



UNIVERSITAS INDONESIA

**PENJADWALAN KAMAR BEDAH DENGAN ALGORITMA
GENETIKA**

SKRIPSI

**ELTINA W HUTAHAEAN
NPM 0806337730**

**FAKULTAS TEKNIK
PROGRAM STUDI TEKNIK INDUSTRI
DEPOK
JUNI 2012**



UNIVERSITAS INDONESIA

**PENJADWALAN KAMAR BEDAH DENGAN ALGORITMA
GENETIKA**

SKRIPSI

Diajukan sebagai salah satu syarat memperoleh gelar Sarjana Teknik

**ELTINA W HUTAHAEAN
NPM 0806337554**

**FAKULTAS TEKNIK
PROGRAM STUDI TEKNIK INDUSTRI
DEPOK
JUNI 2012**

HALAMAN PERNYATAAN ORISINALITAS

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

Nama : Eltina W Hutahaean

NPM : 0806337554

Tanda Tangan :



Tanggal : Juni 2012

HALAMAN PENGESAHAN

Skripsi ini diajukan oleh :
Nama : Eltina W Hutahaeen
NPM : 0806337554
Program Studi : Teknik Industri
Judul Skripsi : Penjadwalan Kamar Bedah dengan Algoritma Genetika

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

DEWAN PENGUJI

Pembimbing : Ir. Amar Rachman, MEIM (.....)
Penguji : Dr. Akhmad Hidayatno, S.T., MBT (.....)
Penguji : Armand Omar Moeis, S.T., M.Sc (.....)
Penguji : Romadhani Ardi, S.T., M.T. (.....)

Ditetapkan di : Depok
Tanggal : Juni 2012

KATA PENGANTAR

Puji syukur kepada Tuhan Yesus yang Mahakasih atas segala berkat, kasih, dan pemeliharaan-Nya sehingga saya dapat menyelesaikan skripsi ini. Penulisan skripsi ini dilakukan dalam rangka memenuhi salah satu syarat untuk memperoleh gelar Sarjana Teknik pada Program Studi Teknik Industri Universitas Indonesia. Saya menyadari bahwa tanpa bantuan dan bimbingan dari berbagai pihak, sejak mulai berkuliah sampai pada penyusunan skripsi ini, sangatlah sulit bagi saya untuk menyelesaikan skripsi ini. Oleh karena itu, saya ingin mengucapkan terima kasih kepada:

1. Bapak Ir. Amar Rachman, MEIM selaku dosen pembimbing utama yang telah menyediakan waktu, tenaga, pikiran, dan kasih sayang untuk mengarahkan saya dalam penyusunan skripsi ini beserta Bapak Sumarsono, S.T, M.T sebagai dosen pembimbing kedua yang membantu mengarahkan pengerjaan skripsi ini;
2. Seluruh dosen Teknik Industri UI atas ilmu yang telah diberikan selama ini untuk mempersiapkan saya sebagai seorang teknik industri;
3. Apraz Ramatryana dan Adithya Yuda Hasibuan yang telah membantu membuat kode program algoritma saya;
4. Bapak, mama, bang Wendy, bang Wellman, kak Fransiska, bang Siskus yang telah mengasihiku selama ini, senantiasa mengingatkan tentang skripsi;
5. Sahabat SAROHA: Andalusia M N, Andreas Riardi, Andrew Lampatar, Friska Hotmauli, Gabriela Sabaktani, Jessica Stephani, Kristina Yohana, Mariana Sianipar, Paulus Oky, Rizal Himawan, Roberton Siahaan, Stefani Sabadtini atas persahabatan dan dukungan dalam suka dan duka, membantu saya belajar menjadi seorang sahabat;
6. Kak Sisca Pratiwi, PKK saya dan anak-anak POFTUI yang membantu aku mengenal dan mencintai Tuhan;
7. Kak Koci, Bang David, Kak Repi, Kak Nira, Putri, Herold, Kak Eteng, Kak Lasma, Kak Eka, dan semua naposo bulung HKBP Pasar Minggu

yang telah menjadi tempat saya belajar bergaul dan menyadari bahwa persahabatan butuh pengorbanan;

8. Anak-anak TIUI 2008 atas kebersamaan dan masa-masa indah belajar bersama;
9. Teman sekamarku, Yovieta, atas cerita-ceritamu yang membantuku belajar lebih atentif dan menambah wawasan, juga penjaga dan teman-teman kosan Sandi Putri Gahol: mba Lini, Wawa, mas Sarto, Noni, Fristasia, Mandor, Yanika, Novel, Ingrid, Debora, Tina, Duti, Dee, Erika, atas keceriaan tinggal di kosan; dan
10. Semua pihak yang tidak dapat disebutkan satu per satu yang terlibat dan telah membantu saya sehingga skripsi ini bisa terselesaikan dengan baik.

Akhir kata, saya berharap Tuhan Yesus membalas segala kebaikan semua pihak yang telah membantu. Semoga skripsi ini bermanfaat bagi semua pembaca dan bagi pengembangan ilmu pengetahuan.

Depok, 15 Juni 2012

Penulis

**HALAMAN PERNYATAAN PERSETUJUAN
PUBLIKASI KARYA ILMIAH UNTUK KEPENTINGAN AKADEMIS**

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

Nama : Eltina W Hutahaean
NPM : 0806337554
Program Studi : Teknik Industri
Departemen : Teknik Industri
Fakultas : Teknik
Jenis karya : Skripsi

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

Penjadwalan Kamar Bedah dengan Algoritma Genetika

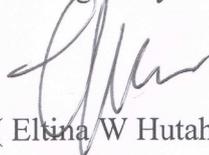
beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Universitas Indonesia berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan memublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta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

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok

Pada tanggal : Juni 2012

Yang menyatakan



(Eltina W Hutahaean)

ABSTRAK

Nama : Eltina W Hutahaeen
Program Studi : Teknik Industri
Judul : Penjadwalan Kamar Bedah dengan Algoritma Genetika

Penelitian ini dilakukan untuk memperoleh jadwal operasi pasien elektif pada sebuah rumah sakit dengan sejumlah ruang operasi sehingga kendala ketersediaan dokter, kapasitas ruangan, dan keterdesakan waktu operasi bisa dipenuhi sebaik mungkin. Optimasi penjadwalan dilakukan dengan algoritma genetika. Algoritma genetika adalah salah satu algoritma yang baik digunakan untuk memecahkan permasalahan dalam skala besar dan memiliki tingkat kerumitan yang tinggi. Oleh karena itu, algoritma ini cocok digunakan untuk menjadwalkan kamar operasi yang harus mempertimbangkan banyak kendala untuk menghasilkan jadwal yang baik. Hasil yang diharapkan dari penelitian ini adalah diperolehnya sebuah sistem penjadwalan kamar operasi yang optimal berdasarkan kapan saja dokter bedah bisa bekerja, waktu istirahat yang dibutuhkan dokter jika harus mengoperasi lebih dari satu pasien pada satu waktu, dan beberapa pasien yang lebih terdesak untuk dioperasi. Penelitian menghasilkan jadwal penggunaan kamar operasi yang cukup baik dalam memenuhi kendala. Algoritma yang dihasilkan juga bisa menampung fleksibilitas data pasien, ruang operasi, rentang hari penjadwalan, ketersediaan dokter.

Kata Kunci:
Optimasi, penjadwalan kamar operasi, algoritma genetika

ABSTRACT

Name : Eltina W Hutahaeen
Study Program : Industrial Engineering
Title : Operating Room Scheduling using Genetic Algorithm

This research aims to obtain optimum surgery schedule for elective patients for a hospital having several operating rooms so as to satisfy the constraints of surgeon availability, room capacity, and the urgency of some patients. Schedule optimization is achieved by using genetic algorithm. Genetic algorithm is an algorithm that is suitable to use for solving highly complex combinatoric problems. Therefore, a genetic algorithm is proposed to solve highly constrained operating room scheduling problem. The result expected from this research is an operating room scheduling system which incorporates the availability of the surgeons, break time inter-scheduling if the surgeon has to operate on more than one patient in a day, and the urgency of some patients. Resulting schedule is satisfactory in satisfying the constraints. The proposed algorithm can also accommodate flexibility in number of patients to be operated on, number of operating rooms used, the day spans of scheduling, and availability of the surgeons.

Keywords:

Optimization, operating room scheduling, genetic algorithm

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PERNYATAAN ORISINALITAS	ii
HALAMAN PENGESAHAN	iii
KATA PENGANTAR	iv
HALAMAN PERSETUJUAN PUBLIKASI KARYA ILMIAH	vi
ABSTRAK	vii
ABSTRACT	viii
DAFTAR ISI	ix
DAFTAR TABEL	x
DAFTAR GAMBAR	xi
DAFTAR LAMPIRAN	xii
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Diagram Keterkaitan Masalah	4
1.3 Rumusan Masalah	4
1.4 Tujuan Penelitian	4
1.5 Ruang Lingkup Penelitian	4
1.6 Metodologi Penelitian	5
1.7 Sistematika Penulisan	5
BAB 2 DASAR TEORI	8
2.1 Sistem Penjadwalan Ruang Operasi	8
2.2 Program Integer	12
2.3 Algoritma Genetika	13
BAB 3 PENGUMPULAN DATA	25
3.1 Data Yang Diperlukan	25
3.2 Model Matematis	26
BAB 4 PENGOLAHAN DATA DAN ANALISIS	30
4.1 Penyusunan Algoritma	30
4.1.1 Langkah-langkah Penyusunan Algoritma Genetika	30
4.1.2 Verifikasi dan Validasi Program	36
4.1.2.1 Hasil Eksekusi Program	37
4.1.2.2 Hasil Perhitungan Manual	37
4.1.3 Input Data	40
4.2 Hasil Pengolahan Data	40
4.3 Analisis	41
4.3.1 Analisis Metode	41
4.3.2 Analisis Program	43
4.3.3 Analisis Skenario Parameter	45
BAB 5 KESIMPULAN DAN SARAN	47
5.1 Kesimpulan	47
5.2 Saran	47
DAFTAR PUSTAKA	49

DAFTAR TABEL

Tabel 2.1 Strategi <i>rank-based selection</i>	20
Tabel 4.1 Data pasien untuk verifikasi dan validasi program	36
Tabel 4.2 Data ketersediaan dokter bedah.....	36
Tabel 4.3 Parameter yang digunakan dalam validasi.....	36
Tabel 4.4 Kromosom terbaik dari proses verifikasi dan validasi	37
Tabel 4.5 Populasi awal (dari program).....	37
Tabel 4.6 Evaluasi individu pertama populasi awal	38
Tabel 4.7 Evaluasi individu anak.....	40



DAFTAR GAMBAR

Gambar 1.1	Diagram keterkaitan masalah	3
Gambar 1.2	Diagram alur metode penelitian.....	6
Gambar 2.1	Pemrosesan solusi pada Algoritma Genetika	14
Gambar 2.2	Tiga jenis skema pengkodean kromosom	15
Gambar 2.3	Pengkodean variabel berdomain kontinyu	16
Gambar 2.4	<i>Trade-off</i> antara ukuran populasi dengan kualitas solusi	17
Gambar 2.5	Strategi seleksi <i>roulette wheel</i>	18
Gambar 2.6	Strategi <i>tournament selection</i>	18
Gambar 2.7	<i>One-point crossover</i>	21
Gambar 2.8	<i>Two-point crossover</i>	21
Gambar 2.9	<i>Uniform crossover</i>	21
Gambar 4.1	Kromosom representasi solusi.....	30
Gambar 4.2	Alur evaluasi kendala kapasitas ruang dan ketersediaan dokter.....	32
Gambar 4.3	Validasi pindah silang secara manual	39
Gambar 4.4	<i>Interface</i> program penjadwalan kamar operasi.....	44

DAFTAR LAMPIRAN

Lampiran 1: Data Pasien dan Dokter.....	50
Lampiran 2: Kode Program MATLAB Algoritma Penjadwalan Kamar Operasi ..	51
Lampiran 3: Jadwal Optimal Hasil 10000 Iterasi.....	57
Lampiran 4: Tabel Verifikasi Program.....	59



BAB 1

PENDAHULUAN

1.1 Latar Belakang

Rumah sakit sebagai penyedia layanan kesehatan bisa dipandang sebagai sebuah sistem produksi multi-layanan yang dibatasi oleh ketersediaan material dan sumber daya manusia yang terbatas (Guinet dan Chaabane, 2003). Jumlah populasi yang terus bertumbuh, meningkatnya jumlah orang lanjut usia, dan munculnya penyakit-penyakit baru menjadi faktor pendorong meningkatnya jumlah pasien dengan cepat. Karena keterbatasan sumber daya, sebagian besar pasien tidak bisa segera ditangani. Semakin lama mereka menunggu untuk dilayani, semakin tidak puas mereka. Di sisi lain, beberapa rumah sakit, terutama yang publik, mengalami defisit anggaran yang cukup besar. Penggunaan sumber daya yang mahal secara optimal menjadi agenda yang cukup mendesak bagi rumah sakit zaman sekarang ini untuk menekan biaya. Namun, hal ini harus dilakukan tanpa mengorbankan kualitas layanan mengingat rumah sakit merupakan institusi yang sensitif terhadap kesehatan dan keselamatan pasien.

Salah satu proses penting dalam operasi sebuah rumah sakit adalah pembedahan/operasi. Proses ini penting karena bukan saja biaya pelaksanaannya yang tinggi, tetapi juga karena pengaruhnya yang dirasakan langsung oleh pasien dan terhadap persepsi kualitas rumah sakit. Kamar bedah merupakan pusat biaya terbesar sekaligus sumber pendapatan terbesar bagi rumah sakit (Gordon *et al.*, 1988; Macario *et al.*, 1995; Health Care Financial Management Association, 2005). Litvak dan Long (2000) menyatakan bahwa kamar bedah bisa dianggap sebagai mesin sebuah rumah sakit. Banyak sumber daya lain, seperti dokter bedah, ahli bius dan tempat tidur pasien, sangat bergantung pada perencanaan kamar bedah. Pengaturan kamar bedah dengan demikian menjadi semakin penting masa sekarang ini dan masa yang akan datang. Namun, pada kenyataannya kamar bedah sebagai sumber daya yang penting sekali sering menjadi tempat pemborosan waktu. Weinbroum, Ekstein, dan Ezri (2003) meneliti beberapa kamar bedah pada sentra kesehatan dan menemukan bahwa pemborosan waktu yang terjadi sebesar 15% dari waktu total. Oleh karena itu, penjadwalan bedah

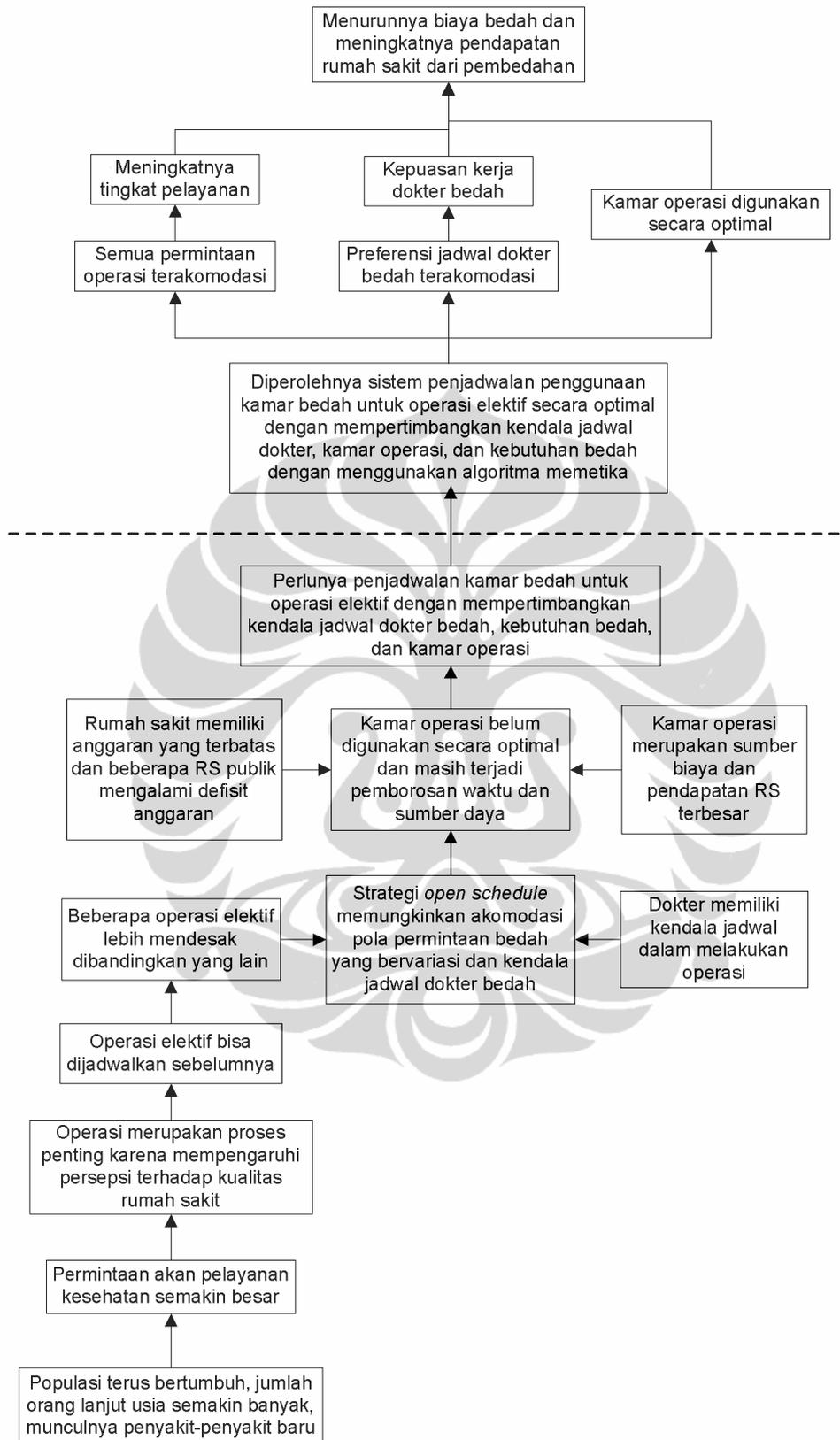
dan pencocokan kebutuhan pasien dengan kamar bedah dan ahli bedah merupakan salah satu topik penelitian yang penting dalam dunia medis.

Ada dua jenis kasus pembedahan: kasus elektif dan kasus urgen. Waktu bedah untuk kasus elektif sudah dijadwalkan jauh-jauh hari sedangkan kasus urgen tidak diduga-duga dan harus diatur mendesak. Oleh karenanya, kasus elektif merupakan salah satu penentu penting kapasitas kamar bedah dan yang akan dibahas dalam persoalan penjadwalan kamar bedah.

Sebelum penjadwalan kamar bedah bisa dilakukan, strategi perencanaan kamar bedah harus ditentukan. Pada dasarnya ada tiga jenis ada tiga jenis strategi penjadwalan kamar bedah yang bisa mempengaruhi biaya dan kualitas pelayanan pasien (Kharraja, 2003), yaitu:

- Strategi *open schedule*. Pada strategi ini, ahli bedah bisa memilih hari kerja apa saja untuk melakukan operasi. Patterson (1996) menyederhanakannya sebagai strategi '*first-come, first-served*'.
- Strategi blok. Ahli bedah atau sekelompok ahli bedah dialokasikan sejumlah blok waktu untuk mengadakan pembedahan. Secara teori, ahli bedah atau kelompok ahli bedah tersebut 'memiliki' blok waktu ini, yang memang dikhususkan dan tidak bisa dilepaskan untuk yang lain selama periode perencanaan sekalipun beberapa mungkin tidak digunakan.
- Strategi blok termodifikasi. Strategi blok bisa dimodifikasi dengan dua cara untuk menyediakan fleksibilitas yang lebih besar: dengan membiarkan beberapa kamar bedah memiliki blok waktu yang sudah dialokasikan dan beberapa lagi *open schedule*, atau dengan membebaskan penggunaan blok waktu yang tidak digunakan kepada yang lain beberapa waktu (misal 72 jam) sebelum pembedahan dilakukan.

Strategi *open schedule* memiliki keunggulan bisa mengakomodasi sebanyak mungkin pasien sesuai permintaan pembedahan pada saat itu. Hal ini sangat baik untuk mengantisipasi pola permintaan bedah yang bervariasi dan juga mengakomodasi kendala-kendala dari pihak dokter bedah. Penelitian ini akan ditujukan untuk penjadwalan dengan menggunakan strategi *open schedule*.



Gambar 1.1 Diagram keterkaitan masalah

1.2 Diagram Keterkaitan Masalah

Untuk mendapat gambaran yang lebih jelas mengenai permasalahan yang akan diteliti, dibuatlah diagram keterkaitan masalah pada Gambar 1.1.

1.3 Rumusan Masalah

Dari latar belakang yang sudah dipaparkan sebelumnya, rumusan masalah dalam penelitian ini adalah perlunya dilakukan penjadwalan kamar operasi dengan mempertimbangkan kendala-kendala dari pihak dokter bedah, permintaan bedah, dan juga kapasitas kamar bedah untuk meningkatkan tingkat pelayanan rumah sakit, kepuasan kerja dokter bedah, menggunakan kamar bedah secara optimal sehingga menurunkan biaya bedah dan meningkatkan pendapatan rumah sakit dari pembedahan.

1.4 Tujuan Penelitian

Penelitian ini bertujuan untuk memperoleh sistem penjadwalan penggunaan kamar operasi untuk operasi elektif yang mempertimbangkan kendala jadwal pihak dokter bedah, kebutuhan bedah, dan kapasitas kamar bedah.

1.5 Ruang Lingkup Penelitian

Penelitian ini akan memiliki batasan-batasan sebagai berikut:

- a. Penjadwalan akan dilakukan terhadap dokter bedah, kamar operasi, dan pasien untuk operasi elektif. Kasus urgen/darurat tidak dipertimbangkan di sini. Sumber daya yang dibutuhkan untuk pembedahan selain dokter bedah dianggap melimpah.
- b. Operasi bersifat *nonpreemptive*, artinya jika sudah dimulai tidak bisa dinterupsi sampai selesai.
- c. Durasi operasi dianggap deterministik.
- d. Kamar operasi yang dijadwalkan bersifat multifungsi, artinya bisa menampung segala jenis operasi, dan terbuka pukul 07.00-17.00 setiap hari dalam rentang waktu penjadwalan.
- e. Strategi yang digunakan dalam penjadwalan kamar bedah adalah strategi penjadwalan *open schedule*.
- f. Horizon penjadwalan adalah lima hari kerja dalam satu minggu.

1.6 Metode Penelitian

Penelitian ini dilakukan dalam lima tahap, yaitu:

a. Perumusan masalah

Pada tahap ini dilakukan identifikasi masalah yang akan diteliti, penentuan tujuan penelitian, pembatasan ruang lingkup penelitian, serta menentukan data-data apa saja yang dibutuhkan.

b. Studi pustaka

Literatur dan teori yang berkaitan dengan penelitian akan dipelajari untuk mendukung penelitian yang dilakukan. Teori yang akan dibahas adalah terkait dengan sistem penjadwalan kamar bedah rumah sakit, model program integer sebagai kerangka pemodelan penjadwalan, dan algoritma genetika sebagai jalan penyelesaian persoalan tersebut.

c. Pengumpulan data

Data-data yang dibutuhkan akan berupa berapa jumlah kamar bedah, jumlah dokter bedah, kendala-kendala yang mungkin menjadi faktor pembatas dokter bedah dalam penjadwalan bedah, lama waktu yang dibutuhkan dalam melakukan bedah, serta jumlah permintaan bedah yang akan dijadwalkan.

d. Pembangunan model, pembuatan program, dan analisis

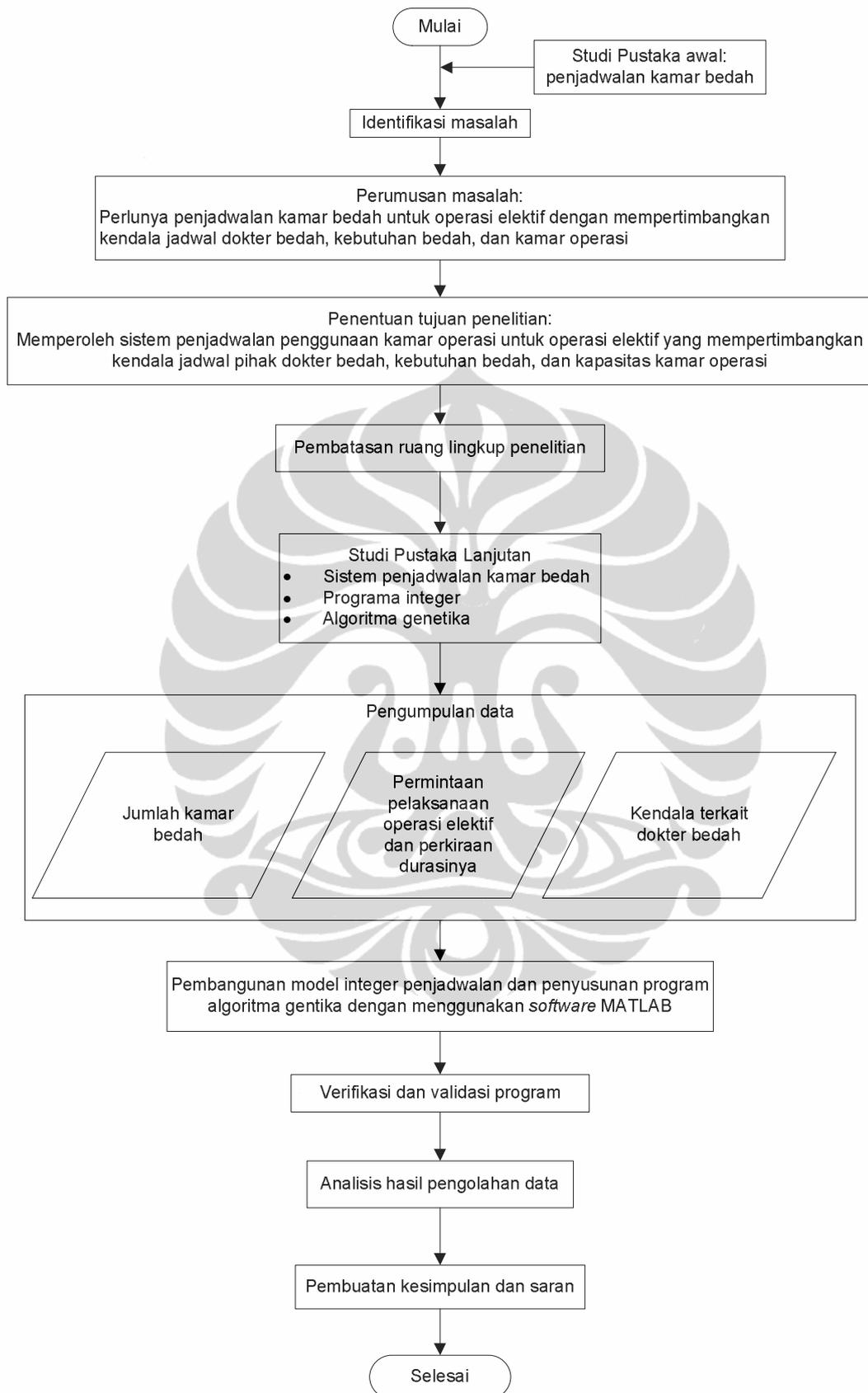
Di sini akan didefinisikan variabel-variabel keputusan, fungsi tujuan, fungsi kendala kemudian dibuatlah sebuah algoritma genetika untuk menyelesaikan model tersebut. Hasil yang didapat kemudian dianalisis.

e. Pembuatan kesimpulan dan saran

Kesimpulan kemudian ditarik dari seluruh proses penelitian dan saran-saran terkait penjadwalan kamar bedah dan penelitian selanjutnya diberikan.

1.7 Sistematika Penulisan

Untuk mempermudah pemahaman alur penelitian ini, maka penulisan penelitian mengenai optimasi penjadwalan kamar operasi ini disajikan beberapa bab. Bab pertama adalah Pendahuluan yang berisikan penjelasan mengenai latar belakang permasalahan yang mendorong dilakukannya penelitian ini. Tujuan, asumsi dan batasan ruang lingkup, serta metodologi penelitian juga dipaparkan dalam bab ini. Pendahuluan juga dilengkapi dengan diagram keterkaitan masalah untuk memahami alur faktor-faktor yang mengakibatkan munculnya masalah



Gambar 1.2 Diagram alir metode penelitian

yang akan diteliti sampai kepada *problem statement* sampai kepada apa yang diharapkan dihasilkan dari penelitian ini.

Bab kedua berisi landasan teori yang digunakan dalam mengerjakan penelitian ini. Landasan teori diperoleh dengan melakukan tinjauan pustaka dari buku dan artikel jurnal. Topik yang dipaparkan berkenaan dengan sistem penjadwalan kamar operasi di rumah sakit, program integer sebagai acuan model matematis, serta algoritma genetika yang digunakan untuk mengoptimasi penjadwalan.

Pada bab ketiga dijelaskan mengenai kebutuhan data dalam penelitian ini. Data tersebut akan berupa jumlah kamar operasi yang akan dijadwalkan, rentang waktu penjadwalan, jumlah pasien yang akan dioperasi beserta durasi masing-masing operasi, jumlah dokter beserta waktu ketersediaan dokter untuk melakukan operasi dalam rentang waktu penjadwalan, serta tingkat keterdesakan waktu operasi masing-masing pasien. Bab ini kemudian akan membahas tentang model matematis penjadwalan kamar operasi sebagai acuan untuk membangun algoritma genetika penjadwalan.

Bab keempat menjelaskan proses pengolahan data dan analisis. Pengolahan data akan berupa langkah-langkah penyusunan algoritma genetika, verifikasi dan validasi kesesuaian program dengan yang diharapkan, penyelesaian penjadwalan dengan data yang ada. Hasil pengolahan data kemudian dianalisis berdasarkan

Setelah pengolahan data dan analisis dilakukan, maka dapat ditarik suatu kesimpulan dari hasil penelitian secara keseluruhan. Kemudian saran-saran yang berguna terkait penjadwalan kamar operasi maupun untuk penelitian selanjutnya diajukan. Kesimpulan dan saran ditemukan pada bab kelima.

BAB 2 DASAR TEORI

2.1 Sistem Penjadwalan Kamar Operasi

Kamar bedah adalah sumber utama pendapatan untuk rumah sakit modern sehingga penjadwalan secara teliti sangat penting terhadap profitabilitas rumah sakit. Selain itu, kamar bedah juga menawarkan area yang luas untuk pengontrolan biaya karena 1) kamar bedah memiliki biaya yang tinggi dan biasanya tingkat utilisasi fasilitas dan sumber daya manusia yang rendah dan 2) pasien bedah merupakan bagian besar dari permintaan yang dilayani oleh departemen lain dalam rumah sakit (Magerlein dan Martin, 1978; Dexter dan Traub, 2002). Penjadwalan pasien kamar bedah menempatkan pasien, staf (dokter bedah, ahli bius/anestesi, perawat, dll), perlengkapan dan peralatan ke ruangan tertentu pada waktu tertentu dalam departemen pembedahan. Penjadwalan secara efisien bisa sekaligus meningkatkan pendapatan dan menurunkan biaya, dengan demikian meningkatkan keuntungan.

Proses pembedahan sendiri bisa dibagi menjadi dua, yaitu:

1. Pembedahan darurat: terjadi pada kasus-kasus mendesak untuk menyelamatkan hidup seseorang dan menjaga fungsi organ tubuhnya dan tidak bisa direncanakan karena tidak terduga. Pembedahan jenis ini harus dilakukan dalam selang waktu 24 jam.
2. Pembedahan elektif: pembedahan yang bisa ditunda sesuai tingkat keseriusan penyakitnya dan telah direncanakan sebelumnya sehingga bisa dijadwalkan.

Dari kedua jenis pembedahan di atas, maka kamar operasi juga dibagi menjadi dua macam, yakni:

1. Kamar *non-emergency*: kamar bedah yang dikhususkan untuk menangani pembedahan elektif dan biasanya dibedakan berdasarkan spesialisasi pembedahan yang dilakukan.
2. Kamar *emergency*: kamar bedah yang dikhususkan untuk menangani operasi darurat dan biasanya dapat melayani segala jenis spesialisasi kasus darurat.

Karena permintaan bedah elektif relatif lebih sering terjadi sepanjang waktu dibanding bedah darurat, rata-rata rumah sakit memiliki lebih banyak kamar

bedah elektif dibanding darurat. Kamar bedah darurat tentu saja akan melayani pasien darurat secepat mungkin tergantung tingkat kegentingan dan waktu tibanya. Ketidakterdugaan kasus darurat menyebabkan penjadwalan jauh sebelumnya menjadi sulit. Oleh karena itu, rumah sakit berfokus menjadwalkan kamar bedah elektif.

Penjadwalan yang tidak efisien menyebabkan terjadinya waktu nganggur antar-kasus, biaya *overtime* yang besar, peningkatan kecemasan pasien karena penundaan, dan sangat mungkin ketidakpuasan ahli bedah. Kepuasan kerja ahli bedah adalah salah satu faktor paling penting untuk dipertimbangkan dalam penjadwalan karena ahli bedah merupakan sumber daya yang mahal dan penting. Kepuasan ini bisa dicapai dengan memastikan probabilitas yang besar untuk waktu mulai dan membuat jadwal yang mereka rasa cukup adil. Di samping itu, penjadwalan kamar bedah harus dikoordinasikan secara hati-hati dengan area lainnya di rumah sakit tersebut, terutama dengan *post-anesthesia care unit* (PACU), kamar rawat, serta jadwal perawat.

Tujuan penjadwalan yang sering digunakan dalam pemilihan sistem penjadwalan kamar bedah yang sering disebutkan dalam literatur mencakup:

- Penggunaan kamar bedah secara efektif dengan mengurangi penundaan dan waktu jeda antara bedah satu dengan bedah selanjutnya.
- Kepuasan kerja ahli bedah.
- Keselamatan dan kepuasan pasien.
- Kepuasan staf kamar bedah.
- Kesederhanaan dan kemudahan penjadwalan.
- Penggunaan PACU (Post Anaesthesia Care Unit) secara efektif.
- Pencapaian tingkat pembatalan bedah yang rendah.

Saat menilai penggunaan kamar operasi, alternatif alat ukur yang digunakan mencakup:

- Total menit kamar bedah digunakan.
- Total waktu utilisasi dibagi dengan total waktu yang tersedia.
- Waktu nganggur perawat sebagai persentase waktu kamar bedah yang tersedia.
- Waktu *turnover*.

- Waktu nganggur ahli bus sebagai persentase waktu kamar bedah yang tersedia.
- Jam utilisasi pada waktu blok dibagi dengan jam blok yang tersedia.

Sistem penjadwalan kamar bedah di rumah sakit menggunakan berbagai macam metode seperti yang dijelaskan di bawah ini.

First come/first served (FC/FS) atau ***open scheduling***. Ini adalah satu dari dua metode paling umum yang digunakan dalam penjadwalan kamar bedah. Metode FC/FS mengalokasikan waktu pembedahan kepada ahli beda yang pertama kali memintanya. Beberapa rumah sakit membuat batasan jumlah waktu yang dialokasikan untuk ahli bedah tersebut atau perkiraan lama bedah. Masalah yang sering terjadi pada metode ini adalah:

- Tingkat pembatalan yang tinggi karena terlalu banyak pemesanan
- Tingkat penggunaan kamar bedah yang berbeda di antara spesialisasi bedah
- Waktu lembur dan nganggur bersamaan: kasus yang dibatalkan menyebabkan waktu nganggur sedangkan kesulitan operasi menyebabkan waktu lembur (Dexter dan Traub, 2002)

Keuntungan terbesar dari metode ini adalah kemudahan penjadwalan dan fleksibilitas yang besar.

Block scheduling. Metode ini adalah metode populer selanjutnya. Di sini blok waktu kamar operasi dialokasikan ke setiap ahli bedah atau kelompok ahli bedah. Panjang satu blok waktu biasanya setengah atau satu hari. Blok tersebut diberikan hak penggunaan eksklusif kepada ahli bedah atau kelompok ahli bedah sampai batas tanggal tertentu, biasanya dua hari sebelum pembedahan, dimana blok yang tidak digunakan akan disediakan untuk ahli bedah lain (Magerlein dan Martin, 1978).

Keuntungan terbesar dengan sistem ini adalah meningkatnya utilisasi melalui penggunaan kamar bedah di sore hari. Sistem ini juga memungkinkan ahli bedah untuk mengetahui waktu mulainya lebih awal dan lebih pasti. Waktu kerja lebih panjang pada sore hari menjadi tanggung jawab ahli beda tersebut, sehingga dia tidak bisa memindahkan tanggung jawab jika terjadi penundaan. Sistem ini juga

mengurangi persaingan ahli bedah atas penjadwalan, dan mungkin mengurangi tugas administratif, pembatalan, dan waktu tunggu pembedahan.

Kekurangan utamanya adalah blok waktu yang tidak digunakan sering ditahan oleh ahli bedah sampai batas tanggal walaupun sebenarnya tidak dibutuhkan. Hal ini menyebabkan waktu nganggur yang mahal. Selain itu, blok yang dikhususkan bisa menunda kasus bedah yang urgen sampai ahli bedah pasien tersebut mendapatkan blok waktunya.

Dynamic block scheduling. Metode ini adalah variasi *block scheduling* dimana masing-masing penggunaan blok waktu ahli bedah ditinjau secara reguler (per kuartel atau semester). Jumlah blok waktu yang akan dialokasikan akan disesuaikan berdasarkan analisis yang dilakukan.

Longest case first (LCF). Metode LCF mengalokasikan prosedur terpanjang ke slot waktu terawal yang ada. Sistem ini memungkinkan spesialis tertentu (misal ahli dada) untuk selalu mendapat slot pagi yang bisa menyebabkan spesialis lain tidak puas. Sistem ini menganggap bahwa semakin panjang lama suatu operasi maka semakin besar tingkat variabilitas waktu operasi. Oleh karena itu, seiring dengan berjalannya waktu dalam satu hari, kasus selanjutnya bisa digeser ke belakang untuk menyelesaikan beban operasi sebisa mungkin tepat waktu.

Shortest case first (SCF). SCF digunakan untuk menjaga beban yang seimbang di PACU; prosedur terpendek dilakukan pada pagi hari. Sistem LCF biasanya menyebabkan rendahnya utilisasi dan banyaknya waktu nganggur PACU pada pagi hari.

Top down/bottom up. Metode ini juga termasuk modifikasi *block scheduling* dimana satu hari dibagi menjadi dua blok. Kasus yang panjang dijadwalkan secara FC/FS pada blok pagi sedangkan kasus pendek dijadwalkan FC/FS diakhir hari kerja. Jika terjadi waktu nganggur pada blok yang lebih panjang, pasien yang datang dengan prosedur pendek dipilih untuk menutup celah waktu. Jika waktu pada blok panjang tidak cukup, maka kasus panjang bisa dijadwalkan pada awal blok kasus pendek. Ahli bedah dengan beberapa kasus bedah dijadwalkan di ruang yang sama untuk mengurangi waktu nganggur antar-bedah.

Multiple room system. Ahli bedah biasanya ditentukan ke sebuah kamar bedah; namun, pada sistem ini, ahli bedah dijadwalkan untuk rotasi dari satu kamar ke kamar lainnya. Sistem ini mencoba mengeliminasi waktu tunggu antar-pembedahan selama masa pembersihan dan penyiapan kamar bedah serta persiapan bedah. Karena waktu antar-kasus sudah diperkirakan antara 20 dan 45 menit, sistem ini bisa menghemat biaya cukup besar untuk ahli bedah dan menurunkan waktu lembur staf bedah.

Jadi, masalah yang dihadapi dalam suatu penjadwalan kamar bedah adalah bagaimana menugaskan sejumlah sumber daya yang tersedia (staf medis atau fasilitas fisik dan material) untuk menangani sejumlah operasi elektif pada periode waktu tertentu beserta urutannya secara optimal dengan mempertimbangkan kapasitas yang ada dengan tujuan meminimalkan biaya, meminimalkan waktu lembur, memaksimalkan utilisasi kamar bedah, kepuasan pasien, atau tujuan lainnya.

2.2 Program Integer

Program integer adalah program linier dimana semua atau beberapa variabel terbatas pada nilai integer (diskrit). Program integer bisa diformulasikan sebagai berikut:

Fungsi tujuan:

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

Dengan kendala:

$$\sum_{j=1}^n a_{ij} \cdot x_j (\leq = \geq) b_j \quad (i = 1, 2, \dots, m), \quad (2.2)$$

$$x_j \geq 0 \quad (j = 1, 2, \dots, n), \quad (2.3)$$

$$x_j \text{ integer} \quad (\text{for some or all } j = 1, 2, \dots, n). \quad (2.4)$$

Saat sebagian variabel saja yang terbatas harus nilai integer, persoalan ini disebut *mixed integer program*. Jika semua variabel keputusan harus memiliki nilai integer, persoalan ini disebut sebagai *pure integer program*.

Program integer bisa diklasifikasikan menjadi dua, yaitu *binary* dan *general*. Pada *binary integer programming* nilai variabel terbatas hanya 0 atau 1. Program jenis ini digunakan untuk memodelkan permasalahan keputusan *go/no-go*, iya atau tidak. Persoalan penjadwalan dimodelkan sebagai *binary integer*

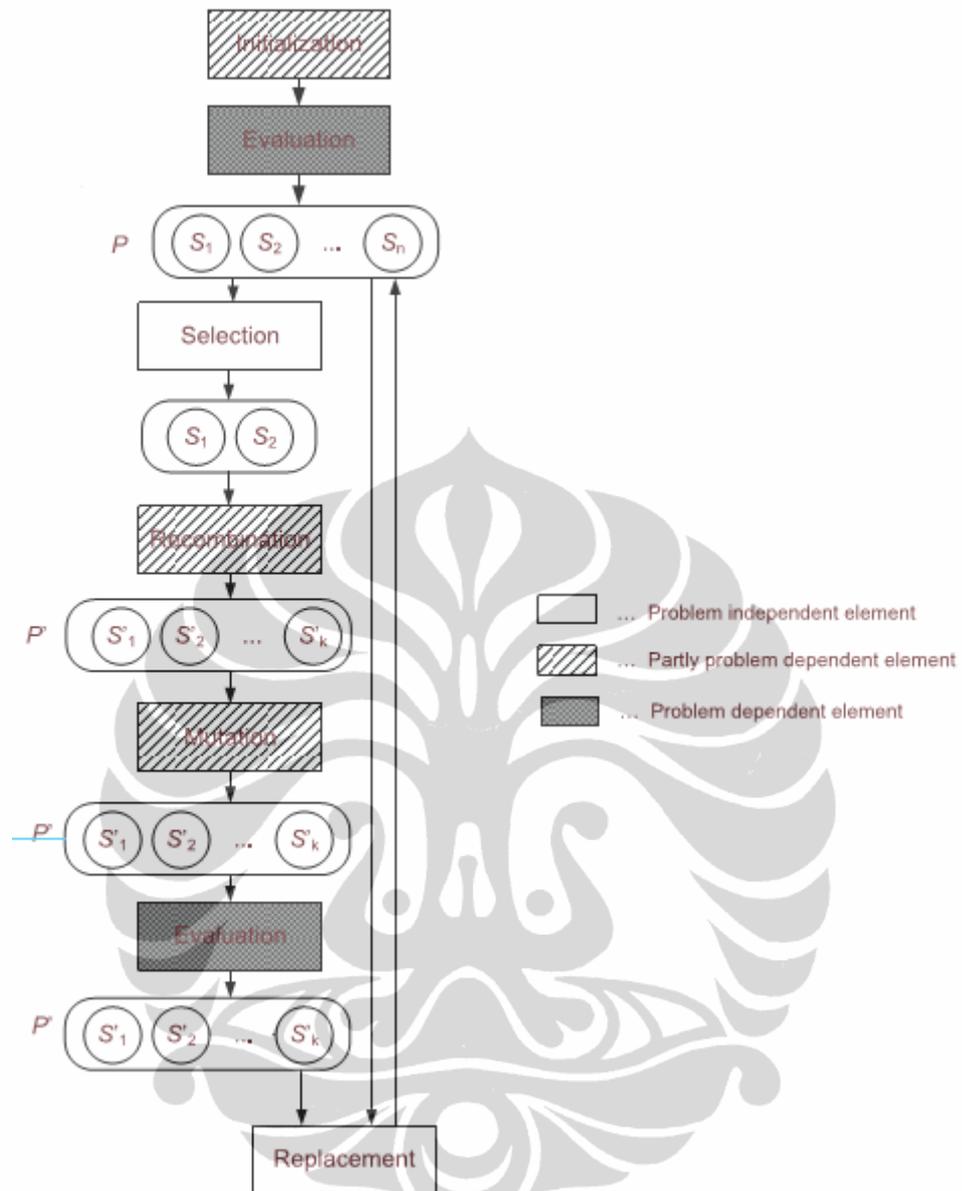
program dengan variabel bernilai 1 berarti manusia/mesin dijadwalkan untuk bekerja pada suatu waktu tertentu sedangkan nilai 0 berarti tidak bekerja. Program integer *general* memiliki nilai variabel integer yang tidak terbatas pada 0 atau 1.

2.3 Algoritma Genetika (AG)

Kemunculan AG untuk optimasi diinspirasi oleh proses evolusi biologi dan menggunakan gagasan seleksi alam, rekombinasi, dan mutasi untuk mengontrol proses pencarian solusi. Rekombinasi memaksa algoritma genetika menggunakan lebih dari satu kandidat solusi. Himpunan kandidat solusi disebut *populasi*. Sebuah solusi dari populasi tersebut disebut *individu* atau *kromosom*. Kromosom terbuat dari gen dan nilai dari sebuah gen disebut alel. Sebuah gen merupakan properti sebuah solusi atau bisa disebut sebagai variabel keputusan, yang bisa mengambil nilai dari domain yang sudah didefinisikan terlebih dahulu. Posisi gen dalam kromosom disebut lokus. Solusi yang akan direkombinasi disebut orangtua dan solusi yang dihasilkan disebut anak, sedangkan proses rekombinasi sendiri disebut pindah silang. Kualitas individu solusi dalam algoritma genetika diukur dari nilai fitness. Nilai fitness setiap individu harus bisa dibandingkan karena hal ini dibutuhkan dalam proses seleksi dan penggantian populasi.

Alur pemrosesan solusi pada algoritma genetika ditunjukkan pada Gambar 2.1. Pertama, metode inisialisasi menciptakan populasi awal. *Fitness* semua individu populasi awal kemudian dievaluasi. Setelah dua langkah ini, fase inisialisasi sudah selesai dan kita masuk ke *loop* pencarian utama.

Dari populasi tadi, metode seleksi menentukan individu mana yang akan mengalami rekombinasi/pindah silang. Pada AG tradisional dua solusi yang akan dipilih. Keduanya akan direkombinasi dengan menggunakan sebuah operator pindah silang. Operator ini menggabungkan properti yang ada pada kedua solusi untuk menghasilkan solusi baru. Setelah rekombinasi, sejumlah persentase tertentu individu (biasanya kecil) akan dimutasi. Artinya, suatu gangguan (*perturbation*) kecil acak akan dilakukan terhadap solusi. Akhirnya, solusi baru akan dievaluasi dan dikumpulkan pada sebuah populasi baru sementara (P').



Gambar 2.1 Pemrosesan solusi pada Algoritma Genetika

(Sumber: Zäpfel, Braune, dan Bögel, 2010)

Dengan cara itu sebuah himpunan solusi baru dihasilkan sampai populasi sementara memiliki jumlah solusi yang cukup untuk mengalami penggantian populasi. Ukuran populasi sementara berbeda-beda antara algoritma genetika yang satu dengan yang lain.

Setelah penciptaan populasi sementara selesai, operator penggantian memilih dari himpunan $P \cup P'$ individu mana yang digunakan untuk membentuk sebuah populasi baru P untuk iterasi selanjutnya. Kita harus memilih solusi mana

dari populasi baru yang akan dimasukkan dan solusi mana dari populasi lama yang akan digantikan oleh yang baru.

Populasi baru akan mengalami proses yang sama lagi sampai kriteria berhenti tertentu terpenuhi.

Komponen-Komponen Algoritma Genetika

Komponen-komponen sebuah algoritma genetika adalah skema pengkodean kromosom, pembentukan populasi awal, evaluasi, seleksi, pindah silang, mutasi, penggantian populasi, serta kriteria berhenti. Berikut penjelasan masing-masing komponen.

a. Skema Pengkodean Kromosom

Pada AG, solusi akan direpresentasikan dalam sebuah kromosom/individu. Oleh karena itu, diperlukan pemilihan sistem pengkodean kromosom yang sesuai untuk masalah yang dihadapi. Secara umum, kromosom berbentuk:

$$\text{Kromosom} = (\text{gen}_1, \text{gen}_2, \dots, \text{gen}_n)$$

dengan n adalah jumlah gen dalam satu kromosom dan setiap gen diisi oleh nilai yang disebut *allele*.

Ada tiga skema pengkodean kromosom yang umum digunakan, yaitu:

- Pengkodean bilangan riil. Nilai setiap gen berada dalam interval $[0, R]$, dengan R merupakan bilangan positif yang biasanya bernilai 1.

1	0	1	1	1	0	0	0	1	1	1	1
gen1			gen2				gen3			gen4	
<i>Pengkodean bilangan biner</i>											
2			40				26			11	
gen1			gen2				gen3			gen4	
<i>Pengkodean bilangan integer</i>											
5		3			1		2		4		
gen1		gen2			gen3		gen4		gen5		
<i>Pengkodean permutasi 1-5</i>											
0.1876			0.9876				0.4523			0.6782	
gen1			gen2				gen3			gen4	
<i>Pengkodean bilangan riil</i>											

Gambar 2.2 Tiga jenis skema pengkodean kromosom

- Pengkodean bilangan diskrit/integer. Nilai setiap gen dipilih dari salah satu bilangan bulat dalam interval $[0, N]$. Pengkodean ini juga mencakup permutasi dari suatu rentang nilai diskrit.
- Pengkodean bilangan biner. Setiap gen akan bernilai satu atau serentetan 0 atau 1.

Domain setiap variabel keputusan akan ditentukan. Rentang domain ini kemudian dikonversi ke nilai gen sesuai dengan skema pengkodean yang digunakan. Misalkan saja variabel x memiliki domain kontinyu $0 \leq x \leq 1$ akan dikodekan dengan menggunakan tiga rentetan bilangan biner. Proses pengkodeannya bisa dilihat pada Gambar 2.3. Rentetan bilangan biner tidak bisa mewakili domain yang kontinyu secara sempurna. Oleh karena itu, nilai satu gen akan diterjemahkan menjadi suatu rentang nilai, bukan satu titik nilai variabel keputusan. Gen bernilai 1110101 pada contoh akan memberikan variabel keputusan rentang nilai 0.500 – 0.625. Untuk mendapatkan rentang yang lebih sempit, kita harus menggunakan rentetan bilangan biner yang lebih banyak. Pengkodean integer biasanya digunakan untuk merepresentasikan variabel keputusan yang bernilai diskrit.

		variable values				
		0.55	0.11	0.95	0.63	
1.000	111			.		0.9375
0.875	110					0.8125
0.750	101				.	0.6875
0.625	100	.				0.5625
0.500	011					0.4375
0.375	010					0.3125
0.250	001					0.1875
0.125	000		.			0.0625
		0.625	0.125	1.000	0.750	quantized hi
		0.500	0.000	0.875	0.625	quantized lo
		0.5625	0.0625	0.9375	0.6875	quantized mid
		100	000	111	101	chromosome

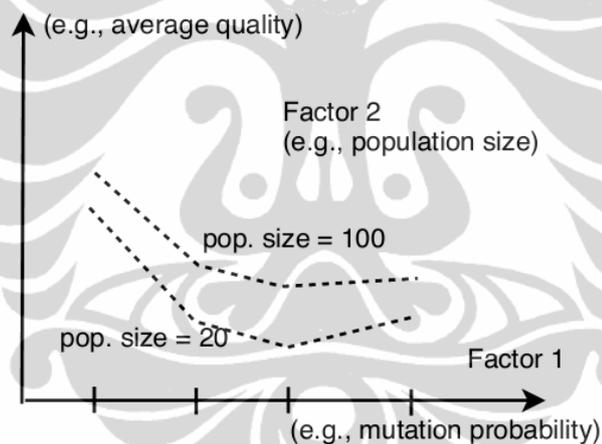
Gambar 2.3 Pengkodean variabel berdomain kontinyu $0 \leq x \leq 1$ dengan tiga rentetan bilangan biner

(Sumber: Haupt dan Haupt, 2004)

b. Pembentukan Populasi Awal

Setelah cara pengkodean solusi sudah ditemukan, dibuatlah suatu populasi awal yang berisikan sekumpulan individu, biasanya berjumlah genap. Tidak ada

aturan khusus berapa seharusnya jumlah individu dalam satu populasi. Hal ini akan bergantung pada kasus yang sedang diselesaikan, parameter operator genetika yang digunakan, serta waktu komputasi yang diharapkan. Semakin besar ukuran suatu populasi, maka konvergensi menuju solusi yang ‘baik’ akan lebih besar. Namun, waktu pencarian AG akan berkembang secara linear sejalan dengan ukuran populasi. Harus dilakukan *trade-off* antara kualitas solusi yang diperoleh dengan ukuran populasi dan waktu pencarian. Pada praktiknya, ukuran populasi yang sering digunakan adalah antara 20 dan 100. Secara teoritis, ukuran populasi harus berkembang secara eksponensial sesuai dengan ukuran individu (banyaknya gen). Di sisi lain, permasalahan yang sangat kompleks dan *constrained* sebaiknya memiliki ukuran populasi yang tidak terlalu besar karena ukuran populasi yang terlalu besar akan menghasilkan lebih banyak individu yang tidak memenuhi kendala.



Gambar 2.4 *Trade-off* antara ukuran populasi dan probabilitas mutasi dengan kualitas solusi
(Sumber: Talbi, 2010)

Nilai gen-gen dari setiap individu dalam populasi awal ini ditentukan secara acak sesuai dengan konversi domain variabel keputusan menjadi nilai gen dan kendala permasalahan. Setelah ukuran populasi ditentukan, dilakukan inisialisasi kromosom sebanyak ukuran tersebut.

c. Evaluasi

Pada setiap generasi dalam AG, semua kromosom dievaluasi dengan menggunakan fungsi *fitness*. Fungsi ini merupakan fungsi untuk mengukur secara kuantitatif kualitas kandidat solusi. Nilai *fitness* sebuah solusi diperlukan dalam

proses seleksi dan penggantian populasi. Semakin tinggi nilai *fitness* suatu individu, maka semakin besar kemungkinan individu tersebut akan bertahan ke generasi selanjutnya. Nilai *fitness* akan menjadi pemandu arah pencarian solusi dalam AG. Fungsi *fitness* tidak harus selalu sama dengan fungsi objektif. Fungsi *fitness* bisa mencakup juga seberapa besar kendala permasalahan tidak dilanggar.

d. Seleksi

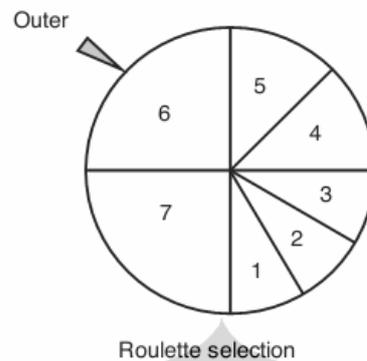
Seleksi adalah proses pemilihan dua kromosom sebagai orangtua untuk mengalami rekombinasi/pindah silang. Prinsip utama suatu metode seleksi adalah semakin baik suatu individu, semakin besar peluangnya untuk menjadi orangtua. Tekanan semacam itu akan mengantarkan populasi menuju solusi yang lebih baik. Namun, individu terburuk tidak harus dibuang dan tetap diberikan peluang untuk terpilih walaupun kecil. Hal ini bisa membawa kepada material genetik yang berguna saat pindah silang.

Strategi seleksi menentukan bagaimana individu dipilih untuk mengalami pindah silang. Ada beberapa strategi seleksi dalam AG, yaitu:

- *Roulette Wheel Selection*. Strategi ini adalah yang paling umum digunakan. Setiap individu diberikan sebuah probabilitas seleksi yang proporsional dengan nilai *fitness* relatifnya. Misalkan f_i adalah nilai *fitness* individu p_i pada populasi P . Probabilitas individu tersebut untuk terpilih adalah $p_i = f_i / (\sum_{j=1}^n f_j)$. Setiap individu kemudian diberi nilai probabilitas kumulatif dari probabilitas yang sudah ditentukan. Anggap sebuah diagram *pie* (roda rulet) dimana setiap individu diberikan sebuah area yang proporsional terhadap nilai *fitness*. Sebuah rulet yang diletakkan di luar keliling diagram tersebut. Seleksi sejumlah individu dilakukan dengan sejumlah putaran roda rulet (membangkitkan sejumlah angka random) secara independen. Satu putaran (angka random) akan memilih individu dengan rentang probabilitas kumulatif yang mencakup angka random tersebut. Individu yang lebih baik memiliki area yang lebih luas sehingga kemungkinan terpilihnya semakin besar.

Pada metode seleksi ini, individu yang sangat baik akan memperkenalkan *bias* pada awal pencarian sehingga bisa mendorong konvergensi lebih awal dan hilangnya keragaman. Hal ini bisa mengarah ke lokal optima. Selain itu, jika

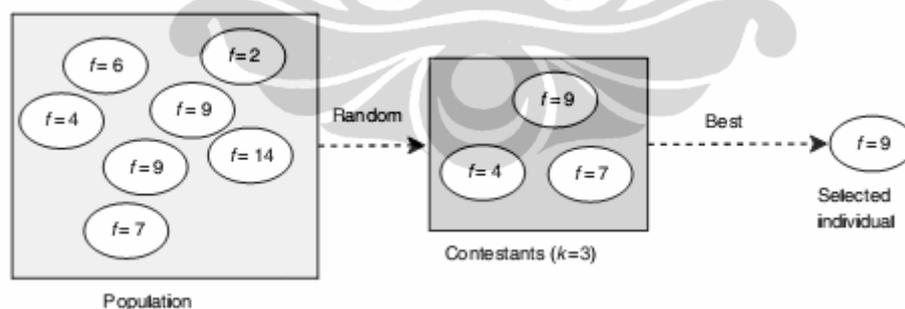
semua individu memiliki nilai *fitness* yang saling berdekatan, strategi ini tidak memperkenalkan tekanan yang cukup untuk memilih individu terbaik.



Gambar 2.5 Strategi seleksi *roulette wheel*

(Sumber: Talbi, 2010)

- *Tournament selection.* Seleksi ini dilakukan dengan memilih sejumlah k individu secara acak; parameter k disebut ukuran grup turnamen. Sebuah turnamen kemudian dilakukan antara k individu tersebut dan dipilihlah satu individu dengan nilai *fitness* terbaik sebagai orangtua. Untuk mendapatkan sejumlah n orangtua, maka harus dilakukan sejumlah n turnamen. Strategi ini bekerja sangat baik pada kasus dengan ukuran populasi yang besar karena mengurutkan berdasarkan nilai *fitness* akan memakan waktu yang lama untuk populasi berukuran besar.



Gambar 2.6 Strategi *tournament selection*

(Sumber: Talbi, 2010)

- *Rank-based selection.* Pada strategi ini, penggunaan nilai *fitness* secara langsung digantikan oleh penggunaan *rank* (peringkat) individual. Pertama, kromosom diurutkan secara menurun dari nilai *fitness* tertinggi sampai terendah. Urutan ini menjadi peringkat masing-masing individu. Probabilitas seleksi kemudian dihitung seperti yang dilakukan pada *roulette selection*

dengan rumus: $p_i = r(s_i) / \sum_{k=1}^n r(s_k)$. Setiap individu diberi nilai probabilitas kumulatif sesuai peringkat. Di bawah ini ditunjukkan sebuah contoh.

Tabel 2.1 Strategi *rank-based selection*

n	Chromosome	P_n	$\sum_{i=1}^n P_i$
1	00110010001100	0.4	0.4
2	11101100000001	0.3	0.7
3	00101111001000	0.2	0.9
4	00101111000110	0.1	1.0

(Sumber: Haupt dan Haupt, 2004)

Sebuah angka random [0,1] akan dibangkitkan dan individu yang terpilih adalah yang rentang probabilitas kumulatifnya mencakup angka random tersebut. Misalkan saja angka random bernilai 0.82, maka individu yang terpilih adalah yang peringkat 3 dengan rentang probabilitas 0.7-0.9.

- *Truncation selection*. Strategi ini mengurutkan semua individu berdasarkan nilai *fitness* dari yang tertinggi sampai yang terendah. Kemudian sejumlah individu urutan teratas dipilih menjadi orangtua. Orangtua dipasangkan dari kromosom indeks ganjil dengan kromosom indeks genap.

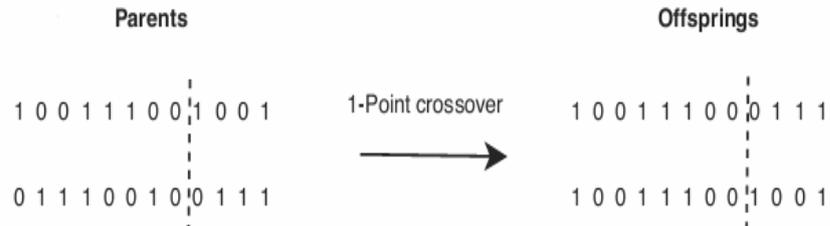
e. Pindah silang

Dua orangtua yang telah dipilih dari proses seleksi akan melalui proses pindah silang oleh operator pindah silang. Tujuan dari pindah silang ini adalah untuk menurunkan beberapa karakteristik orangtua untuk menghasilkan anak. Diharapkan anak mendapatkan karakteristik yang baik dari kedua orangtua sehingga memiliki nilai *fitness* yang lebih baik.

Desain operator pindah silang bergantung pada skema pengkodean kromosom yang digunakan.

- Pindah silang pada skema representasi biner: bisa menggunakan *one-point crossover*, *two-point crossover*, atau *uniform crossover*. Pada *one-point crossover*, sebuah titik pindah silang k dipilih secara acak yang akan memotong masing-masing kromosom orangtua menjadi dua segmen. Anak

pertama akan mendapat segmen pertama dari orangtua pertama dan segmen kedua dari orangtua kedua. Anak kedua akan mendapat segmen pertama dari orangtua kedua dan segmen kedua dari orangtua pertama.



Gambar 2.7 *One-point crossover*

(Sumber: Talbi, 2010)

Pada *two-point crossover*, dua titik pindah silang yang tidak sama akan dipilih secara acak. Dua titik ini akan membagi masing-masing kromosom orangtua menjadi tiga segmen. Anak pertama akan mendapatkan segmen pertama dan ketiga dari orangtua pertama sedangkan segmen keduanya didapat dari orangtua kedua. Anak kedua akan mendapat segmen pertama dan ketiga dari orangtua kedua serta segmen kedua dari orangtua pertama.

*parent*₁ 001 010110 00110

*parent*₂ 011 111000 01100

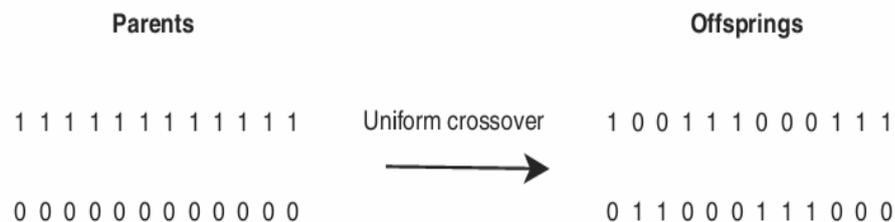
*offspring*₁ 001 111000 00110

*offspring*₂ 011 010110 01100

Gambar 2.8 *Two-point crossover*

(Sumber: Haupt dan Haupt, 2004)

Uniform crossover melakukan pindah silang dengan memilih gen-gen mana yang akan diturunkan ke kromosom anak.



Gambar 2.9 *Uniform crossover*

(Sumber: Talbi, 2010)

- Skema pengkodean bilangan riil menggunakan rekombinasi *mean-centric* dan *parent-centric*. Pada rekombinasi *mean-centric* anak dihasilkan mendekati *centroid* kedua orangtua. Operator ini terdiri dari *intermediate crossover*, *geometrical crossover*, *unimodal normal distribution crossover*, dan *simplex crossover*. Rekombinasi *parent-centric* menghasilkan anak yang salah satu orangtuanya. Setiap orangtua diberikan probabilitas yang sama untuk menghasilkan anak pada *neighborhood*-nya. Operatornya berupa *simulated binary crossover* dan *parent-centric crossover*.
- Pindah silang pada skema pengkodean permutasi tidak bisa menggunakan operator yang digunakan pada skema pengkodean biner karena hasilnya nanti bisa bukan berupa permutasi. Oleh karena itu, perlu operator pindah silang khusus untuk skema pengkodean ini. Operator yang sering digunakan antara lain *order crossover*, *partially mapped crossover*, *two-point crossover*, dll.

f. Mutasi

Mutasi merupakan operator genetika yang mengubah posisi gen dalam kromosom. Tujuan mutasi adalah untuk mendapatkan kembali informasi yang hilang saat pindah silang (menjaga keanekaragaman kromosom), yaitu menghindarkan populasi mengalami konvergensi terlalu cepat. Namun, seperti halnya dalam evolusi biologi, tidak semua individu dalam suatu populasi mengalami mutasi. Pada umumnya, nilai yang kecil direkomendasikan untuk probabilitas mutasi ($p_{mu} \in [0.001, 0.01]$). Nilai probabilitas mutasi bisa ditentukan sebagai $1/k$ dengan k merupakan jumlah variabel keputusan. Oleh karenanya, secara rata-rata hanya satu variabel saja yang dimutasi dari satu kromosom. Jika angka random yang dibangkitkan lebih kecil dari probabilitas mutasi, baru mutasi bisa terjadi pada kromosom tersebut. Satu atau beberapa gen individu yang mengalami mutasi akan mengalami perubahan sehingga individu yang dihasilkan adalah individu baru.

Ada beberapa cara jenis operator mutasi yang tergantung pada skema pengkodean kromosom yang digunakan.

- Mutasi pada skema representasi bilangan biner: operator yang digunakan berupa *bit-flip*. Operator ini bekerja dengan mengubah sebuah bilangan 0 menjadi 1 dan sebaliknya.

- Mutasi pada skema representasi bilangan integer: biasanya operator bekerja dengan mengubah elemen tertentu dengan nilai lain dalam domain variabel keputusan [batas bawah, batas atas], mutasi *swap* (menukarkan dua gen yang berurutan), mutasi *exchange* (menukar dua gen yang berurutan), mutasi *scramble sublist mutation* (memilih dua titik secara acak, kemudian mengubah susunan gen di antara dua titik tersebut secara acak), mutasi *shift* (memilih satu gen secara acak kemudian memasukkannya ke posisi yang dipilih secara acak juga).
- Mutasi pada skema representasi bilangan riil: operator bekerja dengan mengubah gen tertentu dengan nilai lain dalam domain variabel keputusan [batas bawah, batas atas].

g. Pergantian populasi

Setelah rekombinasi, mutasi, dan evaluasi, pada dasarnya ada dua populasi yang berbeda. Karena pada AG biasa ukuran populasi dipertahankan konstan dari generasi ke generasi, maka harus ditentukan individu mana yang akan dipertahankan dan yang mana yang dibuang. Ada dua jenis strategi penggantian populasi, antara lain:

- *Generational replacement*. Semua individu dalam populasi orangtua akan digantikan oleh semua individu anak hasil pindah silang dan mutasi.
- *Steady state replacement*. Pada setiap generasi AG, hanya satu individu anak yang dihasilkan. Kemudian, individu tersebut akan menggantikan individu terburuk dari populasi orangtua.

Di antara kedua strategi penggantian ekstrem tersebut, ada beberapa skema lain yang berupa penggantian sejumlah k individu ($1 < k < n$, dengan n ukuran populasi). Elitisme bercirikan pemilihan individu terbaik dari orangtua dan anak. Individu tersebut dijaga dengan cara dikopi dan disimpan karena mungkin saja nilai *fitness*-nya menurun saat mengalami pindah silang.

h. Kriteria berhenti

Ada dua jenis kriteria berhenti yang bisa digunakan dalam AG:

- Prosedur statis: pada prosedur ini akhir pencarian bisa diketahui sebelumnya. Kriteria yang digunakan bisa berupa jumlah iterasi

(generasi) yang tetap, batas sumberdaya computer, atau jumlah maksimum evaluasi fungsi objektif.

- Prosedur adaptif: pada prosedur ini akhir pencarian tidak bisa diketahui sebelumnya. Kriteria yang digunakan bisa berupa jumlah tertentu iterasi tanpa adanya perbaikan kualitas solusi, saat solusi optimal atau yang memuaskan ditemukan, atau saat nilai *diversity* populasi berada di bawah ambang nilai tertentu yang artinya populasi mengalami stagnasi dan tidak ada gunanya melanjutkan eksekusi.



BAB 3

PENGUMPULAN DATA DAN MODEL MATEMATIS

3.1 Pengumpulan Data

Data yang digunakan dalam penelitian ini ada yang berupa data sekunder dari penelitian sebelumnya dan ada pula yang berupa data *dummy*. Algoritma penjadwalan yang akan dibuat dirancang untuk bisa menampung data yang fleksibel atau bisa berubah dari waktu ke waktu. Hal ini didasari bahwa pada dunia nyata, data jumlah pasien misalnya bisa berubah. Adapun data yang dibutuhkan antara lain:

- a. Data jumlah kamar operasi yang dimiliki rumah sakit. Kamar operasi diasumsikan bisa menampung segala jenis operasi yang akan dijadwalkan. Kasus penjadwalan ini menggunakan 6 kamar operasi.
- b. Data rentang hari penjadwalan. Rentang hari penjadwalan yang digunakan dalam kasus ini adalah 5 hari dari Senin sampai Jumat. Satu hari akan dibagi lagi menjadi pagi dan sore yang disebut setengah-harian (*half-day*). Lima hari kerja dengan demikian akan menjadi 10 setengah-harian. Setengah-harian 1 berarti Senin pagi, 2 berarti Senin sore, 3 berarti Selasa pagi, 4 berarti Selasa sore, dan seterusnya.
- c. Data jumlah dokter bedah serta data ketersediaan masing-masing dokter. Data ketersediaan dokter selama rentang waktu penjadwalan berisi pagi dan/atau sore hari apa saja dokter tersebut tersedia untuk melakukan operasi. Dokter berjumlah 20 orang sedangkan pasien yang akan dijadwalkan berjumlah 40 orang.
- d. Data pasien bedah yang akan dijadwalkan berupa nomor pasien, dokter mana yang menanganinya, serta perkiraan durasi operasi masing-masing pasien. Data durasi operasi pasien akan diambil dari penelitian sebelumnya oleh Tuti Arsyida (2010). Pada penelitian tersebut, data historis durasi masing-masing jenis operasi diambil dari Rumah Sakit Daerah Pasar Rebo Jakarta. Sistem penjadwalan yang digunakan dalam penelitian ini menganggap durasi setiap operasi akan merupakan kelipatan 15 menit. Durasi-durasi data historis tersebut belum tentu merupakan kelipatan 15 menit. Jika durasi yang diambil bukan kelipatan 15 menit, maka akan dibulatkan ke atas. Durasi masing-

masing pasien akan diambil secara acak dari data historis tersebut. Durasi operasi kemudian akan dikonversi menjadi slot durasi dengan satu slot durasi sama dengan 15 menit.

- e. Data tingkat keterdesakan pasien tersebut. Pasien operasi elektif biasanya diputuskan harus menjalani operasi sesudah dia didiagnosis mengalami penyakit yang membutuhkan operasi. Biasanya operasi tidak segera dilaksanakan karena pasien operasi elektif biasanya harus masuk daftar tunggu yang cukup panjang sebelum akhirnya dijadwalkan. Saat akan dijadwalkan, beberapa pasien bisa menjadi lebih terdesak karena kondisi pasien semakin parah atau hal lain. Keterdesakan berarti pada hari (setengah-harian) mana dalam rentang waktu penjadwalan si pasien diusahakan untuk mendapatkan jadwal operasi. Jika seorang pasien diusahakan untuk dioperasi maksimal pada hari kedua rentang penjadwalan, maka keterdesakannya bernilai 4 yang berarti bahwa pasien tersebut diusahakan dioperasi pada setengah-harian 1, 2, 3 atau 4. Jika pasien bisa dioperasi pada hari apa saja dalam rentang penjadwalan, maka nilai keterdesakannya sama dengan nilai setengah-harian pada hari akhir rentang penjadwalan (dalam kasus ini bernilai 10).

Data yang akan digunakan dapat dilihat pada Lampiran 1.

3.2 Model Matematis Penjadwalan Kamar Bedah

Untuk memandu penyusunan algoritma penjadwalan kamar bedah, terlebih dahulu perlu dibangun model matematisnya. Model ini berupa model program integer biner.

Asumsi yang digunakan dalam model penjadwalan kamar bedah ini adalah sebagai berikut:

- Strategi penjadwalan yang digunakan adalah *open scheduling (first come/first served)*;
- Hanya mempertimbangkan operasi elektif;
- Operasi bersifat *nonpreemptive* (jika sudah dimulai, maka tidak bisa diinterupsi sampai selesai);
- Kendala sumber daya yang dipertimbangkan hanyalah dokter bedah, sumber daya lain yang dibutuhkan dalam operasi dianggap tak terbatas;

- Durasi setiap operasi bisa diperkirakan dan dianggap deterministik (kelipatan 15 menit);
- Jika seorang dokter harus melakukan operasi lebih dari satu pasien dalam satu hari, maka selang waktu istirahat dokter tersebut antar-operasi minimal 1 jam;
- Semua ruang operasi dianggap bisa menampung semua jenis operasi;
- Semua ruang operasi selalu terbuka pukul 07.00-17.00 setiap hari dalam periode penjadwalan, yang dibagi lagi menjadi setengah-harian pagi (07.00-12.00) dan setengah-harian sore (12.00-17.00).

Indeks:

i = pasien bedah 1, 2, ..., I

j = ruang operasi 1, 2, ..., J

k = hari 1, 2, ..., K

l = slot waktu 1, 2, ..., T (15 menit per slot waktu, ruang operasi diasumsikan buka pukul 07.00-17.00 $\rightarrow T = 40$)

c = dokter bedah 1, 2, ..., C

Parameter:

p_i = panjang waktu slot operasi pasien i

$O(c)$ = himpunan pasien dokter c

$S(k)$ = himpunan pasien yang seharusnya dioperasi pada hari k

$R^c \in \{0,1\}^{C \times 2K}$ = matriks biner ketersediaan dokter c pada hari k (pagi & sore)

$R_{c,2k-1} = \begin{cases} 1, & \text{dokter } c \text{ dapat bekerja pada pagi hari ke } k \\ 0, & \text{jika tidak} \end{cases}$

$R_{c,2k} = \begin{cases} 1, & \text{dokter } c \text{ dapat bekerja pada sore hari ke } k \\ 0, & \text{jika tidak} \end{cases}$

$D_{i,k}$ = matriks tingkat keterdesakan operasi pasien i

$D_{i,k} = \begin{cases} 1, & \text{pasien } i \text{ seharusnya dioperasi pada hari } k \\ 0, & \text{jika tidak} \end{cases}$

Variabel keputusan:

$x_{ijkl} = \begin{cases} 1, & \text{operasi pasien } i \text{ dilakukan di ruang } j \text{ pada hari ke } k \text{ slot waktu } l \\ 0, & \text{jika tidak} \end{cases}$

Fungsi Tujuan:

Fungsi tujuan model ini adalah untuk meminimumkan jumlah pinalti akibat terunduranya waktu operasi pasien dari tingkat keterdesakannya pada jadwal yang dihasilkan.

$$\begin{aligned} \text{Min } Z = & 1000 \cdot \sum_{i \in S(k)} \sum_j \sum_l x_{ij(k+1)l} + 2000 \cdot \sum_{i \in S(k)} \sum_j \sum_l x_{ij(k+2)l} + \\ & 3000 \cdot \sum_{i \in S(k)} \sum_j \sum_l x_{ij(k+3)l} + 4000 \cdot \sum_{i \in S(k)} \sum_j \sum_l x_{ij(k+4)l} \end{aligned} \quad (3.1)$$

Persamaan di atas berarti bahwa jika seorang pasien i dijadwalkan mundur satu hari dari tingkat keterdesakannya, maka diberikan pinalti 1000. Jika mundur dua hari, diberikan pinalti 2000 dst. Jika jadwal operasi pasien memenuhi tingkat keterdesakannya, solusi yang dihasilkan mendapat pinalti 0.

Kendala:

- Kendala hanya ada paling banyak satu operasi yang berlangsung di ruang j pada hari k pada slot waktu l .

$$\sum_i x_{ijkl} \leq 1, \quad \forall l, \forall j, \forall k \quad (3.2)$$

- Jumlah slot waktu yang dipakai untuk operasi pasien i yang dilakukan di ruang j pada hari k sama dengan panjang durasi operasi tersebut.

$$\sum_l x_{ijkl} \leq p_i, \quad \forall i, \forall j, \forall k \quad (3.3)$$

- Kendala pilihan hubungan antar-slot:

1) Jika operasi pasien i dilaksanakan pada slot l di ruang j pada hari k ($x_{ijkl} = 1$), maka satu slot setelahnya, $x_{ijk(l+1)}$, hanya bisa bernilai lebih kecil atau sama dengan nilai slot tersebut. Jika bernilai sama, maka operasi masih berlanjut pada slot $l+1$. Jika bernilai lebih kecil, maka artinya operasi sudah selesai pada slot $l+1$.

2) Jika operasi pasien i tidak dilaksanakan pada slot l di ruang j pada hari k ($x_{ijkl} = 0$), maka satu slot setelahnya $x_{ijk(l+1)}$, hanya bisa bernilai lebih besar atau sama dengan nilai slot tersebut. Jika bernilai sama, maka operasi masih juga tidak dilaksanakan pada slot $l+1$. Jika bernilai lebih besar, maka artinya operasi akan dilaksanakan pada slot $l+1$.

$$-M \cdot y_{ijkl} + x_{ijk(l+1)} - x_{ijkl} \leq 0 \quad (3.4)$$

$$-M \cdot (1 - y_{ijkl}) - x_{ijk(l+1)} + x_{ijkl} \leq 0 \quad (3.5)$$

dengan

$$y_{ijkl} = \begin{cases} 0, & \text{jika operasi pasien } i \text{ dilaksanakan di ruang } j \text{ pada hari } k \text{ slot } l \\ 1, & \text{jika tidak} \end{cases}$$

M merupakan bilangan yang besar sekali.

- Suatu operasi pasien i yang merupakan tanggung jawab dokter c hanya bisa dilangsungkan pada pagi (dan/atau sore) hari k di ruang manapun pada slot manapun jika dokter tersebut bisa bekerja pada pagi (dan/atau sore) hari tersebut. Operasi yang dimulai pada pagi hari k oleh dokter yang hanya bisa bekerja pada pagi hari tersebut harus selesai pada slot pagi (tidak bisa memasuki slot sore).

$$\sum_{i \in O(c)} \sum_j \sum_{l=1}^{20} x_{ijkl} \leq \sum_{i \in O(c)} p_i \cdot R_{c,2k-1}, \quad \forall k, \forall c \rightarrow \text{hanya bisa pagi} \quad (3.6)$$

$$\sum_{i \in O(c)} \sum_j \sum_{l=21}^{40} x_{ijkl} \leq \sum_{i \in O(c)} p_i \cdot R_{c,2k}, \quad \forall k, \forall c \rightarrow \text{hanya bisa sore} \quad (3.7)$$

$$\sum_{i \in O(c)} \sum_j \sum_l x_{ijkl} \leq \sum_{i \in O(c)} p_i \cdot R_{c,k}, \quad \forall k, \forall c \rightarrow \text{bisa pagi dan sore} \quad (3.8)$$

- Operasi pasien i bisa dijadwalkan pada rentang keterdesakannya atau tidak (tapi terkena pinalti seperti yang terlihat pada fungsi tujuan).

$$\sum_j \sum_l x_{ijkl} \leq p_i \cdot D_{i,k}, \quad \forall k, \forall i \in S(k) \quad (3.9)$$

BAB 4

PENGOLAHAN DATA DAN ANALISIS

4.1 Penyusunan Algoritma Genetika

Model penjadwalan kamar operasi ini diselesaikan dengan menggunakan algoritma genetika. Program algoritma genetika akan dibuat menggunakan *software* MATLAB versi 7.9.0(2009b). MATLAB merupakan bahasa komputasi yang sangat baik dalam mengolah data dalam berbentuk vektor dan matriks. Data akan dibuat dalam bentuk matriks-matriks sehingga bisa diproses dengan mudah.

4.1.1 Langkah-langkah Penyusunan Algoritma Genetika

a. Menentukan parameter/variabel yang digunakan

- Ukuran populasi
Algoritma genetika bekerja dengan membentuk suatu populasi bukan solusi tunggal. Oleh karena itu, perlu ditetapkan ukuran populasi permasalahan yang dihadapi. Talbi (2010) menyatakan bahwa ukuran populasi yang sering digunakan berada dalam rentang 20 sampai 100. Beberapa ukuran populasi yang berbeda akan diuji untuk melihat ukuran populasi mana yang akan memberikan kualitas solusi terbaik. Namun, untuk input awal akan digunakan jumlah populasi 40.
- Probabilitas pindah silang
Probabilitas pindah silang merupakan parameter penting dalam algoritma genetika karena sebuah kromosom yang mengarah pada kualitas solusi yang baik bisa didapatkan dari proses pindah silang.
- Probabilitas mutasi
Biasanya probabilitas mutasi yang digunakan adalah $1/n$ dengan n merupakan jumlah gen dalam kromosom.
- Jumlah iterasi maksimum

b. Penentuan representasi solusi

Setiap individu kandidat solusi dimodelkan dengan sebuah pasangan kromosom δ dan σ yang tidak saling terkait.

$$S = \begin{pmatrix} \delta \\ \sigma \end{pmatrix} = \begin{pmatrix} (i_1, d_1) & \dots & (i_l, d_l) \\ j_1 & \dots & j_l \end{pmatrix}$$

Elemen (i_m, d_m) dari kromosom δ menunjukkan operasi pasien i_m dijadwalkan untuk waktu setengah-harian d_m (pagi atau sore) dan elemen j_m dari kromosom σ menunjukkan bahwa ruang operasinya adalah j_m . Skema pengkodean pasien yang digunakan adalah skema permutasi. Pada skema pengkodean berupa permutasi walaupun waktu setengah-harian muncul tidak berurutan pada kromosom, pasien dengan waktu setengah-harian lebih kecil pasti akan dioperasi lebih awal pada jadwal yang sebenarnya. Pada contoh representasi di bawah ini, pasien nomor 40 dijadwalkan pada setengah-harian 10 tapi muncul lebih dahulu pada kromosom daripada pasien 20 yang dijadwalkan pada setengah-harian 6. Pada jadwal asli pasien 20 tetap akan dioperasi lebih dahulu daripada pasien 40 karena setengah-harian 6 lebih awal daripada setengah-harian 10.

Pasien	40	20	...	3
Setengah-harian	10	6	...	1
Ruang	6	1	...	2

Gambar 4.1 Kromosom representasi solusi

Satu individu akan berupa matriks berukuran 3×40 dengan pasien pada baris pertama, setengah-harian pada baris kedua, dan ruangan operasi pada baris terakhir.

c. Menginisialisasi individu-individu populasi awal

Individu dibentuk dengan menempatkan 40 pasien secara acak pada baris pertama pada kromosom δ . Waktu setengah-harian operasi d_m yang terkait dengan pasien i_m , diinisiasi dengan memilih secara acak waktu dari ketersediaan dokter yang menangani pasien yang bersesuaian dengan mempertimbangkan keterdesakannya. Ruang operasi j_m pasien pada baris ketiga matriks individu dipilih secara acak dari jumlah ruang yang tersedia. Dibuatlah individu sejumlah ukuran populasi yang diinginkan.

d. Menentukan fungsi evaluasi

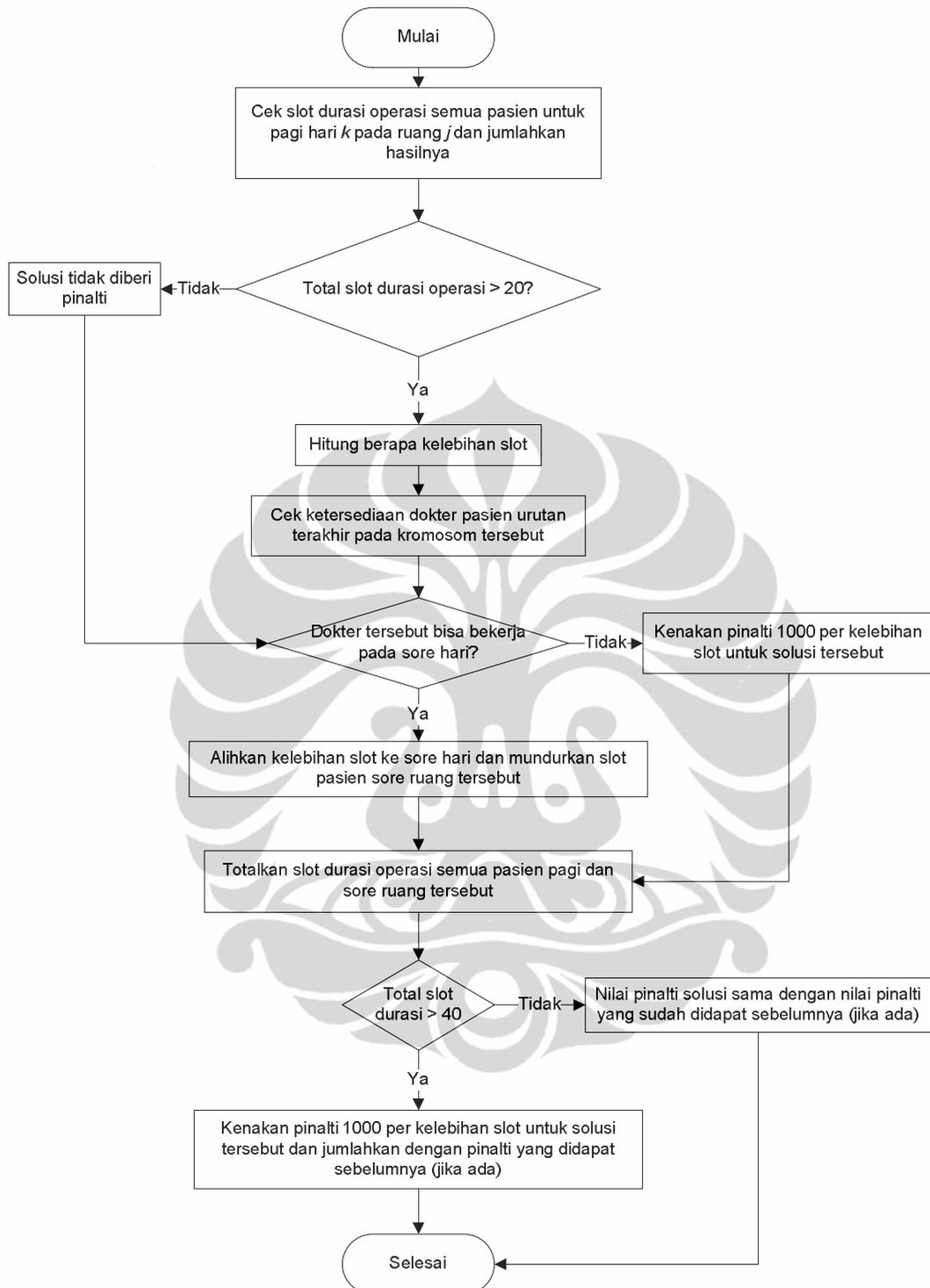
Evaluasi individu S dilakukan dengan melihat kemampuan individu tersebut dalam memenuhi kendala kapasitas ruangan, ketersediaan dokter bedah, serta tingkat keterdesakan waktu operasi.

Kendala kapasitas ruangan dan ketersediaan dokter dievaluasi secara bersamaan. Pertama, kita akan mengevaluasi suatu ruang operasi pada setengah-harian pagi. Jika ternyata durasi total pasien yang akan dioperasi pada pagi hari di ruang tersebut lebih dari 20 slot, maka pasien akan diurutkan berdasarkan posisinya pada kromosom δ . Ketersediaan dokter pasien urutan terakhir akan dicek apakah bisa melakukan operasi pada sore hari yang sama. Jika dokter tersebut bisa, maka kelebihan slot pagi akan dialihkan ke awal slot sore dan waktu mulai pasien sore akan dimundurkan sejumlah kelebihan slot tersebut. Jika tidak bisa, maka individu tersebut diberi pinalti 1000 per kelebihan slot. Hal ini karena ada kendala bahwa dokter yang hanya bisa melakukan operasi pada pagi hari k harus menyelesaikan operasinya pada pagi hari dan tidak bisa melanjutkan operasi pada sore hari k . Selanjutnya kapasitas harian ruang operasi akan dievaluasi. Durasi semua pasien pada yang dijadwalkan di ruang tersebut pada hari k akan didaftarkan dan slot durasinya dijumlahkan. Jika ternyata slot durasi yang terpakai lebih besar daripada 40 slot maka individu solusi akan dikenakan pinalti 1000 per kelebihan slot.

Evaluasi selanjutnya akan menilai seberapa baik individu solusi memenuhi tingkat keterdesakan waktu operasi. Hari operasi masing-masing pasien akan dicek apakah lebih cepat atau sesuai dengan hari keterdesakannya. Jika tidak sesuai individu solusi akan diberikan biaya 1000 per pasien per satu hari kemunduran dari hari keterdesakan. Jika mundur dua hari, maka pinalti yang diberikan adalah 2000, tiga hari pinalti 3000, dan empat hari pinalti 4000. Total pinalti dari evaluasi ini kemudian dijumlahkan dengan total pinalti dari kendala kapasitas ruangan dan ketersediaan dokter.

e. Strategi seleksi

Strategi seleksi orangtua yang digunakan adalah *rank-selection*. Setelah dilakukan evaluasi individu populasi awal, individu diurutkan sesuai nilai *fitness*. Masing-masing individu kemudian diberikan nilai probabilitas seleksi yang proporsional terhadap urutan *fitness*-nya (*rank*-nya). Orangtua kemudian dipilih secara acak untuk mengalami pindah silang sesuai dengan nilai probabilitas seleksi yang sudah ditentukan.



Gambar 4.2 Alur evaluasi kendala kapasitas ruangan dan ketersediaan dokter

f. Prosedur pindah silang

Operator pindah silang menghasilkan dua individu baru S^D dan S^S (disebut *daughter* dan *son*) dari dua individu orangtua S^M dan S^F (disebut *mother* dan *father*) yang sudah dipilih sebelumnya.

Pertama, bangkitkan dua bilangan acak yang berbeda bernilai antara 1 sampai jumlah pasien yaitu q_1 dan q_2 , sehingga $1 \leq q_1$ dan $q_2 \leq I$ dengan $I =$ jumlah pasien. Dengan mempertimbangkan *daughter* $S^D = (\delta^D, \sigma^D)$ terlebih dahulu, kita melanjut dengan dua cara: sejumlah q_1 elemen δ^D diambil dari *mother* S^M :

$$(i_m^D, d_m^D) := (i_m^M, d_m^M) \quad \forall m = 1, \dots, q_1$$

Sisa elemen ke $q_1 + 1$ sampai ke I diturunkan dari *father*, tetapi dengan memastikan nomor pasien yang sudah diambil dari *mother* tidak dipertimbangkan lagi:

$$(i_m^D, d_m^D) := (i, d_n^F) \quad \forall m = q_1 + 1, \dots, I$$

Dengan n merupakan indeks terendah sehingga

$$i_n^F \notin \{i_1^D, \dots, i_{m-1}^D\}$$

Terkait dengan ruang operasi j_i dari kromosom σ , pindah silang terjadi dengan menurunkan sejumlah q_2 elemen dari σ^D dengan j bersesuaian juga diturunkan dari *mother*:

$$\sigma^D(j_m) := \sigma^M(j_m) \quad \forall m = 1, \dots, q_2$$

yang berarti ruang operasi j_m^D dari pasien i_m^D sama dengan ruang operasi untuk i_m^D yang ada pada *mother*; sisanya datang dari *father*. Demikian juga yang terjadi pada *son* S^S , tetapi dengan membalikkan *father* dan *mother*.

Ini adalah contoh untuk satu hari (dua setengah-harian), tiga ruang operasi, dan empat pasien bedah. Dengan memisalkan $q_1 = 2$ dan $q_2 = 3$, kita memperoleh dari orangtua

$$S^M = \begin{pmatrix} \delta^M \\ \sigma^M \end{pmatrix} = \begin{pmatrix} (2,2) & (3,1) & (1,2) & (4,2) \\ 2 & 1 & 3 & 2 \end{pmatrix},$$

$$S^F = \begin{pmatrix} \delta^F \\ \sigma^F \end{pmatrix} = \begin{pmatrix} (3,2) & (4,1) & (1,1) & (2,1) \\ 1 & 1 & 1 & 3 \end{pmatrix},$$

anak sebagai berikut:

$$S^D = \begin{pmatrix} \delta^D \\ \sigma^D \end{pmatrix} = \begin{pmatrix} (2,2) & (3,1) & (4,1) & (1,1) \\ 2 & 1 & 2 & 1 \end{pmatrix},$$

$$S^S = \begin{pmatrix} \delta^S \\ \sigma^S \end{pmatrix} = \begin{pmatrix} (3,2) & (4,1) & (2,2) & (1,2) \\ 1 & 1 & 3 & 3 \end{pmatrix}.$$

Karena $q_1 = 2$, operasi pasien 2 dan 3 pada indeks 1 dan 2 *daughter* S^D diturunkan dari *mother* S^M , sedangkan operasi pasien 4 dan 1 dengan indeks 3 dan 4 pada *daughter* diambil dari *father* S^F sesuai dengan urutan indeks dari yang terendah ke yang tertinggi (pasien 4 memiliki indeks 2 pada *father* sedangkan pasien 1

memiliki indeks 3 pada *father*). Yang terjadi pada *son* juga mirip. Operasi pasien 3 dan 4 pada indeks 1 dan 2 *son* S^S diturunkan dari *father* S^F , sedangkan sisanya pasien 1 dan 2 untuk mengisi indeks 3 dan 4 diambil dari *mother* indeks 3 dan 1. Setengah-harian pelaksanaan operasi dianggap tidak lepas dari pasien.

Untuk ruang operasi pada *daughter*, karena $q_2 = 3$, maka ruang operasi untuk pasien pada indeks 1 – 3 pada *daughter* diambil dari ruang operasi yang bersesuaian dengan nomor operasi tersebut pada *mother*. Pasien 2, 3, dan 4 yang menempati indeks 1, 2, 3 pada *daughter*, masing-masing menempati ruangan 2, 1, dan 2 pada *mother* (dengan mengacu pada indeks 1, 2, 4). Ruangan untuk pasien indeks terakhir akan menyesuaikan dengan ruang mana pasien tersebut dioperasi pada *father*. Pasien 1 pada *father* dioperasi di ruang 1 (lihat indeks 3 *father*) sehingga nilai ini menjadi ruang untuk operasi pasien 1 pada indeks 4 *daughter*.

Hal yang sama berlaku pada *son* dengan membalikkan *mother* dan *father*.

g. Prosedur mutasi

Operator mutasi digunakan untuk setiap individu anak $S = (\delta, \sigma)$ yang baru. Suatu angka acak integer n dipilih [1, jumlah pasien]. Daftar δ dimodifikasi dalam dua langkah. Pertama, kita menukarkan (*switch*) operasi i_n dan i_{n+1} dengan probabilitas p_{mut} . Perhatikan bahwa operasi yang ditukar tetap mempertahankan waktu setengah-harian dan ruang operasinya. Kedua, untuk $n = 1, \dots, \text{jumlah pasien}$, kita memilih mengubah waktu setengah-harian operasi d_n dari antara waktu ketersediaan dokter bedah yang memenuhi kendala keterdesakan, juga dengan probabilitas p_{mut} . Selanjutnya, operator ini akan bekerja pada daftar ruangan σ dengan menggantikan sebuah ruang operasi j_n dengan yang lain secara acak dengan probabilitas yang sama p_{mut} .

h. Penggantian populasi

Penggantian populasi dilakukan dengan mengurutkan semua individu orangtua dan anak berdasarkan nilai pinalti dari yang terkecil sampai yang terbesar. Individu sejumlah ukuran populasi yang memiliki nilai *fitness* terbaik akan dipertahankan, sedangkan sisanya akan dibuang. Individu-individu yang dipertahankan akan menjadi populasi orangtua untuk generasi selanjutnya.

i. Kriteria Berhenti

Kriteria berhenti yang digunakan adalah jumlah iterasi maksimal.

4.1.2 Verifikasi dan Validasi Program

Sebelum menyelesaikan permasalahan penjadwalan kamar operasi rumah sakit, perlu dilakukan verifikasi dan validasi program terlebih dahulu. Tahap verifikasi merupakan tahap melihat kesesuaian antara model program yang didapat dengan model konseptual yang kita inginkan. Parameter model program dikatakan telah terverifikasi apabila telah berjalan sesuai dengan konseptual model, dimana ada perubahan total pinalti. Ketika program dijalankan dengan mengubah-ubah parameter, maka akan didapatkan output yang berbeda-beda. Pada keadaan inilah program disebut terverifikasi. Tabel verifikasi bisa dilihat di Lampiran 4.

Selanjutnya dilakukan validasi program. Validasi dilakukan dengan memasukkan data untuk ukuran kasus yang lebih kecil. Tujuannya adalah untuk mengevaluasi apakah program berjalan sesuai dengan fungsinya. Hasil eksekusi program dibandingkan dengan perhitungan manual untuk memperoleh validasi.

Data kasus yang lebih kecil yang digunakan ada empat, yaitu data ruang operasi, data rentang waktu penjadwalan, data pasien, dan data dokter. Ada dua ruang operasi, rentang perencanaan adalah dua hari (empat pagi dan sore), jumlah pasien lima serta dokter ada dua orang.

Tabel 4.1 Data pasien untuk verifikasi dan validasi program

Pasien	1	2	3	4	5
Dokter	1	1	2	2	2
Durasi operasi	90	60	45	75	60
Slot durasi operasi	6	4	3	5	4
Keterdesakan	2	4	4	4	4

Tabel 4.2 Data ketersediaan dokter bedah

Dokter	Hari 1		Hari 2	
	1 (Pagi)	2 (Sore)	1 (Pagi)	2 (Sore)
1	1	0	1	0
2	1	1	1	1

Tabel 4.3 Parameter yang digunakan dalam validasi

Parameter	Nilai
Ukuran Populasi	2
Maksimum iterasi	1
Probabilitas pindah silang	1
Probabilitas mutasi	1

4.1.2.1 Hasil Eksekusi Program

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

Tabel 4.4 Kromosom terbaik dari proses verifikasi dan validasi

Pasien	3	5	4	1	2
Setengah-harian	3	4	3	1	1
Ruang	2	1	2	1	2

4.1.2.2 Hasil Perhitungan Manual

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

Iterasi yang dilakukan adalah sebagai berikut:

- a. Pembentukan solusi awal

Tabel 4.5 Populasi awal (dari program)

Individu 1	Pasien	3	5	4	1	2
	Setengah-harian	3	4	3	1	1
	Ruang	2	1	2	1	2
Individu 2	Pasien	3	5	1	2	4
	Setengah-harian	3	4	1	1	4
	Ruang	2	2	1	1	1

- b. Evaluasi Individu

Evaluasi dilakukan pertama kali dengan mengecek untuk masing-masing ruangan apakah total slot waktu operasi yang dilakukan pada pagi hari di situ melebihi kapasitas (> 20 slot). Jika tidak, maka individu tidak diberikan pinalti. Jika ya, maka individu dikenakan pinalti 1000 per kelebihan slot. Karena kasus yang digunakan kecil, kemungkinan besar solusi tidak melanggar kendala kapasitas ruangan. Kemudian kendala keterdesakan dievaluasi. Individu dengan pasien yang dijadwal lebih awal daripada keterdesakannya, tidak mendapatkan

pinalti, sedangkan yang mendapat jadwal di luar keterdesakannya mendapat pinalti 1000 per kemunduran.

Tabel 4.6 Evaluasi individu pertama populasi awal

Individu 1

pasien	3	5	4	1	2	
setengah-harian	3	4	3	1	1	
ruang	2	1	2	1	2	
keterdesakan	4	4	4	2	4	
slot durasi	3	4	5	6	4	
total slot	d=1	d=2	d=3	d=4	hari1	hari2
ruang 1	6	0	0	4	6	4
ruang 2	4	0	8	0	4	8
pinalti slot	0					
pinalti keterdesakan	0					

Individu 2

pasien	3	5	1	2	4	
setengah-harian	3	4	1	1	4	
ruang	2	2	1	1	1	
keterdesakan	4	4	2	4	4	
slot durasi	3	4	6	4	5	
total slot	d=1	d=2	d=3	d=4	hari1	hari2
ruang 1	10	0	0	5	10	5
ruang 2	0	0	3	4	0	7
pinalti slot	0					
pinalti keterdesakan	0					

c. Pemilihan orangtua

Karena hanya ada dua orangtua dan tidak mungkin melakukan pindah silang terhadap diri sendiri, kedua individu populasi awal langsung terpilih sebagai orangtua untuk mengalami pindah silang.

d. Pindah silang

Kedua titik silang dipilih secara acak dari interval $[1, \text{jumlah pasien}-1]$. Hasil dari program $q_1 = 1, q_2 = 2$. Sesuai dengan prosedur pindah silang yang sebelumnya sudah dijelaskan pada bagian Penyusunan Algoritma, terjadilah proses pindah silang di bawah ini. Warna pada individu anak yang sama dengan warna pada individu orangtua menunjukkan bahwa elemen tersebut diturunkan dari segemen orangtua tersebut.

Individu orangtua 1

3	5	4	1	2
3	4	3	1	1
2	1	2	1	2

Individu orangtua 2

3	5	1	2	4
3	4	1	1	4
2	2	1	1	1

Individu anak 1

3	5	1	2	4
3	4	1	1	4
2	1	1	1	1

Individu anak 2

3	5	4	1	2
3	4	3	1	1
2	2	2	1	2

Gambar 4.3 Validasi pindah silang secara manual

e. Mutasi

Karena probabilitas mutasi yang dipakai adalah 1, maka pasti kedua anak akan mengalami mutasi. Untuk anak pertama, titik mutasi yang didapat dari program adalah 2. Jadi hasil mutasi anak pertama adalah sebagai berikut:

3	1	5	2	4
3	1	4	1	4
2	2	1	1	1

Untuk anak kedua, titik mutasi yang didapat dari program adalah 3. Jadi hasil mutasi anak kedua adalah sebagai berikut:

3	5	4	2	1
3	4	3	3	1
2	2	2	2	1

f. Evaluasi anak

Anak dari hasil pindah silang yang sudah dimutasi kemudian dievaluasi sebagai berikut:

Tabel 4.7 Evaluasi individu anak

pasien	3	1	5	2	4	
setengah-harian	3	1	4	1	4	
ruang	2	2	1	1	1	
keterdesakan	4	2	4	4	4	
slot durasi	3	6	4	4	5	
total slot	d=1	d=2	d=3	d=4	hari1	hari2
ruang 1	4	0	0	9	4	9
ruang 2	6	0	3	0	6	3
pinalti slot	0					
pinalti keterdesakan	0					
pasien	3	5	4	2	1	
setengah-harian	3	4	3	3	1	
ruang	2	2	2	2	1	
keterdesakan	4	4	4	4	2	
slot durasi	3	4	5	4	6	
total slot	d=1	d=2	d=3	d=4	hari1	hari2
ruang 1	6	0	0	0	6	0
ruang 2	0	12	0	4	12	4
pinalti slot	0					
pinalti keterdesakan	0					

g. Pemilihan kromosom terbaik

Karena semua individu memiliki kualitas solusi yang sama, maka yang dipilih sebagai solusi akhir terbaik adalah individu orangtua pertama. Hal ini dikarenakan pengurutan berdasarkan nilai *fitness* (pinalti) akan mengurutkan berdasarkan individu mana yang lebih dahulu dibentuk karena nilai *fitness* semua individu sama.

4.1.3 Input Data

Data mentah dari masalah penjadwalan kamar operasi ini akan berupa matriks data pasien yaitu nomor pasien, dokter yang menangani masing-masing pasien, durasi operasi, slot durasi operasi, beserta keterdesakannya yang bisa dilihat pada Lampiran 1.

4.2 Hasil Pengolahan Data

Setelah memasukkan data dan menjalankan algoritma yang telah dibuat ke dalam program, maka keluaran yang dihasilkan oleh program adalah jadwal penggunaan kamar operasi yang memiliki nilai pinalti terkecil sehingga dapat dianggap sebagai solusi yang layak.

Data yang dijadikan input mencakup kendala-kendala yang harus dipertimbangkan dalam penjadwalan operasi. Setelah program dijalankan dengan 10000 iterasi untuk kasus 6 kamar operasi, dihasilkan salah satu jadwal terbaik dengan pinalti 5000 seperti yang bisa dilihat pada Lampiran 3. Waktu komputasi yang digunakan adalah 140.1 detik.

4.3 Analisis

Permasalahan penjadwalan kamar operasi merupakan permasalahan yang cukup kompleks karena terdapat banyak aspek yang harus dipertimbangkan dalam menghasilkan jadwal operasi bedah yang baik. Tingkat kerumitan penyelesaian masalah penjadwalan kamar operasi tergantung pada karakteristik yang terdapat di masing-masing rumah sakit, misalnya jumlah pasien yang akan dijadwalkan untuk operasi, jumlah dokter bedah, jumlah ruangan operasi yang tersedia, dan rentang waktu penjadwalan operasi.

Usulan penyelesaiannya adalah dengan menggunakan algoritma metaheuristik 'Algoritma Genetika'. Dalam penelitian ini, algoritma tersebut diadopsi ke dalam bahasa pemrograman MATLAB. Tujuannya adalah untuk mengakomodasi kendala waktu ketersediaan dokter bedah, keterdesakan operasi, kapasitas ruang operasi. Analisis terhadap hasil pengolahan data dibagi menjadi tiga bagian, yakni analisis metode, analisis skenario parameter, dan analisis hasil.

4.3.1 Analisis Metode

Prinsip dasar dari suatu algoritma genetika adalah evolusi biologi Darwinian, yaitu individu yang kuat akan bertahan sedangkan individu yang lemah akan musnah. Untuk mendapatkan populasi awal dilakukan permutasi secara acak. Individu yang akan mengalami proses pindah silang dan mutasi juga dipilih secara acak dengan membandingkan bilangan acak yang dihasilkan dengan probabilitas pindah silang atau mutasi. Apabila bilangan acak yang dihasilkan lebih kecil dari probabilitas pindah silang (probabilitas mutasi), maka individu

tersebut akan mengalami proses pindah silang (mutasi). Titik mutasi juga dipilih secara acak. Inilah yang membuat adanya perbedaan urutan jadwal yang dihasilkan pada setiap *run* program. Kombinasi parameter yang sama belum tentu menghasilkan urutan jadwal yang sama pula dan rentang nilai pinalti yang sama pula. Oleh karenanya, skenario parameter merupakan bagian yang terpenting dalam pengujian metode ini. Karena tiap kasus membutuhkan konfigurasi yang unik untuk memperoleh jadwal yang stabil dalam nilai pinalti yang didapatkan dan waktu *running* yang tidak terlalu panjang (kurang dari 10 menit).

Salah satu metode yang dipakai pada algoritma genetika ini adalah penggantian populasi dengan skema *steady state replacement*, yaitu satu atau beberapa individu terburuk dari populasi orangtua akan diganti oleh satu atau beberapa individu anak yang memiliki nilai *fitness* yang lebih baik.

Alasan yang menyebabkan metode *steady-state replacement* dipilih adalah karena:

- Metode *generational replacement* mengharuskan dihasilkannya individu baru sebanyak ukuran populasi awal yang mengakibatkan populasi untuk generasi selanjutnya bisa dibuat jika jumlah individu baru sama dengan ukuran populasi. Di sisi lain, *steady state replacement* tidak mengharuskan menghasilkan individu baru sebanyak itu, tergantung pada nilai probabilitas pindah silang. Karena menghasilkan individu baru sebanyak ukuran populasi membutuhkan proses evolusi yang lebih panjang, maka waktu komputasi menjadi semakin lambat dan kemungkinan populasi awal untuk dirusak semakin besar.
- *Steady-state replacement* melakukan dua kali evaluasi individu: yang pertama terhadap individu-individu populasi awal dan kedua terhadap individu baru yang dihasilkan sehingga individu yang terpilih benar-benar merupakan individu yang memiliki nilai *fitness* terbaik (pinalti terendah).
- *Steady-state replacement* mengganti satu atau beberapa individu terburuk dalam setiap generasi dengan individu-individu baru yang memiliki kualitas lebih baik. Namun, jika individu baru yang dihasilkan ternyata memiliki kualitas lebih buruk daripada individu lama, maka individu baru tersebut tidak dapat masuk menjadi populasi awal untuk generasi selanjutnya. Hal ini

menyebabkan komposisi individu dalam populasi tidak berubah dari keadaan awal sehingga memastikan bahwa individu-individu dengan kualitas terbaik yang dipertahankan.

- *Fitness* rata-rata metode *generational replacement* lebih buruk dibanding metode *steady-state replacement* krena individu baru selalu menggantikan individu awal tanpa memperhatikan apakah kualitas individu baru lebih baik atau tidak daripada kualitas individu awal yang akan digantikan. Hal ini menyebabkan populasi baru belum tentu merupakan kumpulan individu-individu terbaik dari setiap generasi. Sebaliknya, metode *steady-state replacement* memiliki kumpulan individu terbaik dalam populasi.

Selain dapat menghasilkan solusi yang mendekati optimal, penggunaan algoritma genetika dalam penyelesaian permasalahan penjadwalan ini juga memungkinkan untuk mengakomodasi penambahan atau pengurangan jumlah pasien dan ruangan operasi.

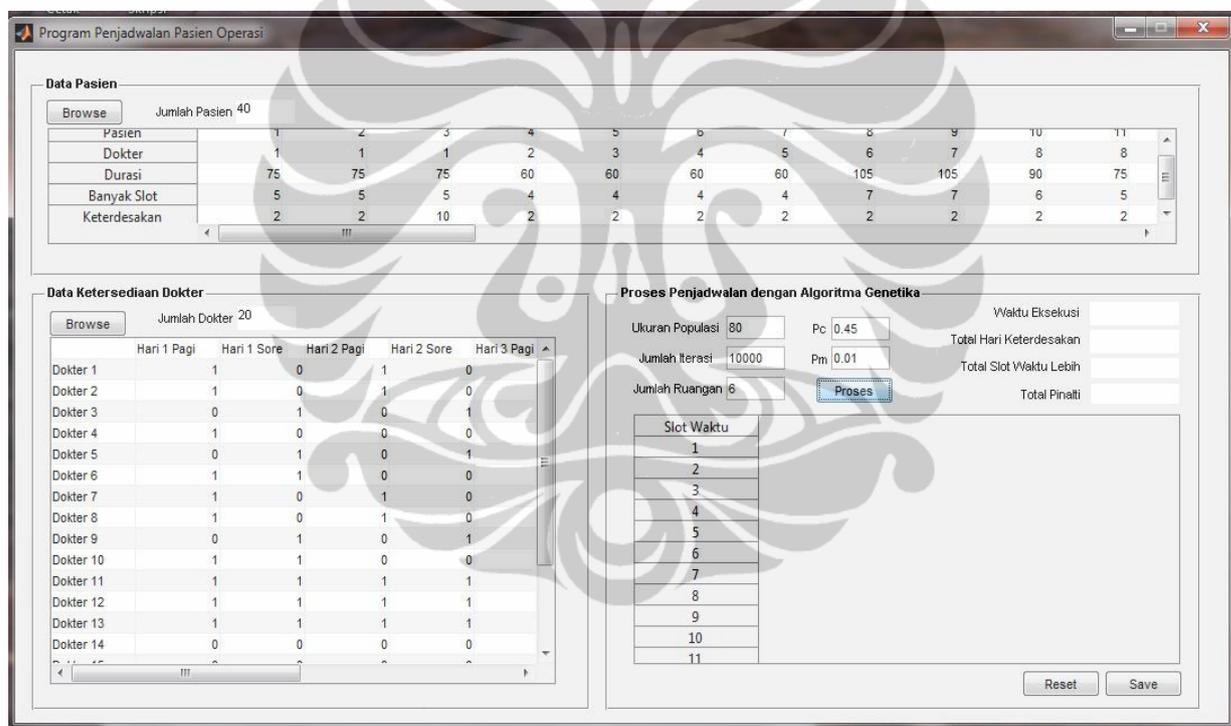
4.3.2 Analisis Program

Hal yang menjadi masalah dalam program adalah saat seorang dokter harus menangani beberapa pasien dengan waktu keterdesakan yang sama. Pada kasus ini, dokter harus mendapat istirahat karena melakukan lebih dari satu operasi dalam sehari bisa menimbulkan kelelahan dan tidak konsentrasi untuk operasi selanjutnya. Kasus seperti ini harus melanggar salah satu kendala agar jadwal bisa tetap dihasilkan.

Contohnya adalah saat tiga orang pasien dengan masing-masing slot durasi operasi 5 harus ditangani secepatnya sementara dokter hanya bisa masuk setengah hari kerja. Dengan asumsi istirahat antaroperasi untuk hari yang sama minimal sebanyak 4 slot (1 jam), maka satu pasien tidak akan bisa dikerjakan karena semuanya akan membutuhkan $5 \times 3 + 4 \times 2 = 23$ slot. Karena dokter hanya bisa bekerja pagi (20 slot) dan operasi bersifat *nonpreemptive*, salah satu pasien tidak akan bisa dioperasi pada hari itu dan harus dijadwalkan di hari lain. Jika memang pasien sangat terdesak, solusinya bisa berupa si dokter mengorbankan waktu sore harinya untuk melakukan operasi. Namun, karena kasus ini adalah kasus elektif dan keterdesakan bukan menjadi sesuatu yang wajib dipenuhi seperti pada kasus

operasi darurat, menjadwalkan pasien terakhir ke hari selanjutnya dokter tersebut tersedia merupakan solusi yang cukup baik.

Algoritma genetika bekerja secara acak. Hal ini terkadang membuat pasien dijadwalkan di hari akhir sementara slot waktu untuk hari awal masih cukup lowong. Ini bisa diatasi dengan mempersempit rentang hari penjadwalan jika jumlah hari tersebut dirasa cukup untuk menampung jumlah pasien yang perlu dijadwalkan. Cara ini memungkinkan ditingkatkannya utilitas kamar operasi dan dengan demikian memusatkan biaya operasional kamar operasi pada hari-hari tertentu. Alternatif lain adalah dengan menutup beberapa kamar tetapi dengan rentang hari penjadwalan yang sama. Biaya operasional kamar operasi yang ditutup sementara dengan demikian bisa dihemat.



Gambar 4.4 Interface program penjadwalan kamar operasi

Durasi operasi juga harus diperhatikan jika ingin menggunakan sistem ini. Karena durasi dianggap deterministik, lewatnya durasi suatu operasi dari yang telah dijadwalkan akan mengganggu jadwal operasi selanjutnya yang akan dilakukan di ruang yang sama. Oleh karena itu, dalam memperkirakan durasi, diusahakan tidak menggunakan kriteria optimis, tetapi juga harus mempertimbangkan kriteria pesimis. Durasi operasi aktual untuk kasus yang

sama perlu direkam untuk digunakan untuk estimasi durasi untuk operasi di masa datang.

Pada kasus ini, kamar operasi juga diasumsikan multifungsi. Artinya kamar tersebut bisa menampung semua jenis operasi. Hal ini terkadang tidak realistis mengingat bahwa investasi untuk fasilitas bedah merupakan hal yang mahal sehingga rumah sakit sering hanya mengkhususkan kamar operasi tertentu untuk jenis operasi tertentu untuk menghemat biaya investasi. Program ini tetap bisa digunakan untuk kasus seperti itu dengan memasukkan jumlah kamar operasi khusus dan data semua pasien yang termasuk dalam jenis operasi yang bisa dikerjakan di kamar tersebut dan dokter yang bisa menangani jenis operasi tersebut.

Bisa disimpulkan bahwa program ini cukup fleksibel dalam mengakomodasi perubahan-perubahan data pasien, jumlah kamar, rentang hari penjadwalan, dan ketersediaan dokter dengan memperhatikan keterbatasan algoritma genetika yang memiliki unsur random.

4.3.3 Analisis Skenario Parameter

Analisis skenario parameter dapat dilakukan dengan menjalankan beberapa percobaan dengan mengganti parameter-parameter yang telah ditentukan sebelumnya untuk diuji. Parameter-parameter yang menghasilkan nilai *fitness* terbaik (pinalti terendah) dalam rentang yang sempit akan digunakan seterusnya dalam program yang dibuat. Parameter-parameter yang diuji terdiri dari ukuran populasi, jumlah iterasi, probabilitas pindah silang, dan probabilitas mutasi. Pengujian dilakukan dengan mengganti parameter satu per satu. Setelah itu baru dikombinasikan dalam pembuatan jadwal.

Setelah menjalankan program yang telah dibuat dengan berbagai kombinasi parameter, didapatlah nilai pinalti terbaik hasil iterasi dengan ukuran populasi 80, probabilitas pindah silang 0.45, probabilitas mutasi 0.01, dan jumlah iterasi 10000.

Dari hasil pengujian kombinasi parameter, dapat disimpulkan beberapa hal berikut ini:

- a. Ukuran populasi yang semakin besar cenderung menghasilkan kualitas solusi yang lebih baik. Hal ini tergambar dari nilai pinalti solusi yang lebih rendah.

Namun, ada ambang batas dimana ukuran populasi yang semakin besar malah akan membuat kualitas solusi menurun. Tidak perlu melakukan *running* program dengan ukuran populasi yang lebih besar daripada ambang batas ini karena hanya akan memperlambat waktu komputasi.

- b. Jumlah iterasi yang makin besar cenderung menghasilkan kualitas solusi yang lebih baik. Hal ini bisa dilihat dari semakin rendahnya nilai pinalti yang didapatkan dari setiap kombinasi setelah melakukan beberapa kali iterasi. Sekalipun demikian, nilai pinalti akan meningkat jika jumlah iterasi melebihi nilai ambang batas tertentu yaitu 10000 iterasi. Waktu komputasi juga semakin meningkat sejalan dengan peningkatan jumlah iterasi. Oleh karena itu, tidak disarankan untuk meningkatkan jumlah iterasi jika dengan nilai jumlah iterasi yang sekarang bisa diperoleh hasil yang memuaskan.
- c. Probabilitas mutasi yang semakin besar cenderung memberikan hasil yang lebih baik. Hal ini bisa dilihat dari semakin kecilnya nilai pinalti yang didapatkan dari setiap kombinasi dengan probabilitas mutasi yang semakin besar setelah melakukan beberapa kali iterasi. Namun, probabilitas mutasi yang lebih besar tidak menjamin hasil yang lebih baik. Probabilitas mutasi tidak mempengaruhi waktu komputasi karena merupakan unsur random.
- d. Probabilitas pindah silang tidak terlalu berpengaruh pada kualitas solusi. Pinalti yang dihasilkan dengan probabilitas pindah silang yang berbeda tidak terlalu berbeda. Probabilitas pindah silang juga tidak terlalu mempengaruhi waktu komputasi karena juga merupakan unsur random.

Setelah menjalankan program dengan berbagai kombinasi parameter dapat diambil kesimpulan bahwa jadwal yang optimal dan memiliki kualitas terbaik hanya dapat diperoleh dengan kombinasi ukuran populasi 80, probabilitas mutasi, probabilitas pindah silang 0.45, dan jumlah iterasi 10000. Program ini juga telah dicoba dijalankan dengan menggunakan perangkat komputer yang memiliki spesifikasi yang berbeda-beda. Hasil yang didapatkan dari program tidak berbeda, bahkan waktu komputasi juga tidak terlalu berbeda sehingga disimpulkan bahwa program ini dapat menghasilkan jadwal optimum asalkan perangkat yang digunakan memenuhi spesifikasi minimum yang dibutuhkan program agar dapat dijalankan.

BAB 5

KESIMPULAN DAN SARAN

5.1 Kesimpulan

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

- Penjadwalan kamar operasi pada rumah sakit menggunakan metode algoritma genetika ini menghasilkan jadwal yang layak secara kapasitas ruangan dan kendala dokter dengan nilai pinalti terbaik. Jadwal yang dihasilkan mendekati kondisi ideal.
- Setelah melakukan analisis parameter bahkan menjalankan program dengan perangkat komputer yang berbeda-beda tetap memberikan hasil yang sama. Hal ini berarti bahwa program yang dibuat dapat digunakan dimanapun asalkan komputer yang digunakan memiliki spesifikasi minimum yang dibutuhkan.
- Program yang dibuat dapat diimplementasikan untuk masalah penjadwalan kamar operasi pada rumah sakit manapun, asalkan memiliki asumsi dan kendala yang sama, dengan mengubah input data sesuai dengan keadaan sebenarnya. Durasi operasi terutama menjadi hal yang harus diperhatikan karena dalam model ini durasi operasi dianggap deterministik.
- Program penjadwalan yang dibuat telah memperhatikan kendala dokter, kapasitas ruangan, dan keterdesakan operasi.

5.2 Saran

Ada beberapa saran yang diberikan kepada rumah sakit yang ingin mengimplementasikan model ini, yaitu:

- Program dikhususkan untuk kamar operasi yang bisa menampung semua jenis operasi yang akan dijadwalkan. Oleh karena itu, untuk kamar operasi yang hanya bisa menampung jenis operasi tertentu harus dijadwalkan secara terpisah dengan pasien diambil dari himpunan pasien dengan jenis operasi yang bisa ditangani di kamar operasi tersebut.

- Rentang hari penjadwalan tidak harus lima hari seperti pada kasus yang digunakan di sini. Rentangnya bisa lebih atau kurang dengan memastikan bahwa data ketersediaan dokter untuk hari-hari itu disediakan.



DAFTAR PUSTAKA

- Arsyida, Tuti. 2010. *Penentuan alokasi ruang operasi dengan metode binary integer nonlinear programming*. Depok: Universitas Indonesia.
- Dexter, F dan R. D. Traub. 2002. How to schedule elective surgical cases into specific operating rooms to maximize the efficiency of use of operating room time. *Anesthesia and Analgesia*, 94, 933-942.
- Macario, A.1995. Where are the costs of perioperative care? *Anesthesiology* 83(6), 2-4.
- Health Care Financial Management Association (HCFMA). 2005. *Achieving operating room efficiency through process integration*. Technical Report.
- Gordon, T. 1988. Surgical unit time review: Resource utilization and management implications. *Journal of Medical System* 12, 169-179.
- Guinet, A. dan S. Chaabane. 2003. Operating theatre planning. *International Journal of Production Economics* 85, 69-81.
- Haupt, Randy L. dan Sue Ellen Haupt. 2004. *Practical Genetic Algorithm*. New Jersey: John Wiley & Sons, Inc.
- Litvak, E dan M. Long. 2000. Cost and quality under managed care: Irreconcilable differences. *The American Journal of Managed Care* 6(3), 305-312.
- Kharraja, S. 2003. *Outils d'aide à la planification et l'ordonnancement des plateau médico-techniques*. Ph.D. thesis, Jean Monnet University.
- Magerlein, J. M. dan J. B. Martin. 1978. Surgical demand scheduling: A review. *Health Services Research* 13, 418-433.
- Roland, B, et al. 2009. *Scheduling an operating theatre under human resource constraints*. *Computers & Industrial Engineering* 58 (2010), 212-220.
- Talbi, El-Ghazali. 2009. *Metaheuristics: From Design to Implementation*. New Jersey: John Wiley & Sons, Inc.
- Weinbroum, A., P. Ekstein, dan T. Ezri. 2003. Efficiency of the operating room suite. *The American Journal of Surgery*, 48(9), 884-891.
- Zäpfel, Günther, et al. 2010. *Metaheuristic Search Concept: A Tutorial with Applications to Production and Logistics*. Berlin: Springer-Verlag.

Lampiran 1: Data Pasien dan Dokter

Data Pasien

Pasien	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				
Dokter	1	1	1	2	3	4	5	6	7	8	8	9	9	9	10	11	12	12	13	13	14	15	15	16	16	17	17	18	18	18	19	19	19	19	19	19	19	20	20	20	20	20	20	
Durasi	75	75	75	60	60	60	105	105	90	75	60	60	60	60	60	105	75	60	60	60	120	75	75	60	60	75	105	60	75	90	75	75	75	60	105	90	60	75	75	60	75	75	60	
Slot durasi	5	5	5	4	4	4	4	7	7	6	5	4	4	4	7	5	4	4	4	4	8	5	5	4	4	5	7	4	5	6	5	5	5	4	7	6	4	5	4	5	4	5	4	
Ketersediaan	2	2	10	2	2	2	2	2	2	2	2	2	2	2	2	2	4	4	4	4	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10

Data Ketersediaan Dokter

Dokter	Hari1		Hari2		Hari3		Hari4		Hari5	
	Pagi	Sore								
1	1	0	1	0	1	0	1	0	1	0
2	1	0	1	0	1	0	1	0	1	0
3	0	1	0	1	0	1	0	1	0	0
4	1	0	0	0	0	0	1	0	0	0
5	0	1	0	1	0	0	0	0	0	0
6	1	1	0	0	0	0	0	0	0	0
7	1	0	1	0	1	0	0	0	0	0
8	1	0	1	0	1	0	1	0	1	0
9	0	1	0	1	0	1	0	1	0	1
10	1	1	0	0	1	1	0	0	1	1
11	1	1	1	1	1	1	1	1	1	1
12	1	1	1	1	1	1	0	0	0	0
13	1	1	1	1	0	0	0	0	0	0
14	0	0	0	0	1	1	0	0	0	0
15	0	0	0	0	1	0	1	0	0	0
16	1	1	1	1	1	1	0	0	0	0
17	0	0	0	0	0	0	1	1	1	1
18	0	0	1	1	0	0	1	1	0	0
19	0	0	0	0	1	1	1	1	1	1
20	0	0	0	0	1	1	1	1	1	1

Lampiran 2: Kode Program MATLAB Algoritma Penjadwalan Kamar Operasi

Program Utama

```
% clc,clear all,close all;
data_pasien=xlsread('Data Operasi.xls',1,'A1:AN5'); %ambil data pasien
dari excell sheet 1 dari B1 sampai A05
data_operasi=xlsread('Data Operasi.xls',2,'B3:K22'); %ambil data
operasi dari excell sheet 2 dari B1 sampai A05
save data_pasien data_pasien %save variabel data_pasien ke
data_pasien.mat
save data_operasi data_operasi %save variabel data_operasi ke
data_operasi.mat
Pc=0.45; %Probabilitas Crossover
Pm=0.01; %Probabilitas Mutasi
maximp=100; %Maksimum Iterasi
pop_size=100; %Ukuran Populasi
no_pasien=40; %Banyak Pasien
no_dokter=size(data_operasi,1);
no_or=6; %Ukuran Ruang
no_half_day=size(data_operasi,2);
no_slot_waktu=40;
[xbest,fbest,f_rata2,f,totalwaktu]=al_gen(Pc,Pm,maximp,pop_size,no_pas
ien,no_or,no_dokter,no_half_day,no_slot_waktu); %Fungsi Algoritma
Genetika
totalwaktu
out=penggambaran(xbest,no_slot_waktu,no_or,no_half_day,no_dokter);
%menggambarkan hasil akhir
xlswrite('hasil.xls',out,'B3:AE42') %tulis ke excell
%=====Menggambarkan Grafik Fitnes=====
figure,plot(f,'k');hold
on;plot(f_rata2,'b');xlabel('Iterasi');ylabel('Pinalti');legend('Pinal
ti Terbaik','Pinalti Rata-rata')
```

Fungsi Mulai Algoritma Genetika

```
function
[xbest,fbest,f_rata2,f,totalwaktu]=al_gen(Pc,Pm,maximp,pop_size,no_pas
ien,no_or,no_dokter,no_half_day,no_slot_waktu)
tic
[kromosom,fkromosom,lebih_durasi,lebih_hari]=inisialisasi_pop(pop_size
,no_pasien,no_or);
for iterasi=1:maximp
    [parent1,parent2]=seleksi_parent(kromosom,fkromosom);
    if rand < Pc
        [anak1,anak2]=crossover(parent1,parent2);
    else
        anak1=parent1;
        anak2=parent2;
    end
    if rand<Pm
        anak1=mutasi(anak1,Pm,no_or);
        anak2=mutasi(anak2,Pm,no_or);
    end
    f_anak1=evaluasi_fitnes(anak1,no_or);
```

```

f_anak2=evaluasi_fitnes (anak2,no_or);
if f_anak1<fkromosom(end)
    kromosom{end,:}=anak1;
    fkromosom(end,1)=f_anak1;
end
if f_anak2<fkromosom(end)
    kromosom{end,:}=anak2;
    fkromosom(end,1)=f_anak2;
end
[fkromosom,index]=sort (fkromosom);
kromosom=kromosom(index,:);
if f_anak1<f_anak2
    fnew=f_anak1;
else
    fnew=f_anak2;
end
if iterasi==1
    xbest=kromosom{1,1};
    fbest=fnew;
else
if fnew<fbest
    xbest=kromosom{1,1};
    fbest=fnew;
end
end
f(iterasi)=fbest
f_rata2(iterasi)=mean (fkromosom);
iterasi
end
totalwaktu=toc;

```

Fungsi Inisialisasi Populasi

```

function
[kromosom, fkromosom, lebih_durasi, lebih_hari]=inisialisasi_pop(pop_size
,no_pasien,no_or)
load data_pasien
load data_operasi
for individu=1:pop_size
    matriks_individu(1,:)=randperm(no_pasien);
    for i=1:no_pasien
        posisi=find(data_pasien(1,:)==matriks_individu(1,i));
        dokter=data_pasien(2,posisi);
        keterdesakan=data_pasien(5,posisi);
        d_terpenuhi=find(data_operasi(dokter,1:keterdesakan)==1);
        random_d=randint(1,1,[1 length(d_terpenuhi)]);
        d=d_terpenuhi(random_d);
        matriks_individu(2,i)=d;
        matriks_individu(3,i)=randint(1,1,[1 no_or]);
    end
    kromosom{individu,1}=matriks_individu;

    [fkromosom(individu,1), lebih_durasi(individu,1), lebih_hari(individu,1)
]=evaluasi_fitnes(matriks_individu,no_or);
end

```

Fungsi Evaluasi Fitness

```

function
[nilai_fitnes,hasil_durasi,hasil_keterdesakan]=evaluasi_fitnes(matriks
_individu,no_or)
load data_patient
load data_operasi
nilai_fitnes=0;
temp_durasi=0;
hasil_durasi=0;
hasil_keterdesakan=0;
for room=1:no_or
    for d=1:10

posisi=find((matriks_individu(2,:)==d)&(matriks_individu(3,:)==room));
    if ~isempty(posisi)
        total_durasi=temp_durasi;
        for i_pasien=1:length(posisi)
            pasien=matriks_individu(posisi(1,i_pasien));
            dokter=data_patient(2,pasien);
            total_durasi=total_durasi+data_patient(4,pasien);
            if total_durasi>20
                if mod(d,2)==1
                    if data_operasi(dokter,d+1)==0
                        nilai_fitnes=nilai_fitnes+(total_durasi-
20)*1000;
                        hasil_durasi=hasil_durasi+(total_durasi-
20);
                        temp_durasi=0;
                    else
                        temp_durasi=total_durasi-20;
                    end
                else
                    nilai_fitnes=nilai_fitnes+(total_durasi-
20)*1000;
                    hasil_durasi=hasil_durasi+(total_durasi-20);
                    temp_durasi=0;
                end
            end
        end
    end
end
end
end
for i_pasien=1:40
    pos_pasien=find(matriks_individu(1,:)==i_pasien);
    d_pasien=matriks_individu(2,pos_pasien);
    hari_pasien=ceil(d_pasien/2);
    keterdesakan=data_patient(5,pos_pasien);
    hari_keterdesakan=ceil(keterdesakan/2);
    if hari_pasien>hari_keterdesakan
        nilai_fitnes=nilai_fitnes+(hari_pasien-
hari_keterdesakan)*1000;
        hasil_keterdesakan=hasil_keterdesakan+(hari_pasien-
hari_keterdesakan);
    end
end
end
end
end

```

```

    end
end

```

Fungsi Seleksi Orangtua

```

function [parent1,parent2]=seleksi_parent (populasi, fkromosom)
[n urut]=sort (fkromosom);
En=fkromosom (urut);
populasi=populasi (urut, :);
for i=1:size (populasi, 1)
    q (i, 1)=sum (En (1:i, 1));
end
q=q/q (end);
qparent=[];
while size (qparent, 1)~=2
    qbaru=[];
    while isempty (qbaru)
        r=rand;
        for i=2:length (q)
            if q (i-1)<r && q (i)>r
                qbaru=q (i);
            end
        end
        end
        ada=find (qparent==qbaru);
        if isempty (ada)
            qparent=[qparent; qbaru];
        end
    end
    pos1=find (q==qparent (1));
    parent1=populasi {pos1, 1};
    pos2=find (q==qparent (2));
    parent2=populasi {pos2, 1};

```

Fungsi Pindah Silang

```

function [hasil1, hasil2]=crossover (solusi1, solusi2)
q1=randint (1, 1, [1 length (solusi1)]);
q2=randint (1, 1, [1 length (solusi1)]);
while q1==q2
    q2=randint (1, 1, [1 length (solusi1)]);
end
hasil1 (1:2, 1:q1)=solusi1 (1:2, 1:q1);
hasil2 (1:2, 1:q1)=solusi2 (1:2, 1:q1);
temp1=[];
temp2=[];
for i=1:length (solusi2)
    pos1=find (hasil1 (1, :) == solusi2 (1, i));
    if isempty (pos1)
        temp1=[temp1, i];
    end
    pos2=find (hasil2 (1, :) == solusi1 (1, i));
    if isempty (pos2)

```

```

        temp2=[temp2,i];
    end
end
hasil1(1:2,q1+1:length(solusi1))=solusi2(1:2,temp1);
hasil2(1:2,q1+1:length(solusi1))=solusi1(1:2,temp2);

for i=1:length(solusi1)
    if i<=q2
        pos1=find(solusi1(1,:)==hasil1(1,i));
        hasil1(3,i)=solusi1(3,pos1);
        pos2=find(solusi2(1,:)==hasil2(1,i));
        hasil2(3,i)=solusi2(3,pos2);
    else
        pos1=find(solusi2(1,:)==hasil1(1,i));
        hasil1(3,i)=solusi2(3,pos1);
        pos2=find(solusi1(1,:)==hasil2(1,i));
        hasil2(3,i)=solusi1(3,pos2);
    end
end
end

```

Fungsi Mutasi

```

function out=mutasi(in,Pmut,no_or)
load data_pasien
load data_operasi
out=in;
for j=1:length(in)-1
    if rand<Pmut
        posisi_mut=randint(1,1,[1 length(in)]);
        pasien1=out(1,j+1);
        pasien2=out(1,posisi_mut);
        posisi=find(data_pasien(1,:)==pasien1);
        dokter1=data_pasien(2,posisi);
        keterdesakan1=data_pasien(5,posisi);
        d_terpenuhi1=find(data_operasi(dokter1,1:keterdesakan1)==1);
        random_d=randint(1,1,[1 length(d_terpenuhi1)]);
        d=d_terpenuhi1(random_d);
        out(1,posisi_mut)=out(1,j+1);
        out(2,posisi_mut)=d;
        out(3,posisi_mut)=randint(1,1,[1 no_or]);
        posisi=find(data_pasien(1,:)==pasien2);
        dokter2=data_pasien(2,posisi);
        keterdesakan2=data_pasien(5,posisi);
        d_terpenuhi2=find(data_operasi(dokter2,1:keterdesakan2)==1);
        random_d=randint(1,1,[1 length(d_terpenuhi2)]);
        d=d_terpenuhi2(random_d);
        out(1,j+1)=pasien2;
        out(2,j+1)=d;
        out(3,j+1)=randint(1,1,[1 no_or]);
    end
end
end

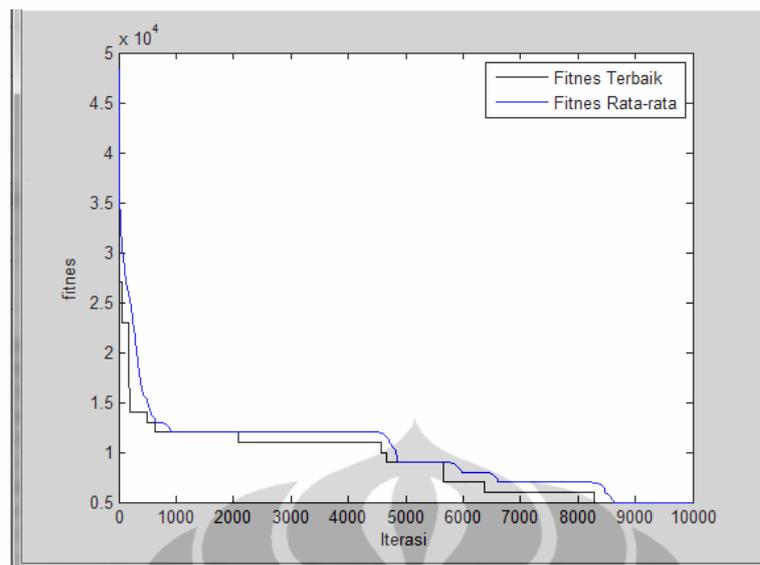
```

Fungsi Output

```

function
out=penggambaran(in,no_slot_waktu,no_or,no_half_day,no_dokter)
load data_pasien
posisi_slot(1,:)=ones(1,no_or*(no_half_day/2));
posisi_slot(2,:)=21*ones(1,no_or*(no_half_day/2));
posisi_dokter(1:no_dokter,:)=ones(no_dokter,no_half_day/2);
posisi_dokter(no_dokter+1:2*no_dokter,:)=ones(no_dokter,no_half_day/2)
;
for i=1:no_slot_waktu
    for j=1:no_or*(no_half_day/2)
        out{i,j}=[];
    end
end
for i=1:length(in)
    pasien=in(1,i);
    durasi=data_pasien(4,pasien);
    dokter=data_pasien(2,pasien);
    half_day=in(2,i);
    day=ceil(half_day/2);
    posisi_halfday=mod(half_day,2);
    ruang=in(3,i);
    if posisi_halfday==1
        mulai_dokter=posisi_dokter(dokter,day);
        if mulai_dokter<posisi_slot(1,(day-1)*no_or+ruang)
            for j=1:durasi
                out{posisi_slot(1,(day-1)*no_or+ruang)+j-1,(day-
1)*no_or+ruang}=pasien;
            end
            posisi_slot(1,(day-1)*no_or+ruang)=posisi_slot(1,(day-
1)*no_or+ruang)+durasi;
        else
            for j=1:durasi
                out{posisi_slot(1,(day-1)*no_or+ruang)+j-1,(day-
1)*no_or+ruang}=pasien;
            end
            posisi_slot(1,(day-1)*no_or+ruang)=posisi_slot(1,(day-
1)*no_or+ruang)+durasi;
            posisi_dokter(dokter,half_day)=mulai_dokter+durasi+4;
        end
    else
        mulai_dokter=posisi_dokter(20+dokter,day);
        for j=1:durasi
            out{posisi_slot(2,(day-1)*no_or+ruang)+j-1,(day-
1)*no_or+ruang}=pasien;
        end
        posisi_slot(2,(day-1)*no_or+ruang)=posisi_slot(2,(day-
1)*no_or+ruang)+durasi;
        posisi_dokter(20+dokter,half_day)=mulai_dokter+durasi+4;
    end
end
end

```

Grafik *fitness* (= nilai pinalti) solusi terbaik

Lampiran 4: Tabel Verifikasi Program

	Ukuran populasi	20		Ukuran populasi	20
	Jumlah iterasi	1000		Jumlah iterasi	10000
	Pc	0.45		Pc	0.45
	Pm	0.01		Pm	0.01
	Pinalti	Waktu		Pinalti	Waktu
	17000	10.4		10000	153.6
	26000	14.6		11000	108.8
	18000	14.4		13000	110.6
	24000	13.3		15000	122.1
	23000	16.2		11000	141.3
Rata-rata	21600	13.78	Rata-rata	12000	127.28
	Ukuran populasi	20		Ukuran populasi	20
	Jumlah iterasi	5000		Jumlah iterasi	15000
	Pc	0.45		Pc	0.45
	Pm	0.01		Pm	0.01
	Pinalti	Waktu		Pinalti	Waktu
	12000	56.7		5000	170.6
	17000	63.3		6000	185.5
	20000	76.2		4000	154.2
	19000	70.2		6000	169.69
	20000	61.1		9000	179.2
Rata-rata	17600	65.5	Rata-rata	6000	171.838

Lampiran 4: Tabel Verifikasi Program (lanjutan)

Ukuran populasi	40	Ukuran populasi	20
Jumlah iterasi	10000	Jumlah iterasi	5000
Pc	0.45	Pc	0.45
Pm	0.01	Pm	0.01
Pinalti	Waktu	Pinalti	Waktu
5000	185.3	12000	56.7
10000	125.6	17000	63.3
11000	138.2	20000	76.2
10000	118.9	19000	70.2
12000	116.7	20000	61.1
Rata-rata	9600 136.94	Rata-rata	17600 65.5
Ukuran populasi	40	Ukuran populasi	20
Jumlah iterasi	15000	Jumlah iterasi	5000
Pc	0.45	Pc	0.7
Pm	0.01	Pm	0.01
Pinalti	Waktu	Pinalti	Waktu
12000	201.4	16000	65.1
6000	198.9	14000	57
7000	209.3	9000	51.7
6000	233.2	15000	59.3
11000	250.4	15000	59.7
Rata-rata	8400 218.64	Rata-rata	13800 58.56
Ukuran populasi	20	Ukuran populasi	20
Jumlah iterasi	5000	Jumlah iterasi	5000
Pc	0.45	Pc	0.45
Pm	0.01	Pm	0.01
Pinalti	Waktu	Pinalti	Waktu
12000	56.7	12000	56.7
17000	63.3	17000	63.3
20000	76.2	20000	76.2
19000	70.2	19000	70.2
20000	61.1	20000	61.1
Rata-rata	17600 65.5	Rata-rata	17600 65.5