solusi permasalahan yang diharapkan. Berdasarkan teori evolusi, individu yang menghasilkan keturunan adalah individu yang terbaik dalam populasi. Untuk itu, pemilihan kromosom yang akan menjadi orang tua dilakukan secara proporsional sesuai dengan nilai *fitness*. Sehingga, semakin besar nilai *fitness* suatu kromosom, maka kemungkinan terpilih sebagai orang tua semakin besar juga. Terdapat beberapa metode pemilihan kromosom terbaik, diantaranya:

- *Roulette wheel selection*

Metode seleksi *roulette wheel* diadaptasi dari permainan *roulette*, dimana setiap individu akan mendapat bagian dari roda *roulette* dengan luas sebanding dengan nilai *fitness*-nya. Semakin tinggi nilai *fitness* individu, maka semakin luas bagian individu tersebut dalam roda *roulette*. Kemudian roda itu diputar dan bagian yang akhirnya ditunjuk akan terpilih menjadi orang tua. Dalam metode ini, individu dengan nilai *fitness* yang tinggi akan memiliki kemungkinan yang lebih besar untuk dipilih menjadi orang tua bagi generasi selanjutnya.

- *Rank selection*

Dengan menggunakan *rank selection*, semua kromosom dalam populasi di-*ranking* dari yang terburuk sampai yang terbaik. Setelah di-*ranking*, dilakukan prosedur yang sama dengan *roulette wheel selection*, dimana kromosom terburuk mendapat satu bagian potongan, kromosom terburuk kedua mendapat dua bagian potongan, dan seterusnya hingga kromosom terbaik mendapat N (jumlah kromosom dalam populasi) bagian potongan. Namun, metode ini membuat AM lambat dalam konvergensinya karena kromosom terbaik memiliki kemungkinan yang tidak terlalu berbeda dengan beberapa kromosom lain.

- *Tournament selection*

Dalam metode ini, diambil dua kromosom secara random dan kemudian menyeleksi salah satu yang bernilai *fitness* paling tinggi untuk menjadi orang tua pertama. Cara yang sama dilakukan lagi untuk mendapatkan orang tua yang kedua. Metode *tournament selection* yang lebih rumit adalah dengan mengambil m kromosom secara random. Kemudian kromosom bernilai *fitness* tertinggi dipilih sebagai orang tua pertama jika bilangan random yang dibangkitkan kurang dari suatu nilai batas yang ditentukan p dalam interval $[0,1]$. Pemilihan orang tua akan dilakukan secara random dari $m - 1$ kromosom yang ada jika bilangan random yang dibangkitkan lebih dari atau sama dengan p . Pada *tournament selection*, variabel m adalah *tournament size*, dan p adalah *tournament probability*. Biasanya m diset sebagai suatu nilai yang sangat kecil, misal 4 atau 5 sedangkan p biasanya diset sekitar 0,75.

- *Truncation selection*

Semua kromosom dalam populasi diurutkan berdasarkan nilai *fitness*-nya, kemudian dipilih sejumlah kromosom terbaik.

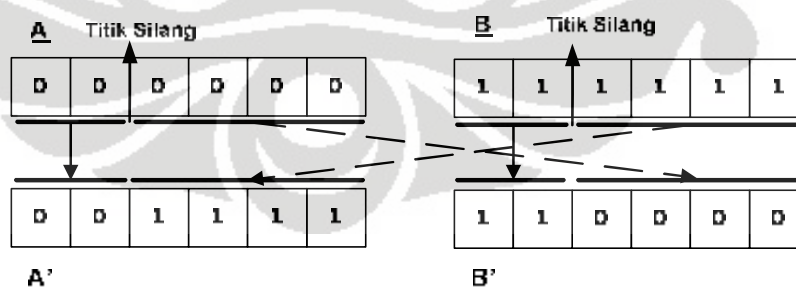
Dalam masalah penjadwalan mata kuliah, metode seleksi yang digunakan adalah *tournament selection*, untuk memastikan solusi yang terbaiklah yang dapat lanjut ke tahapan berikutnya¹⁸.

2.3.6 Crossover

Dalam operator *crossover*, dua orang tua yang telah dipilih akan dilakukan kawin silang untuk menghasilkan dua individu baru yang diharapkan akan mempunyai nilai *fitness* yang lebih tinggi. Terdapat beberapa jenis operator *crossover*, seperti:

- *One-point crossover*

Jenis *crossover* yang paling sederhana namun memiliki keefektifan yang tinggi yaitu *one-point crossover*¹⁹, dimana dalam *one-point crossover* dipilih secara acak titik silang pada kromosom kedua orang tua kemudian menukar silang kedua segmen pada titik ini untuk menghasilkan dua individu baru. Proses ini dilakukan sampai jumlah individu baru yang dihasilkan sama dengan jumlah individu pada populasi awal. Berikut merupakan sebuah contoh dari *one-point crossover* yang dilakukan terhadap dua buah individu yaitu A dan B.



Gambar 2.7 Contoh *One Point Crossover*

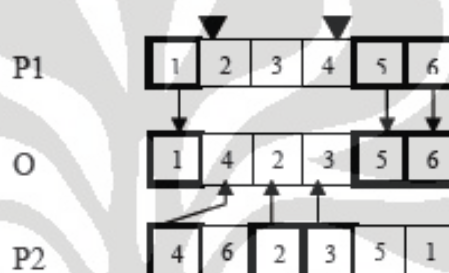
¹⁸ Ibid, hal.1.

¹⁹ Drake, A.E. & Marks, R.E. (1998). *Genetic Algorithms in Economic and Finance Forecasting Stock Market Prices and Foreign Exchange*.

Dari gambar tersebut, dilakukan *one-point crossover* terhadap individu A dan B berdasarkan titik silang yang dipilih secara acak. Dalam contoh ini, *one-point crossover* menghasilkan dua individu baru yaitu A' dan B'.

- *Two-point crossover*

Dengan metode ini dipilih dua titik secara random untuk membagi orang tua yang pertama. Nilai gen di luar sisi dua titik yang terpilih, diturunkan ke kromosom anak sesuai posisinya. Dari orang tua kedua, dilihat nilai gen yang lain, kemudian ditambahkan ke kromosom anak.



Gambar 2.8 Contoh *Two Point Crossover*

- *Uniform crossover*

Dengan metode ini dipilih gen-gen mana yang akan diturunkan ke kromosom anak.

- *Subsequence preservation crossover*

Metode ini dilakukan dengan memilih subbarisan dari orang tua pertama untuk diturunkan ke kromosom anak. Kemudian nilai gen yang lain dimasukkan dari orang tua kedua ke kromosom anak dengan cara *left to right*.

Dalam masalah penjadwalan mata kuliah, metode *crossover* yang digunakan adalah *uniform crossover*.

2.3.7 Mutasi

Mutasi adalah operator dasar genetika yang mengubah posisi gen dalam kromosom. Namun seperti halnya dalam evolusi biologi, tidak semua individu dalam suatu populasi akan mengalami mutasi. Jika terjadi mutasi terhadap suatu individu, nilai gen individu tersebut akan mengalami perubahan sehingga akan

menghasilkan individu yang berbeda. Probabilitas terjadinya mutasi pada suatu individu bernilai sangat kecil, biasanya sekitar 0,001 sampai 0,5. Tujuan dilakukan mutasi adalah untuk memelihara keanekaragaman kromosom pada populasi untuk mencegah terbentuknya populasi yang homogen, sehingga dapat menghindari kondisi optimal lokal. Terdapat beberapa jenis operator mutasi, diantaranya:

- *Swap mutation* : menukarkan dua gen yang berurutan
- *Exchange mutation* : menukar dua gen yang dipilih secara random
- *Scramble sublist mutation* : memilih dua titik secara random, kemudian mengubah susunan gen diantara dua titik tersebut secara random.
- *Shift mutation* : memilih satu gen secara random kemudian memasukkannya ke posisi yang dipilih secara random juga.

Dalam masalah penjadwalan ini, metode mutasi yang digunakan adalah *exchange mutation* dengan $P_m 0,5^{20}$.

2.3.8 Penggantian Populasi

Dalam algoritma genetika dikenal skema penggantian populasi, yang terdiri dari dua tipe, yaitu:

- *Generational replacement*

Metode penggantian populasi dengan mengganti semua individu (N individu) dari populasi lama dengan N individu baru hasil pindah silang dan mutasi. Hal ini menunjukkan bahwa jumlah populasi lama sama dengan populasi baru, atau dapat dinotasikan sebagai berikut:

$$| \text{populasi lama} | = | \text{populasi baru} |$$

- *Steady state replacement*

Metode penggantian populasi dengan hanya mengganti beberapa individu dari populasi lama dengan individu baru hasil pindah silang dan mutasi, sehingga jumlah individu yang diganti pada populasi lama sama dengan jumlah individu baru yang dihasilkan sehingga dapat dinotasikan:

$$| \text{populasi lama} | > | \text{populasi baru} |$$

²⁰ Ibid, hal.1.

Metode penggantian populasi dalam masalah penjadwalan mata kuliah ini adalah *steady state replacement*.

2.3.9 Kriteria Berhenti

Kriteria berhenti dalam AG digunakan untuk menentukan kapan AG akan berhenti. Kriteria berhenti yang umum digunakan yaitu maksimum jumlah generasi. Kriteria berhenti yang digunakan dalam penelitian ini adalah maksimum jumlah generasi.



BAB 3

PENGUMPULAN DATA

Pengumpulan data dalam penelitian ini dilakukan dengan menggunakan data sekunder dari universitas, yaitu berupa data ruang kuliah, data mahasiswa, data dosen, data slot waktu perkuliahan, dan data mata kuliah berdasarkan kurikulum tahun ajaran 2009/2010.

3.1 Data yang Diperlukan

3.1.1 Data Ruang Kuliah

Ruang kuliah memiliki kapasitas masing-masing yang nantinya akan disesuaikan dengan jumlah peserta mata kuliah dalam pengalokasiannya. Ruangan yang digunakan adalah ruangan dengan kapasitas 40 dan lebih (Lampiran 1). Namun karena penjadwalan mata kuliah ini akan dimodelkan dengan menggunakan data Departemen Teknik Industri UI dilakukan pengalokasian kebutuhan ruangan pada tiap departemen di FTUI. Oleh karena itu tidak semua ruangan di FTUI yang akan digunakan sebagai input dalam model penjadwalan ini.

3.1.2 Data Mahasiswa

Data mahasiswa yang dibutuhkan dalam penelitian ini adalah jumlah mahasiswa pada satu angkatan dalam satu jurusan. Untuk memperkecil ruang lingkup, model akan diimplementasikan pada kegiatan perkuliahan mahasiswa Departemen Teknik Industri (DTI). Oleh karena itu, data mahasiswa yang akan digunakan dalam penjadwalan adalah data mahasiswa DTI. Tabel 3.1 merupakan data jumlah mahasiswa DTI pada tahun ajaran 2009/2010.

Tabel 3.1 Jumlah Mahasiswa DTI

Angkatan	Jumlah
2006	85
2007	84
2008	104
2009	80

(Sumber: Departemen Teknik Industri, UI)

3.1.3 Data Dosen

Sama halnya dengan data mahasiswa, data dosen yang dibutuhkan untuk model penjadwalan adalah data dosen yang mengajar untuk kegiatan perkuliahan DTI. Data dosen dapat berupa nama dosen dan mata kuliah yang diajarkan. Terdapat beberapa dosen yang mengajar lebih dari satu mata kuliah tiap semesternya (Lampiran 2).

3.1.4 Data Mata Kuliah

Setiap mata kuliah memiliki bobot berbeda yang akan mempengaruhi jumlah kelas yang akan dibuka. Untuk mata kuliah yang memiliki bobot empat SKS (Sistem Kredit Semester) atau lebih biasanya akan dijadikan dua kali pertemuan sehingga akan menambah jumlah kelas yang akan dibuka. Selain itu, bobot SKS juga mempengaruhi lamanya jam pelajaran masing-masing mata kuliah sehingga penggunaan kelas harus disesuaikan juga dengan bobot setiap mata kuliah. Data mata kuliah diperoleh dari Buku Panduan Akademik S1 Reguler FTUI untuk tahun ajaran 2009-2010 (Lampiran 3).

3.1.5 Data Slot Waktu

Lama jam pelajaran setiap mata kuliah ditentukan berdasarkan bobot (SKS) dari masing-masing mata kuliah. Untuk mata kuliah berbobot empat SKS akan ditransformasikan menjadi dua kali pertemuan dalam seminggu sehingga lama jam pelajarannya menjadi sama dengan mata kuliah berbobot dua SKS, yaitu 110 menit. Sementara untuk mata kuliah berbobot tiga SKS akan dilaksanakan selama 150 menit. Tabel berikut menunjukkan pembagian slot waktu setiap hari perkuliahan.

Tabel 3.2 Slot Waktu Perkuliahan

Hari	Jam	Hari	Jam	Hari	Jam
Senin	08.00-09.50	Rabu	08.00-09.50	Jumat	07.30-09.20
	10.00-11.50		10.00-11.50		09.30-11.20
	13.00-15.30		13.00-15.30		13.30-16.00
	16.00-17.50		16.00-17.50		
Selasa	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		

(Sumber: Pusat Administrasi Fakultas FTUI, telah diolah kembali)

Untuk memperkecil ruang lingkup penjadwalan, model akan diimplementasikan pada kegiatan perkuliahan Departemen Teknik Industri semester ganjil. DTI dipilih sebagai data yang akan digunakan dalam model penjadwalan ini karena penjadwalan mata kuliah di DTI cukup kompleks dengan jumlah mata kuliah yang banyak namun dengan jumlah dosen pengajar yang sangat sedikit. Dimana rata-rata setiap dosen mengajar dua mata kuliah dan bahkan ada yang mengajar lima mata kuliah dalam tiap semesternya. Sementara setiap mata kuliah ditransformasikan menjadi dua kelas sehingga jumlah mata kuliah yang tidak boleh bentrok menjadi semakin bertambah banyak.

Oleh karena itu, agar kegiatan perkuliahan antar departemen tidak mengalami konflik maka dilakukan pembagian pemakaian ruangan kuliah untuk masing-masing departemen. Pembagian ruangan dilakukan berdasarkan banyaknya mata kuliah yang dibuka oleh masing-masing departemen. Jumlah ruangan yang akan dialokasikan adalah ruangan dengan kapasitas lebih atau sama dengan 40 karena dianggap memungkinkan untuk melakukan kegiatan perkuliahan. Tabel 3.3 menunjukkan jumlah kelas yang tersedia berdasarkan perhitungan jumlah ruangan dikalikan dengan jumlah slot waktu dalam seminggu.

Tabel 3.3 Jumlah Kelas yang Tersedia

Alokasi Penggunaan	Kapasitas Kelas	Jumlah Kelas	Penggunaan kelas dalam seminggu	Total kelas yang tersedia	TOTAL
Mata Kuliah Pilihan dan PDPT	40	27	19	513	551
	50	2	19	38	
Mata Kuliah Paralel	70	14	19	266	323
	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		55			1045

(Sumber: Pusat Administrasi Fakultas FTUI, telah diolah kembali)

Perhitungan jumlah kelas yang dibutuhkan oleh tiap departemen dilakukan berdasarkan jumlah mata kuliah yang dibuka oleh masing-masing departemen pada tiap semesternya.

Perhitungan kebutuhan kelas dilakukan dengan mengelompokkan mata kuliah berdasarkan jenis dan bobot SKS-nya. Untuk mata kuliah wajib

Tabel 3.4 Perhitungan Jumlah Kelas yang Dibutuhkan 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 PDPT ≤ 3sks	11		Non PDPT ≤ 3sks	11							
	Non PDPT ≥ 4sks	4	Non PDPT ≥ 4sks	5									
Genap	Wajib	PDPT	2	Wajib	PDPT	2	11	6	3	25	24	22	71
		Non PDPT ≤ 3sks	12		Non PDPT ≤ 3sks	14							
	Non PDPT ≥ 4sks	5	Non PDPT ≥ 4sks	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 PDPT ≤ 3sks	8		Non PDPT ≤ 3sks	14							
	Non PDPT ≥ 4sks	7	Non PDPT ≥ 4sks	3									
Genap	Wajib	PDPT	1	Wajib	PDPT	1	9	3	5	24	26	27	77
		Non PDPT ≤ 3sks	6		Non PDPT ≤ 3sks	7							
	Non PDPT ≥ 4sks	10	Non PDPT ≥ 4sks	10									
	Pilihan		16	Pilihan		7							

3. TEKNIK ELEKTRO/KOMPUTER

Semester	Elektro (Elktr)			Komputer (Kmptr)			Mata Kuliah Gabungan Elkr-Kmptr			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 PDPT ≤ 3sks	9		Non PDPT ≤ 3sks	12							
	Non PDPT ≥ 4sks	7	Non PDPT ≥ 4sks	6									
Genap	Wajib	PDPT	1	Wajib	PDPT	1	12	6	5	17	32	18	67
		Non PDPT ≤ 3sks	10		Non PDPT ≤ 3sks	16							
	Non PDPT ≥ 4sks	6	Non PDPT ≥ 4sks	6									
	Pilihan		19	Pilihan		1							

4. TEKNIK METALURGI

Semester	Metalurgi			Total Kelas yang Dibutuhkan NON-PDPT			TOTAL KEBUTUHAN KELAS
				40-50	70-80	100-140	
Ganjil	Wajib	PDPT	3	6	0	27	33
		Non PDPT ≤ 3sks	21				
	Non PDPT ≥ 4sks	3					
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									
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									
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							

(Sumber: Pusat Administrasi Fakultas FTUI, telah diolah kembali)

Tabel 3.4 Perhitungan Jumlah Kelas yang Dibutuhkan Tiap Departemen
(lanjutan)

7. TEKNIK INDUSTRI				Total Kelas yang Dibutuhkan NON-PDPT			TOTAL KEBUTUHAN KELAS
Semester	TI			40-50	70-80	100-140	
Ganjil	Wajib	PDPT	3	11	44	1	56
		Non PDPT≤3sks	1				
		Non PDPT (paralel)≥4sks	2				
		Non-PDPT Paralel≤3sks	18				
	Pilihan	11					
Genap	Wajib	PDPT	1	9	44	0	53
		Non PDPT≤3sks	0				
		Non PDPT (paralel)≥4sks	2				
		Non-PDPT Paralel≤3sks	18				
	Pilihan	9					

(Sumber: Pusat Administrasi Fakultas FTUI, telah diolah kembali)

dikelompokkan lagi menjadi tiga bagian berdasarkan jenis dan bobot, yaitu Program Dasar Pendidikan Tinggi (PDPT) dan Non-PDPT. Mata kuliah yang tergabung ke dalam PDPT adalah Mata Ajaran Pengembangan Kepribadian Terpadu (MPKT), MPK Agama, Bahasa Inggris, Olah Raga dan Seni. Dalam model penjadwalan mata kuliah PDPT tidak diikutsertakan sehingga dalam perhitungan kebutuhan kelas hanya dilakukan terhadap mata kuliah Non-PDPT. Mata kuliah Non-PDPT dibagi menjadi mata kuliah berbobot kurang dan sama dengan tiga SKS yang akan dialokasikan ke ruangan berkapasitas 100-140 sedangkan untuk mata kuliah berbobot lebih dan sama dengan empat SKS akan dialokasikan ke ruangan berkapasitas 70-80 dengan asumsi bahwa mata kuliah yang bersangkutan akan dibagi dalam dua kelas dan diselenggarakan sebanyak dua kali dalam seminggu. Untuk mata kuliah pilihan akan dialokasikan terhadap ruangan berkapasitas 40-50.

Untuk beberapa departemen yang memiliki bidang peminatan, terlebih dahulu dilakukan perhitungan terhadap mata kuliah yang diajarkan untuk kedua bidang peminatan (disebut dengan mata kuliah gabungan), kemudian dihitung jumlah kelas yang dibutuhkan departemen tersebut. Tabel 3.5 dan 3.6 menunjukkan perhitungan kebutuhan kelas untuk departemen yang memiliki bidang peminatan. Pada kenyataannya kapasitas maksimal mahasiswa untuk masing-masing kuliah adalah kurang dari 80 mahasiswa sehingga sebenarnya semua mata kuliah bisa dialokasikan terhadap kelas dengan berkapasitas kurang dari 80. Namun, untuk mengantisipasi kenaikan jumlah mahasiswa maka mata kuliah berbobot 2 dan 3 SKS dialokasikan terhadap kelas berkapasitas besar.

Tabel 3.5 Perhitungan Kebutuhan Kelas untuk Departemen dengan Bidang Peminatan

Semester	Elektro (k. ltr)			Komputer (kmptr)			Mata Kuliah Gabungan Elin. Kmptr			Total Kelas yang Dibutuhkan NDC-POPT			TOTAL KEBUTUHAN KELAS
	Wajib	POPT	Pilihan	Wajib	POPT	Pilihan	dengan POPT	Non-POPT < 4sk	Non-POPT > 4sk	40-50	70-80	100-140	
Ganjil	Wajib	POPT	11	Wajib	POPT	11	5	2	4	33	20	37	67
		Non-POPT	7		Non-POPT	8							
	Pilihan		13	Pilihan		11							
Genap	Wajib	POPT	11	Wajib	POPT	11	13	8	5	37	32	18	67
		Non-POPT	10		Non-POPT	15							
		Non-POPT	8		Non-POPT	8							
	Pilihan		16	Pilihan		11							

$6 - (12 - 5 - 3)$ $17 - (15 + 1)$ $32 - (6 * 2) + (5 * 2 * 2)$ $18 - (10 - 5) + (6 - 5) * 2 + (16 - 6) + (6 - 5) * 2$

Tabel 3.6 Perbandingan Kebutuhan Kelas dan Jumlah Kelas yang Tersedia

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	

Jumlah kelas yang dibutuhkan tidak lebih banyak dari jumlah kelas yang tersedia, sehingga pengalokasian ruangan untuk Departemen Teknik Industri dapat dilakukan seperti pada Tabel 3.7.

Tabel 3.7 Ruang yang Dialokasikan untuk Departemen Teknik Industri (Berdasarkan Kebutuhan Semester Ganjil)

Jumlah ruangan yang tersedia		Jumlah kelas yang tersedia dalam seminggu		Kebutuhan kelas di TI		Ruang yang dibutuhkan TI		Ruang buat TI berdasarkan banyak Dept.		Ruang yang dialokasikan buat TI
		110 menit	150 menit	110 menit	150 menit					
		R*12	R*7							
40-50	29	348	203	0	11	1.57143	2	4.1429	4	3
70-140	26	312	182	25	20	4.94048	5	3.7143	3	4

Untuk jumlah ruangan yang dibutuhkan TI merupakan hasil pembulatan atas pembagian “kebutuhan kelas di TI” dengan “jumlah slot dalam seminggu”. Terdapat 12 slot waktu yang berdurasi 110 menit dan 7 untuk slot yang berdurasi 150 menit. Sementara untuk “Ruang buat TI berdasarkan banyak Dept.” merupakan pembulatan atas pembagian “jumlah ruangan yang tersedia” dengan “jumlah departemen (yaitu 7)”. Oleh karena itu dipilih 7 kelas yang akan digunakan untuk menyelenggarakan perkuliahan DTI untuk semester ganjil.

Tabel 3.8 Ruang yang Dialokasikan ke DTI

Nama Ruangan	Kapasitas	Lokasi
K.108	40	GEDUNG KUL BERSAMA FT
K.209	50	GEDUNG KUL BERSAMA FT
K.211	50	GEDUNG KUL BERSAMA FT
K.206	80	GEDUNG KUL BERSAMA FT
K.207	80	GEDUNG KUL BERSAMA FT
GK.306	100	Gedung PAF
GK.301	100	Gedung PAF

3.2 Model Penjadwalan Kuliah di Departemen Teknik Industri UI

Seperti yang telah dijelaskan pada bab 2 terdapat dua jenis kendala yang harus diselesaikan dalam masalah penjadwalan ini, yaitu kendala *hard* dan kendala *soft*²¹. Adapun yang termasuk ke dalam kendala *hard* adalah:

²¹ Ibid, hal.13.

1. H1 : Tidak ada mahasiswa yang menghadiri lebih dari satu mata kuliah pada waktu yang bersamaan. Hal ini menyebabkan mata kuliah dalam tiap semester tidak boleh dilangsungkan pada waktu yang bersamaan.
2. H2 : Kapasitas ruangan kelas cukup untuk menyelenggarakan mata kuliah yang bersangkutan
3. H3 : Hanya satu mata kuliah yang dijadwalkan pada periode yang sama. (periode = tiap kombinasi ruangan dan slot)
4. H4 : Tidak ada dosen yang mengajar lebih dari satu mata kuliah pada waktu yang bersamaan. Hal ini menyebabkan mata kuliah-mata kuliah yang diajarkan oleh dosen yang bersangkutan tidak boleh dilangsungkan dalam waktu yang bersamaan.
5. H5 : Mata kuliah harus dilangsungkan pada slot waktu yang telah ditentukan .

Sementara yang menjadi kendala *soft* adalah:

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

Entitas yang akan digunakan dalam model penjadwalan mencakup sebagai berikut:

- Mata kuliah : $K = \{k_1, k_2, k_3, \dots, k_{56}\}$ (Lampiran 3)
- Ruang : $R = \{r_1, r_2, r_3, \dots, r_7\}$ (Tabel 3.8)
- Slot waktu : $T = \{t_1, t_2, t_3, \dots, t_{19}\}$ (Tabel 3.2)
- Dosen : $D = \{d_1, d_2, d_3, \dots, d_{22}\}$ (Lampiran 2)
- Mahasiswa : $M = \{m_1, m_2, m_3, m_4, m_5, m_6\}$

Mahasiswa dikelompokkan menjadi enam karena terdapat mata kuliah yang diselenggarakan sebanyak dua kali untuk mahasiswa tingkat satu dan tingkat dua. Oleh karena itu m_1 dan m_2 merupakan mahasiswa tingkat satu, m_2 dan m_3 merupakan mahasiswa tingkat dua, m_5 merupakan mahasiswa tingkat tiga dan m_6 merupakan mahasiswa tingkat empat.

Komponen yang digunakan dalam formulasi model adalah sebagai berikut²²:

J : Himpunan semua jadwal yang mungkin

$$J = \{j_1, j_2, j_3, \dots, j_n\}$$

H : Himpunan kendala *hard*

$$H = \{h_1, h_2, h_3, h_4, h_5\}$$

S : Himpunan kendala *soft*

$$S = \{s_1, s_2, s_3\}$$

M : Himpunan kelompok mahasiswa

$$M = \{m_1, m_2, m_3, m_4\}$$

K : Himpunan mata kuliah

$$K = \{k_1, k_2, k_3, \dots, k_n\}$$

Jadwal yang mungkin adalah jadwal yang mungkin menjadi solusi, namun ada kemungkinan jadwal tersebut masih melanggar kendala *hard* dan *soft*. Sementara jadwal yang layak adalah jadwal yang mungkin menjadi solusi dan sudah tidak melanggar kendala *hard*, namun masih ada kemungkinan melanggar kendala *soft*.

3.2.1 Fungsi Penalti

Untuk memberikan jadwal yang layak, dalam model ini akan diberlakukan sistem penalti. Dimana jadwal yang merupakan calon solusi akan dikenakan penalti apabila melanggar kendala *hard* dan kendala *soft*. Berikut merupakan sistem penalti yang akan digunakan dalam model.

Penalti kendala *hard*

$$\forall i \in H \text{ dan } i \in S$$

$$f_i(k) = \alpha_i(k) + \beta_i(k) + \gamma_i(k) + \delta_i(k) + 5 \epsilon_i(k)$$

$$f_i(k) = \epsilon_i(k)$$

$$f_i(k) = \begin{cases} 1, & \text{jika kuliah } k \text{ melanggar kendala } h_i \text{ ke } i \\ 0, & \text{jika kuliah } k \text{ tidak melanggar kendala } h_i \text{ ke } i \end{cases}$$

$$i = 1, 2, 3, 4, 5$$

²² Lismanto. (2008). *Penjadwalan Kuliah dengan Algoritma Memetika*. Universitas Indonesia, Fakultas Matematika dan Ilmu Pengetahuan, Departemen Matematika.

Jadwal j disebut layak apabila $\sum f(k) = 0$

$f(k)$ = penalti kendala *hard* untuk kuliah ke- k

$F(j)$ = total penalti kendala *hard* untuk jadwal j

Jadwal j disebut layak apabila $F(j) = 0$

Penalti kendala *soft*

$\forall i \in I$ dan $k \in K$

$$g_i(k) = \alpha_i(k) + \beta_i(k)$$

$$g_i(k) = \begin{cases} 1, & \text{jika kuliah } k \text{ melanggar kendala } i \\ 0, & \text{jika kuliah } k \text{ tidak melanggar kendala } i \end{cases}, \text{ dimana}$$

$$g_i(k) = \begin{cases} 1, & \text{jika kuliah } k \text{ melanggar kendala } i \\ 0, & \text{jika kuliah } k \text{ tidak melanggar kendala } i \end{cases} \text{ ke- } i \\ i = 1, 2$$

$g(k)$ = penalti kendala *soft* untuk kuliah ke- k

$G(j)$ = total penalti kendala *soft* untuk jadwal j

Fungsi tujuan permasalahan ini adalah

$$\min z = F(j) + G(j)$$

BAB 4

PENGOLAHAN DATA DAN ANALISIS

4.1 Penyusunan Algoritma

Model penjadwalan mata kuliah diselesaikan dengan menggunakan algoritma metaheuristik yaitu algoritma memetika yang merupakan penggabungan antara algoritma genetika dengan metode pencarian lokal (*local search*) seperti yang telah dijelaskan pada Bab 2. Terdapat beberapa tahapan yang dibutuhkan dalam penyelesaian algoritma tersebut yang akan dibuat dengan program komputer menggunakan *software* MATLAB 7.6.0(R2008a).

MATLAB adalah bahasa komputasi teknik tinggi dan merupakan lingkungan yang interaktif untuk pengembangan algoritma, visualisasi data, analisis data, dan komputasi numerik. 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 ilmiah²³.

4.1.1 Langkah-langkah Penyusunan Algoritma Memetika

Pembuatan model penjadwalan mata kuliah didasarkan pada fungsi tujuan yaitu meminimumkan jumlah penalti kendala *soft*. Seperti telah dijelaskan sebelumnya bahwa terdapat beberapa tahapan dalam penyusunan algoritma (Gambar 4.1). Prosedur penyelesaiannya dapat diuraikan sebagai berikut:

1. Menentukan parameter/variabel yang digunakan yaitu:
 - Ukuran Populasi

Seperti yang telah dijelaskan pada Bab 2, pembentukan populasi awal merupakan tahap awal karena algoritma memetika bekerja dengan membentuk suatu populasi bukan solusi tunggal, sehingga perlu ditetapkan ukuran populasi permasalahan. Dalam masalah ini, digunakan ukuran

²³ www.mathworks.com

populasi sebesar 16 karena semakin besar ukuran populasi menyebabkan solusi yang dihasilkan kurang baik²⁴.

- Probabilitas mutasi

Dipilih probabilitas mutasi sebesar 0,5 berdasarkan literatur dengan permasalahan yang sama²⁵.

- Maksimum iterasi

Maksimum iterasi dapat dijadikan menjadi kriteria berhenti jalannya program. Jumlah iterasi akan mempengaruhi hasil model, oleh karena itu dalam masalah ini akan dibandingkan hasil dari model dengan jumlah iterasi yang berbeda-beda.

2. Membuat matrik-matrik yang akan menjadi input dalam model penjadwalan.

Data yang menjadi input awal pembuatan model adalah:

- Matrik KM merupakan matrik hubungan antara mata kuliah dengan jumlah mahasiswa pengambil mata kuliah tersebut. Matrik KM merupakan matrik berukuran [jumlah mata kuliah x jumlah kelompok mahasiswa]. Hubungan antara mata kuliah dengan kelompok mahasiswa dilambangkan dengan notasi (n) atau (0), berdasarkan keterangan berikut:

$$KM(i,j) = \begin{cases} 0, & \text{jika mata kuliah ke-} i \text{ tidak diambil oleh kelompok mahasiswa ke-} j \\ n, & \text{jika mata kuliah ke-} i \text{ diambil oleh kelompok mahasiswa ke-} j \end{cases}$$

Dimana n merupakan jumlah mahasiswa pengambil mata kuliah tersebut.

- Matrik KT merupakan matrik hubungan antara mata kuliah dengan slot waktu yang cocok untuk melangsungkan mata kuliah yang bersangkutan. Matrik KT merupakan matrik berukuran [jumlah mata kuliah x jumlah slot] dalam seminggu. Hubungan antara mata kuliah dengan slot waktu dilambangkan dengan notasi (1) atau (0), berdasarkan keterangan berikut:

²⁴ Ibid, hal.22.

²⁵ Ibid, hal.1.

$$KT(i,j) = \begin{cases} 0, & \text{jika mata kuliah ke } - \text{ tidak dapat dilangsungkan pada slot ke } - \\ 1, & \text{jika mata kuliah ke } - \text{ dapat dilangsungkan pada slot ke } - \end{cases}$$

- Matrik KK merupakan matrik hubungan antara mata kuliah yang satu dengan mata kuliah lainnya yang berukuran [jumlah mata kuliah x jumlah mata kuliah]. Hubungannya dapat dilambangkan dengan notasi (1) atau (0), berdasarkan keterangan berikut:

$$KK(i,j) = \begin{cases} 0, & \text{jika mata kuliah ke } - \text{ dapat dilangsungkan pada slot yang sama} \\ & \text{dengan mata kuliah ke } - \\ 1, & \text{jika mata kuliah ke } - \text{ tidak dapat dilangsungkan pada slot yang} \\ & \text{sama dengan mata kuliah ke } - \end{cases}$$

Berdasarkan matrik diatas maka dibuat matrik KR dengan menggunakan MATLAB, matrik KR merupakan matrik hubungan antara mata kuliah dengan ruangan yang dinotasikan dengan (1) dan (0), berdasarkan keterangan berikut:

$$KR(i,j) = \begin{cases} 0, & \text{jika mata kuliah ke } - \text{ tidak dapat dilangsungkan pada ruang} \\ & \text{ke } - \\ 1, & \text{jika mata kuliah ke } - \text{ dapat dilangsungkan pada ruang} \\ & \text{ke } - \end{cases}$$

3. Menentukan kromosom/populasi awal secara acak.

Populasi awal inilah yang nantinya akan diperbaiki dengan menggunakan pencarian lokal (*local search*) dan operator genetika yang ada. Penentuan populasi awal dilakukan berdasarkan matrik KT, KR, dan matrik irisan bentrok (MIB). Matrik irisan bentrok merupakan matrik berukuran [jumlah mata kuliah x jumlah mata kuliah] yang berisi hubungan antar mata kuliah berdasarkan jumlah mahasiswa yang akan mengalami bentrok/konflik apabila mengambil mata kuliah yang bersangkutan. Matrik ini dibuat berdasarkan matrik KM. Perhitungan terhadap matrik irisan bentrok dilakukan dengan membandingkan kelompok mahasiswa pengambil mata kuliah i dan j . Yang menjadi irisan antara kelompok mahasiswa tersebut menjadi notasi yang melambangkan hubungan antar mata kuliah pada matrik tersebut. Perhitungannya dapat dilihat pada contoh pada Tabel 4.1 dan 4.2. Setelah membuat matrik irisan bentrok maka selanjutnya dihitung total bentrok yang terjadi antar mata kuliah disebut dengan matrik total bentrok (MTB). Fungsi matrik ini adalah untuk menentukan mata kuliah mana yang akan dijadwalkan terlebih dahulu. Mata kuliah yang memiliki jumlah bentrok terbesar akan dijadwalkan terlebih dahulu. Dengan

menggunakan input matrik KR, KT, dan MTB dibuat populasi awal dengan ukuran 16. Jadwal yang dihasilkan dalam populasi awal berupa kromosom (Tabel 4.3) dan matrik (Tabel 4.4).

Tabel 4.1 Contoh Matrik KM

Sem	Mata Kuliah	Notasi	Sks	Kelompok Mahasiswa					
				m1	m2	m3	m4	m5	m6
1	Kalkulus A	1	4	35	0	20	0	0	0
	Pengantar Teknik Industri	2	2	35	55	10	0	0	0
3	Fisika Dasar 2 A	3	4	0	0	35	0	20	0
	Programa Linier A	4	3	0	0	35	0	20	0

Tabel 4.2 Contoh Matrik Irisan Bentrok

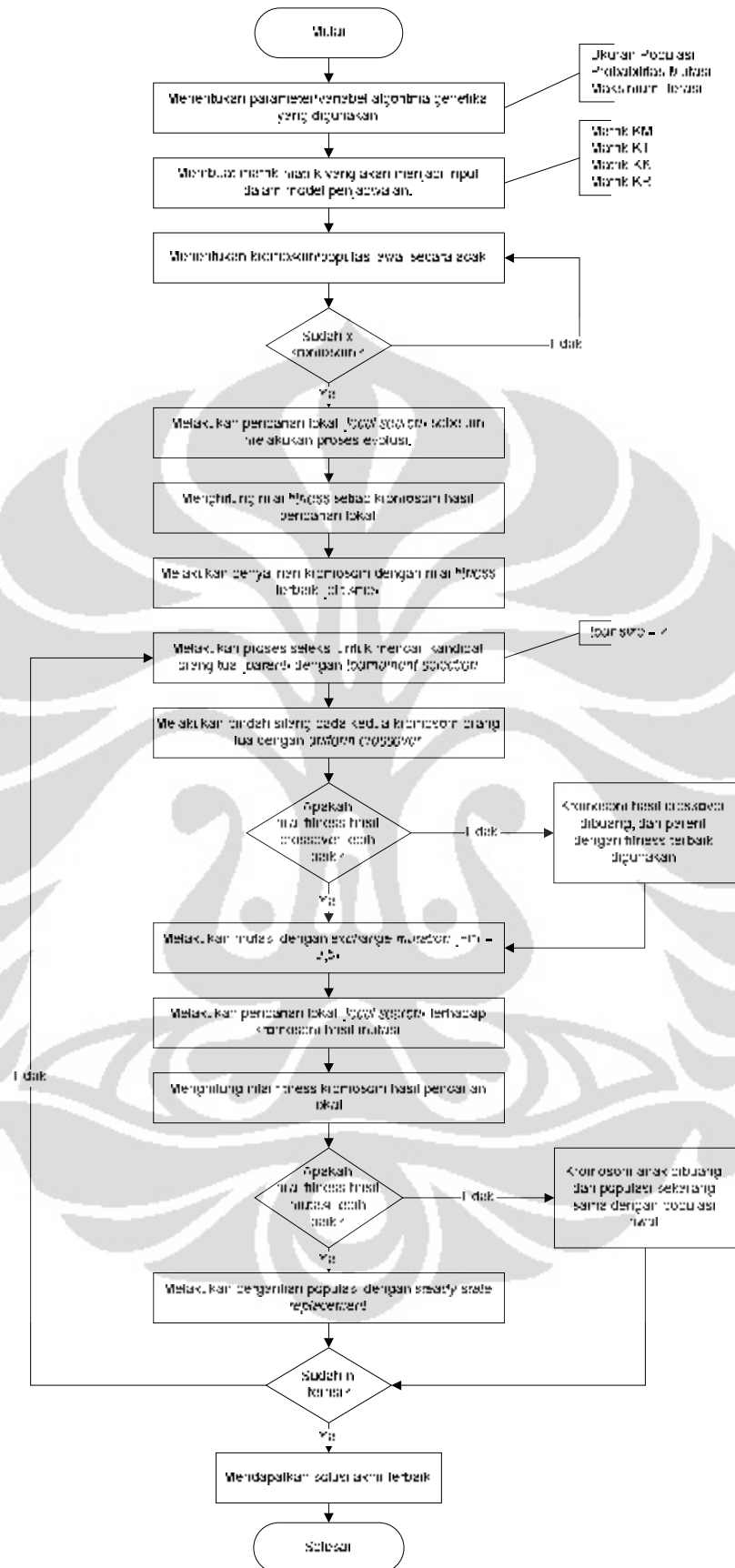
Sem	Mata Kuliah	Notasi	Sks	Notasi			
				1	2	3	4
1	Kalkulus A	1	4	0	45	20	20
	Pengantar Teknik Industri	2	2	45	0	10	10
3	Fisika Dasar 2 A	3	4	20	10	0	55
	Programa Linier A	4	3	20	10	55	0

Tabel 4.3 Matrik Jadwal (MJ berukuran Ruang x Slot)

Jadwal	Slot							
	1	2	18	19
R u a n g	1							
	2							
	3							
	4							
	5							
	6							
	7							

Tabel 4.4 Kromosom Jadwal (KJ berukuran 2 x 56)

	1	2	55	56
Ruang										
Slot										



Gambar 4.1 Algoritma Memetika Penjadwalan Mata Kuliah

4. Melakukan pencarian lokal (*local search*) sebelum melakukan proses evolusi.

Pencarian lokal dilakukan dengan tujuan untuk menghasilkan solusi yang layak (*feasible solution*). Namun karena pada kenyataannya sangat susah dalam menghasilkan jadwal yang layak disebabkan jumlah mata kuliah yang tidak dapat dilangsungkan pada slot yang sama tidak sebanding dengan jumlah slot yang tersedia sehingga kemungkinan melanggar kendala *hard* masih ada. Oleh karena itu, pencarian lokal bertujuan untuk meminimalkan penalti kendala *hard*. Peminimalan penalti kendala *hard* dilakukan dengan cara melakukan pertukaran-pertukaran gen untuk memberikan nilai penalti kendala *hard* terkecil. Yang menjadi input dalam pencarian lokal ini adalah jadwal dalam populasi awal, matrik KK, KT, KM dan KR. Pencarian lokal ini dilakukan terhadap semua kromosom pada populasi. Pencarian lokal diawali dengan melakukan evaluasi nilai penalti kendala *hard* pada jadwal. Evaluasi terhadap penalti kendala *hard* dilakukan dengan:

- $fk1$ = penalti terhadap jadwal yang melanggar kendala *hard* pertama, ketiga, dan keempat

Untuk menghitung penalti yang pertama dilihat dari matrik KK dan jadwal yang berukuran $(R \times T)$. Pada jadwal dicek jika pada slot yang sama terdapat dua mata kuliah atau lebih, maka mata kuliah pada slot tersebut akan dibandingkan satu sama lain berdasarkan matrik KK. Jika pada matrik KK hubungan antar mata kuliah yang bersangkutan adalah (1) maka mata kuliah tersebut akan dipenalti dengan memberikan nilai (1) pada masing-masing mata kuliah yang melanggar kendala *hard* (1), (3), dan (4). Pemberian penalti untuk kendala ini bertujuan untuk memastikan mata kuliah dalam tiap semester dan yang diajarkan oleh dosen yang sama tidak boleh dilangsungkan pada waktu yang bersamaan agar memberikan jadwal yang layak.

- $fk2$ = penalti terhadap jadwal yang melanggar kendala *hard* kedua

Untuk menghitung penalti yang kedua dilihat dari matrik KR dan jadwal yang berukuran $(R \times T)$. Pada jadwal dicek apakah mata kuliah

ke- i yang dijadwalkan pada ruangan ke- j memenuhi matrik KR atau tidak. Jika tidak maka mata kuliah tersebut dipenalti dengan memberikan nilai (1).

- $fk3$ = penalti terhadap jadwal yang melanggar kendala hard kelima

Untuk menghitung penalti yang ketiga dilihat dari matrik KT dan jadwal (RxT). Pada jadwal dicek apakah mata kuliah ke- i yang di-assign pada slot ke- j memenuhi matrik KT atau tidak. Jika tidak maka mata kuliah tersebut dipenalti dengan memberikan nilai (1).

Kemudian dihitung total penalti *hard* terhadap jadwal, $fk = fk1 + fk2 + fk3$. Pencarian lokal dilakukan dengan dua tahap, yaitu:

- Menukar posisi ruang yang memiliki penalti kendala *hard* ($fk2$) pada masing-masing kromosom. Pertukaran tersebut akan menghasilkan jadwal yang tidak melanggar kendala *hard* kedua. Pertama-tama pada setiap gen (setiap mata kuliah) dicek apakah kombinasi mata kuliah dan ruang yang ada melanggar kendala *hard* kedua atau tidak. Jika tidak, maka tidak perlu dilakukan penggantian ruang. Namun jika ya, maka kombinasi tersebut harus diganti dengan melihat kembali kemungkinan-kemungkinan kombinasi yang memungkinkan pada matrik KR.

	1	2	3	55	56
Ruang		2	4							
Slot		2	2							

Walaupun mata kuliah ke 2 memiliki penalti kendala hard 2, maka dilakukan swap, dengan mengeser Matrik KR

Notasi	Nama Ruangan						
	1	2	3	4	5	6	7
1	0	0	0	1	1	0	0
2	0	0	0	1	1	0	0
..							
..							
56	1	1	1	1	1	1	1

Gambar 4.2 Pencarian Lokal Satu

Pada matrik KR dilakukan pengecekan kemungkinan kombinasi mata kuliah-ruang yang cocok, dan ternyata mata kuliah ke-2 dapat dilangsungkan di ruang ke-4 (pencarian lokal dilakukan dengan melihat

kemungkinan pada setiap slot), sehingga gen ruang untuk mata kuliah ke-2 diganti menjadi 4. Begitu seterusnya hingga menghasilkan kombinasi yang tidak melanggar kendala *hard* ke-2.

	1	2	3	55	56
Ruang		4	4							
Slot		2	2							

Gambar 4.3 Hasil Pencarian Lokal Satu

- Menukar kombinasi ruang dan slot dari tiap gen yang memiliki penalti $fk1 + fk3$. Hasil dari pencarian lokal yang pertama akan dievaluasi lagi nilai penaltinya, karena ada kemungkinan memberikan jadwal dengan periode yang sama seperti pada Gambar 4.3. Hal ini mengakibatkan terdapat mata kuliah yang tidak terjadwal karena telah digantikan oleh mata kuliah lainnya yang juga dijadwalkan pada periode (kombinasi ruang dan slot) yang sama. Oleh karena itu terlebih dahulu dilakukan pengumpulan mata kuliah apa saja yang tidak terjadwal kemudian dijadwalkan kembali dengan mengganti kombinasi ruang dan slotnya. Penggantian dilakukan secara acak namun dengan tetap menghindari pelanggaran terhadap kendala *hard*.

	1	2	3	55	56
Ruang		4	4							
Slot		2	2							

Misalnya: mata kuliah ke 2 dan ke 3 melanggar kendala ke-1, maka dilakukan LS2, dengan mengecek Matrik k_{11} dan k_{13} .



Notasi	Slot																		
	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
2	1	1	0	1	1	0	0	1	1	1	0	1	1	0	0	1	1	1	0
3	1	1	0	1	1	0	0	1	1	1	0	1	1	0	0	1	1	1	0
..																			
..																			
55	0	0	1	0	0	1	1	0	0	0	1	0	0	1	1	0	0	0	1
56	0	0	1	0	0	1	1	0	0	0	1	0	0	1	1	0	0	0	1

Gambar 4.4 Pencarian Lokal Dua

Setelah mata kuliah yang tidak terjadwal dikumpulkan maka akan dilakukan penjadwalan ulang dengan menjadwalkan kembali ruang dan

slot yang memenuhi terhadap mata kuliah yang bersangkutan. Pada matrik KT dilakukan pengecekan kemungkinan kombinasi mata kuliah-slot yang cocok, dan ternyata mata kuliah ke-2 dapat juga dilangsungkan di slot ke-1, ke-2, ke-4 dan seterusnya, sehingga gen slot untuk mata kuliah ke-2 diganti secara acak terhadap slot yang memenuhi. Kemudian dilakukan pengecekan lagi terhadap matrik KK, apakah hasil pertukaran tersebut melanggar kendala *hard* ke-1 atau tidak, jika tidak maka hasil pencarian lokal 2 digunakan. Setelah itu, hal yang sama dilakukan lagi namun dengan melihat matrik KR untuk penggantian ruang. Begitu seterusnya hingga dihasilkan jadwal dengan total penalti $fk1 + fk3$ minimal. Hasil dari pencarian lokal kedua ini adalah memastikan semua mata kuliah terjadwal pada periode yang berbeda.

	1	2	3	55	56
Ruang		4	4							
Slot		1	2							

Gambar 4.5 Hasil Pencarian Lokal Dua

5. Membuat salinan dari hasil pencarian lokal (*Elitisme*)

Karena terdapat pemilihan secara random pada tahap berikutnya, 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 dan mutasi. Oleh karena itu perlu dibuat kopian dari individu hasil pencarian lokal.

6. Melakukan evaluasi nilai *fitness* terhadap kromosom hasil pencarian lokal. Evaluasi nilai *fitness* dilakukan dengan menghitung penalti kendala *hard* dan *soft* pada kromosom hasil pencarian lokal.

- $gk1$ = penalti untuk kendala *soft* pertama.

Untuk menghitung penalti yang pertama dilihat dari jadwal (RxT). Pada jadwal dicek berapa kali kemunculan tiap mahasiswa pada tiap *range* slot waktu dengan interval empat. Jika jumlah kemunculan masing-

masing kelompok mahasiswa pada tiap range = 1, maka akan dikenakan penalti dengan memberikan nilai (1). Sehingga pengecekan terhadap penalti kendala *soft* dilihat dari tiap kelompok slot (yaitu ditinjau dari tiap slot pada satu hari).

- $gk2$ = penalti untuk kendala *soft* kedua.

Untuk menghitung penalti yang kedua dilihat dari jadwal (RxT). Pada jadwal dicek apakah pada tiap slot kelipatan empat terdapat jam pelajaran atau tidak. Jika iya maka dikenakan penalti dengan memberikan nilai (1).

Kemudian dihitung total penalti *soft* terhadap jadwal, $gk = gk1 + gk2$. Karena fungsi tujuan dari permasalahan ini adalah meminimalkan total penalti kendala *hard* dan *soft*, maka nilai *fitness* dapat dihitung dengan $1/(fk+gk+1)$.

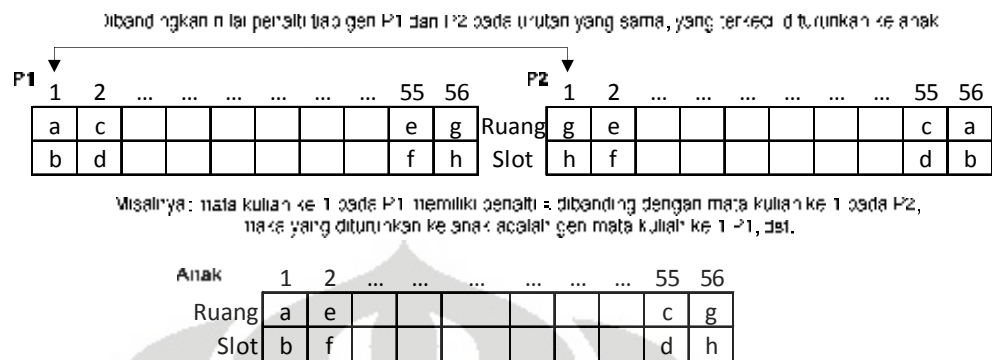
7. Melakukan proses seleksi untuk mencari kandidat orang tua (*parent*).

Proses seleksi dilakukan dengan memilih dua kandidat jadwal (dua kromosom) sebagai orang tua. Yang menjadi input dalam proses seleksi adalah jadwal hasil pencarian lokal. Sistem pemilihan dilakukan dengan *tournament selection* dengan $n = 2$. Dimana dari populasi pertama-tama dipilih dua kromosom secara random kemudian dibandingkan nilai *fitness*-nya. Kromosom dengan nilai *fitness* terbaik dijadikan sebagai orang tua pertama. Hal yang sama dilakukan untuk mencari orang tua kedua.

8. Melakukan pindah silang (*crossover*).

Pindah silang dilakukan dengan tujuan untuk meminimalkan penalti kendala *hard* dan *soft*. Yang menjadi input dalam proses *crossover* adalah dua kromosom orang tua dengan ukuran (2x56). *Crossover* dilakukan dengan metode *uniform crossover*, dimana gen yang akan diturunkan ke individu baru ditentukan sendiri. Dihitung penalti kendala *hard* dan *soft* kedua ($gk2$) kromosom orang tua, dari masing-masing kromosom dicek tiap gennya sehingga yang memiliki penalti terkecil yang akan diturunkan ke kromosom anak. Jika hasil *crossover* memiliki nilai *fitness* yang lebih tinggi dari *parent* maka akan dilanjutkan ke proses mutasi, namun jika

tidak maka yang maju ke tahap berikutnya adalah *parent* dengan *fitness* terbaik.



Gambar 4.6 Crossover

9. Melakukan mutasi.

Mutasi dilakukan dengan menukar-nukarkan gen pada kromosom anak. Metode mutasi yang digunakan yaitu *exchange mutation* dengan $P_m = 0,5$.

10. Melakukan pencarian lokal (*local search*) setelah melakukan proses evolusi (GA).

Pencarian lokal ini bertujuan untuk memperbaiki nilai *fitness* dari kromosom hasil mutasi. Pencarian lokal ini hanya dilakukan untuk kromosom hasil mutasi tidak untuk semua kromosom pada populasi. Metode pencarian lokal yang dilakukan pada tahapan ini sama dengan metode pencarian lokal sebelum proses evolusi. Dalam tahap ini, dipastikan bahwa semua mata kuliah harus terjadwal pada periode yang berbeda. Hasil tahapan ini disebut dengan anak (*child*).

11. Melakukan pergantian populasi.

Pergantian populasi dilakukan dengan mengganti kromosom terburuk pada populasi dengan hasil pencarian lokal kedua (*child*).

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.

4.1.2 Verifikasi dan Validasi Program

Seperti yang telah dijelaskan sebelumnya bahwa masalah penjadwalan ini akan diselesaikan dengan menggunakan *software* MATLAB. Oleh karena itu, perlu dilakukan verifikasi dan validasi terhadap program yang telah dibuat terlebih dahulu. Tahap verifikasi merupakan tahap melihat kesesuaian antara model program yang didapat dengan konseptual model yang kita buat atau yang sesuai dengan yang kita harapkan. Parameter model program dikatakan telah terverifikasi apabila telah berjalan sesuai dengan konseptual model, dimana akan dihasilkan nilai/*output* yang berbeda-beda jika kita mengganti parameter-parameter kontrol pada program. Dengan demikian program dikatakan telah terverifikasi. Untuk melihat perbedaan hasil dengan beberapa kombinasi parameter terdapat pada Tabel 4.5.

Selanjutnya untuk validasi program dapat dilakukan dengan menggunakan data *dummy*. Digunakan data *dummy* bertujuan untuk mengecilkan ruang lingkup permasalahan sehingga memungkinkan untuk menyelesaikannya secara manual. Pengecilan ruang lingkup dilakukan dengan mengurangi jumlah mata kuliah, ruang, dan slot yang akan dijadwalkan. Pada awalnya pengecilan ruang lingkup permasalahan dilakukan dengan menggunakan perbandingan senilai dalam penentuan jumlahnya agar mewakili masalah yang sebenarnya. Namun menyebabkan masalahnya tidak dapat diselesaikan sebab jumlah slot yang tersedia menjadi tidak seimbang dengan jumlah mata kuliah yang akan dijadwalkan. Hal ini dapat dilihat dari perhitungan berikut:

Jumlah Mata Kuliah	Jumlah Ruang	Jumlah Slot (dalam seminggu)
56	7	19
10	2	4

Dapat dilihat dari perbandingan diatas, permasalahan penjadwalan sebenarnya adalah menjadwalkan 56 mata kuliah terhadap 133 periode (periode = kombinasi ruang dan slot, sehingga jumlahnya adalah $7 \times 19 = 133$). Hal ini memungkinkan dilakukannya penjadwalan, karena jumlah periode yang tersedia jauh lebih kecil dibandingkan dengan jumlah mata kuliah yang harus dijadwalkan. Sementara dengan melakukan perbandingan senilai (namun dilakukan pembulatan

keatas), menghasilkan terdapat 10 mata kuliah yang akan dijadwalkan pada 8 periode ($2 \times 4 = 8$). Hal ini jelas tidak dapat diselesaikan dengan karena jumlah periode yang tersedia lebih sedikit dibanding jumlah mata kuliah yang harus dijadwalkan. Oleh karena itu, untuk menentukan jumlah data *dummy* yang akan digunakan dalam pemvalidasian tidak dilakukan dengan menggunakan perbandingan senilai.

Tabel 4.5 Skenario Parameter

No.	Kombinasi			Hasil						Total Penalti Rata-rata (dalam ukPop yang sama)
	Ukpop	Pmutasi	Iterasi	Lama Running (detik)	Penalti Hard	Penalti Soft	Fitness	Mata Kuliah Tidak Terjadwal	Total Penalti	
1	16	0.2	50	14.88	33	8	0.0238	0	41	17.2
2			100	24.53	15	10	0.0385	0	25	
3			500	74.34	8	10	0.0526	0	18	
4			1000	147.43	7	8	0.0625	0	15	
5			1200	161.49	9	9	0.0526	0	18	
6			1300	168.92	4	7	0.0833	0	11	
7			1400	195.65	2	8	0.0909	0	10	
8			1500	199.78	3	6	0.1000	0	9	
9			1600	202.44	3	9	0.0769	0	12	
10			2000	243.72	3	10	0.0714	0	13	
11	16	0.3	50	12.26	22	10	0.0303	0	32	15.6
12			100	16.21	19	13	0.0303	0	32	
13			500	57.89	11	6	0.0556	0	17	
14			1000	121.44	3	8	0.0833	0	11	
15			1200	134.62	5	7	0.0769	0	12	
16			1300	163.84	4	9	0.0714	0	13	
17			1400	169.19	3	7	0.0909	0	10	
18			1500	199.43	2	8	0.0909	0	10	
19			1600	188.10	2	9	0.0833	0	11	
20			2000	238.07	2	6	0.1111	0	8	
21	16	0.4	50	13.30	34	7	0.0238	0	41	16.2
22			100	16.26	15	11	0.0370	0	26	
23			500	67.78	9	12	0.0455	0	21	
24			1000	133.41	5	9	0.0667	0	14	
25			1200	127.37	2	6	0.1111	0	8	
26			1300	164.16	1	6	0.1250	0	7	
27			1400	165.57	4	7	0.0833	0	11	
28			1500	167.57	5	7	0.0769	0	12	
29			1600	187.83	3	8	0.0833	0	11	
30			2000	246.68	7	4	0.0833	0	11	
31	16	0.5	50	18.12	22	9	0.0313	0	31	13.1
32			100	16.38	13	6	0.0500	0	19	
33			500	45.40	6	9	0.0625	0	15	
34			1000	94.30	1	12	0.0714	0	13	
35			1200	103.37	1	6	0.1250	0	7	
36			1300	119.70	2	6	0.1111	0	8	
37			1400	147.74	2	5	0.1250	0	7	
38			1500	153.33	3	6	0.1000	0	9	
39			1600	142.64	4	9	0.0714	0	13	
40			2000	173.96	4	5	0.1000	0	9	

Data *dummy* yang digunakan adalah 10 mata kuliah, 3 ruangan, 6 kelompok mahasiswa, dan 7 slot waktu. Data *dummy* yang digunakan untuk

validasi program ditunjukkan pada Tabel 4.6. Hasil *run* program dengan data *dummy* kemudian akan dibandingkan dengan perhitungan manual. Jika hasil keduanya sama, maka program telah tervalidasi.

Tabel 4.6 Parameter yang Digunakan dalam Validasi

Parameter	Nilai
Ukuran populasi	4
Maksimum iterasi	1
Probabilitas mutasi	0.5

4.1.2.1 Hasil perhitungan manual

Perhitungan manual dilakukan dengan menggunakan populasi awal yang diperoleh dari program. Berikut merupakan tahapan perhitungan manual yang dilakukan:

- Inisialisasi populasi awal

Tabel 4.7 Data *Dummy*

Sem.	Mata Kuliah	Not.	Sks	Matrik KT						Matrik KM						
				Senin			Selasa			m1	m2	m3	m4	m5	m6	
				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
				1	2	3	4	5	6	7						
1	Kalkulus A	1	4	1	1	0	1	1	0	0	35	0	20	0	0	0
	Pengantar Teknik Industri B	2	2	1	1	0	1	1	0	0	0	55	0	0	0	0
3	Fisika Dasar 2A (Gelombang)	3	4	1	1	0	1	1	0	0	0	0	35	0	20	0
	Programa Linier B	4	3	0	0	1	0	0	1	1	0	0	0	55	0	0
5	Pemasaran Industri A	5	3	0	0	1	0	0	1	1	0	0	0	0	35	20
	Sistem Informasi B	6	3	0	0	1	0	0	1	1	0	0	0	0	55	0
7	Perancangan Proses A	7	2	1	1	0	1	1	0	0	0	0	0	0	0	45
	Manajemen Proyek B	8	3	0	0	1	0	0	1	1	0	0	0	0	0	45
Pilihan	Manajemen Energi	9	3	0	0	1	0	0	1	1	0	0	0	0	40	0
	Ekonomi Industri	10	3	0	0	1	0	0	1	1	0	0	0	0	0	40

Sem.	Mata Kuliah	Not.	Sks	Matrik KK										Matrik KR			
				1	2	3	4	5	6	7	8	9	10	K.209 (50)	K.206 (80)	GK.30 6 (100)	
																	1
1	Kalkulus A	1	4	0	0	1	0	0	0	0	0	0	0	0	0	1	1
	Pengantar Teknik Industri B	2	2	0	0	0	1	0	0	0	0	0	0	0	0	1	1
3	Fisika Dasar 2A (Gelombang)	3	4	1	0	0	0	1	0	0	0	0	0	0	0	1	1
	Programa Linier B	4	3	0	1	0	0	0	1	0	0	0	0	0	0	1	1
5	Pemasaran Industri A	5	3	0	0	1	0	0	0	1	0	1	0	0	0	1	1
	Sistem Informasi B	6	3	0	0	0	0	0	0	0	1	1	0	0	0	1	1
7	Perancangan Proses A	7	2	0	0	0	0	1	0	0	0	0	1	1	1	1	1
	Manajemen Proyek B	8	3	0	0	0	0	0	1	0	0	0	1	1	1	1	1
Pilihan	Manajemen Energi	9	3	0	0	0	0	1	1	0	0	0	0	1	1	1	1
	Ekonomi Industri	10	3	0	0	0	0	0	0	0	1	1	0	0	1	1	1

Populasi awal diperoleh dari hasil program yang ditunjukkan pada Gambar 4.7.

- Evaluasi nilai *fitness*

Dilakukan perhitungan nilai *fitness* dari masing-masing kromosom jadwal dengan menghitung terlebih dahulu nilai penalti kendala *hard* dan kendala *soft*-nya. Perhitungan terhadap nilai penalti dapat dilihat pada kromosom-kromosom jadwal dibawah ini.

KJ1											KJ2										
	2	3	2	2	2	3	3	2	1	3		2	3	2	3	2	3	3	1	3	2
	5	2	1	7	6	3	1	3	6	6		5	2	5	7	6	3	1	6	6	3
fk1	0	0	0	0	1	1	0	1	1	0	fk1	1	0	1	0	1	0	0	0	1	0
fk2	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	fk3	0	0	0	0	0	0	0	0	0	0
gk1	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	gk2	0	0	0	0	0	0	0	0	0	0
Hard=	4										Hard=	4									
Soft=	0										Soft=	0									
Fitness=	0.2										Fitness=	0.2									
KJ3											KJ4										
	2	3	3	2	2	3	1	1	2	3		2	3	3	2	2	3	1	1	2	3
	1	5	1	7	6	6	1	6	3	3		1	5	1	7	6	6	1	6	3	3
fk1	1	0	1	0	0	1	0	1	0	0	fk1	1	0	1	0	0	1	0	1	0	0
fk2	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	fk3	0	0	0	0	0	0	0	0	0	0
gk1	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	gk2	0	0	0	0	0	0	0	0	0	0
Hard=	4										Hard=	4									
Soft=	0										Soft=	0									
Fitness=	0.2										Fitness=	0.2									

Gambar 4.7 Kromosom Jadwal 1-4

- Melakukan pencarian lokal sebelum melakukan proses evolusi
Terhadap gen-gen pada kromosom yang mendapatkan penalti *hard* akan dilakukan pencarian lokal dua tahap, yaitu untuk penyesuaian ruang dan penyesuaian slot. Setelah melakukan pencarian lokal, dilakukan perhitungan nilai penalti kendala *hard* dan *soft* terhadap kromosom jadwal. Gambar 4.8 merupakan hasil pencarian lokal untuk masing-masing kromosom jadwal (metode yang dilakukan telah dijelaskan pada subbab sebelumnya).
- Melakukan proses seleksi untuk mencari kandidat orang tua (*parent*).
Dilakukan dengan mengambil dua kromosom jadwal secara acak lalu memilih kromosom dengan nilai *fitness* terbaik (penalti terkecil) sebagai orang tua pertama. Hal yang sama dilakukan untuk memilih orang tua kedua.


Pertama, membandingkan kromosom jadwal 1 dan 3 yang dipilih secara acak. Dipilih kromosom 3 sebagai kromosom orang tua pertama. Kedua, membandingkan kromosom jadwal 1 dan 2 yang dipilih secara acak. Dipilih kromosom 2 sebagai kromosom orang tua kedua.

KJ1_LS										
	2	3	2	2	2	3	3	2	1	3
fk1	5	2	1	7	6	3	1	6	3	6
fk2	0	0	0	0	0	1	0	1	1	1
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
Hard=	4									
Soft=	0									
Fitness=	0.2									
KJ2_LS										
	2	3	2	3	2	3	3	1	3	2
fk1	5	2	1	7	6	3	1	6	6	3
fk2	0	0	0	0	1	0	0	0	0	1
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
Hard=	2									
Soft=	0									
Fitness=	0.3									
KJ3										
	2	3	3	2	2	3	1	1	2	3
fk1	1	5	2	7	6	6	1	3	3	3
fk2	0	0	0	0	0	0	0	1	1	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
Hard=	2									
Soft=	0									
Fitness=	0.3									
KJ4_LS										
	1	2	3	4	5	6	7	8	9	10
	2	3	3	2	2	3	1	1	2	3
	1	5	2	7	6	6	1	3	3	3
	0	0	0	0	0	0	0	0	1	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	0	0	0	0
	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0
Hard=	2									
Soft=	0									
Fitness=	0.33									

Gambar 4.8 Hasil Pencarian Lokal untuk Kromosom Jadwal 1-4

- Melakukan pindah silang (*crossover*).

Dibandingkan nilai penalti masing-masing gen orang tua, yang terbaik diturunkan ke kromosom anak. Berikut merupakan hasil pindah silang.

KJ2											KJ3											
	2	3	2	3	2	3	3	1	3	2		2	3	3	2	2	3	1	1	2	3	
	5	2	1	7	6	3	1	6	6	3		1	5	2	7	6	6	1	3	3	3	
Penalti	0	0	0	0	1	0	0	0	1	0		0	0	0	0	0	0	0	0	1	1	0
Nilai dengan penalti terkecil diturunkan ke kromosom anak																						
																						
Crossover																						
	2	3	2	2	2	3	3	1	3	2												
	5	2	1	7	6	3	1	6	6	3												
Penalti	0	0	0	0	1	0	0	0	1	0												

Gambar 4.9 Hasil Pindah Silang (*Crossover*)

- Melakukan mutasi terhadap hasil pindah silang.

Jika hasil pindah silang masih mendapatkan nilai penalti, maka dilakukan mutasi. Terlebih dahulu dibangkitkan matrik berukuran 1 x 10 secara

random untuk mewakili masing-masing kuliah, kemudian gen yang memiliki nilai random yang lebih kecil dari Pmut akan ditukarkan. Dipilih secara random gen yang akan dimutasi adalah gen ke-8 dan ke-10.

Mutasi	2	3	2	2	2	3	3	2	3	1
	5	2	1	7	6	3	1	3	6	6
Penalti	0	0	0	0	1	1	0	1	1	0

Gambar 4.10 Hasil Mutasi

- Melakukan pencarian lokal terhadap kromosom hasil mutasi. Pencarian lokal dilakukan apabila kromosom hasil mutasi masih memiliki nilai penalti. Metode pencarian lokal yang dilakukan sama halnya dengan pencarian lokal sebelumnya dengan tujuan meminimalkan penalti serta memastikan semua mata kuliah telah terjadwal.

Dari gen 1 hingga 10 dicek dengan alur maju, gen mana saja yang terkena penalti. Pertama dimulai dari gen ke-1, apakah gen ke-1 mengalami konflik dengan gen lain, jika tidak maka dilanjutkan dengan mengecek gen berikutnya hingga ditemukan gen yang terkena penalti. Dari hasil mutasi diatas diperoleh bahwa gen ke-5 mengalami konflik dengan gen ke-9, sehingga slot gen ke-9 akan diganti dengan menyesuaikan terhadap matrik KT.

#LS	2	3	2	2	2	3	3	2	3	1	#LS	2	3	2	2	2	3	3	2	1	1	
	5	2	1	7	6	3	1	6	3	6		5	2	1	7	6	3	1	3	6	6	
fk1	0	0	0	0	1	1	0	1	1	0	fk1	0	0	0	0	1	0	0	0	0	1	1
MK dengan kombinasi R-T sama: 5, 6, 8, 9											MK dengan kombinasi R-T sama: 9 & 10											
MK tidak terjadwal : 8 & 9 >> dijadwalkan kembali dengan LS											MK tidak terjadwal : 10 >> dijadwalkan kembali dengan LS											
LS dilakukan dengan mengganti kombinasi R-T secara acak																						

Child	2	3	2	2	2	3	3	2	1	3
	5	2	1	7	6	3	1	3	6	6

Gambar 4.11 Hasil Pencarian Lokal Hasil Mutasi

- Solusi jadwal.
 Kromosom anak yang diperoleh menjadi solusi jadwal.

Child	2	3	2	2	2	3	3	2	1	3
	5	2	1	7	6	3	1	3	6	6
fk1	0	0	0	0	1	1	0	1	1	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
fk =	4									
gk =	0									
Fitness	0.2									

Gambar 4.12 Solusi Jadwal Perhitungan Manual

4.1.2.2 Hasil perhitungan program

Berikut merupakan jadwal yang diperoleh dari hasil *running* program.

Kromosom Terbaik										
Mata Kuliah	1	2	3	4	5	6	7	8	9	10
Ruang	2	3	2	2	2	3	3	2	1	3
Slot	5	2	1	7	6	3	1	3	6	6
fk1	0	0	0	0	1	1	0	1	1	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
Matrix Jadwal										
Ruang/Slot		1	2	3	4	5	6	7		
1	-1	-1	-1	-1	-1	9	-1			
2	3	-1	8	-1	1	5	4			
3	7	2	6	-1	-1	10	-1			
Total Penalti Hard	4									
Total Penalti Soft	0									
Nilai Fitness Terbaik	0.2									

Gambar 4.13 Solusi Jadwal Hasil *Running* Program

Jadwal yang diperoleh berdasarkan perhitungan manual maupun hasil *running* program memberikan hasil yang sama. Dengan demikian program tervalidasi.

4.2 INPUT DATA

4.2.1 Data

Data yang dibutuhkan dalam pembuatan model penjadwalan mata kuliah telah dijelaskan sebelumnya pada Bab 3, yaitu data mata kuliah, data kelompok mahasiswa, data dosen, data ruang, dan data slot. Namun yang menjadi input

dalam model penjadwalan ini adalah hasil pengolahan dari kelima data tersebut, yaitu matrik KK, matrik KT, matrik KM, dan data ruang.

4.2.2 Parameter

Terdapat empat parameter kontrol yang ditetapkan nilainya, yaitu ukuran populasi, parameter kontrol mutasi, dan jumlah iterasi. Nilai ukuran populasi dan parameter kontrol mutasi diperoleh dari jurnal dengan permasalahan yang sama. Sedangkan untuk jumlah iterasi dilakukan pengecekan hasil *run* program dengan menggunakan beberapa jumlah iterasi yang berbeda. Nilai iterasi yang memberikan hasil terbaik dipilih menjadi parameter dalam model penjadwalan. Namun khusus untuk parameter ukuran populasi tidak akan diubah-ubah lagi sebab berdasarkan jurnal "*A Memetic Algorithm for Solving a Timetabling Problem: An Incremental Strategy*" telah dilakukan percobaan sebelumnya dengan menggunakan beberapa ukuran populasi yang berbeda ternyata ukuran populasi yang besar hanya akan membuat komputasi menjadi semakin lambat namun tidak memberikan hasil yang lebih baik dibandingkan dengan ukuran populasi yang kecil.

Untuk menentukan nilai parameter yang akan digunakan, akan dilakukan perbandingan hasil *running* program dengan beberapa kombinasi parameter yang berbeda. Kombinasi yang memberikan hasil terbaik akan digunakan dalam model. Nilai dari probabilitas mutasi yang digunakan dipilih berdasarkan jurnal dengan permasalahan yang sama. Perbandingan hasil *running* program untuk beberapa kombinasi parameter dapat dilihat pada Tabel 4.5.

Berdasarkan Tabel 4.5, dapat dilihat bahwa parameter dengan ukuran populasi 16 serta probabilitas mutasi sebesar 0.5 memberikan rata-rata total penalti yang paling kecil. Oleh karena itu, dipilih ukuran populasi 16 dengan probabilitas mutasi. Jumlah maksimum iterasi yang dicoba dalam model penjadwalan ini adalah 50, 100, 500, 1000, 1200, 1300, 1400, 1500, 1600, dan 2000. Maksimum iterasi sebesar 1200, 1300, 1400, dan 1500 memberikan hasil yang cukup baik. Oleh karena itu akan dilakukan *running* beberapa kali untuk tiap kombinasi parameter untuk melihat rata-rata total penalti yang dihasilkan.

Tabel 4.8 Skenario Parameter untuk Nilai Maksimum Iterasi

No.	Kombinasi			Hasil				
	Ukpop	Pmutasi	Iterasi	Lama Running (detik)	Penalti Hard	Penalti Soft	Fitness	Total Penalti
1	16	0.5	1200	103.37	1	6	0.1250	7
				106.61	6	4	0.0909	10
				107.34	2	5	0.1250	7
				112.16	5	6	0.0833	11
				Rata-rata	107.37	3.5	5.25	0.1061
2	16	0.5	1300	119.7	2	6	0.1111	8
				167.63	4	11	0.0625	15
				157.58	3	10	0.0714	13
				164.11	2	6	0.1111	8
				Rata-rata	152.255	2.75	8.25	0.0890
3	16	0.5	1400	148.33	2	5	0.1250	7
				160.22	4	5	0.1000	9
				165.25	3	9	0.0769	12
				182.47	1	5	0.1429	6
				Rata-rata	164.0675	2.5	6	0.1112
4	16	0.5	1500	170.21	3	4	0.1250	7
				210.34	2	6	0.1111	8
				199.37	3	5	0.1111	8
				155.32	2	4	0.1429	6
				Rata-rata	183.81	2.5	4.75	0.1225

Berdasarkan tabel diatas dapat dilihat bahwa iterasi dengan 1500 memberikan hasil terbaik dibandingkan dengan nilai maksimum iterasi lainnya. Oleh karena itu, dalam model penjadwalan mata kuliah digunakan nilai parameter seperti yang ditunjukkan oleh tabel berikut.

Tabel 4.9 Parameter Model

Parameter	
Ukpop	16
Pmutasi	0.5
Iterasi	1500

4.3 Pengolahan Data dan Hasil

4.3.1 Hasil Penjadwalan dengan Algoritma Memetika

Setelah ditentukan parameter permasalahan penjadwalan ini, maka program akan di-*run* sebanyak 6 kali. Karena fungsi tujuan permasalahan ini ialah meminimalkan maka akan diambil hasil terbaik dari kesepuluh *run* tersebut

dengan indikator ialah *fitness* terbaik yang ditandai dengan total penalti terkecil. Tabel berikut menunjukkan hasil *run* sebanyak 6 kali.

Tabel 4.10 Hasil *Run* Program

Running ke-	Hasil					
	Lama Running		Penalti Hard	Penalti Soft	Fitness	Total Penalti
	(sekon)	(X menit Y sekon)				
1	125.42	2m 5s	3	8	0.0833	11
2	123.12	2m 3s	3	8	0.0833	11
3	121.49	2m 1s	4	9	0.0714	13
4	118.20	1m 58s	5	6	0.0833	11
5	115.06	1m 55s	0	11	0.0833	11
6	164.82	2m 45s	3	8	0.0833	11

Dari hasil *running* program sebanyak enam kali, hasil *running* ke-5 memberikan hasil terbaik dengan nilai *fitness* terbesar yang artinya memberikan pelanggaran terhadap kendala *hard* dan kendala *soft* terkecil. Dapat dilihat juga bahwa nilai penalti *hard* yang dihasilkan pada *running* ke-5 bernilai 0, yang artinya jadwal yang dihasilkan tidak melanggar kendala *hard* dan telah memberikan jadwal yang layak.

Solusi jadwal yang diperoleh dapat dilihat pada Gambar 4.14 dan 4.15 yang merupakan hasil *running* program dengan nilai *fitness* sebesar 0,0833, total penalti kendala *hard* sebesar 0, total penalti kendala *soft* sebesar 11, dan total waktu *running* selama 115,06 detik. Pada matrik jadwal berikut yang terlihat hanya notasi dari mata kuliah yang telah dijadwalkan ke ruang dan slot. Notasi (-1) berarti menunjukkan tidak ada mata kuliah yang dijadwalkan pada periode tersebut. Tidak terdapat kombinasi ruang dan slot yang berulang yang ditandai dengan tidak terdapat mata kuliah yang tidak terjadwal. Semua mata kuliah telah terjadwal pada periode yang berbeda. Pada Gambar 4.15 ditunjukkan nilai penalti tiap mata kuliah, baik penalti kendala *hard* yang ditunjukkan oleh nilai *fk1*, *fk2*, dan *fk3*, maupun penalti kendala *soft* yang ditunjukkan oleh nilai *gk1* dan *gk2*.

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

Ukuran Populasi	16
Jumlah Iterasi	1500
Probabilitas Mutasi	0.5
Total Penalti Hard	0
Total Penalti Soft	11
Nilai Fitness Terbaik	0.08
Waktu Komputasi	1m 55s

Gambar 4.14 Tampilan Matrix Jadwal Terbaik Hasil *Run* ke-5

Kromosom Terbaik	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	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56				
Mata Kuliah	6	7	7	5	7	7	7	6	4	5	4	6	5	5	5	4	4	6	4	7	4	5	6	6	4	7	5	4	6	6	5	5	6	4	4	5	4	2	5	6	4	6	3	5	7	6	2	7	7	3	4	6	2	5	1	2				
Ruang	2	10	5	8	6	7	4	9	12	10	8	16	9	17	1	18	16	8	5	12	13	2	6	3	4	1	13	7	15	19	18	6	11	10	1	4	2	9	16	13	14	4	9	14	19	1	6	3	14	11	17	10	2	15	5	13				
Slot	2	10	5	8	6	7	4	9	12	10	8	16	9	17	1	18	16	8	5	12	13	2	6	3	4	1	13	7	15	19	18	6	11	10	1	4	2	9	16	13	14	4	9	14	19	1	6	3	14	11	17	10	2	15	5	13				
fk1	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	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	0	0	0			
fk2	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	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	0	0	0	0		
fk3	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	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	0	0	0	0	0	
gk1	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	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	0	0	0	0	0	
gk2	0	0	0	1	0	0	1	0	1	0	1	1	0	0	0	0	1	0	0	1	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Gambar 4.15 Kromosom Jadwal Terbaik Hasil *Run* ke-5 serta Nilai Penalties

4.4 Analisis

Permasalahan penjadwalan mata kuliah pada perguruan tinggi merupakan permasalahan yang cukup kompleks dan rumit karena terdapat banyak aspek yang harus dipertimbangkan dalam menghasilkan jadwal kuliah yang baik. Tingkat kerumitan penyelesaian masalah penjadwalan mata kuliah tergantung pada karakteristik yang terdapat di masing-masing perguruan tinggi, misalnya jumlah mata kuliah yang harus dijadwalkan, jumlah dosen pengajar, jumlah ruangan yang tersedia, dan jumlah slot yang terdapat di tiap minggunya.

Usulan penyelesaian terhadap masalah penjadwalan mata kuliah ini adalah dengan menggunakan algoritma metaheuristik algoritma genetika yang dikombinasikan teknik pencarian lokal yang diharapkan dapat memberikan hasil yang baik. Metode ini disebut dengan algoritma memetika. Dalam penelitian ini, algoritma tersebut akan disajikan ke dalam bahasa pemrograman MATLAB. Adapun fungsi tujuannya adalah minimalisasi total penalti, baik penalti kendala *hard* maupun *soft*. Analisis terhadap hasil pengolahan dibagi menjadi 3, yakni analisis metode, analisis waktu *running* dan analisis hasil.

4.4.1 Analisis Metode

Algoritma memetika didasari oleh teori evolusi biologi *Neo-Darwinian* dan pendapat Dawkin tentang *meme* sebagai unit evolusi kultural yang mampu melakukan perbaikan terhadap dirinya sendiri. Algoritma memetika memiliki karakteristik yang sama dengan algoritma genetika namun dikombinasikan dengan metode pencarian lokal yang diharapkan dapat memberikan hasil yang lebih optimal.

Cara kerja algoritma ini dimulai dari inisialisasi populasi awal yang akan menjadi kandidat solusi. Solusi awal dibuat dengan pemilihan secara acak dari kemungkinan-kemungkinan yang ada. Lalu setiap individu dalam populasi awal tersebut akan mengalami proses pencarian lokal yang diharapkan dapat memperbaiki kualitas dari populasi awal sebelum masuk ke tahap evolusi (algoritma genetika). Pencarian lokal pun dilakukan dengan menggunakan algoritma yang cocok namun masih menggunakan sistem acak dalam melakukan perubahan-perubahan komposisi setiap gen. Populasi hasil pencarian lokal

kemudian akan diteruskan ke tahapan algoritma genetika. Dalam tahapan ini pun terdapat proses seleksi yang dilakukan secara acak kemudian dikawinsilangkan, namun kawin silang tidak dilakukan secara acak karena memiliki konsep yang agak berbeda dengan proses mutasi. Sementara mutasi juga dipilih secara acak dimana apabila bilangan random yang dihasilkan lebih kecil dari probabilitas mutasi maka individu tersebut akan mengalami proses mutasi. Lalu dilakukan lagi perbaikan hasil mutasi dengan pencarian lokal.

Terdapatnya sistem pemilihan secara acak dalam algoritma ini akan membuat perbedaan urutan jadwal yang dihasilkan pada setiap *run* program yang dilakukan. Kombinasi parameter yang sama sekalipun belum tentu dapat menghasilkan urutan jadwal yang sama jika di-*run* beberapa kali, sebagaimana terlihat pada Tabel 4.8. Oleh karena itu skenario parameter merupakan bagian yang penting dalam pengujian metode ini, karena tiap kasus membutuhkan kombinasi parameter yang unik untuk menghasilkan jadwal yang sesuai dengan fungsi tujuan (total penalti yang minimum).

Salah satu metode yang dipakai pada algoritma memetika ini adalah pergantian populasi dengan skema *steady state replacement*, yaitu metode penggantian populasi dengan hanya mengganti beberapa individu dari populasi lama dengan individu baru hasil pencarian lokal tahap dua, yaitu setelah mutasi, sehingga jumlah individu yang diganti pada populasi lama sama dengan jumlah individu baru yang dihasilkan. Dalam model penjadwalan ini, jumlah individu yang akan menggantikan individu terburuk dalam populasi adalah satu.

Alasan yang menyebabkan metode ini dipilih adalah karena didasari oleh jenis permasalahan yang dimodelkan. Hal ini disebabkan karena:

- Metode *generational replacement* mengharuskan untuk menghasilkan individu baru sebanyak ukuran populasi awal mengakibatkan populasi bisa di-*update* apabila jumlah individu baru sama dengan ukuran populasi. Sementara proses genetika yang dilakukan dalam hal ini dirancang untuk menghasilkan satu individu dalam tiap satu kali proses evolusi. Oleh karena itu untuk menghasilkan individu baru sebanyak ukuran populasi maka jumlah proses evolusinya dilakukan sebanyak ukuran populasi dan hal ini membuat

komputasi menjadi lambat dan kemungkinan populasi awal untuk dirusak semakin besar.

- Metode *generational replacement* hanya melakukan satu kali evaluasi individu yaitu hanya pada individu dalam populasi awal sedangkan *steady-state reproduction* melakukan dua kali evaluasi individu yaitu pertama pada individu populasi awal, kedua pada individu baru yang dihasilkan sehingga memastikan bahwa individu dengan nilai *fitness* terbaik akan selalu disimpan.
- Jika dengan menggunakan metode *generational replacement* maka kemungkinan hasil yang diberikan adalah individu dengan nilai *fitness* yang lebih buruk dibandingkan dengan *steady state replacement* karena individu baru selalu menggantikan individu awal tanpa memperhatikan apakah nilai *fitness* individu baru lebih baik dibandingkan dengan individu awal yang digantikannya. Hal ini menyebabkan populasi baru belum tentu merupakan kumpulan individu-individu terbaik setiap generasi.

4.4.2 Analisis Waktu *Running*

Hal yang mempengaruhi waktu *run* ialah maksimum iterasi. Pada Tabel 4.7 ditunjukkan bahwa semakin besar nilai iterasi maka waktu *running* yang dibutuhkan juga semakin besar, namun dapat dilihat bahwa waktu yang dibutuhkan program penjadwalan ini dalam tiap iterasinya tidak terlalu lama. Nilai iterasi terkecil yang digunakan dalam *trial* adalah 50 dan memerlukan waktu *running* tercepat selama 12,26 detik. Sementara iterasi terbesar yang digunakan yaitu 2000 dan memerlukan waktu *running* terlama sebesar 246,68 detik. Ketika model penjadwalan di-*running* dengan menggunakan maksimum iterasi sebesar 10000, waktu komputasi yang dibutuhkan berkisar 16 menit, namun tidak memberikan hasil yang baik. Oleh karena itu tidak dapat disimpulkan bahwa semakin banyak jumlah iterasi maka semakin baik pula hasil yang diberikan.

4.4.3 Analisis Hasil

Hasil jadwal terbaik yang diperoleh dari program ini dapat dilihat pada Gambar 4.14. Dari jadwal yang dihasilkan dapat dilihat bahwa ke-56 mata kuliah telah terjadwal pada periode yang berbeda. Tidak terdapat mahasiswa yang

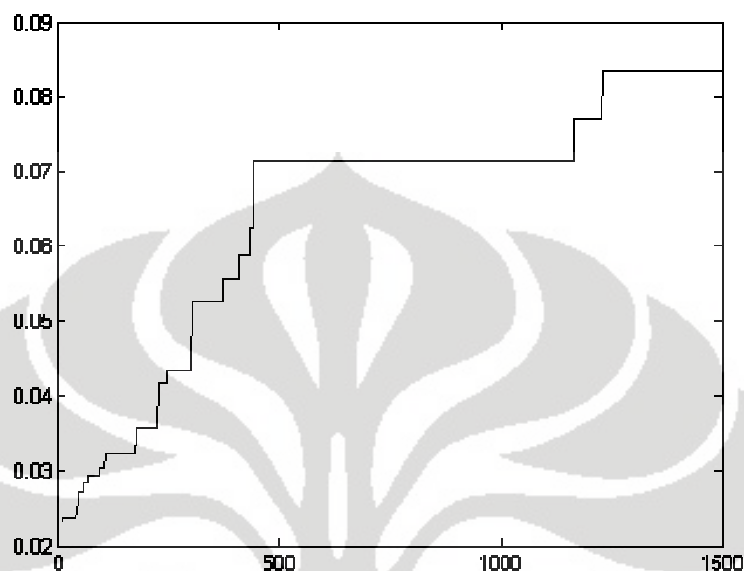
menghadiri lebih dari satu mata kuliah pada waktu yang bersamaan. Hal ini menunjukkan bahwa mata kuliah dalam tiap semester tidak mengalami bentrok/konflik. Selain itu, tidak terdapat dosen yang mengajar lebih dari satu mata kuliah pada waktu yang bersamaan, yang berarti bahwa mata kuliah-mata kuliah yang diajarkan oleh dosen yang sama dijadwalkan pada slot waktu yang berbeda.

Seperti yang dijelaskan pada bab sebelumnya bahwa sebaiknya mata kuliah yang berada pada semester yang beriringan tidak mengalami bentrok, tidak dapat dipenuhi secara sempurna dalam penjadwalan ini, karena hal ini sangat terkait dengan data yang dibutuhkan, yaitu jumlah mata kuliah, dosen, dan slot. Penjadwalan mata kuliah dengan mengambil data mata kuliah Departemen Teknik Industri tidak memungkinkan untuk menjadwalkan mata kuliah dalam semester yang beriringan tidak mengalami konflik seutuhnya dikarenakan terbatasnya jumlah dosen dan slot waktu yang tersedia. Terdapat 56 mata kuliah yang harus dijadwalkan terhadap 22 dosen dalam 19 slot waktu dan jumlah dosen yang cukup sedikit mengakibatkan hampir semua dosen mengajarkan lebih dari satu mata kuliah dalam tiap semesternya. Oleh karenanya jumlah mata kuliah yang tidak boleh berlangsung dalam waktu yang bersamaan semakin banyak, sementara jumlah slot yang tersedia hanya 19. Dan hal ini sangat tidak memungkinkan apabila dilakukan penjadwalan dengan kendala demikian.

Dengan tetap menggunakan kendala “mata kuliah pada tiap semester yang beriringan tidak boleh mengalami bentrok” dalam penjadwalan mata kuliah dengan data Departemen Teknik Industri (DTI) maka dihasilkan jadwal yang tidak layak, karena antar mata kuliah dalam satu semester pun terjadi konflik. Hal tersebut mengakibatkan terdapat dosen maupun mahasiswa yang menghadiri dua mata kuliah pada waktu yang bersamaan. Sementara hal ini merupakan kendala *hard* yang harus dipenuhi. Dengan demikian jadwal dikatakan tidak layak. Oleh karena itu, untuk membuat penjadwalan mata kuliah yang layak untuk DTI, mata kuliah dalam satu semester dan mata kuliah yang tidak diajarkan oleh dosen yang sama lebih diprioritaskan agar tidak dilangsungkan pada waktu yang bersamaan.

Tujuan penjadwalan ini adalah meminimalkan total penalti terhadap kendala *hard* dan *soft*. Dengan demikian jadwal dengan total penalti terkecil

adalah jadwal yang optimal dengan menggunakan algoritma memetika, yaitu jadwal yang memiliki nilai *fitness* terbaik. Berikut merupakan grafik yang menunjukkan peningkatan nilai *fitness* dengan beberapa iterasi.



Gambar 4.16 Grafik Peningkatan Nilai *Fitness* Jadwal

Dari grafik dapat dilihat bahwa dengan iterasi kurang dari 1500, yaitu berkisar pada iterasi ke-1400 an nilai *fitness*-nya sama dengan nilai *fitness* iterasi 1500 kali, namun masing-masing total penalti *hard* dan total penalti *soft* yang dihasilkan berbeda-beda. Oleh karena itu walaupun nilai *fitness* yang diberikan sama, namun lebih diutamakan jadwal dengan total penalti *hard* terkecil sebab hal ini membuat jadwal semakin layak. Setelah me-*running* program beberapa kali dengan jumlah iterasi 1500 kali, ternyata dihasilkan jadwal dengan tidak melanggar kendala *hard* (penalti *hard* = 0) dan penalti *soft* sebesar 11. Oleh karena itu jadwal ini dinyatakan layak.

Program ini juga dicoba untuk mengetes apakah jadwal masih layak apabila jumlah ruangan yang dialokasikan adalah enam (Gambar 4.17 dan 4.18). Setelah melakukan beberapa kali *running* dengan menggunakan parameter yang sama, hasil yang terbaik yang diperoleh adalah jadwal dengan total penalti *hard* sebesar 1, penalti *soft* sebesar 7 (nilai *fitness* 0.111) dengan waktu *running* 181 detik (3 menit 1 detik). Jadwal ini memang memberikan total penalti yang sedikit,

Tabel 4.11 Solusi Jadwal

Nama Ruang	Senin				Selasa
	08.00-09.50 1	10.00-11.50 2	13.00-15.30 3	16.00-17.50 4	08.00-09.50 5
K.108					Analisa Kelayakan Industri Ir. Hj. Erlinda Muslim,MEE/Ir. Djoko Sihono Gabriel,MT M6B
K.209		Keterampilan Interpersonal Akhmad Hidayatno,ST.,MBT M6B			
K.211					
K.206	Sistem Informasi A Ir.M.Dachyar,MSc M5A	Total Quality Management (TQM) Ir. Djoko Sihono Gabriel,MT/Ir. Rahmat Nurcahyo,M.Eng.Sc./Dr.Ir. T .Yuri. M.z.,MEngSc M5A		Mekanika Teknik A Dosen Mesin M3A	Psikologi Industri A Ir. Rahmat Nurcahyo,M.Eng.Sc. M3A
K.207	Akuntansi dan Biaya A Arian Dhini,ST.,MT M3A	Ekonomi Teknik B Ir.M.Dachyar,MSc M4B		Sistem Informasi B Ir.M.Dachyar,MSc M5B	
GK.306	Kapita Selekt Industri Ir. Isti Surjandari,MT.,MA.,PhD M6	Kalkulus A Dosen MIPA (Kalkulus) M1A	Programa LinierB Farizal.,PhD M4B	Perancangan Proses A Ir.Fauzia Dianawati,Msi M6A	
GK.301	Mekanika Teknik B Dosen Mesin M4B		Sistem Pemeliharaan B Ir. Boy Nurtjahyo,MSIE/Ir.Fauzia Dianawati,Msi M6B	Pengantar Teknik Industri A Ir. Djoko Sihono Gabriel,MT/Dr.Ir. T .Yuri. M.z.,MEngSc M1A	Kalkulus A' Dosen MIPA (Kalkulus) M1A

Tabel 4.11 Solusi Jadwal (lanjutan)

	Selasa			Rabu	
10.00-12.30 6	13.30-16.00 7	16.00-17.50 8	08.00-09.50 9	10.00-11.50 10	13.00-15.30 11
Sistem Pemeliharaan A Ir. Boy Nurtjahyo,MSIE/Ir.Fauzia Dianawati,Msi M6A			Analisa Multivariat Ir. Isti Surjandari,MT.,MA.,PhD M5B		
			Perancangan Proses B Ir.Fauzia Dianawati,Msi M6B		Manajemen Teknologi B Akhmad Hidayatno,ST.,MBT M6B
	Faktor Manusia Dalam Rekayasa Disain B Ir. Boy Nurtjahyo,MSIE M5B	Fisika Dasar 2A (Optik) Dosen MIPA2 (Fisdas2 Optik) M3A		Pemasaran Industri B Ir. Hj. Erlinda Muslim,MEE M5B	
Pemodelan Sistem + Praktikum B Akhmad Hidayatno,ST.,MBT/Armand Omar M.,ST.,MSc/ Komarudin,ST.,M.Eng M5B		Kalkulus B' Dosen MIPA (Kalkulus) M2B	Kimia Dasar A Dosen Kimia M3A	Fisika Dasar 2B (Gelombang) Dosen MIPA1 (Fisdas2Gelombang) M4B	
Programa LinierA Ir.Amar Rachman,MEIM M3A		Fasilitas Industri B Ir.Sri Bintang Pamungkas.,MSISE.,PhD M4B	Pengantar Teknik Industri B Ir. Djoko Sihono Gabriel,MT/Dr.Ir. T .Yuri. M.z.,MEngSc M2B	Sistem Distribusi dan Logistik Ir.Amar Rachman,MEIM M6A	Pemasaran Industri A Ir. Hj. Erlinda Muslim,MEE M5A
Statistik & Probabilitas A Arian Dhini,ST.,MT M1A	Statistik & Probabilitas B Arian Dhini,ST.,MT M2B			Kalkulus B Dosen MIPA (Kalkulus) M2B	

Tabel 4.11 Solusi Jadwal (lanjutan)

Rabu	Kamis			
16.00-17.50 12	08.00-09.50 13	10.00-12.30 14	13.30-16.00 15	16.00-17.50 16
	Ekonomi Industri Ir. Rahmat Nurcahyo M6A			
Fisika Dasar 2A (Gelombang) Dosen MIPA1 (Fisdas2 Gelombang) M3A	Ekonomi Teknik A Ir.M.Dachyar,MSc M3A	Manajemen Energi Ir. Hj. Erlinda Muslim,MEE/Armand Omar M.,ST.,MSc M5B		Fasilitas Industri A Ir.Sri Bintang Pamungkas.,MSISE.,PhD M3A
	Faktor Manusia Dalam Rekayasa Disain A Ir. Boy Nurtjahyo,MSIE M5A	Manajemen Proyek A Ir.M.Dachyar,MSc M6A	Berfikir Sistem Akhmad Hidayatno,ST.,MBT/Armand Omar M.,ST.,MSc M6B	Manajemen Pengetahuan Akhmad Hidayatno,ST.,MBT/Armand Omar M.,ST.,MSc M5A
	Sistem Keuangan Perusahaan Ir.Sri Bintang Pamungkas.,MSISE.,PhD M5B		Perencanaan & Pengendalian Produksi A Ir.Yadrifil,Msc M5A	Fisika Dasar 2B' (Optik) Dosen MIPA2 (Fisdas2 Optik) M4B
Psikologi Industri B Ir. Rahmat Nurcahyo,M.Eng.Sc. M4B		Manajemen Teknologi A Akhmad Hidayatno,ST.,MBT M6A		

Tabel 4.11 Solusi Jadwal (lanjutan)

Jumat		
07.30-09.20 17	09.30-11.20 18	13.30-16.00 19
Teori Keputusan Ir. Isti Surjandari, MT., MA., PhD M6A	Akuntansi dan Biaya B Arian Dhini, ST., MT M4B	
Kimia Dasar B Dosen Kimia M4B	Pemodelan Sistem + Praktikum A Akhmad Hidayatno, ST., MBT/Armand Omar M., ST., MSc/ Komarudin, ST., M.Eng M5A	
		Perencanaan & Pengendalian Produksi B Ir. Yadrifil, Msc M5B
		Manajemen Proyek B Ir. M. Dachyar, MSc M6B

BAB 5

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan penelitian optimasi penjadwalan mata kuliah di perguruan tinggi menggunakan algoritma memetika dengan bahasa pemrograman MATLAB, diperoleh kesimpulan sebagai berikut:

- Diperoleh model penjadwalan ruang kuliah dan slot untuk mata kuliah yang memberikan penjadwalan yang layak.
- Dari uji coba yang telah dilakukan dengan menggunakan beberapa kombinasi parameter. Penjadwalan mata kuliah dengan algoritma memetika telah memenuhi semua kendala *hard* dengan ukuran populasi 16, probabilitas mutasi 0,5 dan 1500 iterasi dengan waktu *running* kurang dari 3 menit.
- Jadwal yang dihasilkan juga telah mendekati optimal dengan memenuhi tujuan penelitian, yaitu meminimalkan penalti kendala *hard* dan *soft* yang berarti memberikan nilai *fitness* jadwal terbesar.
- Algoritma memetika dapat menyelesaikan masalah penjadwalan mata kuliah dengan memenuhi semua kendala *hard* yang berarti telah menghasilkan jadwal yang layak. Namun tidak semua kendala *soft* dapat dipenuhi.

5.2 Saran

Berdasarkan penelitian optimasi penjadwalan mata kuliah, terdapat saran untuk penelitian berikutnya, yaitu untuk memperoleh penjadwalan mata kuliah dengan tidak melanggar kendala *hard*, maka diusulkan untuk melakukan penjadwalan secara bertahap pada inisialisasi populasi awal. Penjadwalan mata kuliah terhadap ruang dan slot waktu dilakukan dengan memperkecil ruang pencarian solusi, yaitu hanya mempertimbangkan ruang dan slot yang memungkinkan untuk melangsungkan mata kuliah yang bersangkutan. Dengan demikian diharapkan jadwal yang dihasilkan akan selalu memenuhi kendala *hard*,

sehingga tujuan penelitian adalah meminimalkan total penalti kendala *soft* yang akan mengoptimalkan jadwal.



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, İstanbul, 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: Penerbit Andi
- 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. Data Ruang Kuliah

NO	Nama ruangan	Kapasitas	Lokasi	NO	Nama ruangan	Kapasitas	Lokasi
1	A.101	40	GEDUNG PASCA SARJANA FT	52	EC 108	20	GEDUNG EC
2	A.102	40	GEDUNG PASCA SARJANA FT	53	EC 109	20	GEDUNG EC
3	A.103	40	GEDUNG PASCA SARJANA FT	54	EC 110	20	GEDUNG EC
4	A.104	40	GEDUNG PASCA SARJANA FT	55	EC 202	120	GEDUNG EC
5	A.201	30	GEDUNG PASCA SARJANA FT	56	EC 203	20	GEDUNG EC
6	A.202	30	GEDUNG PASCA SARJANA FT	57	EC 302	120	GEDUNG EC
7	A.203	30	GEDUNG PASCA SARJANA FT	58	EC 303	20	GEDUNG EC
8	A.204	20	GEDUNG PASCA SARJANA FT	59	EC STUDIO	20	GEDUNG EC (DEPAN ATM)
9	A.205	20	GEDUNG PASCA SARJANA FT	60	RS CHEVRON		GEDUNG DEKANAT
10	A.601	40	GEDUNG PASCA SARJANA FT	61	GB 101	70	GEDUNG BARU
11	A.602	40	GEDUNG PASCA SARJANA FT	62	GB 102	40	GEDUNG BARU
12	A.603	40	GEDUNG PASCA SARJANA FT	63	GB 201	70	GEDUNG BARU
13	A.604	40	GEDUNG PASCA SARJANA FT	64	GB 202	70	GEDUNG BARU
14	A.605	40	GEDUNG PASCA SARJANA FT	65	GB 203	70	GEDUNG BARU
15	A.606	40	GEDUNG PASCA SARJANA FT	66	GB 204	40	GEDUNG BARU
16	SEM PASCA	20	GEDUNG PASCA SARJANA FT	67	GB 205	40	GEDUNG BARU
17	GK.201	40	Gedung PAF	68	GB 301	70	GEDUNG BARU
18	GK.301	100	Gedung PAF	69	GB 302	40	GEDUNG BARU
19	GK.302	20	Gedung PAF	70	GB 303	40	GEDUNG BARU
20	GK.303	20	Gedung PAF	71	GB 304	140	GEDUNG BARU
21	GK.304	20	Gedung PAF	72	GB 401	70	GEDUNG BARU
22	GK.305	20	Gedung PAF	73	GB 402	40	GEDUNG BARU
23	GK.306	100	Gedung PAF	74	GB 403	40	GEDUNG BARU
24	LAB PUSGIWA	70	Gedung PAF	75	GB 404	140	GEDUNG BARU
25	K.101	40	GEDUNG KUL BERSAMA FT	76	GB 501	70	GEDUNG BARU
26	K.102	70	GEDUNG KUL BERSAMA FT	77	GB 502	140	GEDUNG BARU
27	K.103	70	GEDUNG KUL BERSAMA FT	78	GB 503	140	GEDUNG BARU
28	K.104	40	GEDUNG KUL BERSAMA FT	79	S 201	40	GEDUNG BARU
29	K.105	40	GEDUNG KUL BERSAMA FT	80	S 202	40	GEDUNG BARU
30	K.106	70	GEDUNG KUL BERSAMA FT	81	S 203	40	GEDUNG BARU
31	K.107	70	GEDUNG KUL BERSAMA FT	82	S 204	40	GEDUNG BARU
32	K.108	40	GEDUNG KUL BERSAMA FT	83	S 301	70	GEDUNG BARU
33	K.201	30	GEDUNG KUL BERSAMA FT	84	S 303	40	GEDUNG BARU
34	K.202	80	GEDUNG KUL BERSAMA FT	85	RDM S	10	
35	K.203	30	GEDUNG KUL BERSAMA FT	86	R SEM S	15	
36	K.204	70	GEDUNG KUL BERSAMA FT	87	BP3 101		
37	K.205	30	GEDUNG KUL BERSAMA FT	88	BP3 102		
38	K.206	80	GEDUNG KUL BERSAMA FT	89	C. 102		
39	K.207	80	GEDUNG KUL BERSAMA FT	90	C. 102		
40	K.208	30	GEDUNG KUL BERSAMA FT	91	C. 301		
41	K.209	50	GEDUNG KUL BERSAMA FT	92	C. 302		
42	K.210	70	GEDUNG KUL BERSAMA FT	93	C. 303		
43	K.211	50	GEDUNG KUL BERSAMA FT	94	C. 304		
44	K.301	130	GEDUNG KUL BERSAMA FT	95	C. 401		
45	EC 101	20	GEDUNG EC	96	C. 402		
46	EC 102	20	GEDUNG EC	97	STUDIO DA		
47	EC 103	20	GEDUNG EC	98	MULTIMEDIA DA		
48	EC 104	20	GEDUNG EC	99	R. SEMINAT DA		
49	EC 105	20	GEDUNG EC	100	DA 101		
50	EC 106	20	GEDUNG EC	101	DA 501		
51	EC 107	20	GEDUNG EC				

Lampiran 2. Data Dosen Pengajar di Departemen Teknik Industri UI

No	Nama dosen	Mata Kuliah
1	Akhmad Hidayatno,ST.,MBT	Sistem Penjaminan dan Pengendalian Kualitas Pemodelan Sistem + Praktikum Simulasi Industri Keterampilan Interpersonal Perancangan Permainan Simulasi Manajemen Pengetahuan Berfikir Sistem
2	Ir.Amar Rachman,MEIM	Programa Linier Fasilitas Pabrik Penelitian Operasional Elemen Mesin Sistem Distribusi dan Logistik
3	Armand Omar M.,ST.,MSc	Menggambar Teknik Perancangan Kerja, Metode dan Standar Pemodelan Sistem + Praktikum Manajemen Teknologi Prinsip-prinsip Rekayasa Sistem Industri Manajemen Pengetahuan Berfikir Sistem Manajemen Energi
4	Arian Dhini,ST.,MT	Statistik & Probabilitas Akuntansi Biaya Statistik Industri
5	Ir. Boy Nurtjahyo,MSIE	Dasar Komputer Perancangan Kerja, Metode dan Standar Faktor Manusia Dalam Rekayasa Disain Tata Letak Pabrik Kesehatan, Keselamatan Kerja & Lindung Lingkungan Sistem Pemeliharaan
6	Ir. Dendi P. Ishak,MSIE	Perancangan Organisasi Manajemen Investasi
7	Ir. Djoko Sihono Gabriel,MT	Tata Letak Pabrik <i>Total Quality Management (TQM)</i> Analisa Kelayakan Industri
8	Ir. Hj. Erlinda Muslim,MEE	Pemasaran Industri Perancangan Produk Perancangan Proses Manajemen Strategi Industri Analisa Kelayakan Industri
9	Ir.Fauzia Dianawati,Msi	Pengetahuan Bahan Perancangan Organisasi Perancangan Proses Perancangan Produk Sistem Pemeliharaan
10	Farizal.,PhD	Penelitian Operasional Programma Linier
11	Ir. Isti Surjandari,MT.,MA.,Ph	Pengantar Ilmu Ekonomi Statistik Industri Kapita Selekta Industri Teori Keputusan Analisa Multivariat

(lanjutan)

No	Nama dosen	Mata Kuliah
12	Komarudin,ST.,M.Eng	Simulasi Industri Pemodelan Industri Teknik Proses dalam Industri Keputusan, Ketidakpastian dan Risiko
13	Ir.M.Dachyar,MSc	Ekonomi Teknik Sistem Informasi Manajemen Proyek Teori Keputusan Manajemen Inovasi
14	Ir. Rahmat Nurcahyo,M.Eng.Sc.	Psikologi Industri <i>Total Quality Management (TQM)</i> Ekonomi Industri
15	Dr.Ir. T .Yuri. M.z.,MEngSc	Pengantar Teknik Industri Etika Enjiniring <i>Total Quality Management (TQM)</i>
16	Ir.Yadrifil,Msc	Perencanaan & Pengendalian Produksi Sistem Produksi
17	Ir.Sri Bintang Pamungkas.,MSISE.,	Pengantar Ilmu Ekonomi Fasilitas Industri Proses Produksi + Praktikum Teknik Proses dalam industri Sistem Keuangan Perusahaan Kebijakan Industri
18	Dosen Teknik Kimia	Kimia
19	Dosen MIPA1 (Fisdas1 Mekanika)	Fisika Dasar1 - Mekanika
20	Dosen MIPA2 (Fisdas1 Listrik)	Fisika Dasar1 - Listrik
21	Dosen MIPA (Kalkulus)	Kalkulus
22	Dosen Teknik Mesin	Mekanika Teknik

Lampiran 3. Struktur Mata Ajaran Program Studi Teknik Industri

KODE	MATA AJARAN	SUBJECT	SKS
Semester 1		1st Semester	
UIII1001	MPK Terintegrasi	Integrated Characteristic Building Subject	0
UIII1010	Bahasa Inggris	English	3
UIII2030	Olah Raga / Seni	Sport / Art	1
ENG10001	Kalkulus	Calculus	4
ENG20001	Statistik & Probabilitas	Statistics and Probability	2
IES110001	Pengantar Teknik Industri	Introduction to Industrial Engineering	2
Sub Total			19
Semester 2		2nd Semester	
UIII2020-24	Agama	Religion	2
ENG10004	Aljabar Linear	Linear Algebra	4
ENG10005	Fisika Dasar 1	Basic Physics 1	4
ENG10007	Dasar Komputer	Basic Computer	3
IES12001	Pengertian Bahan	Material Science	2
ENG10000	Kongambar Teknik	Engineering Drawing	2
IES12007	Pengantar Ilmu Ekonomi	Introduction to Economics	2
Sub Total			19
Semester 3		3rd Semester	
ENG20002	Fisika Dasar 2	Basic Physics 2	4
ENG10002	Kimia Dasar	Basic Chemistry	2
IES11001	Akuntansi dan Biaya	Cost Accounting	2
IES11002	Fasilitas Industri	Industrial Facilities	2
IES11003	Psikologi Industri	Industrial Psychology	2
ENG20004	Ekonomi Teknik	Engineering Economics	2
IES11004	Programa Linier	Linear Programming	3
IES11005	Mekanika Teknik	Engineering Mechanics	2
Sub Total			19
Semester 4		4th Semester	
IES21001	Fasilitas Pabrik	Plant Facilities	2
IES21002	Proses Produksi + Praktikum	Production Process + Lab	2
IES21003	Statistik Industri	Industrial Statistics	2
IES21004	Penelitian Operasional	Operation Research	3
IES21005	Sistem Penjaminan dan Pengendalian Kualitas	Quality Control and Assurance Systems	3
IES21006	Perancangan Organisasi	Organizational Design	2
IES21007	Perencanaan Kerja, Metode dan Standar Kerja	Work Design, Methods and Standards	2
IES21008	Elemen Mesin	Machine Elements	2
Sub Total			20
Semester 5		5th Semester	
IES21001	Faktor Manusia Dalam Pekerjaan Dipan	Human Factors in Engineering Design	2
IES21002	Perencanaan & Pengendalian Produksi	Production Planning and Control	3
IES21003	Pemodelan Sistem + Praktikum	Systems Modeling + Lab	3
IES21004	Pemasaran Industri	Industrial Marketing	2
IES21005	Sistem Informasi	Information System	3
	Pilihan 1	Elective 1	1
	Pilihan 2	Elective 2	1
Sub Total			21
Semester 6		6th Semester	
IES21001	Perancangan Produk	Product Design	2
IES21002	Data Letak Pabrik	Plant Lay Out	2
ENG30003	Etika Engineering	Engineering Ethics	2
ENG30002	Kesehatan, keselamatan kerja & lingkungan	Occupational, health, safety & environment	2
IES21003	Sistem Produksi	Production System	3
IES21004	Simulasi Industri	Industrial Simulation	3
	Pilihan 3	Elective 3	2
	Pilihan 4	Elective 4	2
Sub Total			21
Semester 7		7th Semester	
IES41001	Perancangan Proses	Process Design	2
IES41002	Manajemen Proyek	Project Management	2
IES41003	Kapita Selekt Industri	Special topic in Industrial Engineering	2
IES41004	Filter: Pemeliharaan	Maintenance System	3
IES41005	Manajemen Teknologi	Management of Technology	3
IES40001	Internship	2	
	Pilihan 5	Elective 5	2
	Pilihan 6	Elective 6	2
Sub Total			21
Semester 8		8th Semester	
IES40001	Skripsi dan Seminar	Final Project in Ind.Eng	5
Sub Total			5

(lanjutan)

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 Strategic Management	3
IEF300806	Sistem Distribusi dan Logistik	Logistic and Distribution System	3
IEF300807	Manajemen Inovasi	Inovation 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

Lampiran 4. Matrik KK (Matrik Hubungan antar Mata Kuliah)

Nama Mata Kuliah	Sem	sks	Notasi	Notasi Mata Kuliah																									
				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	
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	
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	
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	
Statistik & Probabilitas A	1	3	5	1	0	1	0	0	1	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	1	0	0	
Statistik & Probabilitas B	1	3	6	0	1	0	1	1	0	0	1	0	0	0	0	0	1	1	0	0	0	0	0	0	0	1	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	
Pengantar Teknik Industri B	1	2	8	0	1	0	1	0	1	1	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	1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	
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	
Fisika Dasar 2A (Optik)	3	4	11	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	
Fisika Dasar 2B' (Optik)	3	4	12	0	0	0	0	0	0	0	1	1	0	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	
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	
Kimia Dasar B	3	2	14	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	
Akuntansi dan Biaya A	3	2	15	0	0	0	0	1	1	0	0	1	0	1	0	1	0	0	1	0	1	0	1	0	1	0	1	0	
Akuntansi dan Biaya B	3	2	16	0	0	0	0	1	1	0	0	0	1	0	1	0	1	0	0	1	0	1	0	1	0	1	0	1	
Fasilitas Industri A	3	2	17	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	
Fasilitas Industri B	3	2	18	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1	0	1	1	0	1	0	1	0	1	0	
Psikologi Industri A	3	2	19	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 B	3	2	20	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	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	
Ekonomi Teknik B	3	2	22	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	
Programa LinierA	3	3	23	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 LinierB	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	
Mekanika Teknik A	3	2	25	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	
Mekanika Teknik B	3	2	26	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	
Faktor Manusia Dalam Reayasa 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	
Faktor Manusia Dalam Reayasa 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	
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	
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	
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	
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	
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	
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	
Sistem Informasi A	5	3	35	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	
Sistem Informasi B	5	3	36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	
Total Quality Management (TQM)	5	3	37	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	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	
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	
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	1	1	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	
Perancangan Proses A	7	2	42	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	
Perancangan Proses B	7	2	43	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 Proyek A	7	3	44	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	
Manajemen Proyek B	7	3	45	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	
Kapita Selekta Industri	7	2	46	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 Pemeliharaan A	7	3	47	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 Pemeliharaan B	7	3	48	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 Teknologi A	7	3	49	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 Teknologi B	7	3	50	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	
Teori Keputusan	7	3	51	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 Distribusi dan Logistik	7	3	52	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	
Keterampilan Interpersonal	7	3	53	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	
Berfikir Sistem	7	3	54	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	
Analisa Kelayakan Industri	7	3	55	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	
Ekonomi Industri	7	3	56	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	

Lampiran 5. Matrik KT (Hubungan antara Mata Kuliah dengan Slot Waktu)

Mata Kuliah	Sem	SKS	Notasi	Senin				Selasa				Rabu			
				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	08.00-09.50	10.00-11.50	13.00-15.30	16.00-17.50
				1	2	3	4	5	6	7	8	9	10	11	12
Kalkulus A	1	4	1	1	1	0	1	1	0	0	1	1	1	0	1
Kalkulus B		4	2	1	1	0	1	1	0	0	1	1	1	0	1
Kalkulus A'		4	3	1	1	0	1	1	0	0	1	1	1	0	1
Kalkulus B'		4	4	1	1	0	1	1	0	0	1	1	1	0	1
Statistik & Probabilitas A		3	5	0	0	1	0	0	1	1	0	0	0	1	0
Statistik & Probabilitas B		3	6	0	0	1	0	0	1	1	0	0	0	1	0
Pengantar Teknik Industri A		2	7	1	1	0	1	1	0	0	1	1	1	0	1
Pengantar Teknik Industri B		2	8	1	1	0	1	1	0	0	1	1	1	0	1
Fisika Dasar 2A (Gelombang)		4	9	1	1	0	1	1	0	0	1	1	1	0	1
Fisika Dasar 2B (Gelombang)		4	10	1	1	0	1	1	0	0	1	1	1	0	1
Fisika Dasar 2A (Optik)	4	11	1	1	0	1	1	0	0	1	1	1	0	1	
Fisika Dasar 2B' (Optik)	4	12	1	1	0	1	1	0	0	1	1	1	0	1	
Kimia Dasar A	2	13	1	1	0	1	1	0	0	1	1	1	0	1	
Kimia Dasar B	2	14	1	1	0	1	1	0	0	1	1	1	0	1	
Akuntansi dan Biaya A	2	15	1	1	0	1	1	0	0	1	1	1	0	1	
Akuntansi dan Biaya B	2	16	1	1	0	1	1	0	0	1	1	1	0	1	
Fasilitas Industri A	3	2	17	1	1	0	1	1	0	0	1	1	1	0	1
Fasilitas Industri B		2	18	1	1	0	1	1	0	0	1	1	1	0	1
Psikologi Industri A		2	19	1	1	0	1	1	0	0	1	1	1	0	1
Psikologi Industri B		2	20	1	1	0	1	1	0	0	1	1	1	0	1
Ekonomi Teknik A		2	21	1	1	0	1	1	0	0	1	1	1	0	1
Ekonomi Teknik B		2	22	1	1	0	1	1	0	0	1	1	1	0	1
Programa LinierA		3	23	0	0	1	0	0	1	1	0	0	0	1	0
Programa LinierB		3	24	0	0	1	0	0	1	1	0	0	0	1	0
Mekanika Teknik A		2	25	1	1	0	1	1	0	0	1	1	1	0	1
Mekanika Teknik B		2	26	1	1	0	1	1	0	0	1	1	1	0	1
Faktor Manusia Dalam Rekrutasi Disain A	5	3	27	1	1	1	1	1	1	1	1	1	1	1	1
Faktor Manusia Dalam Rekrutasi Disain B		3	28	1	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	1
Perencanaan & Pengendalian Produksi B		3	30	1	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	1
Pemodelan Sistem + Praktikum B		3	32	1	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	1
Pemasaran Industri B		3	34	1	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	1
Sistem Informasi B		3	36	1	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	1	
Analisa Multivariat	3	38	1	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	1	
Sistem Keuangan Perusahaan	3	40	1	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	1	
Perancangan Proses A	7	2	42	1	1	0	1	1	0	0	1	1	1	0	1
Perancangan Proses B		2	43	1	1	0	1	1	0	0	1	1	1	0	1
Manajemen Proyek A		3	44	0	0	1	0	0	1	1	0	0	0	1	0
Manajemen Proyek B		3	45	0	0	1	0	0	1	1	0	0	0	1	0
Kapita Selekt Industri		2	46	1	1	0	1	1	0	0	1	1	1	0	1
Sistem Pemeliharaan A		3	47	0	0	1	0	0	1	1	0	0	0	1	0
Sistem Pemeliharaan B		3	48	0	0	1	0	0	1	1	0	0	0	1	0
Manajemen Teknologi A		3	49	0	0	1	0	0	1	1	0	0	0	1	0
Manajemen Teknologi B		3	50	0	0	1	0	0	1	1	0	0	0	1	0
Teori Keputusan		3	51	1	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	1	
Keterampilan Interpersonal	3	53	1	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	1	
Analisa Kelayakan Industri	3	55	1	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	1	

(lanjutan)

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

Lampiran 6. Matrik KM (Hubungan antara Mata Kuliah dengan Mahasiswa)

KELOMPOK MAHASISWA

Mata Kuliah	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	
Kalkulus B		4	2	0	55	0	0	0	0	
Kalkulus A'		4	3	35	0	20	0	0	0	
Kalkulus B'		4	4	0	55	0	0	0	0	
Statistik & Probabilitas A		3	5	35	0	20	0	0	0	
Statistik & Probabilitas B		3	6	0	55	0	0	0	0	
Pengantar Teknik Industri A		2	7	35	0	20	0	0	0	
Pengantar Teknik Industri B		2	8	0	55	0	0	0	0	
Fisika Dasar 2A (Gelombang)		3	4	9	0	0	35	0	20	0
Fisika Dasar 2B (Gelombang)			4	10	0	0	0	55	0	0
Fisika Dasar 2A (Optik)	4		11	0	0	35	0	20	0	
Fisika Dasar 2B' (Optik)	4		12	0	0	0	55	0	0	
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	
Fasilitas Industri A	2		17	0	0	35	0	20	0	
Fasilitas Industri B	2		18	0	0	0	55	0	0	
Psikologi Industri A	5	2	19	0	0	35	0	20	0	
Psikologi Industri B		2	20	0	0	0	55	0	0	
Ekonomi Teknik A		2	21	0	0	35	0	20	0	
Ekonomi Teknik B		2	22	0	0	0	55	0	0	
Programa LinierA		3	23	0	0	35	0	20	0	
Programa LinierB		3	24	0	0	0	55	0	0	
Mekanika Teknik A		2	25	0	0	35	0	20	0	
Mekanika Teknik B		2	26	0	0	0	55	0	0	
Faktor Manusia Dalam Reayasa Disain A		3	27	0	0	0	0	35	20	
Faktor Manusia Dalam Reayasa Disain B		3	28	0	0	0	0	55	0	
Perencanaan & Pengendalian Produksi A	7	3	29	0	0	0	0	35	20	
Perencanaan & Pengendalian Produksi B		3	30	0	0	0	0	55	0	
Pemodelan Sistem + Praktikum A		3	31	0	0	0	0	35	20	
Pemodelan Sistem + Praktikum B		3	32	0	0	0	0	55	0	
Pemasaran Industri A		3	33	0	0	0	0	35	20	
Pemasaran Industri B		3	34	0	0	0	0	55	0	
Sistem Informasi A		3	35	0	0	0	0	35	20	
Sistem Informasi B		3	36	0	0	0	0	55	0	
Total Quality Management (TQM)		3	37	0	0	0	0	40	0	
Analisa Multivariat		3	38	0	0	0	0	20	20	
Manajemen Pengetahuan	3	39	0	0	0	0	40	0		
Sistem Keuangan Perusahaan	3	40	0	0	0	0	40	0		
Manajemen Energi	3	41	0	0	0	0	40	0		
Perancangan Proses A	7	2	42	0	0	0	0	0	45	
Perancangan Proses B		2	43	0	0	0	0	0	45	
Manajemen Proyek A		3	44	0	0	0	0	0	45	
Manajemen Proyek B		3	45	0	0	0	0	0	45	
Kapita Selekt Industri		2	46	0	0	0	0	0	90	
Sistem Pemeliharaan A		3	47	0	0	0	0	0	45	
Sistem Pemeliharaan B		3	48	0	0	0	0	0	45	
Manajemen Teknologi A		3	49	0	0	0	0	0	45	
Manajemen Teknologi B		3	50	0	0	0	0	0	45	
Teori Keputusan		3	51	0	0	0	0	0	40	
Sistem Distribusi dan Logistik	3	52	0	0	0	0	0	40		
Keterampilan Interpersonal	3	53	0	0	0	0	0	25		
Berfikir Sistem	3	54	0	0	0	0	0	40		
Analisa Kelayakan Industri	3	55	0	0	0	0	0	40		
Ekonomi Industri	3	56	0	0	0	0	0	40		

Fungsi Program Utama

```

clear;

fprintf('Algoritma MA untuk penjadwalan mata kuliah\n');
MaxIterasi=input('Masukkan maximum iterasi MA yang diinginkan: ');
Pmut=input('Masukkan nilai probabilitas mutasi yang diinginkan: ');
fileOutput=input('Nama file output yang diinginkan (dalam .xls): ', 's');

% memulai penghitungan waktu
tic;

% Ambil data
MatrixKK=xlsread('matrixKK', 'F4:BI59');
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);']);
end

```

(lanjutan)

```
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

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

(lanjutan)

```
eval(['KJ', int2str(z), '=yantifk2(KJ', int2str(z), ', MatrixKR);'])
eval(['KJ', int2str(z), '=yantifk3(KJ', int2str(z), ', MatrixKT);'])
eval(['KJ', int2str(z), '=yantigk1(MJ', int2str(z), ', KJ',
int2str(z), ', MatrixKM);'])

eval(['KJ', int2str(z), '=yantigk2(KJ', int2str(z), ');'])

%Pencarian Lokal terhadap UkPop
eval(['KJ', int2str(z), '=LocalSearch1(MJ', int2str(z), ', KJ',
int2str(z), ', MatrixKK, MatrixKT, MatrixKR, MatrixKM);'])

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

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

    % 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
    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
end

% Mutasi
```

(lanjutan)

```
if rand<Pmut
    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), '=yantifk1(MJ', int2str(outlayer), ',
KJ', int2str(outlayer), ', MatrixKK);'])

eval(['KJ', int2str(outlayer), '=yantifk2(KJ', int2str(outlayer), ',
MatrixKR);'])

eval(['KJ', int2str(outlayer), '=yantifk3(KJ', int2str(outlayer), ',
MatrixKT);'])

eval(['KJ', int2str(outlayer), '=yantigk1(MJ', int2str(outlayer), ',
KJ', int2str(outlayer), ', MatrixKM);'])

eval(['KJ', int2str(outlayer), '=yantigk2(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);

%Pencarian Lokal memperbaiki hasil mutasi
eval(['KJ', int2str(outlayer), '=LocalSearch1(MJ', int2str(z), ',
KJ', int2str(outlayer), ', MatrixKK, MatrixKT, MatrixKR, MatrixKM);'])

[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);

toc;
waktu=toc;

% Output
% Kromosom Terbaik
xlswrite(fileOutput, {'Kromosom Terbaik'}, 'Output', 'A1:A1')
```

(lanjutan)

```
matrixKeterangan = {'Mata Kuliah';'Ruang';'Slot'};
xlswrite(fileOutput, matrixKeterangan, 'Output', 'A2:A4')
xlswrite(fileOutput, [[1:56]; JadwalOpt2(1:2,:)], 'Output', 'B2:BE4')

% Nilai fk1, fk2, fk3, gk1, dan gk2 tiap gen
matrixKeterangan = {'fk1';'fk2';'fk3'; 'gk1'; 'gk2'};
xlswrite(fileOutput, matrixKeterangan, 'Output', 'A6:A10')
xlswrite(fileOutput, JadwalOpt2(3:7,:), 'Output', 'B6:BE10')

% Matrix Jadwal Terbaik
xlswrite(fileOutput, {'Matrix Jadwal'; 'Ruang/Slot'}, 'Output',
'A12:A13')
xlswrite(fileOutput, [1:19], 'Output', 'B13:T13')
xlswrite(fileOutput, [1:7]', 'Output', 'A14:A20')
xlswrite(fileOutput, JadwalOpt1, 'Output', 'B14:T20')

% Rincian Parameter
matrixKeterangan = {'Ukuran Populasi';
    'Jumlah Iterasi';
    'Probabilitas Mutasi';
    'Total Penalti Hard';
    'Total Penalti Soft';
    'Nilai Fitness Terbaik';
    'Waktu Komputasi'};
xlswrite(fileOutput, matrixKeterangan, 'Output', 'A22:A28')
xlswrite(fileOutput, [16; MaxIterasi; Pmut; PHT; PST; FitnessOpt],
'Output', 'B22:B27')
waktu=[int2str(floor(waktu/60)), 'm ', int2str(mod(waktu,60)), 's'];
xlswrite(fileOutput, {waktu}, 'Output', 'B28:B28')

plot([1:MaxIterasi], F, '-r')
```