



UNIVERSITAS INDONESIA

**OPTIMASI PENJADWALAN PEMELIHARAAN PESAWAT
TERBANG DENGAN METODE ALGORITMA GENETIK**

SKRIPSI

**RAKHMAT ABU MUSA
0706201216**

**FAKULTAS TEKNIK
PROGRAM STUDI TEKNIK INDUSTRI
DEPOK
DESEMBER 2009**



UNIVERSITAS INDONESIA

**OPTIMASI PENJADWALAN PEMELIHARAAN PESAWAT
TERBANG DENGAN METODE ALGORITMA GENETIK**

SKRIPSI

Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik

**RAKHMAT ABU MUSA
0706201216**

**FAKULTAS TEKNIK
PROGRAM STUDI TEKNIK INDUSTRI
DEPOK
DESEMBER 2009**

ii

HALAMAN PERNYATAAN ORISINALITAS

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

Nama : Rakhmat Abu Musa

NPM : 0706201216

Tanda Tangan :

Tanggal : 30 Desember 2009

HALAMAN PERSETUJUAN

Skripsi dengan judul:

“Optimasi Penjadwalan Pemeliharaan Pesawat Terbang dengan Metode Algoritma Genetik”

Dibuat untuk melengkapi sebagian salah persyaratan menjadi Sarjana Teknik pada Program Teknik Industri Program Pendidikan Sarjana Fakultas Teknik Universitas Indonesia, dan disetujui untuk diajukan dalam sidang ujian skripsi.

Depok, 30 Desember 2009

Pembimbing Skripsi


Ir. Amal Rachman, MEIM

130 702 238

HALAMAN PENGESAHAN

Skripsi ini diajukan oleh :

Nama : Rakhmat Abu Musa
NPM : 0706201216
Program Studi : Teknik Industri
Judul Skripsi : Optimasi Penjadwalan Pemeliharaan Pesawat
Terbang Dengan Metode Algoritma Genetik

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

DEWAN PENGUJI

Pembimbing : Ir. Amar Rachman, MEIM

Penguji : Ir. Yadrifil, Msc

Penguji : Ir. Fauzia Dianawati, M.Si

Penguji : Ir. Akhmad Hidayatno, MBT

Ditetapkan di : Salemba, Jakarta

Tanggal : 30 Desember 2009

KATA PENGANTAR

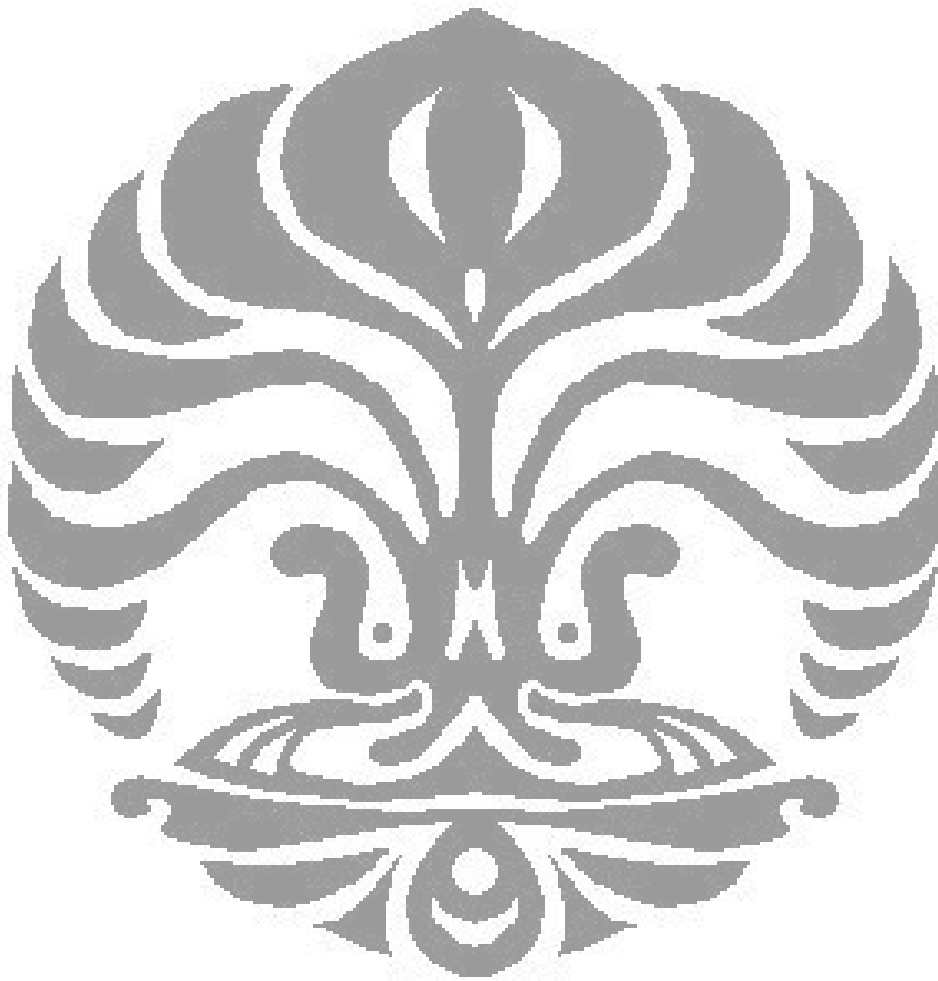
Puji syukur penulis panjatkan kepada Allah swt, karena atas berkat dan rahmat-Nya, penulis dapat menyelesaikan skripsi ini. Penulisan skripsi ini dilakukan dalam rangka memenuhi salah satu syarat untuk mencapai gelar Sarjana Teknik Jurusan Teknik Industri pada Fakultas Teknik Universitas Indonesia. Penulis menyadari bahwa tanpa bantuan dan bimbingan dari berbagai pihak, dari masa perkuliahan sampai pada masa penyusunan skripsi ini sangatlah sulit bagi penulis untuk menyelesaikan skripsi ini. Oleh karena itu, penulis mengucapkan terima kasih kepada:

1. Bapak Ir. Amar Rachman, MEIM , selaku dosen pembimbing skripsi yang selalu memberikan kepercayaan, semangat, bimbingan, dan bantuan yang luar biasa.
2. Ibu Ir. Betrianis, M.Si, selaku pembimbing akademis atas perhatiannya.
3. Bapak Yadrifil, Bapak Boy, Ibu Fauzia, dan Ibu Arian Dhini atas semua masukan dan kritiknya selama masa seminar.
4. Bapak Komarudin yang telah memberikan ilmu dan pengetahuan kepada penulis selama penyusunan skripsi.
5. Segenap jajaran Dosen Departemen Teknik Industri yang telah memberikan ilmu dan pengetahuan kepada penulis selama masa perkuliahan.
6. Bagian Administrasi Departemen Teknik Industri yang selalu siap sedia membantu penulis dalam segala urusan.
7. Semua pihak yang membantu memberikan masukan kepada penulis: Bapak Subiat, Bapak Suharsono, Bapak Tatang, dan rekan-rekan kerja di Sekolah Tinggi Penerbangan Indonesia (STPI).
8. Keluarga penulis Bapak, Ibu serta kakak yang selalu memberikan kasih sayang dan perhatiannya tanpa mengharap balasan.
9. Teman-teman TI ekstensi salemba angkatan 2007 yang selalu memberikan keceriaan dan persahabatan selama masa perkuliahan.

Akhir kata saya berharap Tuhan Yang Maha Esa berkenan membalas segala kebaikan semua pihak yang telah membantu. Semoga skripsi ini membawa manfaat bagi pengembangan ilmu ke depannya.

Depok, Desember 2009

Penulis



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

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

Nama : Rakhmat Abu Musa
NPM : 0706201216
Program Studi : Teknik Industri
Departemen : Teknik Industri
Fakultas : Teknik
Jenis karya : Skripsi

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

“Optimasi Penjadwalan Pemeliharaan Pesawat Terbang dengan Metode Algoritma Genetik”

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 mempublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilih Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di: Depok

Pada tanggal : 30 Desember 2009

Yang menyatakan



(Rakhmat Abu Musa)

ABSTRAK

Nama : Rakhmat Abu Musa
Program Studi : Teknik Industri
Judul Skripsi : Optimasi Penjadwalan Pemeliharaan Pesawat
Terbang Dengan Metode Algoritma Genetik

Menumpuknya pesawat terbang pada saat pemeliharaan berkala menyebabkan kurangnya pesawat terbang yang dapat dioperasikan untuk pelatihan bagi mahasiswa penerbang. Untuk itu diperlukan jadwal pemakaian dan jadwal pemeliharaan pesawat terbang agar diperoleh jadwal yang optimal. Penelitian ini bertujuan untuk memperoleh suatu sistem penjadwalan pemeliharaan pesawat terbang yang optimal dengan meminimumkan terjadinya pemeliharaan 2000 jam secara bersamaan dengan metode algoritma genetik. Hasil yang dicapai adalah jadwal pemeliharaan pesawat terbang dengan meminimumkan terjadinya pemeliharaan 2000 jam selama 10 tahun dengan standard deviasi sebesar 0,43483.

Kata kunci:

Optimasi, Penjadwalan, Pemeliharaan Pesawat, Algoritma Genetik

ABSTRACT

Name : Rakhmat Abu Musa
Study Program : Industrial Engineering
Title : Optimization Aircraft Maintenance Scheduling
With Genetic Algorithm Method

Stacked of aircraft at the time in a periodic aircraft maintenance causing the lack of aircraft which is operated for the aviation training students. This require the schedule usage and aircraft maintenance schedule in order to obtain the optimal schedule. This study aims to obtain a maintenance scheduling system of the optimal aircraft to minimize the occurrence of 2000-hour maintenance in conjunction with genetic algorithm method. The results are an aircraft maintenance schedule to minimize the occurrence of 2000-hour maintenance for 10 years with a standard deviation of 0.43483.

Keywords:

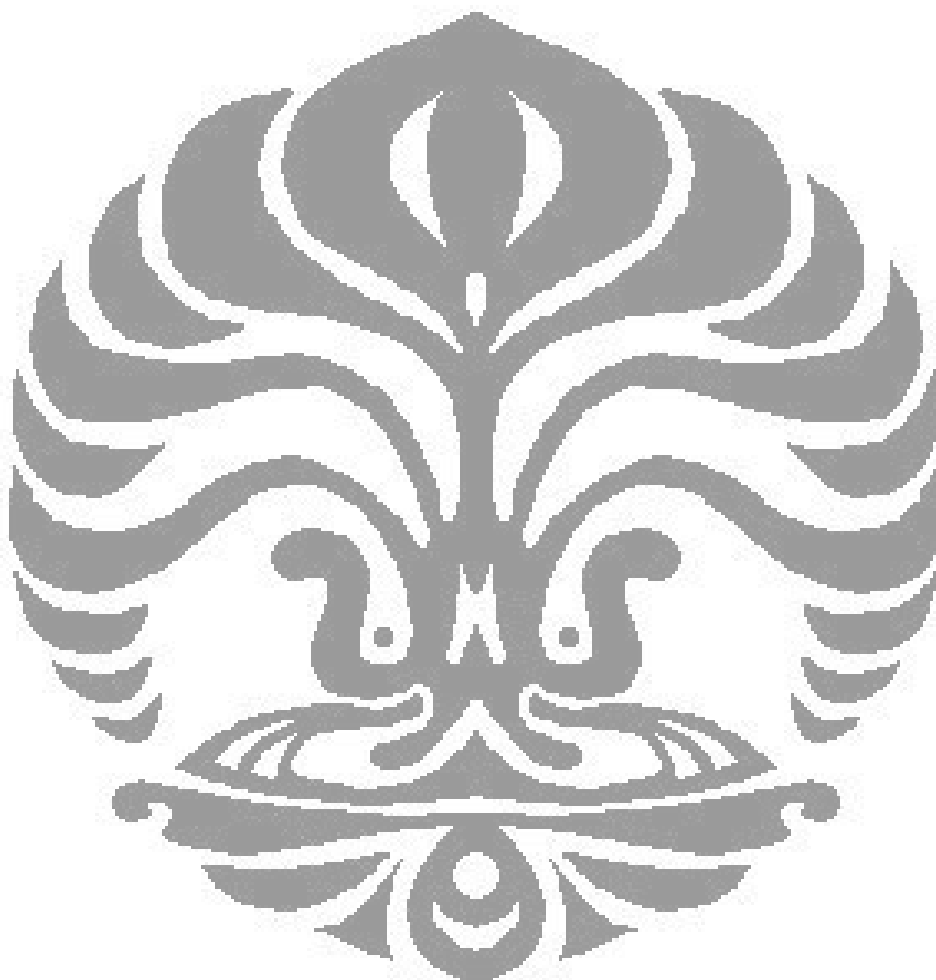
Optimization, Scheduling, Aircraft Mintenance, Genetic Algorithm

DAFTAR ISI

| | |
|--|----------|
| HALAMAN JUDUL | i |
| HALAMAN PERNYATAAN ORISINALITAS | iii |
| HALAMAN PERSETUJUAN | iv |
| HALAMAN PENGESAHAN | v |
| KATA PENGANTAR | vi |
| HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI | viii |
| ABSTRAK | ix |
| DAFTAR ISI | xi |
| DAFTAR TABEL | xiv |
| DAFTAR GAMBAR | xv |
| DAFTAR RUMUS | xvi |
| DAFTAR LAMPIRAN | xvii |
| 1. PENDAHULUAN | 1 |
| 1.1 Latar Belakang | 1 |
| 1.2 Diagram Keterkaitan Masalah | 3 |
| 1.3 Perumusan Masalah | 4 |
| 1.4 Tujuan Penelitian | 4 |
| 1.5 Batasan Masalah | 4 |
| 1.6 Metodologi Penelitian | 4 |
| 1.7 Sistematika Penulisan | 6 |
| 2. LANDASAN TEORI | 8 |
| 2.1 Penjadwalan | 8 |
| 2.2 Pemeliharaan | 8 |
| 2.2.1 Jenis Pemeliharaan | 8 |
| 2.3 Algoritma Genetik | 9 |
| 2.3.1 Struktur Umum Algoritma Genetik | 10 |
| 2.3.2 Komponen Utama Algoritma Genetik | 10 |
| 2.3.2.1 Prosedur Inisialisasi | 12 |

| | |
|--|-----------|
| 2.3.2.2 Teknik Pengkodean | 12 |
| 2.3.2.3 Fungsi Evaluasi | 12 |
| 2.3.2.4 Seleksi | 13 |
| 2.3.2.5 Pindah Silang | 14 |
| 2.3.2.6 Mutasi | 15 |
| 2.3.2.7 Penentuan Parameter | 16 |
| 2.3.2.8 Penggantian Populasi | 16 |
| 2.4 Design Of Experiment | 17 |
| 2.4.1 Prinsip Dasar dalam design of experiment | 18 |
| 2.4.2 Langkah-langkah Percobaan | 19 |
| 2.4.3 Factorial Design | 20 |
| 3. PENGUMPULAN DATA | 23 |
| 3.1 Profil STPI | 23 |
| 3.2 Pengumpulan Data Penelitian | 23 |
| 3.2.1 Data Pesawat | 24 |
| 3.2.2 Data Pemeliharaan | 25 |
| 3.2.3 Data Penerbangan | 26 |
| 3.2.4 Data Penggunaan Pesawat | 29 |
| 3.2.5 Data Pemeliharaan Pesawat | 29 |
| 4. PENGOLAHAN DATA DAN ANALISIS | 30 |
| 4.1 Fungsi Tujuan | 30 |
| 4.2 Algoritma Genetik | 31 |
| 4.2.1 Langkah-langkah Penyusunan Algoritma Genetik | 33 |
| 4.2.2 Verifikasi dan Validasi | 34 |
| 4.2.2.1 Verifikasi Program | 34 |
| 4.2.2.2 Validasi Program | 35 |
| 4.3 Percobaan Kombinasi Parameter | 35 |
| 4.3.1 Persiapan dan Perancangan Percobaan | 35 |
| 4.3.2 Hasil Percobaan Parameter | 37 |

| | |
|--|-----------|
| 4.3.3 Susunan Penjadwalan yang Terbaik | 38 |
| 4.4 Analisa | 40 |
| 4.4.1 Analisa Percobaan Parameter | 40 |
| 4.4.2 Analisa Rotasi Penjadwalan | 43 |
| 5. KESIMPULAN | 44 |
| DAFTAR REFERENSI | 45 |
| LAMPIRAN | 47 |



DAFTAR TABEL

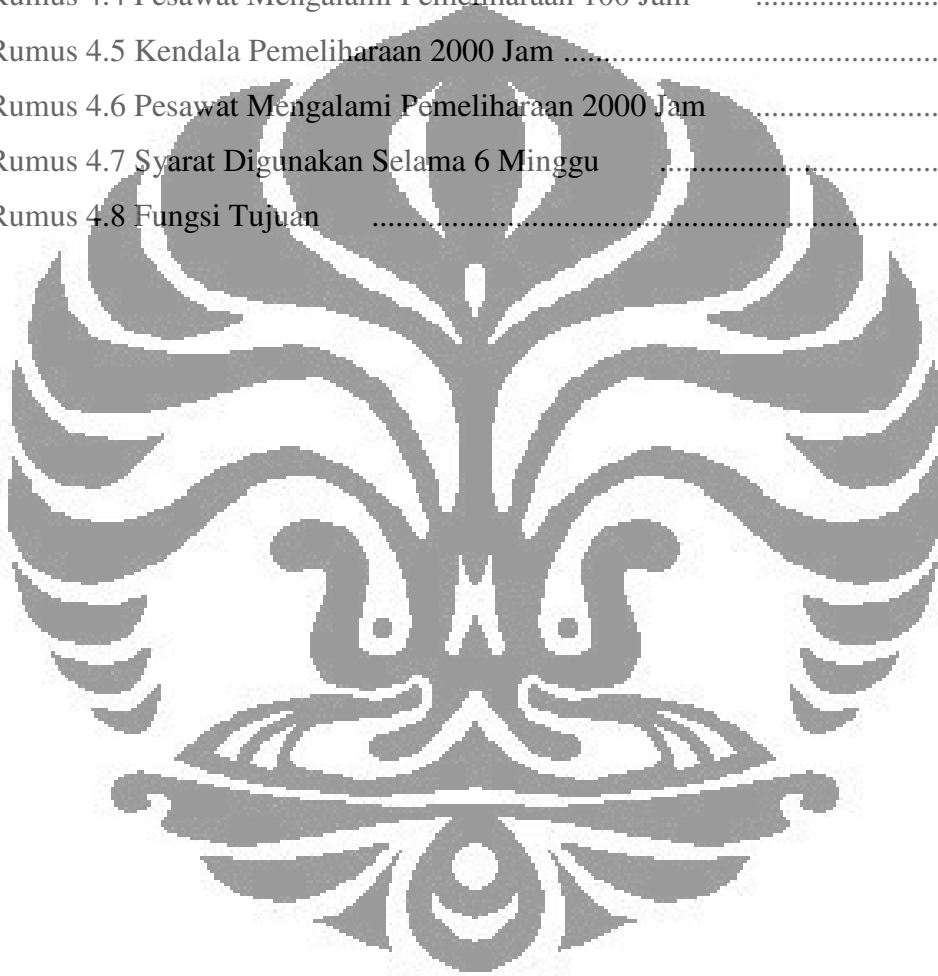
| | |
|---|----|
| Tabel 3.1 Daftar Pesawat Dan Umur Peswat | 24 |
| Tabel 3.2 Data Pekerjaan Divisi Pemeliharaan | 25 |
| Tabel 3.3 Data Inspeksi Pesawat | 25 |
| Tabel 3.4 Data Jurusan Penerbang | 26 |
| Tabel 3.5 Silabus Penggunaan Pesawat Jurusan Penerbang | 27 |
| Tabel 3.6 Data Penggunaan Pesawat | 28 |
| Tabel 4.1 Kombinasi Parameter yang Digunakan dalam Verifikasi | 34 |
| Tabel 4.2 Hasil Validasi dengan Penghitungan Manual | 35 |
| Tabel 4.3 Kombinasi Level Tiap Faktor Parameter | 36 |
| Tabel 4.4 Kombinasi Parameter Terbaik | 38 |
| Tabel 4.6 Rotasi Pemakaian Pesawat Terbaik | 39 |
| Tabel 4.7 Hasil Penjadwalan Pemeliharaan Pesawat Terbang | 43 |

DAFTAR GAMBAR

| | |
|--|---------|
| Gambar 1.1 Diagram Keterkaitan Masalah |3 |
| Gambar 1.2 Diagram Alir Metodologi Penelitian |5 |
| Gambar 2.1 Diagram Alir Algoritma Genetik Sederhana |11 |
| Gambar 2.2 Pseudocode Algoritma Genetik |11 |
| Gambar 2.3 Skema pengkodean <i>binary encoding</i> |12 |
| Gambar 2.4 Contoh Proses Pindah Silang Satu Titik Potong |15 |
| Gambar 2.5 Contoh Proses Mutasi |16 |
| Gambar 2.6 Contoh Perancangan 2^2 <i>factorial design</i> |21 |
| Gambar 2.7 Perubahan <i>Factorial</i> tanpa Interaksi |21 |
| Gambar 2.8 Perubahan <i>Factorial</i> dengan Interaksi |22 |
| Gambar 3.1 Kebutuhan Pesawat per Minggu di Bulan x |26 |
| Gambar 3.2 Data Pemeliharaan Pesawat |29 |
| Gambar 4.1 Diagram Alir Algoritma Penjadwalan Perawatan Pesawat |32 |
| Gambar 4.2 Grafik Hasil Pengolahan Data |39 |
| Gambar 4.3 Grafik <i>Main Effects Plot</i> untuk ketiga Faktor Parameter |42 |

DAFTAR RUMUS

| | |
|---|----|
| Rumus 2.1 Nilai <i>Fitness</i> | 13 |
| Rumus 2.2 Nilai <i>Fitness</i> Baru | 14 |
| Rumus 4.1 Umur Pesawat | 30 |
| Rumus 4.2 Kendala Pemeliharaan 50 Jam | 30 |
| Rumus 4.3 Kendala Pemeliharaan 100 Jam | 30 |
| Rumus 4.4 Pesawat Mengalami Pemeliharaan 100 Jam | 30 |
| Rumus 4.5 Kendala Pemeliharaan 2000 Jam | 30 |
| Rumus 4.6 Pesawat Mengalami Pemeliharaan 2000 Jam | 30 |
| Rumus 4.7 Syarat Digunakan Selama 6 Minggu | 30 |
| Rumus 4.8 Fungsi Tujuan | 31 |



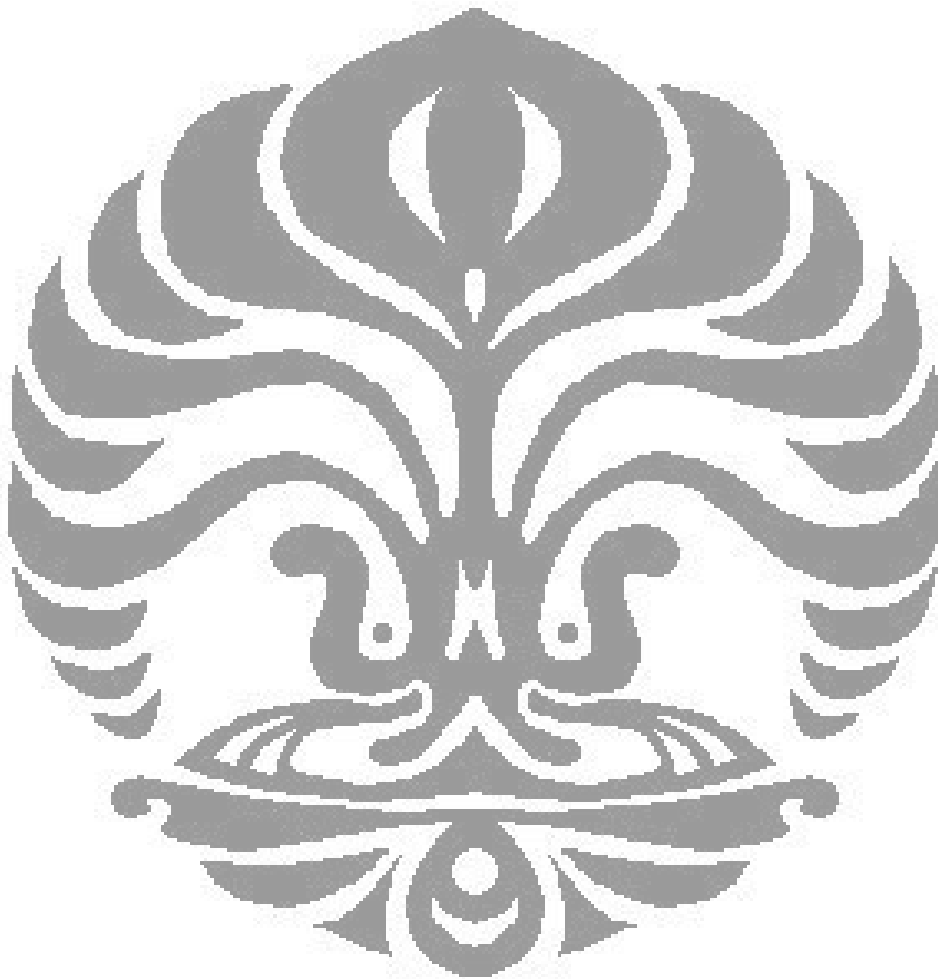
DAFTAR LAMPIRAN

Lampiran 1: *Script* Program Algoritma Penyelesaian

Lampiran 2: Hasil Penjadwalan Terbaik

Lampiran 3: Program Untuk Validasi

Lampiran 4: Hasil DOE



BAB I

PENDAHULUAN

1.1 LATAR BELAKANG

Penjadwalan merupakan hal yang terpenting dalam penerbangan. Pada pesawat terbang penjadwalan penerbangan dan penjadwalan pemeliharaan adalah hal yang utama untuk diperhatikan, karena penjadwalan yang optimal sangatlah diperlukan dalam penerbangan. Apabila di dalam suatu perusahaan terdapat suatu penjadwalan yang optimal, maka penggunaan waktu, tenaga, produksi, dan keuntungan yang dimiliki juga akan lebih optimal.

Setiap pesawat udara selama beroperasi pasti mempunyai jadwal untuk pemeliharaan. Pemeliharaan ini harus dilakukan karena setiap komponen mempunyai batas usia tertentu sehingga komponen tersebut harus diganti atau diperbaiki. Pada dasarnya pemeliharaan dibagi menjadi dua, pemeliharaan preventif yaitu pemeliharaan untuk mencegah terjadinya kegagalan komponen sebelum komponen itu rusak, pemeliharaan korektif yaitu pemeliharaan komponen yang rusak agar bisa digunakan kembali.

Pemeliharaan pesawat biasanya dikelompokkan berdasarkan interval sepadan dalam paket-paket kerja atau sering disebut *clustering*. Hal ini dilakukan agar tugas pemeliharaan lebih mudah, efektif, dan efisien. Interval yang dilakukan untuk melaksanakan paket-paket tersebut adalah:

- *Flight Hour*
Merupakan interval inspeksi yang didasarkan pada jumlah jam operasional suatu pesawat terbang.
- *Flight Cycle*
Merupakan interval inspeksi yang didasarkan pada jumlah *takeoff-landing* yang dilakukan suatu pesawat terbang. Satu kali *takeoff-landing* dihitung satu cycle.
- *Calender Time*
Merupakan interval inspeksi yang dilakukan sesuai dengan jadwal tertentu.

Lembaga pendidikan penerbangan adalah lembaga yang memiliki tanggung jawab untuk mendidik para mahasiswa dibidang pendidikan penerbangan, meliputi jurusan penerbang, teknik penerbangan, keselamatan penerbangan, dan manajemen penerbangan. Jurusan penerbang membutuhkan kondisi pesawat latih yang siap pakai setiap hari, agar jadwal terbang mahasiswa tidak terganggu dengan adanya jadwal pesawat latih yang sedang diinspeksi. Pemeliharaan preventif pesawat latih di lembaga pendidikan penerbangan biasa dikenal dengan pemeliharaan *scheduling*. Untuk *flight hour* dibagi menjadi pemeliharaan kelipatan 50, 100, dan 2000 jam. Setiap kelompok mempunyai *job list* yang berbeda yang harus dikerjakan. Untuk *calender time* biasanya di lembaga pendidikan penerbangan tidak dilakukan karena pemakaian pesawat latih yang sering, jadi pemeliharaannya mengikuti *flight hour*. Untuk pemeliharaan korektif biasa dikenal dengan pemeliharaan *unscheduling*. Contoh pemeliharaan *unscheduling* adalah *hard landing*, dimana pilot (mahasiswa) melaporkan kejadian kerusakan kepada pihak pemeliharaan.

Tujuan penjadwalan pemeliharaan pesawat latih adalah menyusun urutan pemeliharaan pesawat latih yang terjadwal (50, 100, dan 2000 jam) dalam satu interval waktu, misal 1 atau 2 tahun agar kondisi pesawat latih selalu siap jika digunakan dan tidak ada penumpukan pesawat latih pada saat pemeliharaan.

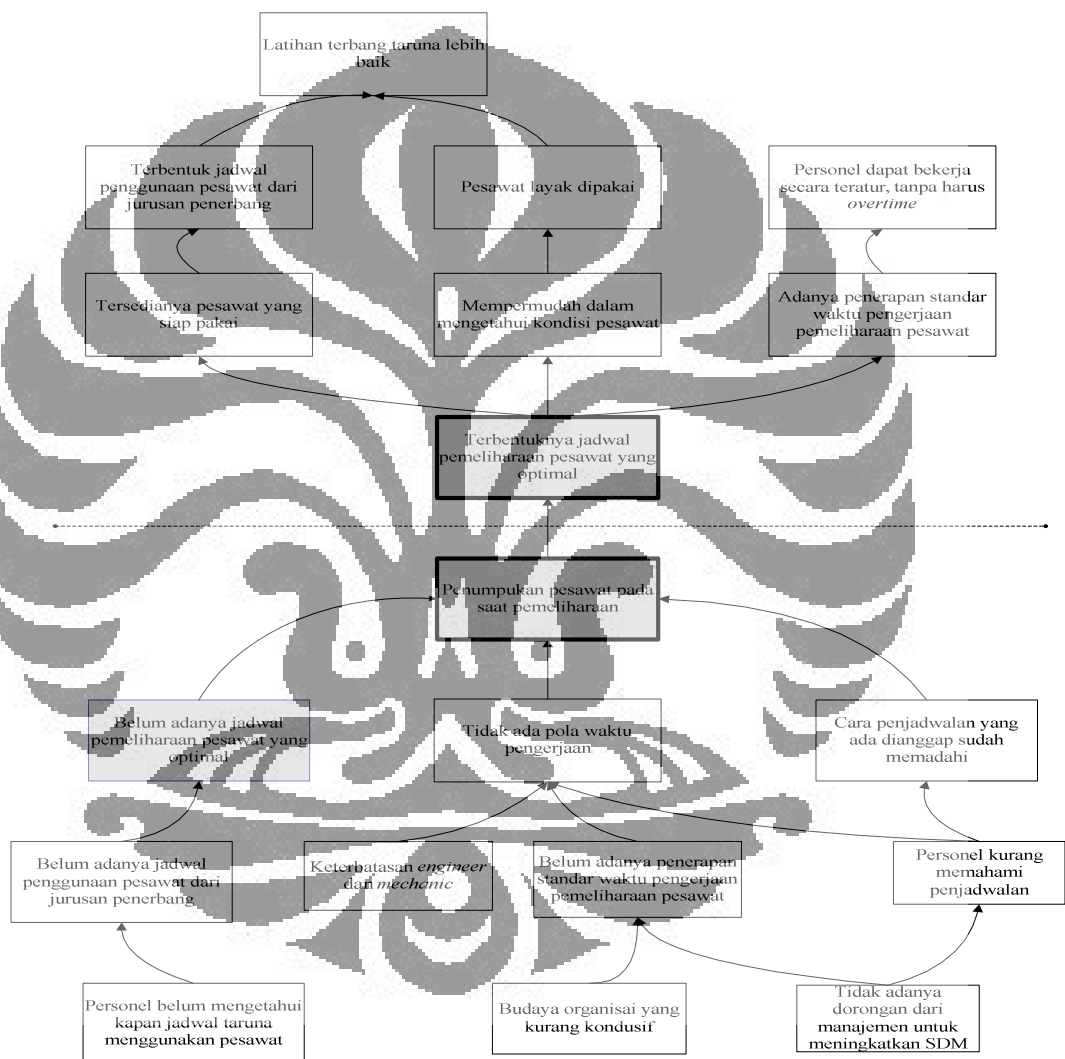
Sekolah Tinggi Penerbangan Indonesia (STPI) adalah lembaga pendidikan penerbangan. Permasalahan yang dihadapi oleh STPI adalah penumpukan pesawat pada saat pemeliharaan, mengakibatkan jumlah pesawat yang siap digunakan tidak memenuhi permintaan dari jurusan penerbang. Dan juga tidak adanya jadwal kebutuhan pemakaian pesawat yang diberikan dari pihak jurusan penerbang kepada divisi pemeliharaan. Maka dari itu penjadwalan harus disesuaikan dari pihak pemeliharaan maupun penerbang.

Pada penelitian ini akan dicari solusi untuk menyelesaikan permasalahan penjadwalan pemeliharaan pesawat latih dengan menggunakan metode Algoritma Genetika, yang merupakan metode meta-heuristik. Prinsipnya adalah berdasarkan

analogi evolusi biologi, yang terdiri dari proses penginisialisasian populasi, proses mutasi, proses penyilangan, dan proses penyeleksian¹.

1.2 DIAGRAM KETERKAITAN PERMASALAHAN

Masalah-masalah dalam penelitian ini digambarkan pada diagram keterkaitan masalah, yang akan ditampilkan pada gambar 1.1.



Gambar 1.1 Diagram Keterkaitan Masalah

¹ Mattila V., Virtanen K. 2005. A Simulation-Based Optimization Model To Schedule Periodic Maintenance Of A Fleet Of Aircraft

1.3 RUMUSAN MASALAH

Penjadwalan pemeliharaan pesawat latih yang kurang baik dapat mengakibatkan terhambatnya jadwal latihan terbang pada jurusan penerbang. Pokok permasalahan yang akan dibahas adalah perlunya perancangan suatu sistem penjadwalan pesawat latih yang baik.

1.4 TUJUAN PENELITIAN

Memperoleh suatu jadwal pemeliharaan yang lebih baik (lebih mendekati optimal) dengan meminimumkan waktu pemeliharaan melalui algoritma genetika.

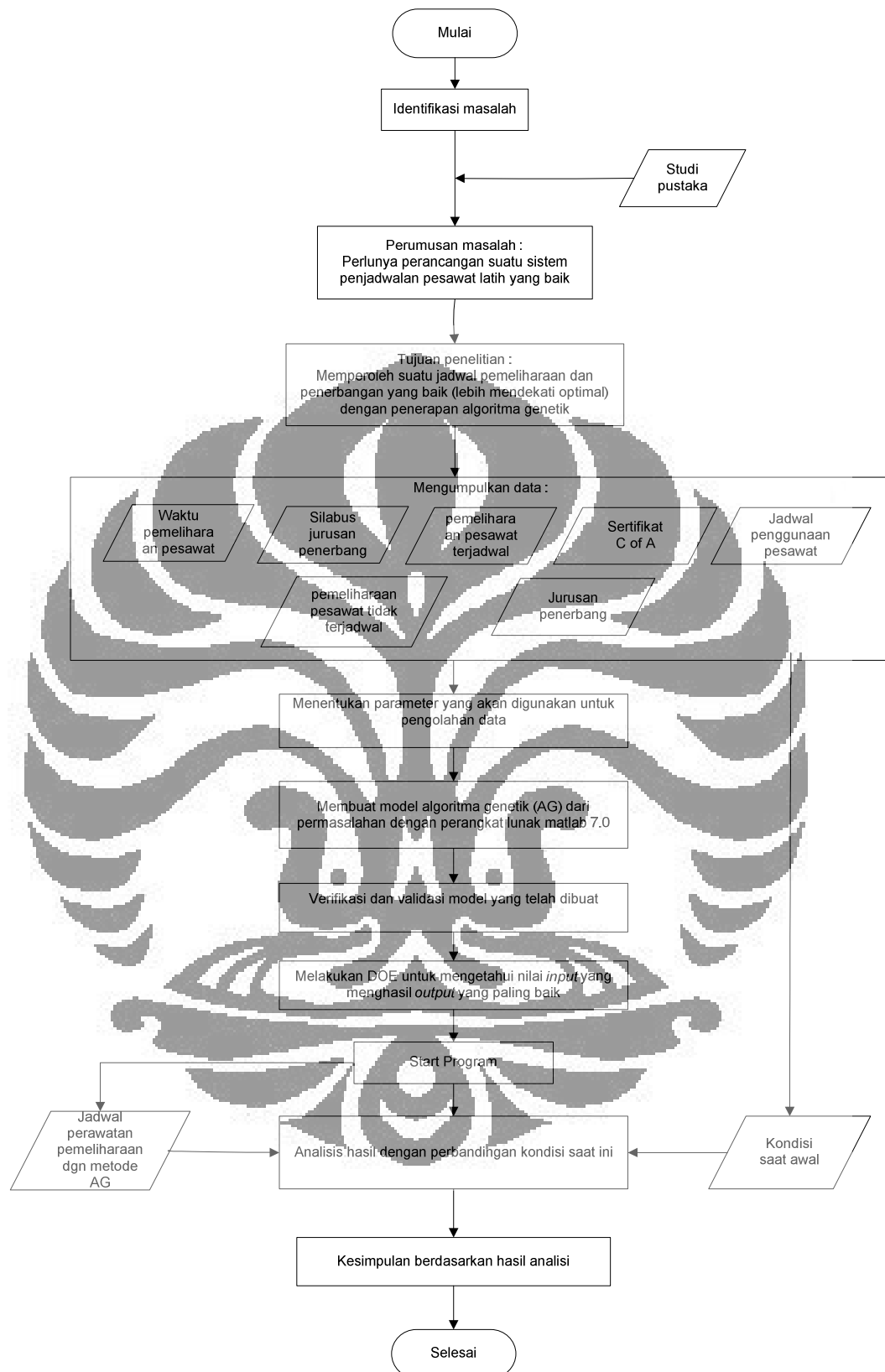
1.5 BATASAN MASALAH

Untuk mendapatkan hasil penelitian yang spesifik dan terarah, maka ruang lingkup permasalahan dari penelitian ini adalah sebagai berikut :

1. Penelitian dilakukan di Sekolah Tinggi Penerbangan Indonesia (STPI), dengan fokus kepada penjadwalan pemeliharaan pesawat terbang.
2. Pesawat latih yang digunakan untuk penelitian adalah Socata TB10, Sundowner C23, dan Dakota PA28.
3. Pengumpulan data dari Januari 2008 – September 2009.
4. Pesawat yang terbang berarti pesawat digunakan satu minggu selama 25 jam.
5. Jumlah karyawan untuk *mechanic* pemeliharaan pesawat 15 orang.
6. Jumlah pesawat 24 pesawat.
7. Jumlah Mahasiswa 25 orang per semester.

1.6 METODOLOGI PENELITIAN

Berikut adalah langkah-langkah metodologi yang digunakan dalam penelitian, sebagaimana tergambar pada diagram alir metodologi penelitian (Gambar 1.2) :



Gambar 1.2 Diagram Alir Metodologi Penelitian

Penjelasan dari diagram alir metodologi :

1. Melakukan identifikasi permasalahan perusahaan.
2. Mengumpulkan dan menyusun studi literatur yang berkaitan dengan masalah yang telah diidentifikasi.
3. Merumuskan masalah, yaitu perlunya perancangan suatu sistem penjadwalan yang mendekati optimal.
4. Menentukan tujuan, yaitu memperoleh suatu sistem penjadwalan yang lebih baik yang mendekati optimal.
5. Mengidentifikasi data yang dibutuhkan dan selanjutnya mengumpulkan data sekunder dari perusahaan.
6. Menentukan parameter yang akan digunakan untuk pengolahan data.
7. Membuat model algoritma genetik dari permasalahan dengan perangkat lunak matlab 7.8.
8. Melakukan verifikasi dan validasi terhadap program yang telah dibuat.
9. Melakukan DOE untuk mengetahui nilai *input* yang menghasilkan nilai *output* yang paling baik.
10. Membandingkan dan menganalisis solusi jadwal dengan kondisi awal.
11. Kesimpulan berdasarkan hasil analisa.

1.7 SISTEMATIKA PENULISAN

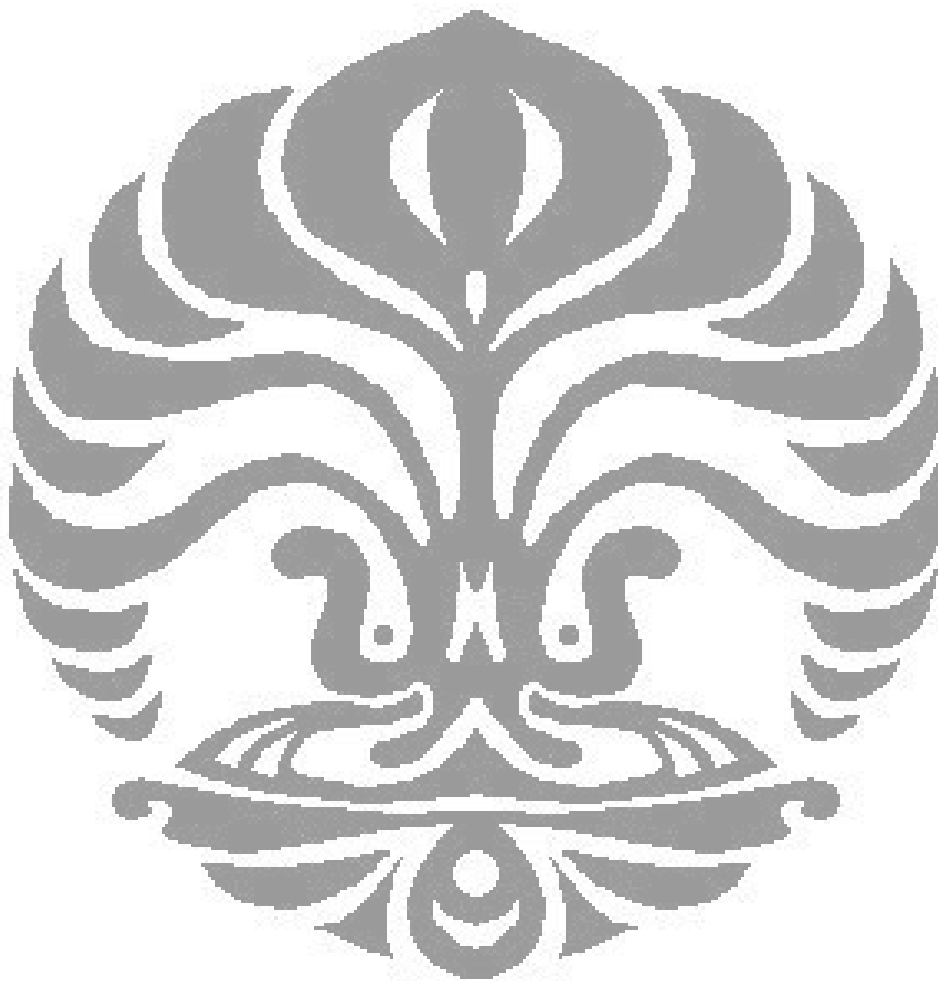
Penulisan laporan penelitian ini dibagi menjadi lima bab.

- BAB I** Merupakan bab pendahuluan, menjelaskan mengenai latar belakang permasalahan, diagram yang menggambarkan keterkaitan masalah, perumusan masalah, tujuan penelitian yang ingin dicapai, batasan masalah yang dilakukan, metodologi penelitian yang dilakukan oleh penulis, dan sistematika penulisan.
- BAB II** Merupakan bab landasan teori, berisikan mengenai teori-teori yang berkaitan dengan penjadwalan pemeliharaan pesawat latih dan algoritma genetik.
- BAB III** Merupakan bab pengumpulan data, menjelaskan mengenai data yang diambil oleh penulis selama penelitian yang akan dijadikan

input dalam pengolahan data yang dilakukan pada tahap selanjutnya.

BAB IV Merupakan pengolahan data dan analisis hasil yang diperoleh. Berisikan tentang analisa terhadap hasil pengolahan data yang telah dilakukan sehingga diketahui hasil akhir tujuan.

BAB V Merupakan kesimpulan yang diambil berdasarkan hasil penelitian dan analisa.



BAB II

LANDASAN TEORI

2.1 PENJADWALAN

Penjadwalan memiliki pengertian secara khusus sebagai durasi dari waktu kerja yang dibutuhkan untuk melakukan serangkaian aktivitas kerja yang ada dalam kegiatan penerbangan. Penjadwalan juga merupakan proses penyusunan daftar pekerjaan yang akan dilakukan untuk mencapai atau mewujudkan suatu tujuan tersebut yang juga memuat tabel untuk pelaksanaannya.

2.2 PEMELIHARAAN

Setiap pesawat udara selama beroperasi pasti mempunyai jadwal untuk pemeliharaan. Perawatan ini harus dilakukan karena setiap komponen mempunyai batas usia tertentu sehingga komponen tersebut harus diganti. Selain itu, komponen juga harus diperbaiki bila ditemukan telah mengalami kerusakan.

2.2.1 Jenis Pemeliharaan

Secara garis besar, program pemeliharaan dapat dibagi menjadi dua kelompok besar, yaitu pemeliharaan preventif dan pemeliharaan korektif. Pemeliharaan preventif adalah pemeliharaan yang mencegah terjadinya kegagalan komponen sebelum komponen tersebut rusak. Pemeliharaan korektif adalah pemeliharaan yang memperbaiki komponen yang rusak agar dapat ke kondisi awal / digunakan lagi.

Perawatan preventif dapat dibagi menjadi 2 jenis yaitu:

1. Perawatan periodik atau hard time, merupakan perawatan yang dilakukan berdasarkan batas waktu dari umur maksimum suatu komponen pesawat. Dengan kata lain, perawatan ini merupakan perawatan pencegahan dengan cara mengganti komponen pesawat meskipun komponen tersebut belum mengalami kerusakan.

2. Perawatan on-condition, merupakan perawatan yang memerlukan inspeksi untuk menentukan kondisi suatu komponen pesawat. Setelah itu ditentukan tindakan selanjutnya berdasarkan hasil inspeksi tersebut. Bila ada gejala kerusakan, komponen tersebut dapat diganti bila alasan-alasan teknik dan ekonominya memenuhi.

Perawatan korektif dikenal pula dengan nama condition monitoring yaitu perawatan yang dilakukan setelah ditemukan kerusakan pada suatu komponen, dengan cara memperbaiki komponen tersebut. Bila cara perbaikan tidak dapat dilakukan dengan alasan teknik maupun ekonomi, maka harus dilakukan penggantian.

2.3 ALGORITMA GENETIK

Algoritma genetik adalah algoritma pencarian heuristik yang didasarkan pada mekanisme evolusi biologi. Keberagaman pada evolusi biologis adalah variasi dari kromosom dalam individu organisme. Variasi kromosom ini akan mempengaruhi laju reproduksi dan tingkat kemampuan organisme untuk tetap hidup². Pada dasarnya ada empat kondisi yang mempengaruhi proses evaluasi, yaitu :

1. Kemampuan organisme untuk melakukan reproduksi.
2. Keberadaan populasi organisme yang bisa melakukan reproduksi.
3. Keberagaman organisme dalam suatu populasi.
4. Perbedaan kekuatan dan kemampuan organisme untuk bertahan hidup.

Individu yang lebih kuat (*fit*) akan memiliki tingkat *survival* atau tingkat daya bertahan hidup yang lebih tinggi. Selain itu individu yang semakin kuat akan memiliki tingkat reproduksi yang lebih tinggi jika dibandingkan dengan individu yang kurang *fit*. Pada kurun waktu tertentu (sering dikenal dengan istilah generasi), populasi secara keseluruhan akan memuat lebih banyak organisme yang *fit*.

Algoritma genetik pertama kali dikembangkan oleh John Holland dari Universitas Michigan (1975). John Holland mengatakan bahwa setiap masalah

² Kusumadewi, 2005, hal 231

yang berbentuk adaptasi (alami maupun buatan) dapat diformulasikan dalam terminologi genetik. Algoritma genetik adalah simulasi dari proses evolusi Darwin dan operasi genetika atas kromosom.

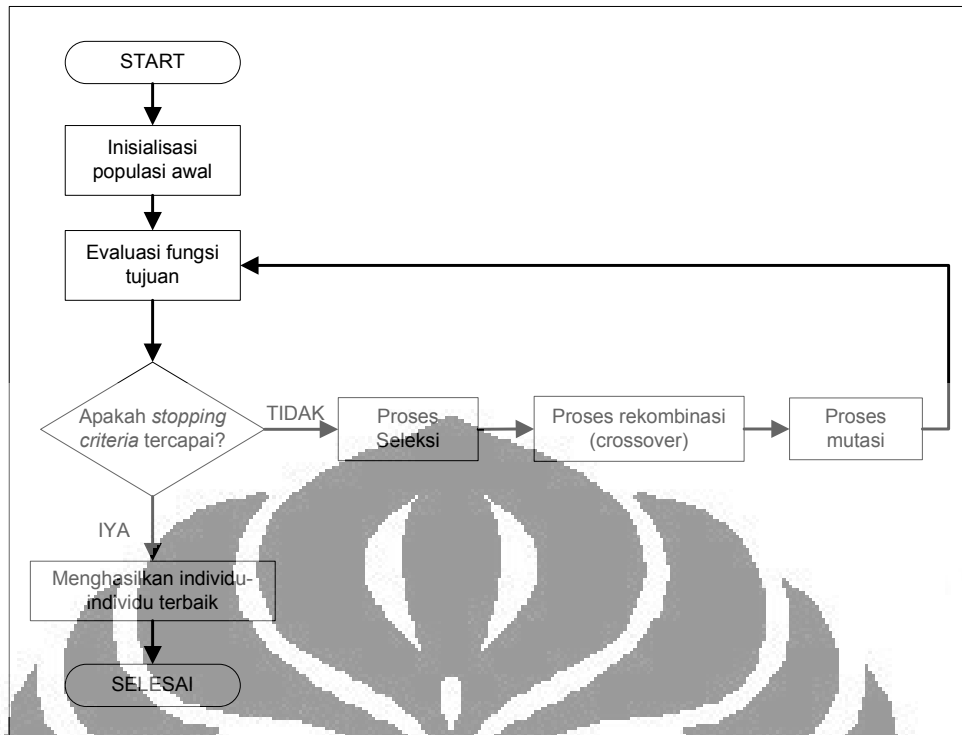
2.3.1 Struktur Umum Algoritma Genetik

Pada algoritma ini, teknik pencarian dilakukan sekaligus atas sejumlah solusi yang mungkin yang dikenal dengan istilah populasi. Individu yang terdapat dalam satu populasi disebut dengan kromosom. Kromosom ini merupakan suatu solusi yang masih berbentuk simbol. Populasi awal dibangun secara acak, sedangkan populasi berikutnya merupakan hasil evolusi kromosom-kromosom melalui iterasi yang disebut dengan generasi. Pada setiap generasi, kromosom akan melalui proses evaluasi dengan menggunakan alat ukur yang disebut dengan fungsi *fitness*. Nilai *fitness* dari suatu kromosom akan menunjukkan kualitas kromosom dari populasi tersebut. Generasi berikutnya dikenal dengan istilah anak (*offspring*) terbentuk dari gabungan dua kromosom generasi sekarang yang bertindak sebagai induk (*parent*) dengan menggunakan operator penyilangan (*crossover*). Selain operator penyilangan, suatu kromosom dapat juga dimodifikasi dengan menggunakan operator *mutasi*.

Populasi generasi yang baru dibentuk dengan menyeleksi nilai *fitness* dari kromosom induk (*parent*) dan nilai *fitness* dari kromosom anak (*offspring*), serta menolak kromosom-kromosom yang lainnya sehingga ukuran populasi (jumlah kromosom dalam suatu populasi) konstan. Setelah melalui beberapa generasi, maka algoritma ini akan konvergen ke kromosom terbaik.

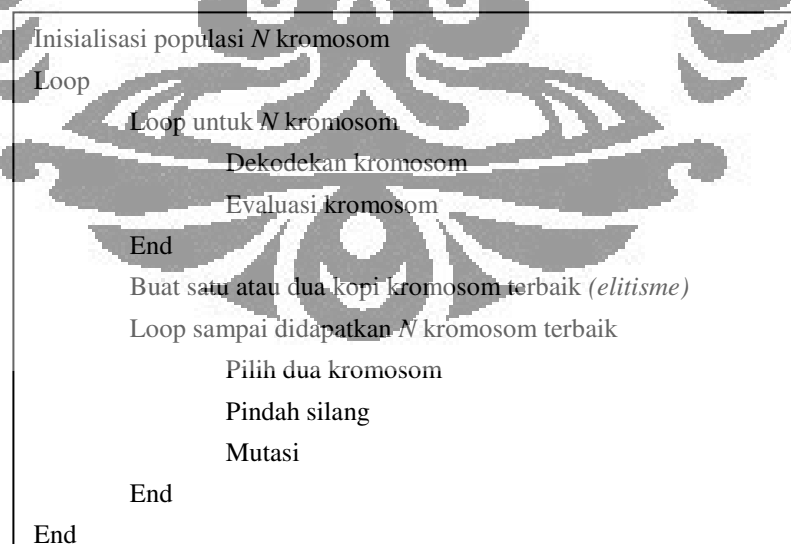
2.3.2 Komponen Utama Algoritma Genetik

Secara umum, diagram alir algoritma genetik sederhana seperti terlihat pada gambar 2.1 berikut ini :



Gambar 2.1 Diagram Alir Algoritma Genetik Sederhana

Untuk memudahkan pemahaman alur algoritma genetik, algoritma genetik sederhana mempunyai *pseudocode* seperti pada gambar 2.2 di bawah yang memperlihatkan langkah-langkah yang dilakukan.



Gambar 2.2 Pseudocode Algoritma Genetik

Ada beberapa komponen utama dalam algoritma genetik, yaitu prosedur inisialisasi, teknik pengkodean, fungsi evaluasi, seleksi, operator genetika, penentuan parameter, dan penggantian populasi. Operator genetika dapat terbagi menjadi operator pindah silang dan operator mutasi.

2.3.2.1 Prosedur Inisialisasi

Ukuran populasi tergantung pada masalah yang akan dipecahkan dan jenis operator genetika yang akan diimplementasikan. Setelah ukuran populasi ditentukan, kemudian harus dilakukan inisialisasi terhadap kromosom yang terdapat pada populasi tersebut. Inisialisasi kromosom dilakukan secara acak, namun demikian harus tetap memperhatikan domain solusi dan kendala permasalahan yang ada.

2.3.2.2 Teknik Pengkodean

Teknik pengkodean di sini meliputi pengkodean gen dan kromosom. Gen merupakan bagian dari kromosom. Satu gen biasanya akan mewakili satu variabel. Gen dapat direpresentasikan dalam bentuk : *string bit*, pohon, *array* bilangan real, daftar aturan, elemen permutasi, elemen program, atau representasi lainnya yang dapat diimplementasikan untuk operator genetika.

Yang akan dijelaskan lebih lanjut adalah *binary encoding*, yaitu setiap gen hanya bisa bernilai 0 dan 1. Skema pengkodean jenis *binary encoding* pada kromosom yang mempunyai 9 gen dapat dilihat pada gambar 2.3 di bawah.

| | | | | | | | | |
|----|----|----|----|----|----|----|----|----|
| 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 |
| g1 | g2 | g3 | g4 | g5 | g6 | g7 | g8 | g9 |

Gambar 2.3

Skema

pengkodean *binary encoding*

2.3.2.3 Fungsi Evaluasi (Nilai Fitness)

Ada dua hal yang harus dilakukan dalam melakukan evaluasi kromosom, yaitu: evaluasi fungsi objektif (fungsi tujuan) dan konversi fungsi objektif ke

Universitas Indonesia

dalam fungsi *fitness*. Secara umum, fungsi *fitness* diturunkan dari fungsi objektif dengan nilai yang tidak negatif. Apabila ternyata fungsi-fungsi objektif mempunyai nilai yang negatif, maka perlu ditambahkan suatu konstanta C agar nilai *fitness* yang terbentuk menjadi tidak negatif.

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

$$f = \frac{1}{(h+a)} \quad (2.1)$$

Di mana a adalah bilangan yang dianggap sangat kecil dan bervariasi sesuai dengan masalah yang akan diselesaikan.

2.3.2.4 Seleksi

Seleksi akan menentukan individu-individu *parent* mana saja yang akan dipilih untuk dilakukan operator genetika. Langkah pertama yang dilakukan dalam seleksi adalah pencarian nilai *fitness*. Pemilihan dua buah kromosom *parent* yang akan dilakukan operator genetika (pindah silang dan mutasi), biasanya dilakukan secara proporsional sesuai dengan nilai *fitness*nya. Beberapa metode yang digunakan untuk melakukan seleksi adalah : *Rank-based fitness assignment*, *Roulette wheel selection*, *Stochastic universal sampling*, *Local selection*, *Truncation selection*, dan *Tournament selection*.

Metode seleksi yang akan dibahas lebih lanjut adalah *rank-based fitness*, yaitu populasi diurutkan sesuai dengan nilai objektifnya. Nilai *fitness* dari tiap-tiap individu hanya tergantung pada posisi individu tersebut dalam urutan, dan tidak dipengaruhi oleh nilai objektifnya. Mekanisme ini bertujuan untuk melakukan penskalaan nilai-nilai *fitness*. Ada dua cara melakukan penskalaan nilai *fitness* yaitu *Linear fitness ranking* yang mengurutkan nilai *fitness* secara linear dan *Non-linear fitness ranking* yang mengurutkan nilai *fitness* sebagai akar polinomial.

Pada mekanisme *Linear fitness ranking*, individu bernilai *fitness* tertinggi diberi nilai *fitness* N (jumlah individu dalam populasi). Individu bernilai *fitness* tertinggi kedua diberi nilai *fitness* $N-1$, dan seterusnya sehingga individu bernilai *fitness* terendah diberi nilai *fitness* 1. Misalkan $R(i)$ menyatakan ranking individu ke i , $R(i) = N$ jika i adalah individu yang bernilai *fitness* terendah, maka nilai *fitness* yang baru adalah :

$$f(i) = (N + 1 - R(i)) \quad (2.2)$$

2.3.2.5 Pindah Silang (*Crossover*)

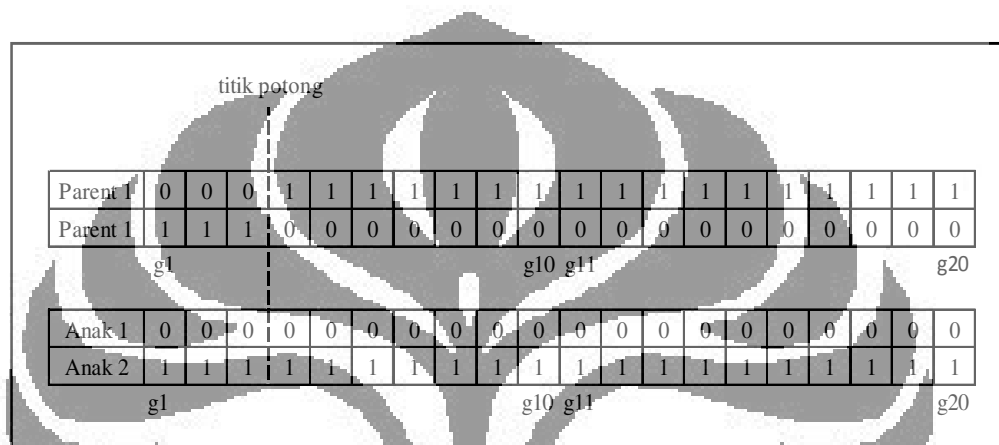
Salah satu komponen paling penting dalam algoritma genetik adalah *crossover* atau pindah silang. Sebuah kromosom yang mengarah pada solusi yang bagus bisa diperoleh dari proses memindahsilangkan dua buah kromosom.

Pindah silang dapat berakibat buruk jika ukuran populasinya sangat kecil. Dalam suatu populasi yang sangat kecil, suatu kromosom dengan gen-gen yang mengarah ke solusi akan sangat cepat menyebar ke kromosom-kromosom lainnya. Untuk mengatasi masalah ini digunakan suatu aturan bahwa pindah silang hanya bisa dilakukan dengan satu probabilitas tertentu p_c . Artinya, pindah silang bisa dilakukan hanya jika suatu bilangan random antara 0 sampai 1 yang dibangkitkan kurang dari p_c yang ditentukan. Pada umumnya, p_c diset mendekati 1, misalnya 0.8.

Pindah silang bisa dilakukan dalam beberapa cara yaitu pindah silang satu titik potong (*one-point crossover*), pindah silang banyak titik (*multi-point*

crossover atau *n-point crossover*), penyilangan seragam (*uniform crossover*), dan penyilangan dengan permutasi (*permutation crossover*).

Proses pindah silang paling sederhana adalah pindah silang satu titik potong (*one-point crossover*). Suatu bilangan titik potong dipilih secara random, kemudian bagian pertama dari *parent 1* digabungkan dengan bagian kedua dari *parent 2*, seperti yang terlihat pada gambar 2.4.



Gambar 2.4 Contoh Proses Pindah Silang Satu Titik Potong (*one-point crossover*).

Untuk kromosom yang sangat panjang, misalnya 1000 gen, mungkin saja diperlukan beberapa titik potong. Pindah silang lebih dari satu titik potong disebut *n-point crossover*, di mana *n* titik potong dipilih secara random dan bagian-bagian kromosom dipilih dengan probabilitas 0.5 dari salah satu orang tuanya. Satu skema pindah silang lainnya adalah *uniform crossover*, yang merupakan kasus khusus dari *n-point crossover* di mana *n* sama dengan jumlah gen dalam kromosom dikurangi satu.

2.3.2.6 Mutasi

Setelah mengalami proses pindah silang, pada *offspring* dapat dilakukan mutasi. Variabel *offspring* dimutasi dengan menambahkan nilai random yang sangat kecil (ukuran langkah mutasi), dengan probabilitas yang rendah. Peluang mutasi (p_m) didefinisikan sebagai presentasi dari jumlah total gen pada populasi

yang mengalami mutasi. Peluang mutasi mengendalikan banyaknya gen baru yang akan dimunculkan untuk dievaluasi. Jika peluang mutasi terlalu kecil, banyak gen yang mungkin berguna tidak pernah dievaluasi. Tetapi bila peluang mutasi ini terlalu besar, maka akan terlalu banyak gangguan acak, sehingga anak akan kehilangan kemiripan dari induknya, dan juga algoritma akan kehilangan kemampuan untuk belajar dari histori pencarian.

Biasanya p_m diset sebagai $1/n$, di mana n adalah jumlah gen dalam kromosom. Dengan p_m sebesar ini, berarti mutasi hanya terjadi pada sekitar satu gen saja. Pada algoritma sederhana, nilai p_m adalah tetap selama evolusi. Gambar 2.5 mengilustrasikan proses mutasi yang terjadi pada gen ke-10 pada kromosom yang mempunyai jumlah gen g_{20} .

| | | | | | | | | | | | | | | | | | | | | |
|---------------|----|---|---|-----|---|---|---|---|---|---|-----|---|---|---|---|---|---|---|---|--|
| Kromosom asal | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | |
| | g1 | | | g10 | | | | | | | g20 | | | | | | | | | |
| Hasil mutasi | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | |
| | g1 | | | g10 | | | | | | | g20 | | | | | | | | | |

Gambar 2.5 Contoh Proses Mutasi

Ada beberapa pendapat mengenai laju mutasi ini. Ada yang berpendapat bahwa, laju mutasi sebesar $1/n$ akan memberikan hasil yang cukup baik. Ada juga yang beranggapan bahwa laju mutasi tidak tergantung pada ukuran populasinya. Kromosom hasil mutasi harus diperiksa, apakah masih berupa domain solusi, dan bila perlu bisa dilakukan perbaikan.

2.3.2.7 Penentuan Parameter

Yang disebut dengan parameter disini adalah parameter kontrol algoritma genetika, yaitu ukuran populasi (U_{kpop}), peluang *crossover* (p_c), dan peluang mutasi (p_m). Nilai parameter ini ditentukan juga berdasarkan permasalahan yang akan dipecahkan. Selain itu, penentuan fungsi *stopping criteria* atau keadaan di mana proses *loop* akan berhenti.

2.3.2.8 Penggantian Populasi

Dalam algoritma genetika dikenal skema penggantian populasi yang disebut *generational replacement*, yang berarti semua individu (misal N individu dalam satu populasi) dari suatu generasi digantikan sekaligus oleh N individu baru hasil pindah silang dan mutasi. Skema penggantian ini tidak realistis dari sudut pandang biologi. Di dunia nyata, individu-individu dari generasi berbeda bisa berada dalam waktu yang bersamaan. Fakta lainnya adalah individu-individu muncul dan hilang secara konstan, tidak pada generasi tertentu. Secara umum, skema penggantian populasi dapat dirumuskan berdasarkan suatu ukuran yang disebut *generation gap* (G). Ukuran ini menunjukkan presentase populasi yang digantikan dalam setiap generasi. Pada skema *generation replacement*, $G = 1$.

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

2.4 DESIGN OF EXPERIMENT

Design of experiment atau perancangan percobaan (DOE) merupakan ilmu statistik yang banyak digunakan oleh banyak industri di dunia. DOE adalah teknik ampuh yang melibatkan proses perencanaan dan pendesainan suatu percobaan sehingga data yang tepat bisa dikumpulkan dan diolah secara statistik yang pada akhirnya dapat ditarik kesimpulan yang valid³.

³ Antony et al, 2001, hal 51

Tujuan dari dilakukannya perancangan percobaan adalah :

1. Menentukan variabel paling berpengaruh pada output.
2. Menentukan nilai optimum variabel x agar dicapai nilai y yang ideal.
3. Menentukan nilai optimum variabel x agar variansi nilai y minimum.
4. Menentukan nilai optimum variabel x agar pengaruh dari faktor yang tidak dapat dikendalikan z_1, z_2, \dots, z_q minimum.

2.4.1 Prinsip Dasar dalam *Design of Experiment*

Tiga prinsip dasar dalam melakukan perancangan percobaan adalah *replication*, *blocking*, dan *randomization*. Dua prinsip awal bertujuan untuk meningkatkan keakuratan percobaan, dan prinsip yang terakhir bertujuan mengurangi terjadinya *bias*.

- *Replication* (Replikasi)
 Dengan melakukan replikasi berarti kita mengulangi percobaan beberapa kali. Contohnya, apabila kita menguji 5 buah bahan percobaan pada satu media tertentu, berarti kita memiliki 5 replikasi. Replikasi mempunyai dua peranan penting. Pertama, orang yang melakukan percobaan dapat memperoleh *error*. Kedua, replikasi juga berguna untuk mendapatkan perkiraan percobaan yang lebih akurat.
- *Blocking*
Blocking adalah cara untuk meningkatkan keakuratan dari sebuah percobaan⁴. Dengan memblok, kita membagi percobaan kedalam kelompok atau grup. Sistem blok diberlakukan karena ada kemungkinan terjadinya perbedaan nilai akhir yang cukup jauh apabila percobaan tersebut tidak dikelompokkan.
- *Randomization* (Randomisasi)
 Tujuan melakukan randomisasi adalah untuk menghindari terjadinya *bias*. Dengan randomisasi, percobaan dilakukan secara acak. Metode statistik harus dilakukan dengan melakukan percobaan yang terdistribusi secara

⁴ Montgomery, 1997, hal 13

acak. Dengan melakukan hal ini, kita bisa mencegah terjadinya efek luar yang dapat mempengaruhi hasil percobaan. Apabila kita tidak melakukan randomisasi, maka ada kemungkinan percobaan tersebut bisa dipengaruhi oleh faktor lingkungan, kelelahan operator, dan kelainan material yang digunakan, dll.

2.4.2 Langkah-langkah Percobaan

Langkah-langkah dalam melakukan percobaan adalah sebagai berikut⁵ :

1. Mempersiapkan percobaan (*plan the experiment*)

Tahap ini kerja sama dilakukan oleh orang yang ahli dalam DOE dan orang yang berpengalaman dalam masalah yang akan diamati. Tahap persiapan terdiri dari beberapa bagian :

- Mengidentifikasi variabel *input* dan *output*.
- Menterjemahkan variabel *output* ke dalam suatu hal yang bisa diukur secara kuantitatif.
- Menentukan faktor-faktor yang berpengaruh terhadap hasil akhir.
- Menentukan jumlah level atau nilai untuk tiap faktor dan level apa saja yang harus diujicobakan.
- Mengidentifikasi kemungkinan terjadinya interaksi/hubungan antara faktor.

2. Merancang percobaan (*design the experiment*)

Dalam merancang percobaan dipilih tipe rancangan apa yang akan dipakai, apakah *full factorial* atau *fractional factorial*. Ketika memutuskan untuk menggunakan *fractional factorial*, kita harus menentukan pula beberapa banyak *fractioning*/pengurangan yang sesuai. Hal lain yang juga harus diperhatikan adalah apakah perlu diberlakukan sistem *blocking* atau tidak.

3. Menjalankan percobaan (*perform the experiment*)

Tahap ini menjalankan percobaan tersebut untuk mendapatkan data untuk diolah nantinya. Yang perlu diperhatikan adalah bahwa percobaan tersebut

⁵ Berger dan Murer (2002, hal 3)

harus dilakukan secara *random*/acak untuk mendapatkan hasil yang akurat dan menghindari terjadinya *bias*.

4. Analisis data dari hasil percobaan (*analyze data from the experiment*)

Dalam menganalisa data, kita perlu melakukan analisis secara statistik, antara lain dengan melakukan pengujian hipotesis sehingga kesimpulan yang didapatkan lebih valid dan akurat. Melalui analisis secara statistik, kita bisa mengetahui faktor mana yang berpengaruh dalam suatu proses dan mengetahui konsistensi suatu proses. Metode statistik yang biasa dipakai dalam DOE adalah *analysis of variance* (ANOVA),

Terdapat banyak piranti lunak (*software*) yang baik untuk membantu dalam analisis secara statistik. Metode grafik juga bisa dipakai untuk mendapatkan interpretasi hasil yang lebih baik dan menarik.

5. Mengkonfirmasi hasil percobaan (*confirm the result of the experiment*)

Setelah didapatkan kesimpulan percobaan, ada baiknya apabila kita melakukan verifikasi terhadap kesimpulan tersebut. Verifikasi berarti melakukan percobaan kembali untuk pembuktian akan kesimpulan yang kita dapatkan. Apabila hasil verifikasi kita sesuaikan dengan kenyataan yang ada, maka dapat dikatakan bahwa percobaan serta model matematis yang kita buat adalah valid.

6. Mengevaluasi kesimpulan percobaan (*evaluate the conclusion of the experiment*)

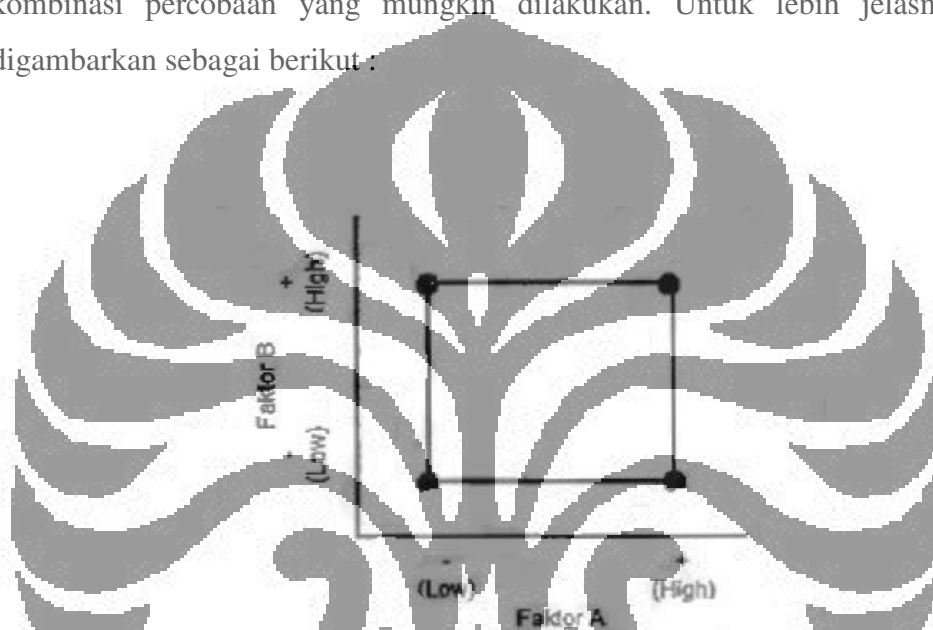
Langkah terakhir adalah mengevaluasi keseluruhan percobaan yang kita lakukan. Evaluasi ini penting untuk dipertimbangkan apakah percobaan akan perlu terus dilakukan untuk masalah-masalah berikutnya atau untuk melihat apakah dari sisi ekonomi percobaan ini mungkin dilakukan kembali atau tidak.

2.4.3 Factorial Design

Terdapat banyak percobaan yang melibatkan 2 atau lebih faktor. Umumnya, *factorial design* adalah suatu jenis percobaan yang paling efektif dan efisien untuk mengatasi hal ini. Dengan melakukan *factorial design*, berarti kita

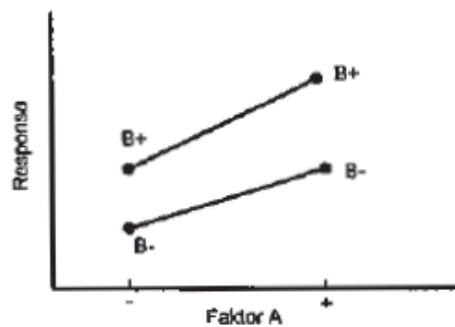
melakukan percobaan utuh dari setiap kombinasi level yang mungkin dari faktor yang kita coba. Sebagai contoh, apabila terdapat a level dari faktor A dan b level dari faktor B, maka setiap percobaan memerlukan kombinasi ab .

Level sendiri menggambarkan tingkatan nilai dalam suatu faktor. Biasanya dinotasikan dengan -1 (nilai rendah) atau 1 (nilai tinggi). Apabila terdapat 2 level dengan 2 faktor yang ingin dilakukan eksperimen, maka perancangan terbaik yang dapat digunakan adalah 2^2 factorial design. Dengan demikian, terdapat 4 kombinasi percobaan yang mungkin dilakukan. Untuk lebih jelasnya akan digambarkan sebagai berikut :

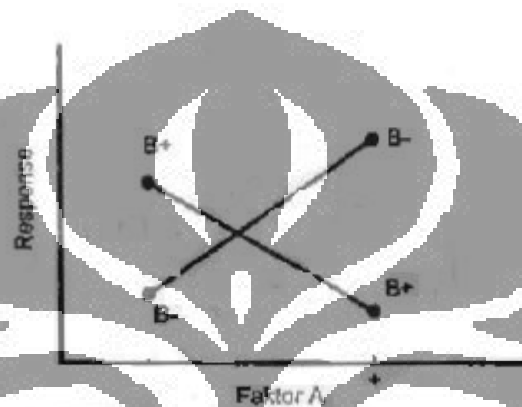


Gambar 2.6 Contoh Perancangan 2^2 factorial design

Ada istilah yang disebut *main effects*, atau faktor utama, yaitu pengaruh masing-masing faktor dalam suatu percobaan. Apabila kita memainkan/merubah nilai dari faktor utama tersebut. Maka dapat terjadi perubahan secara signifikan pada hasil yang keluar. Terjadinya interaksi apabila ada perbedaan hasil diantara level dalam satu faktor yang tidak sama dengan level dari faktor lain. Lebih mudahnya akan digambarkan dalam suatu grafik di bawah ini :



Gambar 2.7 Perubahan *Factorial* tanpa Interaksi



Gambar 2.8 Perubahan *Factorial* dengan Interaksi

Pada gambar 2.7, dapat dilihat bahwa faktor A dengan 2 level apabila dikombinasikan dengan faktor B yang juga memiliki 2 level akan menghasilkan suatu keadaan yang paralel. Akan tetapi, apabila kita perhatikan gambar 2.8, faktor A dengan level + bisa dikondisikan dengan faktor B dengan level +, menghasilkan respon yang justru berkurang dibandingkan apabila faktor A+ dikombinasikan dengan B-. Sebaliknya, apabila faktor A- dikombinasikan dengan B+ juga menghasilkan respon yang mirip dengan faktor A+B-. Dengan kata lain, penentuam kedua faktor tersebut pada level yang tinggi tidak menjamin terjadinya kenaikan respon. Hal ini dikarenakan adanya interaksi diantara faktor-faktor tersebut.

BAB III

PENGUMPULAN DATA

3.1 PROFIL STPI

Sekolah Tinggi Penerbangan Indonesia (STPI) merupakan salah satu perguruan tinggi kedinasan yang berada di bawah Departemen Perhubungan Republik Indonesia. STPI terletak di kecamatan Legok Kewedanaan Curug kabupaten Tangerang provinsi Banten.

STPI Curug memiliki tugas dan fungsi mendidik putra putri terbaik bangsa Indonesia untuk menjadi sumber daya manusia yang ahli dan terampil di bidang penerbangan, yang diakui secara nasional maupun internasional.

Dalam melaksanakan tugas dan fungsinya STPI Curug memiliki 4 (empat) jurusan pendidikan, yaitu Jurusan Penerbang, Jurusan Teknik Penerbangan, Jurusan Keselamatan Penerbangan dan Jurusan Manajemen Penerbangan. Setiap jurusan pendidikan terbagi dalam beberapa program studi, sesuai dengan minat dan bakat peserta pendidikan dan pelatihan.

3.2 PENGUMPULAN DATA PENELITIAN

Data yang digunakan dalam penelitian ini merupakan data sekunder yang diperoleh dari Sekolah Tinggi Penerbangan Indonesia (STPI). Data yang digunakan untuk penelitian mengenai penjadwalan pemeliharaan pesawat terdiri dari :

1. Data pesawat
2. Data pemeliharaan
3. Data jurusan penerbang
4. Data pemakaian pesawat 2008-2009
5. Data perawatan pesawat 2008-2009

3.2.1 Data Pesawat

Data pesawat yang diperoleh adalah data jenis pesawat, tipe registrasi, *total time in air*, *time since overhaul* (TSO). Setiap pesawat memiliki umur jam masing-masing, *total time in air* maksudnya adalah umur jam penggunaan pesawat mulai dari pesawat itu pertama kali digunakan, *time since overhaul* adalah umur jam pesawat setelah overhaul 2000 jam, artinya pesawat tersebut akan kembali ke-0 jam setelah dilakukan inspeksi 2000 jam (untuk jenis-jenis inspeksi dapat di lihat pada 3.2.2). Tabel 3.1 adalah daftar dan umur pesawat di STPI.

Tabel 3.1 Daftar Pesawat Dan Umur Pesawat Januari 2008

| NO | TYPE OF REGISTRATION | TOTAL TIME IN AIR (HRS) | TSO (HRS) |
|-----------------------|----------------------|-------------------------|------------|
| SOCATA TB-10 | | | |
| 1 | PK-AGI | 2000 | 0:00:00 |
| 2 | PK-AGJ | 2001 | 0:00:00 |
| 3 | PK-AGK | 2000 | 0:00:00 |
| 4 | PK-AGL | 3997 | 0:00:00 |
| 5 | PK-AGM | 2001 | 0:00:00 |
| 6 | PK-AGN | 1996 | 0:00:00 |
| 7 | PK-AGP | 1680 | 675:00:00 |
| 8 | PK-AGS | 1996 | 0:00:00 |
| 9 | PK-AGQ | 1800 | 700:00:00 |
| 10 | PK-AGT | 1988 | 0:00:00 |
| 11 | PK-AGU | 1999 | 0:00:00 |
| 12 | PK-AGV | 1999 | 0:00:00 |
| 13 | PK-AMN | 1999 | 0:00:00 |
| 14 | PK-AMO | 2000 | 0:00:00 |
| 15 | PK-AMP | 2000 | 0:00:00 |
| SUNDOWNER C-23 | | | |
| 16 | PK-ANN | 10541 | 800:00:00 |
| 17 | PK-ANQ | 7081 | 1175:00:00 |
| 18 | PK-ANR | 13709 | 1850:00:00 |
| 19 | PK-ANT | 12666 | 875:00:00 |
| 20 | PK-ANW | 10676 | 1200:00:00 |
| 21 | PK-ANX | 11293 | 1000:00:00 |
| DAKOTA P-28 | | | |
| 22 | PK-ADA | 5300 | 1300:00:00 |
| 23 | PK-ADG | 7650 | 1650:00:00 |
| 24 | PK-ADK | 6000 | 0:00:00 |

Sumber : STPI

3.2.2 Data Pemeliharaan

Yang dimaksud data pemeliharaan disini adalah data divisi pemeliharaan dan jenis-jenis inspeksi pada pesawat.

Tabel 3.2 Data Pekerjaan Divisi Pemeliharaan

| NO | LAPORAN DATA | DATA | KETERANGAN |
|----|----------------------|---------------|------------|
| 1 | JUMLAH MEKANIK | 15 ORANG | |
| 2 | HARI KERJA | SENIN - JUMAT | |
| 3 | WAKTU KERJA SHIFT 1 | 05:00 - 13:00 | 5 ORANG |
| 4 | WAKTU KERJA SHIFT 2 | 10:00 - 18:00 | 10 ORANG |
| 5 | WAKTU KERJA INSPEKSI | 07:00 - 18:00 | |
| 6 | JUMLAH PESAWAT | 24 UNIT | |

Sumber : STPI

Jumlah setiap shift adalah 5 orang dimana setiap pesawat yang di inspeksi dikerjakan oleh 5 orang mekanik. Di STPI memiliki 15 mekanik, artinya ada 3 team yang akan bekerja.

Tabel 3.3 Data Inspeksi Pesawat

| NO | JENIS INSPEKSI | WAKTU Pengerjaan | KETERANGAN |
|----|----------------|------------------|---|
| 1 | 50 JAM | 5 JAM | |
| 2 | 100 JAM | 10 JAM | |
| 3 | 300 JAM | 10 JAM | |
| 4 | 400 JAM | 10 JAM | |
| 5 | 500 JAM | 10 JAM | |
| 6 | 1000 JAM | 10 JAM | |
| 7 | 2000 JAM | 6 MINGGU | TOTAL 4 BULAN (INCLUDE WAITING PART) |

Sumber : STPI

Pada setiap 50 jam terbang, pesawat akan masuk inspeksi hingga mencapai 2000 jam dan *time since overhaul* kembali ke-0 jam lagi. Inspeksi 2000 jam untuk pengerjaannya membutuhkan waktu 6 minggu, tetapi waktu pengerjaan ditambah *waiting part* menjadi 4 bulan.

3.2.3 Data Penerbang

Berbicara masalah pemeliharaan pesawat tidak lepas dari penggunaan pesawat, karena untuk pemeliharaan pesawat ada yang namanya pemeliharaan *preventive* berdasarkan *flight hour* yaitu pemeliharaan pesawat berdasarkan jam terbang atau jam pemakaian pesawat. Di STPI pengguna pesawat adalah taruna yang sedang pendidikan di sana. Oleh karena itu data penerbang dibutuhkan untuk mengetahui kapan pesawat akan digunakan.

Tabel 3.4 Data Jurusan Penerbang

| NO | LAPORAN DATA | DATA |
|----|----------------------------------|---------------|
| 1 | JUMLAH TARUNA /Semester | 25 ORANG |
| 2 | HARI TERBANG | SENIN-JUMAT |
| 3 | WAKTU TERBANG | 07:00 - 17:00 |
| 4 | TOTAL JAM TERBANG (4 SEMESTER) | 162 JAM |
| 6 | WAKTU PERGANTIAN TARUNA /PESAWAT | 30 MENIT |

Sumber : STPI

| Bulan (x) | 1 | 2 | 3 | 4 | 5 | 6 |
|--------------------------|---------------|-----|-----|-----|-----|-----|
| Semester 1 | Ground School | | | | | |
| Semester 2 (jam) | 5 | 5 | 5 | 5 | 10 | 10 |
| Semester 3 (jam) | 10 | 17 | 18 | 18 | 15 | 10 |
| Semester 4 (jam) | 10 | 10 | 10 | 4 | | |
| Total (jam) | 25 | 32 | 33 | 27 | 25 | 20 |
| persemester (jam) | 625 | 800 | 825 | 675 | 625 | 500 |
| Kebutuhan Pesawat (unit) | 7 | 8 | 8 | 7 | 6 | 5 |

Gambar 3.1 Kebutuhan Pesawat per Minggu di Bulan x

Tabel 3.5 Silabus Penggunaan Pesawat Jurusan Penerbang

| | GROUND SCHOOL | | | HOLIDAY | GROUND SCHOOL | SIMULATION FLIGHT PREPARATION | |
|------------|-------------------|------------------|------------------|---------|------------------|-------------------------------|------------------|
| SEMESTER 1 | Bulan 1 | Bulan 2 | Bulan 3 | | Bulan 4 | Bulan 5 | Bulan 6 |
| | PRE SOLO | | | | | SOLO | |
| SEMESTER 2 | Bulan 1 / 5 jam | Bulan 2 / 5 jam | Bulan 3 / 5 jam | | Bulan 4 / 5 jam | Bulan 5 / 10 jam | Bulan 6 / 10 jam |
| | SOLO (NOV 12 HRS) | | PPL | HOLIDAY | PPL | | |
| SEMESTER 3 | Bulan 1 / 10 jam | Bulan 2 / 17 jam | Bulan 3 / 18 jam | | Bulan 4 / 18 jam | Bulan 5 / 15 jam | Bulan 6 / 10 jam |
| | PPL | | | | | CPL | |
| SEMESTER 4 | Bulan 1 / 10 jam | Bulan 2 / 10 jam | Bulan 3 / 10 jam | | Bulan 4 / 4 jam | Bulan 5 | Bulan 6 |

Sumber : STPI

Tabel 3.6 Data Penggunaan Pesawat

| Type of aircraft | 2008 | | | | | | | | | | | | Total |
|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|-------------------|------------------|-------------------|
| | Januari | Feb | Maret | April | Mei | Juni | Juli | Agust | Sept | Okt | Nov | Des | |
| Socata TB10 | 30:52:00 | 18:45:00 | 0:00:00 | 95:05:00 | 300:01:00 | 453:30:00 | 417:34:00 | 283:22:00 | 496:42:00 | 555:51:00 | 857:00:00 | 679:48:00 | 4188:30:00 |
| Sundowner C23 | 132:53:00 | 158:26:00 | 163:20:00 | 204:35:00 | 126:12:00 | 38:30:00 | 53:30:00 | 177:10:00 | 81:20:00 | 51:35:00 | 155:45:00 | 196:27:00 | 1539:43:00 |
| Dakota P28 | 140:40:00 | 3:10:00 | 0:00:00 | 0:00:00 | 0:00:00 | 45:25:00 | 215:30:00 | 116:25:00 | 31:25:00 | 0:00:00 | 61:40:00 | 25:45:00 | 640:00:00 |
| Total | 304:25:00 | 180:21:00 | 163:20:00 | 299:40:00 | 426:13:00 | 537:25:00 | 686:34:00 | 576:57:00 | 609:27:00 | 607:26:00 | 1074:25:00 | 902:00:00 | 6368:13:00 |

| Type of aircraft | 2009 | | | | | | | | | | | | Total |
|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|----------------|----------------|----------------|-------------------|
| | Januari | Feb | Maret | April | Mei | Juni | Juli | Agust | Sept | | | | |
| Socata TB10 | 634:36:00 | 485:25:00 | 461:41:00 | 634:39:00 | 295:14:00 | 364:21:00 | 99:10:00 | 127:20:00 | 245:21:00 | | | | 3347:47:00 |
| Sundowner C23 | 175:27:00 | 105:30:00 | 242:55:00 | 145:52:00 | 2:15:00 | 15:10:00 | 1:00:00 | 0:00:00 | 0:00:00 | | | | 688:09:00 |
| Dakota P28 | 0:00:00 | 10:50:00 | 9:09:00 | 27:00:00 | 0:00:00 | 1:30:00 | 0:00:00 | 0:00:00 | 0:00:00 | | | | 48:29:00 |
| Total | 810:03:00 | 601:45:00 | 713:45:00 | 807:31:00 | 297:29:00 | 381:01:00 | 100:10:00 | 127:20:00 | 245:21:00 | 0:00:00 | 0:00:00 | 0:00:00 | 4084:25:00 |

Sumber : STPI

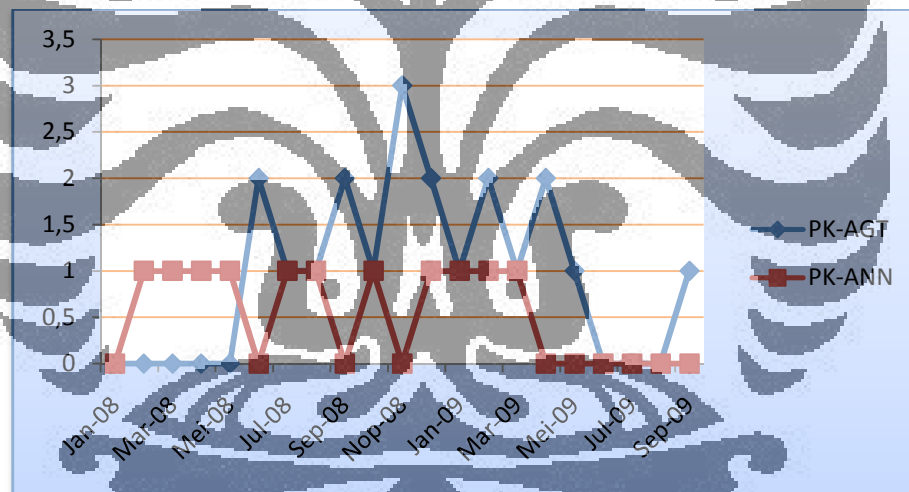
3.2.4 Data Penggunaan Pesawat

Data pemakaian pesawat dari Januari 2008 – September 2009. Data ini adalah data pemakaian pesawat berdasarkan pengelompokkan jenis pesawat per bulan dalam jam, yaitu Socata, Sundowner, Dakota.

Tabel 3.6 adalah hasil dari data setiap penggunaan pesawat per hari yang dilaporkan ke divisi pemeliharaan pesawat.

3.2.5 Data Pemeliharaan Pesawat

Data pemeliharaan pesawat dari Januari 2008 – September 2009. Data pemeliharaan ini sangat tergantung pada data pemakaian pesawat, karena setiap pesawat dipelihara setiap 50 jam setelah pesawat itu digunakan.



Gambar 3.2 Data Pemeliharaan Pesawat

Contoh pengambilan data pada pesawat PK-AGT dan PK-ANN, dimana terlihat pesawat PK-AGT pada bulan november 2008 mengalami inspeksi 3 kali dan PK-ANN pada bulan tersebut tidak mengalami inspeksi sama sekali.

BAB IV

PENGOLAHAN DATA DAN ANALISIS

4.1 FUNGSI TUJUAN

Penjadwalan pemeliharaan pesawat di STPI mengikuti *flight hour* yaitu pemeliharaan pesawat sesuai dengan jam pemakaian pesawat, artinya pemeliharaan tergantung dari pemakaian pesawat oleh mahasiswa STPI. Kendalanya adalah pada saat pemeliharaan 2000 jam terjadi penumpukan pesawat, maka dari itu fungsi tujuan dari penjadwalan pemeliharaan adalah meminimalkan terjadinya pemeliharaan 2000 jam secara bersamaan.

Variabel : C_{ij} = Umur pesawat i pada akhir periode j

$$i = 1, 2, 3, \dots, 24, j = 0, 1, 2, 3, \dots, 480$$

$X_{ij} = 1$, jika pesawat i pada periode j dipakai.

$= 0$, jika pesawat i pada periode j tidak dipakai.

$$i = 1, 2, 3, \dots, 24, j = 0, 1, 2, 3, \dots, 480$$

$C_{ij=0} = U_i$, umur awal dari pesawat i

$$C_{ij} = C_{ij-1} + 25 X_{ij} \quad (4.1)$$

Jika $(C_{ij}) / 50 = \text{Integer}$, maka $p_{ij} = 1$, pemeliharaan
 $p_{ij} = 0$, tidak pemeliharaan (4.2)

Jika $(C_{ij}) / 100 = \text{Integer}$, maka $q_{ij} = 1$, pemeliharaan
 $q_{ij} = 0$, tidak pemeliharaan (4.3)

Jika $(C_{ij}) / 100 = \text{Integer}$, maka $p_{ij} = 0$ (4.4)

Jika $(C_{ij}) / 2000 = \text{Integer}$, maka $r_{ij} = 1$, pemeliharaan
 $r_{ij} = 0$, tidak pemeliharaan (4.5)

Jika $r_{ij} = 1$, maka $\sum_j^{j+6} r_{ij} = 6$ (4.6)

$\sum_i 5p_{ij} + \sum_i 10q_{ij} + \sum_i 5r_{ij} \leq 450$ (4.7)

Fungsi Tujuan :

$$\text{f.t } \boxed{\min Z = \sum_i \sum_j r_{ij}} \quad (4.8)$$

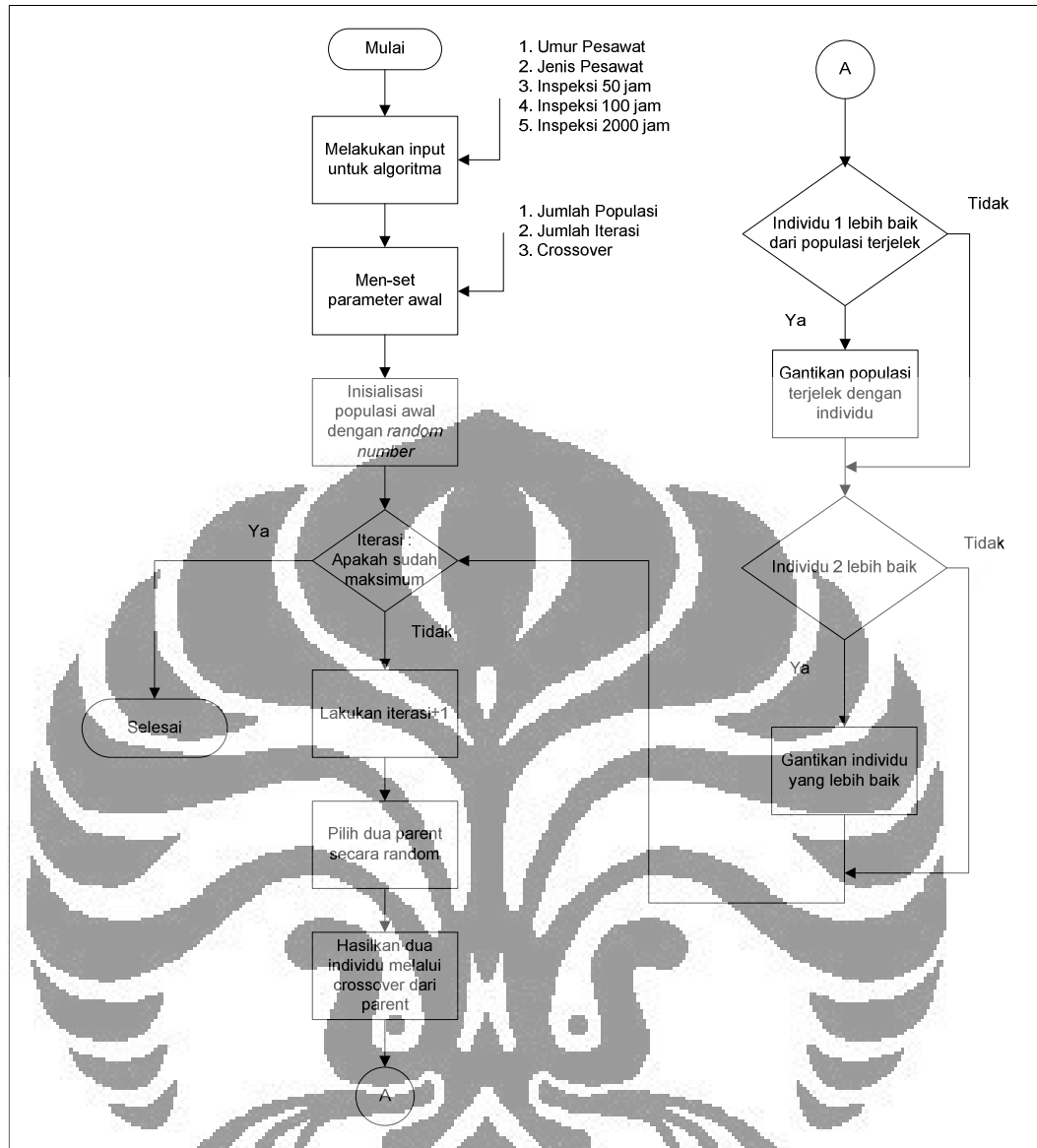
Umur pesawat akan dihitung pada setiap periode, ini untuk menentukan pesawat mana saja yang dapat digunakan pada periode berikutnya. Cara pembacaannya adalah :

1. Jika umur pesawat di akhir periode dibagi dengan 50 jam hasilnya integer, maka untuk periode berikutnya pesawat tersebut $P_{ij} = 1$ (pemeliharaan), $P_{ij} = 0$ (tidak pemeliharaan).
2. Jika umur pesawat di akhir periode dibagi dengan 100 jam hasilnya integer, maka untuk periode berikutnya pesawat tersebut $q_{ij} = 1$ (pemeliharaan), $q_{ij} = 0$ (tidak pemeliharaan).
3. Jika umur pesawat di akhir periode dibagi dengan 100 jam hasilnya integer, maka $P_{ij} = 0$.
4. Jika umur pesawat di akhir periode dibagi dengan 2000 jam hasilnya integer, maka untuk periode berikutnya pesawat tersebut $r_{ij} = 1$ (*overhaul*), $r_{ij} = 0$ (tidak *overhaul*).
5. Jika umur pesawat di akhir periode dibagi dengan 2000 jam hasilnya integer, maka $\sum_j^{j+6} r_{ij} = 6$

4.2 ALGORITMA GENETIK

Model penyelesaian penjadwalan pemeliharaan pesawat latih pada Sekolah Tinggi Penerbangan Indonesia menggunakan algoritma genetik, seperti yang telah dijelaskan pada bab 2. Berdasarkan tahapan algoritma penyelesaian tersebut, dibuat program komputer dengan bantuan software MATLAB 7.8.

Penjadwalan pemeliharaan pesawat terbang pada STPI akan menggunakan algoritma genetik yang dibuat berdasarkan fungsi optimasi yang meminimumkan terjadinya pemeliharaan 2000 jam pada saat yang bersamaan dengan menggunakan bahasa pemrograman MATLAB 7.8.



Gambar 4.1 Diagram Alir Algoritma Penjadwalan Perawatan Pesawat

MATLAB adalah bahasa komputasi teknis tingkat tinggi dan merupakan lingkungan yang interaktif untuk pengembangan algoritma, visualisasi data, analisis data, dan komputasi numerik. Konstruksi penyelesaian komputasi teknis dengan MATLAB dapat dilakukan lebih cepat dibandingkan dengan bahasa pemrograman tradisional seperti C, C++, dan Fortran. MATLAB menyediakan fungsi-fungsi matematis untuk aljabar linier, statistik, optimasi, dan lainnya. Selain itu, MATLAB juga menyediakan fitur-fitur dokumentasi dan integrasi algoritma berbasis MATLAB dengan bahasa dan aplikasi lain, seperti C, C++, Fortran, Java, COM, dan Microsoft Excel. Bahasa MATLAB memudahkan

operasi-operasi vektor dan matriks yang merupakan dasar bagi permasalahan di bidang teknik dan ilmiah⁶.

4.2.1 Langkah-langkah Penyusunan Algoritma Genetik

1. Menentukan parameter/variabel yang digunakan

Input yang dimasukkan adalah nomor pesawat, umur jam pesawat, inspeksi. Parameter yang digunakan dalam algoritma adalah ukuran populasi atau jumlah individu dalam populasi (U_{pop}), jumlah iterasi maksimum ($MaxG$), dan probabilitas terjadinya proses kawin silang/*crossover* (p_c).

- Ukuran populasi

Algoritma genetik bekerja dengan membentuk suatu populasi solusi bukan solusi tunggal, sehingga perlu ditetapkan ukuran populasi permasalahan. Menurut Suyanto (2005), ukuran populasi berkisar dari 30 sampai 1000. Dipilih 30 menjadi ukuran populasi terkecil karena dianggap sampel mengikuti distribusi normal.

- Probabilitas kawin silang

Merupakan parameter penting dalam algoritma genetik karena sebuah kromosom yang mengarah pada kualitas solusi yang baik bisa diperoleh dari proses kawin silang. Pada umumnya, P_c diset mendekati 1, misalnya 0,8.

- Maksimum itersi

2. Inisialisasi populasi awal dengan *random number*

Dalam permasalahan ini populasi yang telah ditentukan akan menghasilkan populasi sesuai dengan parameter yang di *input* pada awal program. Pada permasalahan penjadwalan pemeliharaan pesawat populasi yang diambil adalah 30, 50, dan 100.

3. Lakukan iterasi

Setiap populasi yang telah ditentukan di atas, akan dilakukan iterasi sebanyak yang kita inginkan, setiap populasi akan mengulang sebanyak

⁶ www.mathworks.com

iterasi yang di *input*. Funsinya untuk mendapatkan populasi yang terbaik dari proses iterasi tersebut.

Pada permasalahan penjadwalan pemeliharaan pesawat iterasi yang diambil adalah 1000, 5000, 10000.

4. Hasilkan dua individu melalui *crossover* dari *parent*

Dari beberapa individu yang dihasilkan, dipilih 2 individu terbaik untuk di kawin silang (*crossover*) yang akan menggantikan populasi yang terjelek.

Pada permasalahan penjadwalan pemeliharaan pesawat *crossover* yang diambil adalah 0,5 , 0,7 , 0,95.

4.2.2 Verifikasi dan Validasi

4.2.2.1 Verifikasi Program

Verifikasi adalah menjalankan program yang telah dibuat dalam MATLAB. Program dijalankan dengan memasukkan *input* dan menggunakan konfigurasi parameter.

Tabel 4.1 Kombinasi Parameter yang Digunakan dalam Verifikasi

| Parameter | Nilai |
|------------------------------------|-------|
| Ukuran populasi (Ukpop) | 100 |
| Iterasi maksimum (MaxG) | 5000 |
| probabilitas <i>crossover</i> (pc) | 0.95 |

Hasil dari *run* program adalah sebagai berikut :

JumlahTahun : 2

Indeks Populasi Terbaik: 1

StandarDeviasi (fitness) Terbaik: 0.28233

JumlahTambahanPesawat : 0

JumlahInspeksi2000 =

Columns 1 through 12

0 0 0 0 1 0 0 0 0 0 0 0

Columns 13 through 24

1 0 0 0 0 0 0 0 0 0 0 0

4.2.2.2 Validasi Program

Validasi adalah pembuktian apakah *script* program yang dibuat sudah berjalan sesuai dengan fungsinya dengan melakukan penghitungan secara manual dari hasil yang didapatkan pada verifikasi program. Jika hasil perhitungan manual dan peranti lunak tidak memiliki perbedaan, maka dapat dinyatakan bahwa model telah tervalidasi atau dapat berjalan sesuai dengan fungsinya. Hasil perhitungan dengan piranti lunak adalah :

Tabel 4.2 Hasil Validasi dengan Penghitungan Manual

| | |
|-------------------------|---------|
| Jumlah Tahun | 2 |
| Jumlah Inspeksi 2000 | 2 |
| Jumlah Pesawat Tambahan | 0 |
| Standard Deviasi | 0,28233 |

4.3 PERCOBAAN KOMBINASI PARAMETER

4.3.1 Persiapan dan Perancangan Percobaan

Untuk menentukan kombinasi yang tepat, dibutuhkan studi parameter untuk ketiga parameter (Ukpop, MaxG, Pc), sebagai faktor percobaan dengan membandingkan rata-rata hasil akhir (standard deviasi) dengan menggunakan *design of experiment*.

Analisis hasil studi parameter dilakukan dengan *design of experiment* jenis *factorial design* pada piranti lunak MINITAB bertujuan untuk :

1. Mencari apakah perubahan level dari ketiga faktor input parameter algoritma genetik mempunyai pengaruh yang signifikan terhadap output *run* program.
2. Mencari nilai level dari tiap faktor yang memberikan *output* dengan biaya minimum.

Prinsip dasar yang digunakan dalam percobaan adalah replikasi sebanyak 10 kali. Jumlah level yang diujicobakan untuk ketiga faktor adalah sebagai berikut:

1. Faktor ukuran populasi (Ukpop), faktor Ukpop dibagi menjadi tiga level yaitu 30, 50, 100.

Menurut (Suyanto,2005,hal 43) ukuran populasi biasanya berkisar antara 30 sampai 1000. Jika ukuran populasi terlalu kecil, algoritma genetik akan cepat konvergen disebabkan oleh rendahnya variasi pada kromosom-kromosom dalam populasi. Tetapi ukuran populasi yang terlalu besar akan menyebabkan proses algoritma genetik menjadi lambat.

2. Faktor Iterasi Maksimum (MaxG), faktor MaxG dibagi menjadi tiga level yaitu 1000, 5000, 10000.

Semakin banyak iterasi maka semakin besar peluang untuk mendapatkan solusi yang lebih baik, namun dapat menurunkan efisiensi pengolahan dari segi waktu.

3. Faktor probabilitas crossover (Pc), faktor P crossover dibagi menjadi tiga level yaitu 0.5, 0.7, 0.95.

Probabilitas *crossover* yang baik berada pada kisaran 0.5 sampai dengan 0.95⁷.

Dari ketiga faktor dan tiga level untuk tiap faktor, didapatkan jumlah kombinasi sebanyak $= 3 \times 3 \times 3 = 27$ kombinasi.

Tabel 4.3 Kombinasi Level Tiap Faktor Parameter

| | Ukpop | MaxG(iterasi) | Crossover |
|-------------|-------|---------------|-----------|
| Kombinasi 1 | 30 | 1000 | 0,5 |
| Kombinasi 2 | 30 | 1000 | 0,7 |
| Kombinasi 3 | 30 | 1000 | 0,95 |
| Kombinasi 4 | 30 | 5000 | 0,5 |
| Kombinasi 5 | 30 | 5000 | 0,7 |
| Kombinasi 6 | 30 | 5000 | 0,95 |
| Kombinasi 7 | 30 | 10000 | 0,5 |

⁷ Achmad Basuki, Strategi Menggunakan Algoritma Genetik, 2003

Tabel 4.3 Kombinasi Level Tiap Faktor Parameter (sambungan)

| | | | |
|--------------|-----|-------|------|
| Kombinasi 8 | 30 | 10000 | 0,7 |
| Kombinasi 9 | 30 | 10000 | 0,95 |
| Kombinasi 10 | 50 | 1000 | 0,5 |
| Kombinasi 11 | 50 | 1000 | 0,7 |
| Kombinasi 12 | 50 | 1000 | 0,95 |
| Kombinasi 13 | 50 | 5000 | 0,5 |
| Kombinasi 14 | 50 | 5000 | 0,7 |
| Kombinasi 15 | 50 | 5000 | 0,95 |
| Kombinasi 16 | 50 | 10000 | 0,5 |
| Kombinasi 17 | 50 | 10000 | 0,7 |
| Kombinasi 18 | 50 | 10000 | 0,95 |
| Kombinasi 19 | 100 | 1000 | 0,5 |
| Kombinasi 20 | 100 | 1000 | 0,7 |
| Kombinasi 21 | 100 | 1000 | 0,95 |
| Kombinasi 22 | 100 | 5000 | 0,5 |
| Kombinasi 23 | 100 | 5000 | 0,7 |
| Kombinasi 24 | 100 | 5000 | 0,95 |
| Kombinasi 25 | 100 | 10000 | 0,5 |
| Kombinasi 26 | 100 | 10000 | 0,7 |
| Kombinasi 27 | 100 | 10000 | 0,95 |

4.3.2 Hasil Percobaan Parameter

DOE untuk parameter diuji dengan menggunakan program MINITAB 14. Setelah dilakukan percobaan sebanyak 270 kali, maka diperoleh hasil yang dapat dilihat pada lampiran 4 dimana yang menjadi *output* adalah standard deviasi.

Dengan melihat hasil percobaan yang telah dilakukan, maka dibuat suatu kombinasi parameter yang terbaik tabel 4.4 yang mampu memberikan kualitas solusi terbaik untuk masalah penjadwalan pemeliharaan pesawat terbang ini.

Tabel 4.4 Kombinasi Parameter Terbaik

| Faktor | Nilai |
|--------------------------------|-------|
| Ukuran Populasi (Ukpop) | 100 |
| Maksimum Iteraksi (MaxG) | 5000 |
| Probabilitas kawin silang (Pc) | 0,95 |

4.3.3 Susunan Penjadwalan yang Terbaik

Berdasarkan hasil pengolahan alternatif kombinasi parameter di atas, dicari susunan rotasi penjadwalan pemakaian pesawat yang terbaik dengan standard deviasi pemeliharaan 2000 jam yang minimum. Kombinasi yang digunakan adalah kombinasi ke-24 yaitu Ukpop 100, MaxG 5000, Pc 0.95. *Run* program dilakukan sebanyak 10 kali dan didapatkan solusi terbaik dengan standard deviasi 0.43483. Hasil pengolahan data untuk susunan rotasi penjadwalan pemeliharaan pesawat terbaik dalam bentuk grafik dapat dilihat Gambar 4.4 serta hasil pengolahannya adalah sebagai berikut :

JumlahInspeksi2000 =

Columns 1 through 20

0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0

Columns 21 through 40

1 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0

Columns 41 through 60

1 1 0 0 0 0 0 0 1 0 1 0 0 0 1 0 0 1 0

Columns 61 through 80

1 1 1 1 0 1 1 1 0 1 0 0 1 0 1 1 0 0 1 0

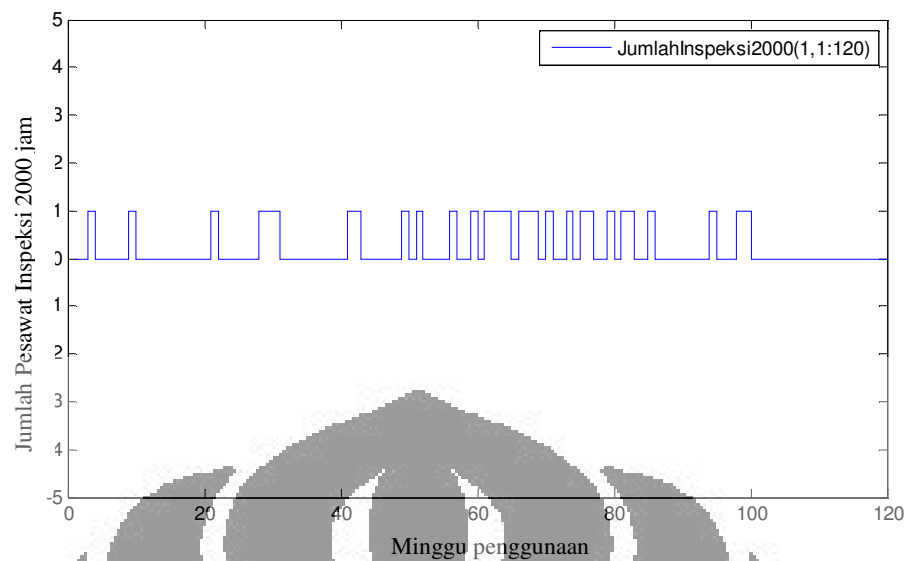
Columns 81 through 100

1 1 0 0 1 0 0 0 0 0 0 0 1 0 0 0 1 1 0

Columns 101 through 120

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

JumlahTambahanPesawat = 0



Gambar 4.2 Grafik Hasil Pengolahan Data

Berdasarkan dari hasil solusi optimum dari *run* program, dapat diketahui bahwa penjadwalan pemakaian pesawat yang dapat meminimumkan pemeliharaan 2000 jam adalah seperti Tabel 4.6 rotasi pemakaian pesawat terbaik, dimana hanya diperlihatkan selama tahun pertama, untuk tahun kedua sampai dengan kesepuluh terlihat di lampiran 2.

Tabel 4.6 Rotasi Pemakaian Pesawat Terbaik

| Tahun | sem | Bulan | Urutan Pesawat | Jumlah |
|-------------------|-----|-------|---|--------|
| 2008 / 2009 | 1 | Apri | 1 20 21 16 7 14 9 13 4 21 24 16 15 12 4 11 15 20 3 19 13 9 22 10 11 17 18 21 | 28 |
| | | Mei | 22 2 12 3 18 6 5 24 23 20 23 9 9 4 2 5 7 15 14 21 8 3 18 18 24 15 20 7 12 22 6 23 | 32 |
| | | Juni | 14 5 16 5 21 11 21 22 4 24 15 17 16 12 3 8 13 2 9 5 18 22 1 20 2 18 24 7 23 19 6 19 | 32 |
| | | Juli | 21 13 10 11 6 12 17 7 3 24 8 15 4 14 3 10 21 2 13 13 23 15 19 23 15 22 10 1 | 28 |
| | | Agst | 1 7 17 13 12 6 16 17 2 8 19 4 11 9 10 23 13 24 20 15 22 5 19 23 | 24 |
| | | Sept | 6 16 4 14 22 10 2 11 14 4 8 21 23 12 20 16 23 8 6 9 | 20 |
| | 2 | Okt | 17 21 4 7 9 15 1 23 2 22 20 6 17 3 9 12 22 11 23 7 10 13 5 4 2 11 19 8 | 28 |
| | | Nov | 18 24 13 20 15 10 21 20 5 8 19 22 22 15 17 4 24 19 18 2 21 13 20 12 9 8 13 24 1 16 15 17 | 32 |
| | | Des | 20 4 18 18 23 23 15 10 6 10 14 21 2 11 1 1 17 20 8 2 6 13 17 4 12 13 11 15 15 1 20 14 | 32 |
| | | Jan | 19 4 21 19 3 10 6 8 13 22 7 22 10 15 17 11 17 21 1 9 8 4 18 12 4 19 11 7 | 28 |
| | | Feb | 18 17 10 2 6 14 17 2 8 13 1 4 14 4 15 21 16 5 7 17 5 19 1 18 | 24 |
| | | Mar | 15 3 9 18 15 19 14 8 22 24 11 21 1 6 17 9 22 5 2 9 | 20 |

4.4 ANALISA

4.4.1 Analisa Percobaan Parameter

Hasil dari pengolahan data pada peranti lunak MINITAB adalah sebagai berikut.

22/12/2009 13:01:41

Welcome to Minitab, press F1 for help.

Results for: Worksheet 2

Multilevel Factorial Design

Factors: 3 Replicates: 10
 Base runs: 27 Total runs: 270
 Base blocks: 1 Total blocks: 1
 Number of levels: 3; 3; 3

22/12/2009 18:03:47

Welcome to Minitab, press F1 for help.
 Retrieving project from file: 'C:\Program Files\MINITAB 14\Data\skripsi
 std.
 deviasi new.MPJ'

Results for: Worksheet 2

General Linear Model: std.deviasi versus Ukpap; MaxG; Pc

| Factor | Type | Levels | Values |
|--------|-------|--------|---------|
| Ukpap | fixed | 3 | 1; 2; 3 |
| MaxG | fixed | 3 | 1; 2; 3 |
| Pc | fixed | 3 | 1; 2; 3 |

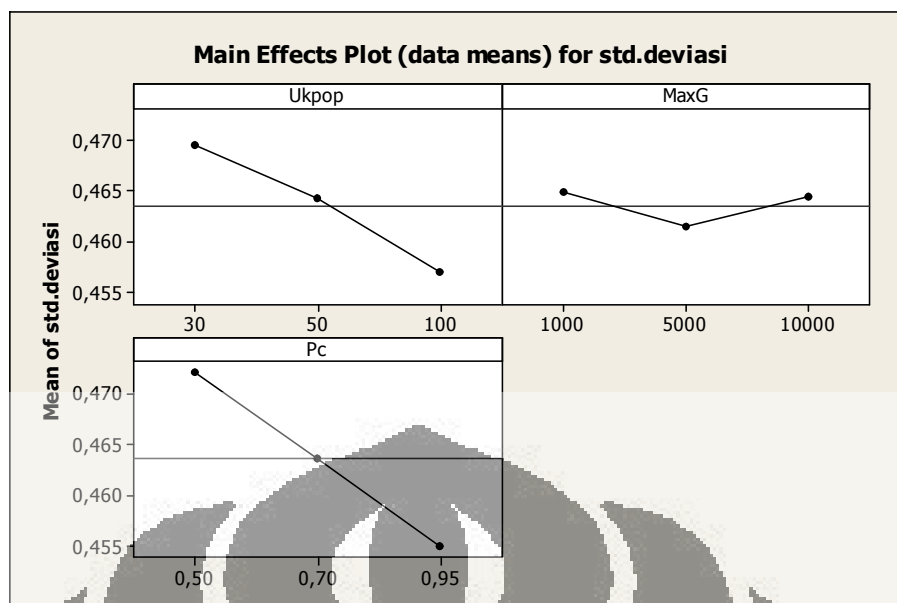
Analysis of Variance for std.deviasi, using Adjusted SS for Tests

| Source | DF | Seq SS | Adj SS | Adj MS | F | P |
|---------------|-----|-----------|-----------|-----------|-------|-------|
| Ukpap | 2 | 0,0070650 | 0,0070650 | 0,0035325 | 24,52 | 0,000 |
| MaxG | 2 | 0,0006393 | 0,0006393 | 0,0003196 | 2,22 | 0,111 |
| Pc | 2 | 0,0133567 | 0,0133567 | 0,0066784 | 46,36 | 0,000 |
| Ukpap*MaxG | 4 | 0,0007121 | 0,0007121 | 0,0001780 | 1,24 | 0,296 |
| Ukpap*Pc | 4 | 0,0028930 | 0,0028930 | 0,0007233 | 5,02 | 0,001 |
| MaxG*Pc | 4 | 0,0002678 | 0,0002678 | 0,0000669 | 0,46 | 0,762 |
| Ukpap*MaxG*Pc | 8 | 0,0038566 | 0,0038566 | 0,0004821 | 3,35 | 0,001 |
| Error | 243 | 0,0350051 | 0,0350051 | 0,0001441 | | |
| Total | 269 | 0,0637956 | | | | |

Pada *analysis of variante* (ANOVA), nilai P-value tiap faktor berfungsi untuk mengetahui apakah perubahan level dari faktor tersebut mempunyai pengaruh yang signifikan terhadap *output* yaitu standard deviasi. Apabila angka P-value kurang dari 0,05 maka faktor tersebut mempunyai pengaruh yang signifikan terhadap output. P-value dari faktor Ukpops adalah 0,00 , MaxG 0,111 , Pc 0,00. Hal ini menunjukkan bahwa faktor Ukpops (populasi) dan Pc (probabilitas *crossover*) yang berpengaruh terhadap algoritma penyelesaian.

Nilai P-value pada interaksi faktor Ukpops dengan MaxG adalah 0,296, faktor Ukpops dengan Pc adalah 0,001, faktor MaxG dengan Pc adalah 0,762, dan interaksi antara ketiga faktor adalah 0,001. Nilai P di atas 0,05 menunjukkan tidak ada interaksi tiap level pada faktor yang dapat mempunyai pengaruh signifikan terhadap *output*. Jadi faktor yang menunjukkan adanya interaksi adalah faktor Ukpops dengan Pc dan ketiga faktor tersebut.

Dari hasil penghitungan P-value, faktor Ukpops dan Pc yang mempunyai pengaruh yang signifikan terhadap hasil akhir, sehingga pemilihan nilai level pada kedua faktor tersebut menjadi penting. Nilai level yang optimum pada tiap faktor dapat dilihat pada grafik *main effects plot* pada gambar 4.2. yang menunjukkan nilai rata-rata output. Karena tujuan akhir adalah untuk meminimumkan standard deviasi, maka diambil level yang menghasilkan output minimum. Tiap titik pada grafik *main effects plot* menunjukkan nilai rata-rata standard deviasi pada tiap level untuk tiap faktor. Gradien atau kemiringan tiap garis menunjukkan besar pengaruh perubahan level terhadap hasil akhir. Semakin besar gradien atau semakin tajam peningkatan atau penurunan garis, maka semakin besar pengaruh efek dari perubahan level tersebut terhadap hasil akhir.



Gambar 4.3 Grafik *Main Effects Plot* untuk ketiga Faktor Parameter

Dari grafik *main effect plot* untuk faktor ukuran populasi (Ukpop), diambil level ketiga yaitu populasi yang optimum adalah level ketiga (100) yang memberikan hasil akhir minimum pada standard deviasi. Pada grafik *main effect plot*, garis rata-rata standard deviasi untuk faktor Ukpop menurun tajam dari level pertama (30) ke level kedua (50) kemudian ke level ketiga (100). Ini menunjukkan perubahan level populasi mempunyai pengaruh yang signifikan terhadap hasil akhir.

Untuk faktor *crossover* (Pc), diambil level ketiga yaitu probabilitas *crossover* 0,95 yang memberikan hasil akhir yang minimum. Pada grafik *main effect plot*, gradien garis dari level pertama (0,5) ke level kedua (0,7) kemudian ke level ketiga (0,95) hampir sama dengan gradien pada faktor Ukpop, artinya perubahan level probabilitas *crossover* mempunyai pengaruh yang signifikan terhadap hasil akhir.

Untuk faktor iterasi (MaxG), diambil level ke dua yaitu iterasi 5000 yang memberikan hasil akhir yang minimum. Pada grafik *main effect plot*, gradien dari level pertama (1000) ke level kedua (5000) kemudian ke level ketiga (10000) hampir sama pada standard deviasi, artinya perubahan level iterasi tidak terlalu mempengaruhi hasil akhir.

4.4.2 Analisa Rotasi Penjadwalan

Permasalahan penjadwalan pemeliharaan pesawat terbang pada STPI adalah terjadinya penumpukan pesawat pada saat pemeliharaan dikarenakan tidak adanya jadwal kebutuhan pemakaian pesawat. Hal ini akan menyulitkan divisi pemeliharaan untuk menjadwalkan pemeliharaan setiap pesawat.

Usulan penyelesaian penjadwalan pemeliharaan adalah dengan menggunakan algoritma genetik. Dalam penelitian ini algoritma genetik diadopsi kedalam bahasa pemograman MATLAB. Fungsi tujuannya adalah meminimalkan pemeliharaan 2000 jam pada periode yang bersamaan.

Dari tabel 4.6 menampilkan urutan pemakaian pesawat terbang yang terbaik dari hasil *run* MATLAB dengan *Ukpop* 100, *MaxG* 5000, dan *crossover* 0,95 hasil lebih lengkapnya dapat dilihat pada lampiran 2. Jadwal tersebut akan menghasilkan jadwal pemeliharaan pesawat untuk 2000 jam secara merata gambar 4.2, dan jurusan penerbang tidak akan kekurangan pesawat dikarenakan pesawat sedang dalam pemeliharaan. Hal ini dapat dilihat pada jumlah tambahan pesawat / penalti sebesar nol.

Tabel 4.7 Hasil Penjadwalan Pemeliharaan Pesawat Terbang

| | |
|------------------------------|---------|
| Jumlah Pesawat | 24 |
| Panjang Periode (minggu) | 480 |
| Jumlah Pemeliharaan 2000 jam | 30 |
| Jumlah Tambahan Pesawat | 0 |
| Standard Deviasi (bulan) | 0,43483 |

BAB V

KESIMPULAN

Berdasarkan hasil analisis dari permasalahan penjadwalan pemeliharaan pesawat terbang latih di Sekolah Tinggi Penerbangan Indonesia menggunakan algoritma genetik dengan bantuan bahasa pemrograman MATLAB 7.8, diperoleh kesimpulan sebagai berikut :

1. Tidak ada kekurangan pesawat selama 10 tahun dengan urutan penjadwalan pemakaian pesawat kombinasi 24.
2. Jadwal pemeliharaan 2000 jam pesawat terbang selama 10 tahun kedepan dengan standard deviasi 0.43483.
3. Waktu *run* program rata-rata untuk memperoleh usulan jadwal yaitu adalah 34,4759 detik dengan menggunakan *notebook centrino duo* yang memiliki *memory* 2GB.
4. Dari kondisi ketersediaan pesawat pada tabel 3.1, Sekolah Tinggi Penerbangan Indonesia dapat menerima mahasiswa hingga 30 orang per semester.

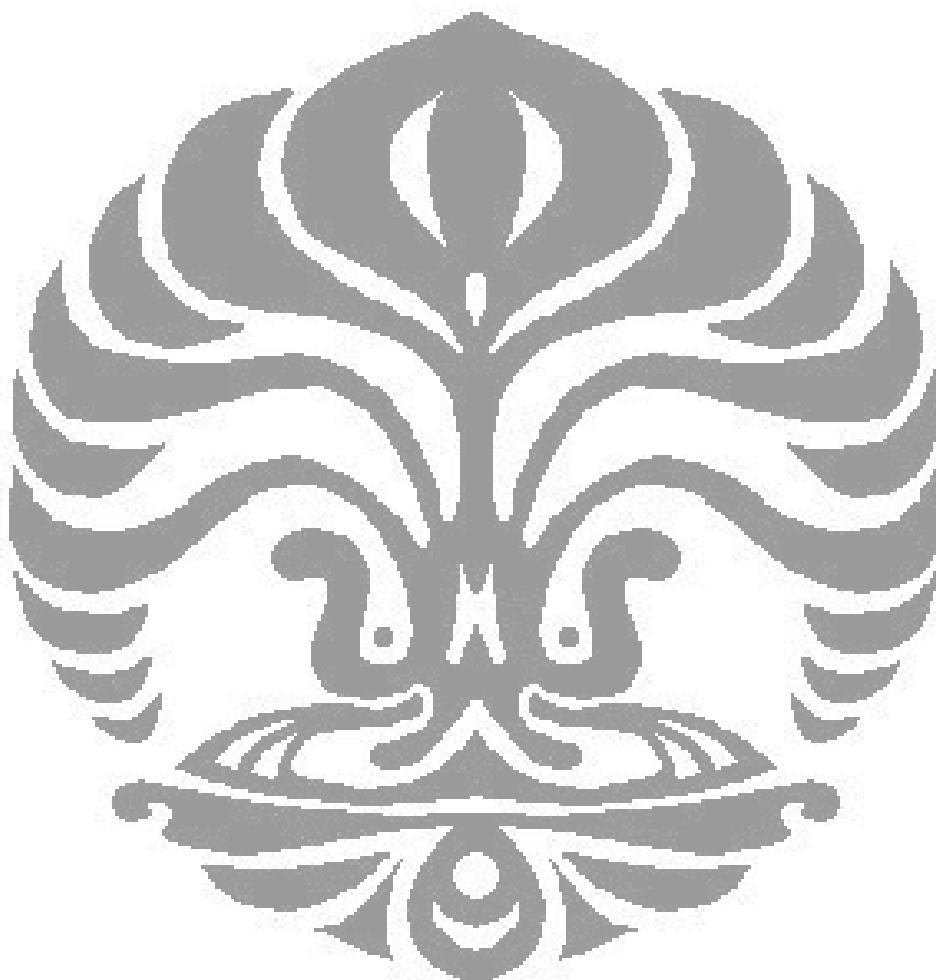
DAFTAR PUSTAKA

- Anthony, J., Warwood, S., Fernandez, K., Rowlands, H., 2001. *Process Optimisation Using Taguchi Method of Experimental Design*, Work Study, Vol.50,No.2.
- Basuki, Achmad. 2003. Strategi Menggunakan Algoritma Genetik.
- Berger, P.D dan Murer, R.E, 2002. *Experimental Design with Applications in Management, Engineering, and Science*, Thompson, New York.
- Fadlisyah, & Arnawan, & Faisal. 2009. Algoritma Genetik. Penerbit Graha Ilmu Yogyakarta.
- Hakim, Azki. 2008. <http://aeroblog.wordpress.com/category/perawatan-pesawat/>
- Haupt, Randy L. Haupt, Sue Elen. 2004. *Practical genetic algorithms*. New Jersey.
- Hessburg, Jack. 2001. *Air Carrier MRO Handbook*. United State of America.
- Kusumadewi, Sri. 2005. Penyelesaian Masalah Optimasi Dengan Teknik-teknik Heuristik, Penerbit Graha Ilmu : Yogyakarta.
- Mattila, Ville. Virtanen, Kai. 2005. *A Simulation-Based Optimization Model To Schedule Periodic Maintenance Of A Fleet Of Aircraft*
http://www.automaatioseura.fi/confprog/downloadfile_public.php?conference=12&filename=12-12017.pdf
- Mitchell, Melanie. 1996. *An Introduction to Genetic Algorithms*. Cambridge, Massachusetts. London, England.

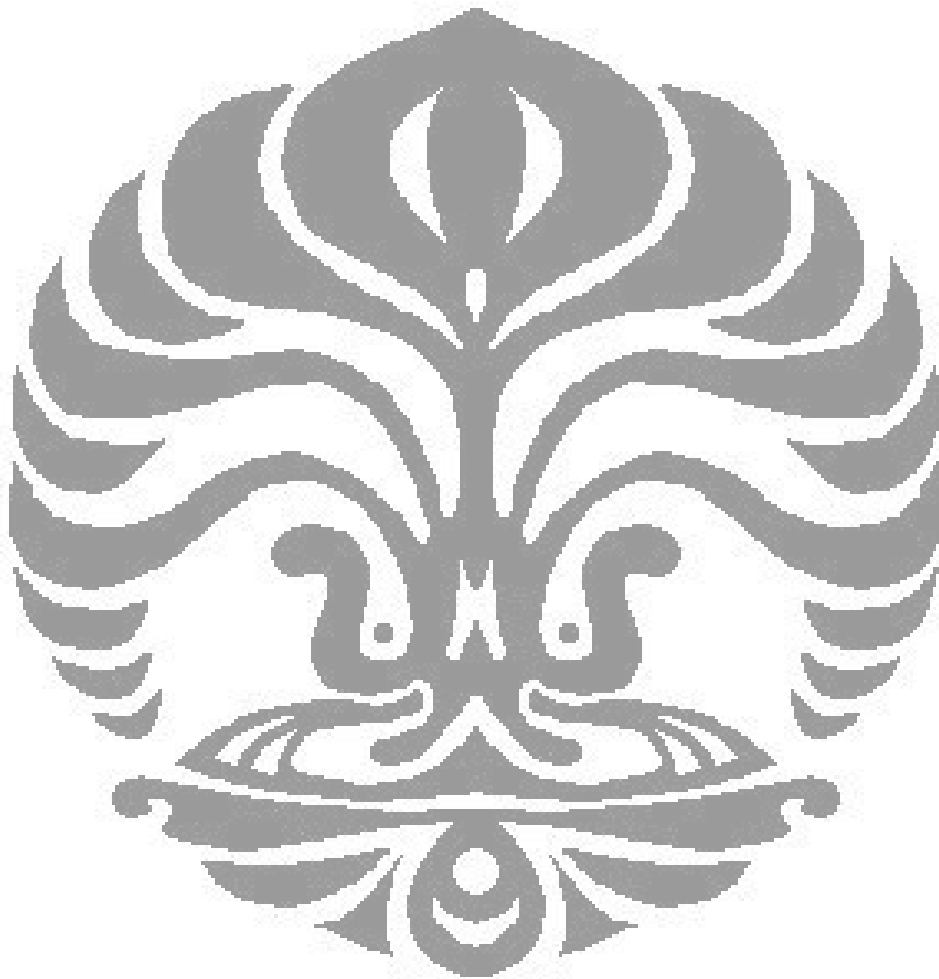
Montgomery, D.C., 1997. *Design and Analysis of Experiments*, Fourth Edition. John Wiley and Sons, New York.

Suyanto. 2005. *Algoritma Genetik dalam MATLAB*. Penerbit Andi:Yogyakarta.

www.mathworks.com



LAMPIRAN 1
SCRIPT PROGRAM ALGORITMA PENYELESAIAN



Lampiran 1 : Script Program Algoritma Penyelesaian

```
%Main Program

clear all;
tic;
disp(' ');
disp('>>> Membersihkan memory');

%----- INPUT DATA -----
----
disp('>>> Input Data dan parameter');
%Definisikan input data kebutuhan maintenance
JumlahWaktuMaintenance50 = 1; % 5 jam (dianggap satu minggu tidak
bisa digunakan)
JumlahWaktuMaintenance100 = 1; % 10 jam (dianggap satu minggu
tidak bisa digunakan)
JumlahWaktuMaintenance2000 = 16; % dalam minggu (4 bulan)

%definisikan jadwal penggunaan pesawat
KebutuhanPesawat = [7 8 8 7 6 5 7 8 8 7 6 5]; %dalam setahun

TetapanPenalti = 10;

JumlahPesawat = 24;
UmurPesawat = [0 0 0 0 0 0 675 0 700 0 0 0 0 0 0 800 1175 1850 875
1200 1000 1300 1650 0];
JumlahMingguSedangDimaintenance = [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0];
JumlahTahun = 10;
PanjangSolusi = (sum(KebutuhanPesawat) * 4 * JumlahTahun) +
JumlahPesawat;

%----- INPUT PARAMETER GA -----
----
JumlahPopulasi = 100;
JumlahIterasi = 5000;
TingkatCrossover = 0.95;
%TingkatMutasi = 0.5;
%JumlahLocalSearch = 100;

%----- INISIALISASI POPULASI -----
----
disp(' ');
disp('>>> Inisialisasi Populasi');
PopulasiAwal = zeros(JumlahPopulasi, PanjangSolusi);
SolusiUrutanPenggunaanawal = zeros(1, PanjangSolusi);
NomorPesawat = 0;
for i = 1 : PanjangSolusi
    NomorPesawat = NomorPesawat + 1;
    if NomorPesawat > 24
        NomorPesawat = NomorPesawat - 24;
    end
    SolusiUrutanPenggunaanawal(i) = NomorPesawat;
end
PopulasiAwal(1,:) = SolusiUrutanPenggunaanawal;
for i = 2 : JumlahPopulasi
    PopulasiAwal(i,:) =
SolusiUrutanPenggunaanawal(randperm(PanjangSolusi));
```

Lampiran 1 : Script Program Algoritma Penyelesaian

```
end

%----- Hitung Fitness Populasi Awal -----
---
FitnessPopulasiAwal = zeros(1, JumlahPopulasi);
SolusiUrutanPenggunaanPopulasiAwalRevisi = zeros(JumlahPopulasi,
sum(KebutuhanPesawat) * JumlahTahun * 4);
TotalOBJPopulasi = zeros(JumlahPopulasi, 12 * JumlahTahun);
JumlahInspeksi2000Populasi = zeros(JumlahPopulasi, 12 *
JumlahTahun);
JumlahTambahanPesawatPopulasi = zeros(JumlahPopulasi, 12 *
JumlahTahun);
for individuKe = 1 : JumlahPopulasi
    %coba hitung standard deviasi
    [StandarDeviasi, SolusiUrutanPenggunaanRevisi, TotalOBJ,
JumlahInspeksi2000, JumlahTambahanPesawat] =
HitungStandarDeviasi(PopulasiAwal(individuKe,:), JumlahPesawat,
UmurPesawat, JumlahMingguSedangMaintenance, JumlahTahun,
KebutuhanPesawat, JumlahWaktuMaintenance50,
JumlahWaktuMaintenance100, JumlahWaktuMaintenance2000,
TetapanPenalti);
    FitnessPopulasiAwal(individuKe) = StandarDeviasi;
    SolusiUrutanPenggunaanPopulasiAwalRevisi(individuKe, 1 :
numel(SolusiUrutanPenggunaanRevisi)) =
SolusiUrutanPenggunaanRevisi;
    TotalOBJPopulasi(individuKe, :) = TotalOBJ;
    JumlahInspeksi2000Populasi(individuKe, :) =
JumlahInspeksi2000;
    JumlahTambahanPesawatPopulasi(individuKe, :) =
JumlahTambahanPesawat;
end

%----- Tampilkan fitness populasi awal terbaik -----
---
indeksPopulasiTerbaik = min(find(FitnessPopulasiAwal ==
min(FitnessPopulasiAwal)));
disp(' ');
disp('Populasi Awal Terbaik');
disp(['Indeks Populasi Terbaik: '
num2str(indeksPopulasiTerbaik)]);
disp(['StandarDeviasi (fitness) Terbaik: '
num2str(FitnessPopulasiAwal(indeksPopulasiTerbaik))]);
disp(['SolusiUrutanPenggunaanRevisi Terbaik: '
num2str(SolusiUrutanPenggunaanPopulasiAwalRevisi(indeksPopulasiTer
baik, :))]);
disp(['TotalOBJ Terbaik: '
num2str(TotalOBJPopulasi(indeksPopulasiTerbaik, :))]);
disp(['JumlahInspeksi2000 Terbaik: '
num2str(JumlahInspeksi2000Populasi(indeksPopulasiTerbaik, :))]);
disp(['JumlahTambahanPesawat Terbaik: '
num2str(JumlahTambahanPesawatPopulasi(indeksPopulasiTerbaik,
:))]);
disp(['Total waktu sampai saat ini: ' num2str(toc)]);

%----- Mulai Iterasi -----
---
disp(' ');
disp('Mulai iterasi GA');
```

Lampiran 1 : Script Program Algoritma Penyelesaian

```
FitnessPopulasiIterasi = FitnessPopulasiAwal;
PopulasiIterasi = PopulasiAwal;
SolusiUrutanPenggunaanPopulasiIterasiRevisi =
SolusiUrutanPenggunaanPopulasiAwalRevisi;
TotalOBJPopulasiIterasi = TotalOBJPopulasi;
JumlahInspeksi2000PopulasiIterasi = JumlahInspeksi2000Populasi;
JumlahTambahPesawatPopulasiIterasi =
JumlahTambahPesawatPopulasi;
%sedikan memori
SolusiUrutanPenggunaanIndividu1 = zeros(1, (sum(KebutuhanPesawat)
* JumlahTahun * 4) + JumlahPesawat);
SolusiUrutanPenggunaanIndividu2 = zeros(1, (sum(KebutuhanPesawat)
* JumlahTahun * 4) + JumlahPesawat);
for IterasiKe = 1 : JumlahIterasi

    %pilih dua individu parent secara random
    IndeksParent1 = round(rand * (JumlahPopulasi - 1)) + 1;
    IndeksParent2 = IndeksParent1;
    while IndeksParent2 == IndeksParent1
        IndeksParent2 = round(rand * (JumlahPopulasi - 1)) + 1;
    end

    %pilih indeks solusi yang ingin dipotong
    IndeksKromosom1 = round(rand * (PanjangSolusi - 1)) + 1;
    IndeksKromosom2 = IndeksKromosom1;
    while IndeksKromosom1 >= IndeksKromosom2
        IndeksKromosom1 = round(rand * (PanjangSolusi - 1)) + 1;
        IndeksKromosom2 = round(rand * (PanjangSolusi - 1)) + 1;
    end

    %Buat individu baru
    SolusiUrutanPenggunaanIndividu1(1, 1 : IndeksKromosom1 - 1) =
PopulasiIterasi(IndeksParent1, 1 : IndeksKromosom1 - 1);
    if rand() < TingkatCrossover
        SolusiUrutanPenggunaanIndividu1(1, IndeksKromosom1 :
IndeksKromosom2) = PopulasiIterasi(IndeksParent2, IndeksKromosom1
: IndeksKromosom2);
    else
        SolusiUrutanPenggunaanIndividu1(1, IndeksKromosom1 :
IndeksKromosom2) = PopulasiIterasi(IndeksParent1, IndeksKromosom1
: IndeksKromosom2);
    end
    SolusiUrutanPenggunaanIndividu1(1, IndeksKromosom2 + 1 :
PanjangSolusi) = PopulasiIterasi(IndeksParent1, IndeksKromosom2 +
1 : PanjangSolusi);
    SolusiUrutanPenggunaanIndividu2(1, 1 : IndeksKromosom1 - 1) =
PopulasiIterasi(IndeksParent2, 1 : IndeksKromosom1 - 1);
    if rand() < TingkatCrossover
        SolusiUrutanPenggunaanIndividu2(1, IndeksKromosom1 :
IndeksKromosom2) = PopulasiIterasi(IndeksParent1, IndeksKromosom1
: IndeksKromosom2);
    else
        SolusiUrutanPenggunaanIndividu2(1, IndeksKromosom1 :
IndeksKromosom2) = PopulasiIterasi(IndeksParent2, IndeksKromosom1
: IndeksKromosom2);
    end
end
```

Lampiran 1 : Script Program Algoritma Penyelesaian

```
SolusiUrutanPenggunaanIndividu2(1, IndeksKromosom2 + 1 :  
PanjangSolusi) = PopulasiIterasi(IndeksParent2, IndeksKromosom2 +  
1 : PanjangSolusi);  
  
%hitung fitnes individu yg baru pertama  
[StandarDeviasi, SolusiUrutanPenggunaanRevisi, TotalOBJ,  
JumlahInspeksi2000, JumlahTambahanPesawat] =  
HitungStandarDeviasi(SolusiUrutanPenggunaanIndividu1,  
JumlahPesawat, UmurPesawat, JumlahMingguSedangDimaintenance,  
JumlahTahun, KebutuhanPesawat, JumlahWaktuMaintenance50,  
JumlahWaktuMaintenance100, JumlahWaktuMaintenance2000,  
TetapanPenalti);  
indeksPopulasiTerjelek = min(find(FitnessPopulasiIterasi ==  
max(FitnessPopulasiIterasi), 1-));  
%Jika individu yg baru lebih baik daripada individu populasi  
terjelek,  
%maka update populasinya  
if StandarDeviasi <  
FitnessPopulasiIterasi(indeksPopulasiTerjelek)  
PopulasiIterasi(indeksPopulasiTerjelek, :) =  
SolusiUrutanPenggunaanIndividu1;  
FitnessPopulasiIterasi(indeksPopulasiTerjelek) =  
StandarDeviasi;  
SolusiUrutanPenggunaanPopulasiIterasiRevisi(indeksPopulasiTerjelek  
, :) = 0;  
  
SolusiUrutanPenggunaanPopulasiIterasiRevisi(indeksPopulasiTerjelek  
, 1 : numel(SolusiUrutanPenggunaanRevisi)) =  
SolusiUrutanPenggunaanRevisi;  
TotalOBJPopulasiIterasi(indeksPopulasiTerjelek, :) =  
TotalOBJ;  
JumlahInspeksi2000PopulasiIterasi(indeksPopulasiTerjelek,  
:) = JumlahInspeksi2000;  
  
JumlahTambahanPesawatPopulasiIterasi(indeksPopulasiTerjelek, :) =  
JumlahTambahanPesawat;  
end  
  
%hitung fitnes individu yg baru yg kedua  
[StandarDeviasi, SolusiUrutanPenggunaanRevisi, TotalOBJ,  
JumlahInspeksi2000, JumlahTambahanPesawat] =  
HitungStandarDeviasi(SolusiUrutanPenggunaanIndividu2,  
JumlahPesawat, UmurPesawat, JumlahMingguSedangDimaintenance,  
JumlahTahun, KebutuhanPesawat, JumlahWaktuMaintenance50,  
JumlahWaktuMaintenance100, JumlahWaktuMaintenance2000,  
TetapanPenalti);  
indeksPopulasiTerjelek = min(find(FitnessPopulasiIterasi ==  
max(FitnessPopulasiIterasi))));  
%Jika individu yg baru lebih baik daripada individu populasi  
terjelek,  
%maka update populasinya  
if StandarDeviasi <  
FitnessPopulasiIterasi(indeksPopulasiTerjelek)  
PopulasiIterasi(indeksPopulasiTerjelek, :) =  
SolusiUrutanPenggunaanIndividu2;  
FitnessPopulasiIterasi(indeksPopulasiTerjelek) =  
StandarDeviasi;
```

Lampiran 1 : Script Program Algoritma Penyelesaian

```
SolusiUrutanPenggunaanPopulasiIterasiRevisi (indeksPopulasiTerjelek
, :) = 0;

SolusiUrutanPenggunaanPopulasiIterasiRevisi (indeksPopulasiTerjelek
, 1 : numel(SolusiUrutanPenggunaanRevisi)) =
SolusiUrutanPenggunaanRevisi;
    TotalOBJPopulasiIterasi (indeksPopulasiTerjelek, :) =
TotalOBJ;
    JumlahInspeksi2000PopulasiIterasi (indeksPopulasiTerjelek,
:) = JumlahInspeksi2000;

JumlahTambahanPesawatPopulasiIterasi (indeksPopulasiTerjelek, :) =
JumlahTambahanPesawat;
    end

end
disp(['Total waktu sampai saat ini: ' num2str(toc)]);

%-----Tampilkan Nilai terbaik-----
indeksPopulasiTerbaik = min(find(FitnessPopulasiIterasi ==
min(FitnessPopulasiIterasi)));
disp(' ');
disp('Hasil Terbaik GA');
disp(['Indeks Populasi Terbaik: '
num2str(indeksPopulasiTerbaik)]);
disp(['StandarDeviasi (fitness) Terbaik: '
num2str(FitnessPopulasiIterasi (indeksPopulasiTerbaik))]);
disp(['SolusiUrutanPenggunaanRevisi Terbaik: '
num2str(SolusiUrutanPenggunaanPopulasiIterasiRevisi (indeksPopulasi
Terbaik, :))]);
disp(['TotalOBJ Terbaik: '
num2str(TotalOBJPopulasiIterasi (indeksPopulasiTerbaik, :))]);
disp(['JumlahInspeksi2000 Terbaik: '
num2str(JumlahInspeksi2000PopulasiIterasi (indeksPopulasiTerbaik,
:))]);
disp(['JumlahTambahanPesawat Terbaik: '
num2str(JumlahTambahanPesawatPopulasiIterasi (indeksPopulasiTerbaik
, :))]);
disp(['Total waktu: ' num2str(toc)]);

%-----HitungStandardDeviasi-----

function [StandarDeviasi, SolusiUrutanPenggunaanRevisi, TotalOBJ,
JumlahInspeksi2000, JumlahTambahanPesawat] =
HitungStandarDeviasi (SolusiUrutanPenggunaan, JumlahPesawat,
UmurPesawat, JumlahMingguSedangDimaintenance, JumlahTahun,
KebutuhanPesawat, JumlahWaktuMaintenance50,
JumlahWaktuMaintenance100, JumlahWaktuMaintenance2000,
TetapanPenalti)

JumlahInspeksi2000 = zeros(1, 12 * JumlahTahun);
JumlahTambahanPesawat = zeros(1, 12 * JumlahTahun); %penalti
```

Universitas Indonesia

Lampiran 1 : Script Program Algoritma Penyelesaian

```
SolusiUrutanPenggunaanRevisi = zeros(1, sum(KebutuhanPesawat) *
JumlahTahun * 4); %penggunaan pesawat mingguan sebenarnya
UmurPesawatSekarang = UmurPesawat;
PanjangUrutan = numel(SolusiUrutanPenggunaan);
UrutanDigunakan = 0;
UrutanDigunakanSebenarnya = 0;
for TahunKe = 1 : JumlahTahun
    for BulanKe = 1 : 12

        for MingguKe = 1 : 4

            %cek apakah pesawat yang digunakan sedang diinspeksi,
gunakan
            %jika sedang tidak diinspeksi
            PesawatKe = 0;
            JumlahPesawatDitest = 0;
            while PesawatKe < KebutuhanPesawat(BulanKe) &&
JumlahPesawatDitest < JumlahPesawat
                JumlahPesawatDitest = JumlahPesawatDitest + 1;

                %update urutan penggunaan
                UrutanDigunakan = UrutanDigunakan + 1;
                if UrutanDigunakan > PanjangUrutan
                    UrutanDigunakan = UrutanDigunakan -
PanjangUrutan;
                end

                if
                    JumlahMingguSedangDimaintenance (SolusiUrutanPenggunaan(UrutanDigun
akan)) == 0 %artinya bisa digunakan atau tidak sedang
                    dimaintenance
                        PesawatKe = PesawatKe + 1;
                        UrutanDigunakanSebenarnya =
UrutanDigunakanSebenarnya + 1;
                        %catat pesawat yg digunakan

                        SolusiUrutanPenggunaanRevisi (UrutanDigunakanSebenarnya) =
                        SolusiUrutanPenggunaan (UrutanDigunakan);
                        %update umur pesawat

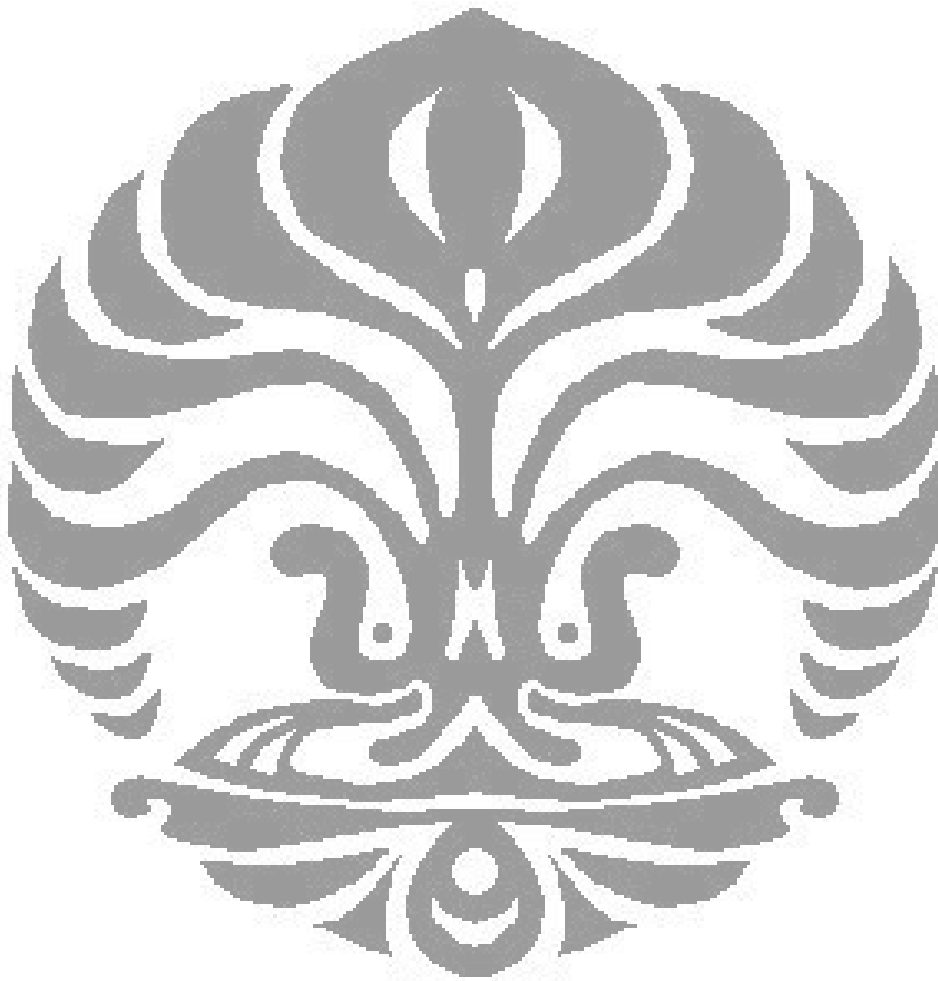
                        UmurPesawatSekarang (SolusiUrutanPenggunaan (UrutanDigunakan)) =
                        UmurPesawatSekarang (SolusiUrutanPenggunaan (UrutanDigunakan)) + 25;
                        %update kebutuhan maintenance
                        if
                            mod (UmurPesawatSekarang (SolusiUrutanPenggunaan (UrutanDigunakan)),
                            2000) == 0 %artinya diinspeksi 2000

                                JumlahMingguSedangDimaintenance (SolusiUrutanPenggunaan (UrutanDigun
                                akan)) =
                                JumlahMingguSedangDimaintenance (SolusiUrutanPenggunaan (UrutanDigun
                                akan)) + JumlahWaktuMaintenance2000 + 1;
                                %catat kebutuhan inspeksi 2000
                                JumlahInspeksi2000 ((TahunKe - 1) * 12 +
                                BulanKe) = JumlahInspeksi2000 ((TahunKe - 1) * 12 + BulanKe) + 1;
                                elseif
                                    mod (UmurPesawatSekarang (SolusiUrutanPenggunaan (UrutanDigunakan)),
                                    100) == 0 %artinya diinspeksi 100
```

Lampiran 1 : Script Program Algoritma Penyelesaian

```
JumlahMingguSedangDimaintenance (SolusiUrutanPenggunaan (UrutanDigun  
akan)) =  
JumlahMingguSedangDimaintenance (SolusiUrutanPenggunaan (UrutanDigun  
akan)) + JumlahWaktuMaintenance100 + 1;  
elseif  
mod (UmurPesawatSekarang (SolusiUrutanPenggunaan (UrutanDigunakan)),  
50) == 0 %artinya diinspeksi 100  
JumlahMingguSedangDimaintenance (SolusiUrutanPenggunaan (UrutanDigun  
akan)) =  
JumlahMingguSedangDimaintenance (SolusiUrutanPenggunaan (UrutanDigun  
akan)) + JumlahWaktuMaintenance50 + 1;  
end  
end  
end  
end  
%update jumlah tambahan pesawat yg diperlukan  
(penalti)  
JumlahTambaharPesawat((TahunKe - 1)* 12 + BulanKe) =  
JumlahTambaharPesawat((TahunKe - 1)* 12 + BulanKe) +  
KebutuhanPesawat (BulanKe) - PesawatKe;  
%update JumlahMingguSedangDimaintenance  
for PesawatKe = 1 : JumlahPesawat  
if JumlahMingguSedangDimaintenance (PesawatKe) > 0  
JumlahMingguSedangDimaintenance (PesawatKe) =  
JumlahMingguSedangDimaintenance (PesawatKe) - 1;  
end  
end  
end  
end  
end  
end  
%Total obj (kebutuhan inspeksi 2000 dan tambahan pesawat  
TotalOBJ = JumlahInspeksi2000 + (JumlahTambaharPesawat *  
TetapanPenalti);  
%hitung standar deviasi  
StandarDeviasi = std(TotalOBJ);
```

LAMPIRAN 2
HASIL PENJADWALAN TERBAIK



Jadwal Pemakaian Pesawat Terbaik

| Tahun | Sem | Bulan | Urutan Pesawat | Jumlah |
|-----------|-----|-------|---|--------|
| 2008-2009 | 1 | April | 1 20 21 16 7 14 9 13 4 21 24 16 15 12 4 11 15 20 3 19 13 9 22 10 11 17 18 21 | 28 |
| | | Mei | 22 2 12 3 18 6 5 24 23 20 23 9 9 4 2 5 7 15 14 21 8 3 18 18 24 15 20 7 12 | 32 |
| | | Juni | 14 5 16 5 21 11 21 22 4 24 15 17 16 12 3 8 13 2 9 5 18 22 1 20 2 18 24 7 2 | 32 |
| | | Juli | 21 13 10 11 6 12 17 7 3 24 8 15 4 14 3 10 21 2 13 13 23 15 19 23 15 22 10 1 | 28 |
| | | Agust | 1 7 17 13 12 6 16 17 2 8 19 4 11 9 10 23 13 24 20 15 22 5 19 23 | 24 |
| | | Sept | 6 16 4 14 22 10 2 11 14 4 8 21 23 12 20 16 23 8 6 9 | 20 |
| | 2 | Okt | 17 21 4 7 9 15 1 23 2 22 20 6 17 3 9 12 22 11 23 7 10 13 5 4 2 11 19 8 | 28 |
| | | Nov | 18 24 13 20 15 10 21 20 5 8 19 22 22 15 17 4 24 19 18 2 21 13 20 12 9 8 13 24 | 32 |
| | | Des | 20 4 18 18 23 23 15 10 6 10 14 21 2 11 1 1 17 20 8 2 6 13 17 4 12 13 11 15 15 | 32 |
| | | Jan | 19 4 21 19 3 10 6 8 13 22 7 22 10 15 17 11 17 21 1 9 8 4 18 12 4 19 11 7 | 28 |
| | | Feb | 18 17 10 2 6 14 17 2 8 13 1 4 14 14 15 21 16 5 7 17 5 19 1 18 | 24 |
| | | Maret | 15 3 9 18 15 19 14 8 22 24 11 21 1 6 17 9 22 5 2 9 | 20 |
| 2009-2010 | 1 | April | 20 24 1 19 10 4 3 12 11 7 13 17 23 17 7 19 3 19 15 24 18 16 14 15 1 21 18 2 | 28 |
| | | Mei | 5 6 13 22 10 9 20 4 11 14 22 3 2 16 21 9 7 6 17 5 2 6 11 4 22 7 20 22 18 | 32 |
| | | Juni | 6 18 2 16 23 14 5 10 4 17 6 20 24 12 7 12 11 19 13 21 10 15 16 5 24 10 6 9 2 | 32 |
| | | Juli | 13 21 22 16 18 8 5 24 12 8 11 17 4 14 24 9 6 7 1 13 5 8 11 17 16 18 3 10 | 28 |
| | | Agust | 4 24 21 19 12 10 14 17 13 11 8 18 17 1 9 22 12 15 5 6 13 20 4 10 | 24 |
| | | Sept | 20 1 23 15 11 22 16 5 11 2 4 15 8 20 21 16 6 16 1 17 | 20 |
| | 2 | Okt | 21 24 10 13 15 9 15 22 12 19 1 2 16 4 9 9 21 4 21 6 1 3 2 17 7 19 13 8 | 28 |
| | | Nov | 15 6 20 9 7 22 11 24 16 13 22 11 8 4 10 18 15 18 7 6 23 19 5 24 2 4 8 7 9 | 32 |
| | | Des | 20 24 23 5 10 12 15 22 13 13 4 3 21 17 16 15 11 2 1 12 16 8 21 23 9 20 19 7 | 32 |
| | | Jan | 16 13 4 7 11 23 11 1 5 1 21 9 8 16 10 11 20 6 21 24 24 8 20 12 18 1 13 16 | 28 |
| | | Feb | 21 10 21 13 15 10 19 16 23 17 15 8 14 11 19 10 8 4 2 18 16 12 3 7 | 24 |
| | | Maret | 20 20 3 15 9 5 8 21 15 13 18 13 12 23 6 4 10 5 21 24 | 20 |

Jadwal Pemakaian Pesawat Terbaik (sambungan)

| Tahun | Sem | Bulan | Urutan Pesawat | Jumlah |
|-----------|--|---|--|--------|
| 2010-2011 | 1 | April | 14 7 9 24 4 12 5 10 2 8 22 13 23 1 17 13 20 6 23 4 18 4 7 12 12 8 16 15 | 28 |
| | | Mei | 14 15 14 19 13 21 21 23 23 3 6 3 24 19 10 7 20 22 14 1 9 4 5 8 9 4 2 2 12 5 12 24 | 32 |
| | | Juni | 18 10 10 22 22 8 19 16 21 3 16 5 18 15 17 4 8 19 3 10 21 2 1 12 18 5 12 16 17 18 4 10 | 32 |
| | | Juli | 22 7 6 9 24 1 7 23 3 18 10 13 2 9 20 8 4 20 16 13 1 21 9 18 19 22 5 4 | 28 |
| | | Agust | 8 2 7 14 16 23 10 1 3 13 5 3 1 19 22 17 15 11 10 12 13 12 8 22 | 24 |
| | Sept | 14 2 3 18 5 8 21 2 19 21 5 13 1 13 6 1 15 6 2 10 | 20 | |
| | 2 | Okt | 16 2 5 8 23 12 11 15 14 24 6 1 22 8 2 12 23 1 5 3 10 11 10 8 4 11 9 15 | 28 |
| | | Nov | 1 2 13 1 14 23 3 4 23 16 22 12 11 15 12 24 11 2 14 13 19 7 18 24 10 16 15 3 15 22 9 10 | 32 |
| | | Des | 23 19 23 13 9 22 19 5 6 8 16 16 4 12 11 18 5 4 1 10 12 7 13 3 14 20 14 9 10 15 8 1 | 32 |
| | | Jan | 9 22 24 18 4 6 13 15 24 14 20 19 2 11 10 18 13 12 6 22 19 15 5 12 5 10 9 9 | 28 |
| Feb | | 18 21 24 4 22 23 21 22 8 20 1 13 11 23 13 17 4 14 8 21 17 9 5 1 | 24 | |
| Maret | 14 24 3 21 14 22 5 20 9 19 16 10 18 7 15 1 4 8 21 10 | 20 | | |
| 2011-2012 | 1 | April | 20 9 23 3 7 14 14 23 8 15 24 8 24 1 9 21 5 15 10 22 10 17 17 14 4 19 18 8 | 28 |
| | | Mei | 9 6 22 4 13 11 10 12 1 7 1 8 15 13 22 15 10 6 20 2 21 5 17 12 4 16 8 15 13 18 19 4 | 32 |
| | | Juni | 5 9 22 8 22 12 2 15 1 10 19 17 3 23 10 1 20 11 2 18 23 24 14 21 5 2 22 13 3 8 22 24 | 32 |
| | | Juli | 23 4 12 6 7 9 19 21 1 21 2 10 18 17 18 4 16 20 8 6 19 16 2 1 10 3 14 13 | 28 |
| | | Agust | 7 4 15 22 9 22 4 13 14 2 8 12 23 10 18 18 17 19 12 7 3 19 24 7 | 24 |
| | Sept | 1 17 20 14 3 12 17 9 8 24 23 23 14 9 11 5 3 7 12 2 | 20 | |
| | 2 | Okt | 8 24 24 7 20 17 1 22 23 12 23 16 22 6 16 18 13 4 11 21 8 21 17 1 18 13 10 14 | 28 |
| | | Nov | 5 20 4 23 11 20 12 3 16 10 8 15 10 21 24 7 14 1 7 2 12 6 4 18 4 5 23 5 20 17 21 8 | 32 |
| | | Des | 17 11 3 2 3 7 13 22 21 15 1 8 6 10 1 24 4 6 18 10 23 13 14 4 17 14 2 20 23 5 22 15 | 32 |
| | | Jan | 7 16 3 18 17 19 12 22 1 2 6 8 20 4 23 21 18 10 3 10 19 1 9 7 22 23 20 11 | 28 |
| Feb | | 10 10 13 12 4 13 11 21 2 17 11 9 8 15 4 22 15 10 5 22 21 20 13 16 | 24 | |
| Maret | 7 2 12 24 11 14 18 20 4 9 14 12 15 18 24 5 17 12 22 10 | 20 | | |

Jadwal Pemakaian Pesawat Terbaik (sambungan)

| Tahun | Sem | Bulan | Urutan Pesawat | Jumlah |
|-----------|-----|-------|--|--------|
| 2012-2013 | 1 | April | 6 16 20 7 19 1 21 21 10 7 20 11 5 10 19 12 3 1 17 9 8 15 11 21 18 10 13 17 | 28 |
| | | Mei | 7 6 4 22 11 2 20 24 1 5 14 13 12 18 17 19 15 7 24 22 6 11 11 21 22 4 19 8 18 8 15 13 | 32 |
| | | Juni | 18 14 6 20 21 7 12 3 2 9 8 22 1 17 10 4 5 4 19 15 21 22 19 3 1 3 10 15 11 7 14 9 | 32 |
| | | Juli | 24 23 13 11 12 17 22 15 3 13 2 20 4 3 19 14 20 11 18 18 6 24 11 13 9 13 1 9 | 28 |
| | | Agust | 5 6 6 5 18 15 20 8 2 10 19 18 15 6 11 24 12 15 23 10 1 13 6 14 | 24 |
| | | Sept | 24 11 2 2 22 20 13 5 5 23 17 11 22 18 6 9 16 13 2 11 | 20 |
| | 2 | Okt | 12 15 3 16 23 24 15 1 13 10 19 3 2 22 14 12 20 4 24 9 19 10 6 11 15 12 2 2 | 28 |
| | | Nov | 10 4 17 20 4 22 14 22 16 23 2 23 9 17 14 1 19 24 15 22 13 8 1 6 5 5 8 6 7 7 3 22 | 32 |
| | | Des | 24 18 2 15 17 20 20 11 22 23 8 1 3 21 23 17 5 8 19 13 20 4 21 9 18 16 1 2 14 17 12 20 | 32 |
| | | Jan | 2 3 21 21 8 9 11 1 19 4 16 5 22 17 11 7 3 16 2 21 19 5 1 24 23 13 21 6 | 28 |
| | | Feb | 2 13 22 17 24 22 17 21 9 5 21 12 24 18 8 6 1 20 24 20 16 7 2 16 | 24 |
| | | Maret | 5 2 4 23 11 7 8 1 21 3 14 6 3 24 12 9 12 20 11 1 | 20 |
| 2013-2014 | 1 | April | 23 21 17 5 4 8 1 19 12 20 12 2 17 9 8 7 5 10 15 18 16 22 24 10 5 19 22 17 | 28 |
| | | Mei | 9 6 23 20 7 7 17 20 12 8 2 15 12 10 18 22 16 21 11 9 11 15 19 19 22 23 18 14 2 24 20 3 | 32 |
| | | Juni | 3 9 11 5 6 7 9 11 24 2 21 13 23 8 19 14 12 5 17 8 6 12 20 19 14 9 22 11 22 9 1 21 | 32 |
| | | Juli | 6 2 14 2 20 23 3 5 6 20 18 3 5 7 1 9 17 12 22 12 13 20 5 7 10 23 24 7 | 28 |
| | | Agust | 3 21 2 6 11 16 3 6 23 4 20 10 11 18 9 21 11 13 12 17 6 12 13 4 | 24 |
| | | Sept | 9 5 7 11 22 24 6 23 12 14 16 10 2 22 21 13 11 7 23 6 | 20 |
| | 2 | Okt | 4 15 6 9 24 20 12 21 20 3 11 16 19 5 16 19 24 22 13 23 14 9 7 3 7 5 21 4 | 28 |
| | | Nov | 10 13 11 6 18 9 6 14 8 5 17 21 7 4 19 4 21 16 24 7 13 8 14 9 19 4 1 9 10 20 23 16 | 32 |
| | | Des | 15 18 1 21 8 24 21 23 20 6 4 10 5 18 8 17 1 10 7 24 9 1 13 7 13 24 14 8 18 23 15 3 | 32 |
| | | Jan | 9 7 10 15 5 14 23 8 6 13 7 3 4 18 20 9 17 9 22 4 13 5 19 24 20 10 19 18 | 28 |
| | | Feb | 16 14 8 15 17 8 19 1 13 9 17 22 19 13 20 8 16 12 14 2 7 4 15 24 | 24 |
| | | Maret | 20 2 12 8 7 17 16 14 21 11 21 22 12 23 19 22 3 20 3 20 | 20 |

Jadwal Pemakaian Pesawat Terbaik (sambungan)

| Tahun | Sem | Bulan | Urutan Pesawat | Jumlah |
|-----------|--|---|---|--------|
| 2014-2015 | 1 | April | 9 13 2 4 11 24 18 7 10 10 17 7 23 19 15 16 13 11 6 11 8 9 9 24 14 17 8 3 | 28 |
| | | Mei | 2 12 4 23 16 6 1 20 14 23 13 7 17 18 13 22 14 18 11 4 19 16 6 7 13 3 11 10 9 21 2 10 | 32 |
| | | Juni | 22 5 7 7 2 5 20 18 12 18 11 15 3 17 21 23 6 9 2 8 3 22 19 17 11 13 21 23 7 21 16 12 | 32 |
| | | Juli | 18 20 20 22 4 1 5 14 14 7 10 11 3 18 5 12 20 3 10 16 19 23 12 22 11 9 7 4 | 28 |
| | | Agust | 2 10 9 7 15 1 10 11 22 17 8 23 20 11 7 16 2 13 21 21 3 2 12 12 | 24 |
| | Sept | 13 11 3 11 4 16 19 15 23 23 7 12 14 10 1 11 2 24 21 22 | 20 | |
| | 2 | Okt | 12 24 3 21 13 13 3 23 11 22 16 10 14 8 12 24 16 24 21 20 5 19 12 11 14 5 21 14 | 28 |
| | | Nov | 10 24 15 20 15 17 24 11 5 23 6 2 19 10 5 12 14 8 22 12 16 21 22 6 17 7 24 19 5 19 9 1 | 32 |
| | | Des | 10 23 12 14 5 8 16 2 4 8 15 1 18 23 12 6 7 22 24 20 2 17 5 10 1 5 4 8 6 21 12 19 | 32 |
| | | Jan | 20 2 12 16 22 23 7 21 11 5 1 17 16 7 15 18 23 1 2 11 6 16 21 7 20 10 7 9 | 28 |
| Feb | | 17 4 24 19 6 2 1 2 16 1 20 9 19 15 6 18 13 13 10 20 4 17 17 1 | 24 | |
| Maret | 15 18 5 13 24 16 16 8 10 11 6 13 4 1 24 22 2 15 20 21 | 20 | | |
| 2015-2016 | 1 | April | 19 24 9 11 8 21 15 10 8 22 3 1 12 16 16 9 4 20 18 17 24 19 7 20 7 17 5 15 | 28 |
| | | Mei | 8 15 1 12 24 13 8 18 17 3 21 6 7 2 4 6 17 24 7 5 15 14 24 8 10 16 11 10 18 6 20 13 | 32 |
| | | Juni | 24 15 8 5 3 17 14 17 4 12 11 5 6 2 4 13 1 2 21 19 14 14 18 3 1 12 17 13 5 18 20 6 | 32 |
| | | Juli | 5 24 10 8 7 2 14 17 18 11 15 9 10 14 11 23 15 6 4 13 18 23 1 20 20 8 3 9 | 28 |
| | | Agust | 1 21 16 7 14 9 13 4 21 24 15 12 11 15 20 3 19 13 4 9 22 4 10 11 | 24 |
| | Sept | 17 18 21 22 2 12 3 18 6 5 24 23 22 20 23 9 3 4 2 5 | 20 | |
| | 2 | Okt | 7 15 14 21 8 18 18 24 15 20 7 12 22 6 23 14 5 16 5 21 20 11 21 22 4 24 15 17 | 28 |
| | | Nov | 16 12 3 5 8 13 2 9 5 18 22 3 1 20 18 24 7 23 19 1 6 2 21 13 10 11 6 12 17 7 3 24 | 32 |
| | | Des | 8 15 4 14 3 18 21 2 13 13 23 19 23 22 10 1 18 22 1 7 17 12 6 16 2 8 19 4 11 9 10 23 | 32 |
| | | Jan | 13 24 20 15 22 5 23 6 16 4 14 22 10 2 11 14 4 8 21 23 12 16 23 8 6 9 17 21 | 28 |
| Feb | | 4 7 9 15 1 2 22 6 17 3 12 22 11 23 23 7 10 13 5 4 2 12 11 19 | 24 | |
| Maret | 8 18 24 13 15 11 10 21 5 8 19 24 22 22 15 17 4 24 18 2 | 20 | | |

Jadwal Pemakaian Pesawat Terbaik (sambungann)

| Tahun | Sem | Bulan | Urutan Pesawat | Jumlah |
|-----------|-----|-------|---|--------|
| 2016-2017 | 1 | April | 21 15 8 13 12 2 9 8 19 24 1 24 19 16 9 15 13 18 17 4 18 23 23 10 6 14 21 2 | 28 |
| | | Mei | 11 1 1 17 8 2 6 13 17 4 12 15 15 23 20 23 8 14 19 4 21 11 19 3 10 6 13 22 7 22 10 15 | 32 |
| | | Juni | 17 11 17 23 1 9 8 4 18 12 4 1 19 23 7 18 17 10 2 6 14 17 2 8 13 1 4 14 23 4 15 16 | 32 |
| | | Juli | 2 5 7 19 1 18 3 9 18 15 14 8 22 24 11 1 6 22 5 2 20 24 1 19 10 4 3 12 | 28 |
| | | Agust | 11 7 13 23 7 19 3 15 24 18 16 14 1 7 18 2 5 6 13 22 15 10 16 9 | 24 |
| | | Sept | 20 4 11 6 5 20 13 14 22 5 3 2 16 9 7 6 5 11 4 22 | 20 |
| | 2 | Okt | 7 20 22 18 3 9 9 6 18 2 16 23 14 5 10 4 6 16 20 24 12 7 11 19 13 21 15 5 | 28 |
| | | Nov | 24 10 6 9 21 23 1 24 18 13 22 16 13 18 8 5 24 12 11 17 4 16 14 24 9 6 7 1 8 13 5 8 | 32 |
| | | Des | 11 17 16 18 3 10 4 24 21 19 12 14 13 11 8 18 17 1 9 22 15 5 6 20 4 17 10 20 23 15 11 16 | 32 |
| | | Jan | 5 11 2 4 8 21 6 16 1 17 16 24 10 13 15 1 9 15 22 12 19 2 24 16 4 22 21 4 | 28 |
| | | Feb | 21 6 12 1 3 2 17 7 19 13 8 15 6 20 9 22 11 8 24 16 13 7 22 11 | 24 |
| | | Maret | 6 8 1 10 18 15 18 7 6 23 22 19 5 24 2 4 8 7 9 3 | 20 |
| 2017-2018 | 1 | April | 22 11 20 24 23 5 10 12 7 3 15 13 4 3 21 17 16 15 11 2 1 12 16 8 21 23 9 20 | 28 |
| | | Mei | 19 7 2 13 10 22 20 24 12 11 8 16 13 4 22 7 20 11 23 1 3 21 9 16 10 3 20 6 21 24 8 8 | 32 |
| | | Juni | 12 18 1 13 16 13 22 15 10 19 16 23 17 10 15 21 8 14 11 19 8 4 2 18 16 12 3 7 20 20 3 15 | 32 |
| | | Juli | 18 9 5 8 21 15 13 18 13 22 12 23 6 4 10 23 5 21 24 14 7 9 24 4 12 22 5 10 | 28 |
| | | Agust | 2 8 22 7 13 2 23 24 1 17 13 20 6 6 23 4 18 4 7 12 8 16 15 14 | 24 |
| | | Sept | 15 14 19 13 21 23 23 3 6 3 14 24 19 10 7 20 22 14 1 9 | 20 |
| | 2 | Okt | 23 4 5 8 9 4 2 2 12 5 12 24 18 10 22 8 19 16 13 16 15 17 4 8 19 3 10 21 | 28 |
| | | Nov | 2 1 12 18 5 12 16 2 18 4 10 22 7 6 5 9 24 1 23 3 13 2 9 20 8 4 20 16 13 1 21 18 | 32 |
| | | Des | 19 22 18 5 4 8 2 7 14 16 21 23 14 10 1 3 13 23 8 5 3 20 1 19 22 17 15 11 10 12 13 12 | 32 |
| | | Jan | 8 19 22 14 2 3 18 12 5 21 2 19 17 5 13 1 13 6 1 15 6 2 10 16 2 21 5 8 | 28 |
| | | Feb | 23 10 17 12 11 15 14 24 6 1 2 16 22 8 24 2 12 21 20 23 14 1 5 3 | 24 |
| | | Maret | 24 8 10 11 10 1 12 5 8 5 4 1 11 23 9 17 15 10 2 13 | 20 |

Lampiran 2 : Hasil Penjadwalan Terbaik

Hasil Penjadwalan Pemeliharaan 2000 jam
JumlahInspeksi2000 =

Columns 1 through 20

0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0

Columns 21 through 40

1 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0

Columns 41 through 60

1 1 0 0 0 0 0 0 1 0 1 0 0 0 0 1 0 0 1 0

Columns 61 through 80

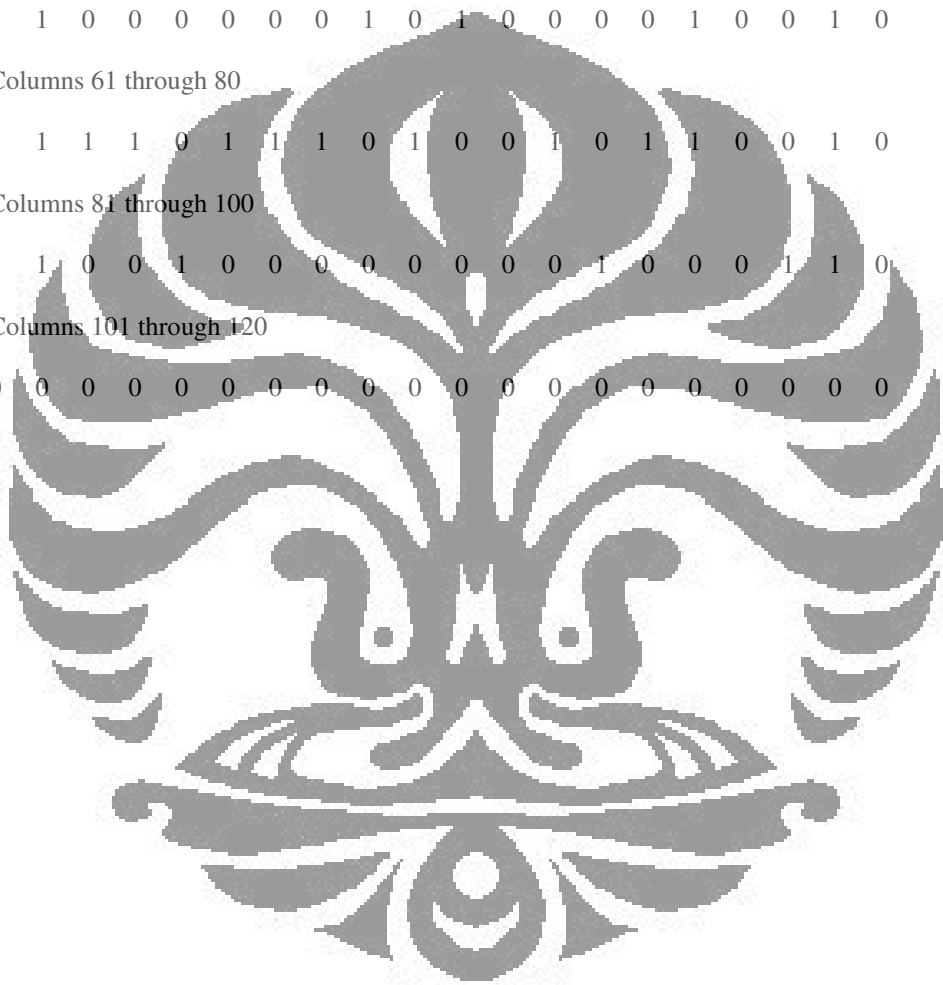
1 1 1 1 0 1 1 1 0 1 0 0 1 0 1 1 0 0 1 0

Columns 81 through 100

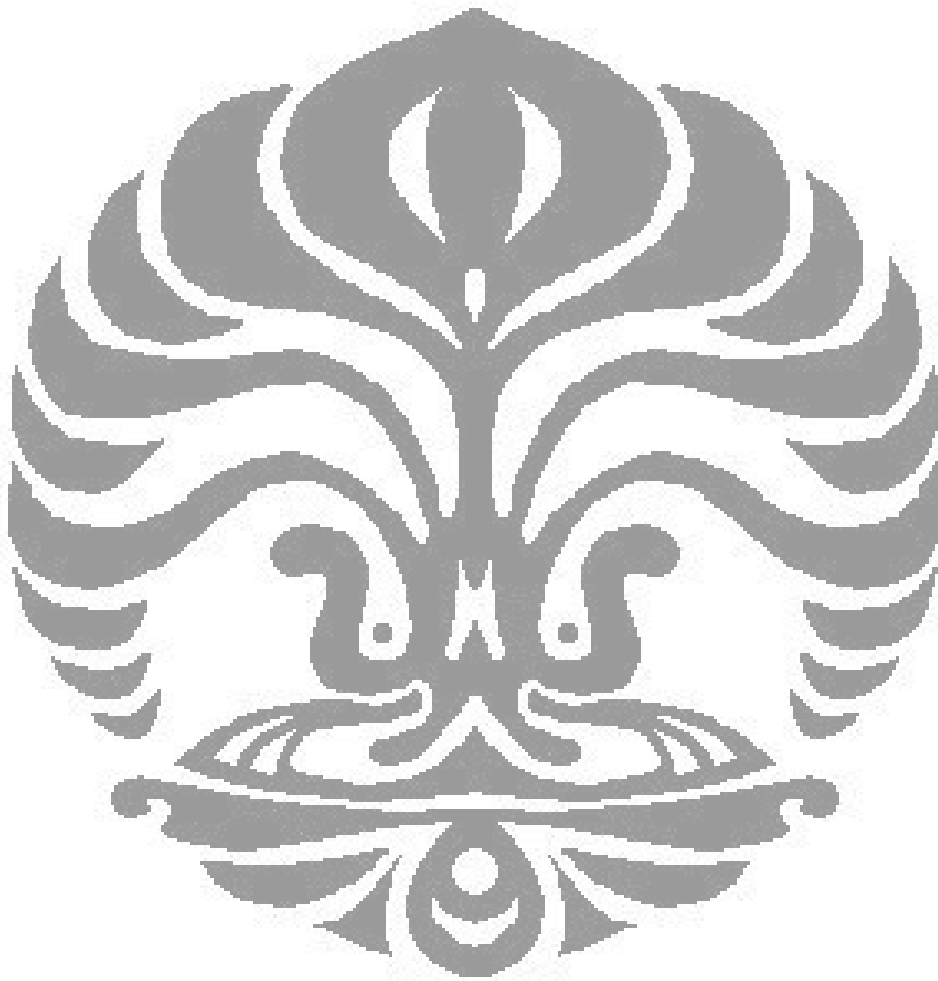
1 1 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 1 1 0

Columns 101 through 120

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0



LAMPIRAN 3
PROGRAM UNTUK VALIDASI



Hasil Terbaik GA

Indeks Populasi Terbaik: 1

StandarDeviasi (fitness) Terbaik: 0.28233

SolusiUrutanPenggunaanRevisi Terbaik: 1 2 3 4 5 6 7 8 9 10 11 12
 13 14 15 16 17 18 19 20 21 22 23 24 1 2 3 4 5 6 7 8 9 10 11
 12 13 14 15 16 17 18 19 20 21 22 23 24 1 2 3 4 5 6 7 8 9 10
 11 12 13 14 15 16 17 18 19 20 21 22 23 24 1 2 3 4 5 6 7 8 9
 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 1 2 3 4 5 6 7 8
 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 1 2 3 4 5 6 7 8
 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 1 2 3 4 5 6 7 8
 9 10 11 12 13 14 15 16 17 19 20 21 22 23 24 1 2 3 4 5 6 7 8 9
 10 11 12 13 14 15 16 17 19 20 21 22 23 24 1 2 3 4 5 6 7 8 9
 10 11 12 13 14 15 16 17 19 20 21 22 23 24 1 2 3 4 5 6 7 8 9
 10 11 12 13 14 15 16 17 19 20 21 22 23 24 1 2 3 4 5 6 7 8 9
 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 1 2 3 4 5 6 7 8
 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 1 2 3 4 5 6 7 8
 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 1 2 3 4 5 6 7 8
 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 1 2 3 4 5 6 7 8
 9 10 11 12 13 14 15 16 17 18 19 20 21 22 24 1 2 3 4 5 6 7 8 9
 10 11 12 13 14 15 16 17 18 19 20 21 22 24 1 2 3 4 5 6 7 8 9
 10 11 12 13 14 15 16 17 18 19 20 21 22 24 1 2 3 4 5 6 7 8 9
 10 11 12 13 14 15 16 17 18 19 20 21 22 24 1 2 3 4 5 6 7 8 9
 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 1 2 3 4 5 6 7 8
 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 1 2 3 4 5 6 7 8
 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 1 2 3 4 5 6 7 8
 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 1 2 3 4 5 6 7 8
 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 1 2 3 4 5 6 7 8
 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 1 2 3 4 5 6 7 8
 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 1 2 3 4 5 6 7 8
 9 10 11 12 13 14 15 16 17

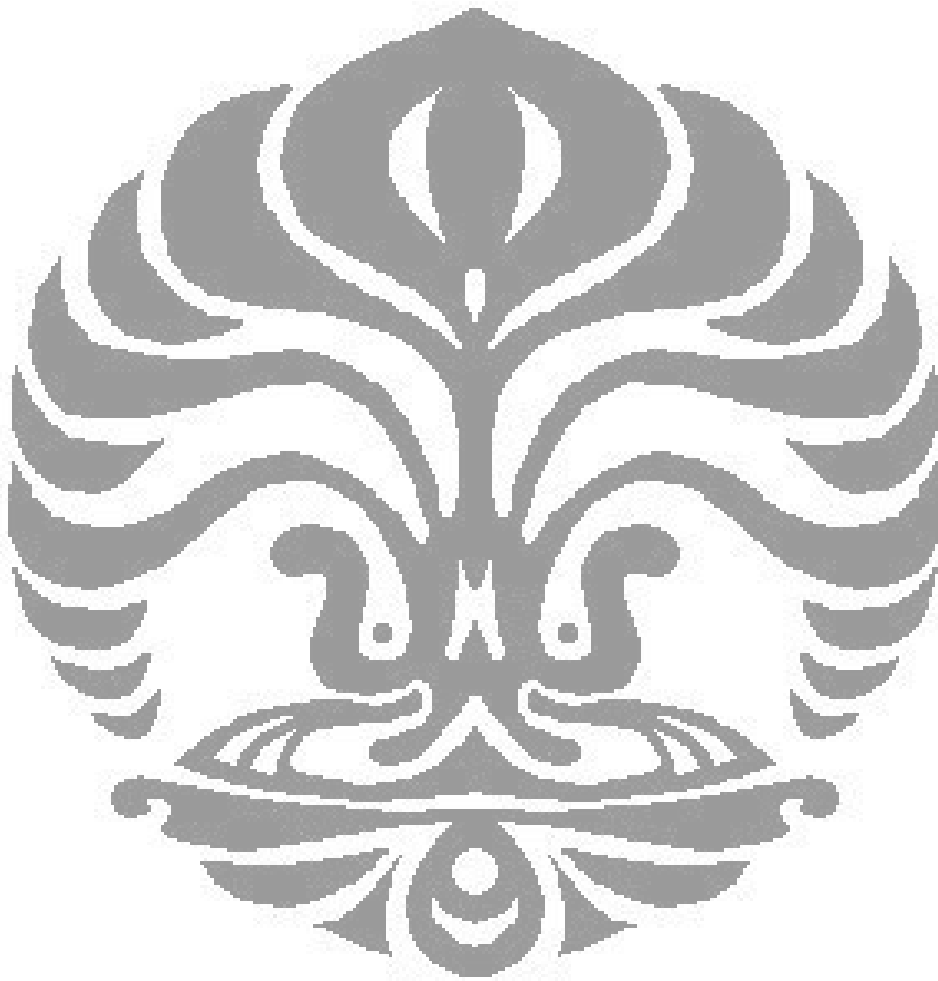
TotalOBJ Terbaik: 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0

JumlahInspeksi2000 Terbaik: 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
 0 0 0

JumlahTambahannyaPesawat Terbaik: 0
 0 0 0 0 0

LAMPIRAN 4

HASIL DOE



 22/12/2009 13:01:41

Welcome to Minitab, press F1 for help.

Results for: Worksheet 2

Multilevel Factorial Design

Factors: 3 Replicates: 10
 Base runs: 27 Total runs: 270
 Base blocks: 1 Total blocks: 1

Number of levels: 3; 3; 3

Design Table (randomized)

| Run | Blk | A | B | C |
|-----|-----|---|---|---|
| 1 | 1 | 2 | 2 | 1 |
| 2 | 1 | 2 | 1 | 2 |
| 3 | 1 | 3 | 2 | 3 |
| 4 | 1 | 2 | 2 | 2 |
| 5 | 1 | 1 | 3 | 3 |
| 6 | 1 | 2 | 3 | 1 |
| 7 | 1 | 2 | 2 | 1 |
| 8 | 1 | 3 | 2 | 1 |
| 9 | 1 | 3 | 1 | 1 |
| 10 | 1 | 1 | 1 | 1 |
| 11 | 1 | 1 | 1 | 3 |
| 12 | 1 | 1 | 2 | 3 |
| 13 | 1 | 1 | 2 | 1 |
| 14 | 1 | 2 | 1 | 3 |
| 15 | 1 | 3 | 2 | 2 |
| 16 | 1 | 3 | 1 | 1 |
| 17 | 1 | 2 | 2 | 2 |
| 18 | 1 | 3 | 2 | 2 |
| 19 | 1 | 2 | 1 | 1 |
| 20 | 1 | 2 | 3 | 2 |
| 21 | 1 | 1 | 2 | 3 |
| 22 | 1 | 1 | 1 | 3 |
| 23 | 1 | 2 | 2 | 3 |
| 24 | 1 | 1 | 3 | 1 |
| 25 | 1 | 1 | 2 | 2 |
| 26 | 1 | 1 | 1 | 3 |
| 27 | 1 | 1 | 2 | 2 |
| 28 | 1 | 2 | 1 | 1 |
| 29 | 1 | 2 | 1 | 2 |
| 30 | 1 | 2 | 3 | 2 |
| 31 | 1 | 2 | 2 | 3 |
| 32 | 1 | 1 | 2 | 2 |
| 33 | 1 | 2 | 2 | 3 |
| 34 | 1 | 2 | 2 | 3 |
| 35 | 1 | 3 | 1 | 3 |
| 36 | 1 | 3 | 3 | 1 |
| 37 | 1 | 1 | 1 | 3 |
| 38 | 1 | 3 | 1 | 2 |
| 39 | 1 | 1 | 3 | 3 |
| 40 | 1 | 2 | 2 | 1 |
| 41 | 1 | 3 | 2 | 1 |
| 42 | 1 | 1 | 1 | 2 |
| 43 | 1 | 2 | 2 | 2 |
| 44 | 1 | 3 | 1 | 2 |

Lampiran 4 : Hasil DOE

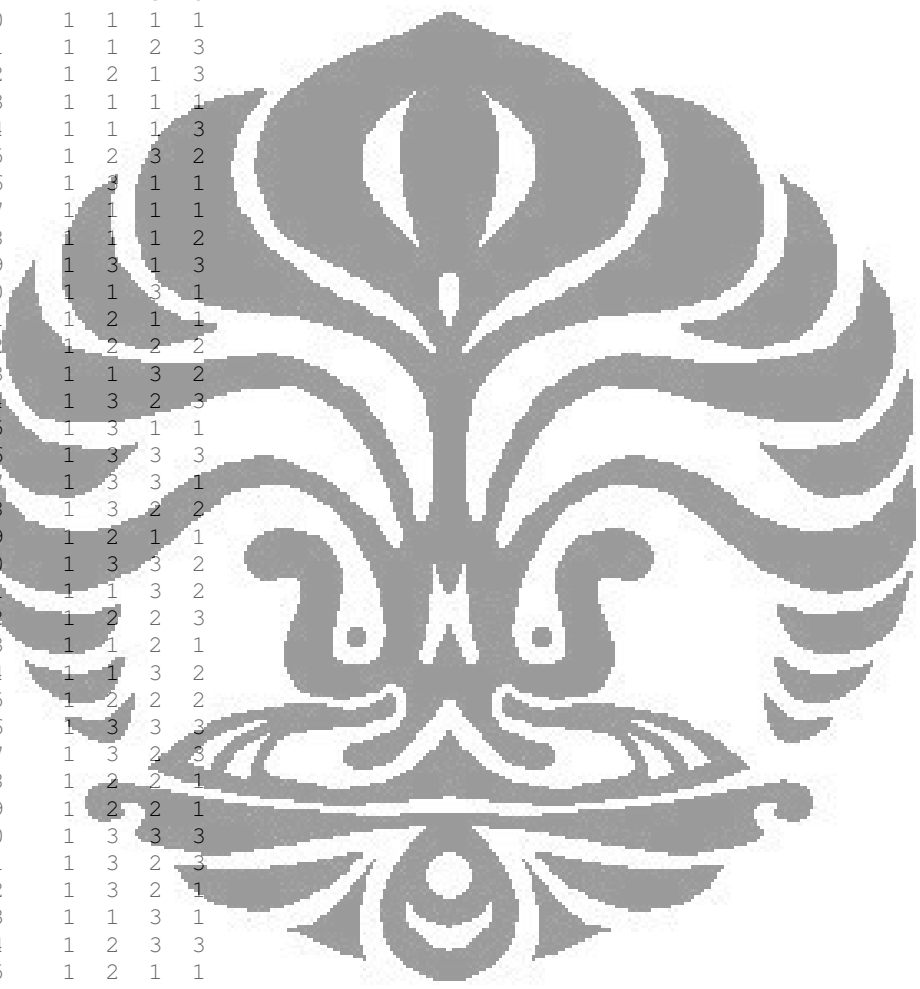
| | | | | |
|-----|---|---|---|---|
| 45 | 1 | 2 | 1 | 3 |
| 46 | 1 | 3 | 2 | 3 |
| 47 | 1 | 2 | 3 | 3 |
| 48 | 1 | 2 | 3 | 3 |
| 49 | 1 | 3 | 2 | 2 |
| 50 | 1 | 1 | 3 | 1 |
| 51 | 1 | 1 | 1 | 1 |
| 52 | 1 | 3 | 1 | 2 |
| 53 | 1 | 1 | 1 | 1 |
| 54 | 1 | 3 | 2 | 3 |
| 55 | 1 | 3 | 1 | 3 |
| 56 | 1 | 1 | 2 | 3 |
| 57 | 1 | 3 | 1 | 3 |
| 58 | 1 | 3 | 1 | 2 |
| 59 | 1 | 2 | 2 | 2 |
| 60 | 1 | 1 | 1 | 2 |
| 61 | 1 | 3 | 2 | 2 |
| 62 | 1 | 1 | 1 | 1 |
| 63 | 1 | 3 | 3 | 3 |
| 64 | 1 | 2 | 2 | 3 |
| 65 | 1 | 3 | 1 | 2 |
| 66 | 1 | 1 | 1 | 2 |
| 67 | 1 | 3 | 3 | 1 |
| 68 | 1 | 3 | 3 | 3 |
| 69 | 1 | 3 | 1 | 3 |
| 70 | 1 | 2 | 2 | 1 |
| 71 | 1 | 1 | 3 | 1 |
| 72 | 1 | 3 | 2 | 2 |
| 73 | 1 | 2 | 1 | 2 |
| 74 | 1 | 2 | 1 | 2 |
| 75 | 1 | 3 | 2 | 1 |
| 76 | 1 | 3 | 3 | 2 |
| 77 | 1 | 1 | 1 | 1 |
| 78 | 1 | 1 | 2 | 1 |
| 79 | 1 | 2 | 1 | 3 |
| 80 | 1 | 3 | 3 | 2 |
| 81 | 1 | 3 | 3 | 2 |
| 82 | 1 | 1 | 2 | 1 |
| 83 | 1 | 3 | 2 | 1 |
| 84 | 1 | 1 | 2 | 1 |
| 85 | 1 | 2 | 2 | 1 |
| 86 | 1 | 1 | 2 | 3 |
| 87 | 1 | 2 | 3 | 3 |
| 88 | 1 | 3 | 3 | 2 |
| 89 | 1 | 1 | 1 | 3 |
| 90 | 1 | 3 | 1 | 3 |
| 91 | 1 | 3 | 2 | 2 |
| 92 | 1 | 3 | 3 | 1 |
| 93 | 1 | 3 | 3 | 3 |
| 94 | 1 | 1 | 1 | 2 |
| 95 | 1 | 1 | 1 | 2 |
| 96 | 1 | 1 | 2 | 2 |
| 97 | 1 | 1 | 1 | 2 |
| 98 | 1 | 3 | 3 | 1 |
| 99 | 1 | 1 | 3 | 2 |
| 100 | 1 | 2 | 3 | 2 |
| 101 | 1 | 1 | 2 | 2 |
| 102 | 1 | 1 | 2 | 3 |
| 103 | 1 | 1 | 3 | 3 |
| 104 | 1 | 3 | 2 | 2 |
| 105 | 1 | 2 | 3 | 1 |
| 106 | 1 | 1 | 1 | 1 |
| 107 | 1 | 2 | 1 | 3 |
| 108 | 1 | 1 | 2 | 2 |
| 109 | 1 | 3 | 3 | 1 |

Lampiran 4 : Hasil DOE

| | | | | |
|-----|---|---|---|---|
| 110 | 1 | 2 | 3 | 2 |
| 111 | 1 | 1 | 3 | 1 |
| 112 | 1 | 1 | 3 | 3 |
| 113 | 1 | 3 | 2 | 3 |
| 114 | 1 | 1 | 3 | 3 |
| 115 | 1 | 1 | 2 | 2 |
| 116 | 1 | 2 | 2 | 1 |
| 117 | 1 | 3 | 3 | 3 |
| 118 | 1 | 2 | 3 | 1 |
| 119 | 1 | 1 | 1 | 3 |
| 120 | 1 | 1 | 1 | 2 |
| 121 | 1 | 3 | 2 | 3 |
| 122 | 1 | 2 | 1 | 1 |
| 123 | 1 | 2 | 1 | 3 |
| 124 | 1 | 2 | 3 | 1 |
| 125 | 1 | 3 | 2 | 1 |
| 126 | 1 | 3 | 1 | 1 |
| 127 | 1 | 2 | 3 | 2 |
| 128 | 1 | 1 | 2 | 1 |
| 129 | 1 | 1 | 1 | 3 |
| 130 | 1 | 3 | 1 | 3 |
| 131 | 1 | 1 | 3 | 2 |
| 132 | 1 | 1 | 3 | 3 |
| 133 | 1 | 2 | 3 | 3 |
| 134 | 1 | 2 | 1 | 1 |
| 135 | 1 | 2 | 1 | 2 |
| 136 | 1 | 3 | 3 | 2 |
| 137 | 1 | 1 | 1 | 3 |
| 138 | 1 | 1 | 3 | 2 |
| 139 | 1 | 1 | 1 | 1 |
| 140 | 1 | 1 | 2 | 3 |
| 141 | 1 | 2 | 2 | 2 |
| 142 | 1 | 2 | 1 | 3 |
| 143 | 1 | 2 | 1 | 1 |
| 144 | 1 | 1 | 2 | 1 |
| 145 | 1 | 2 | 2 | 3 |
| 146 | 1 | 2 | 3 | 3 |
| 147 | 1 | 2 | 3 | 2 |
| 148 | 1 | 2 | 3 | 1 |
| 149 | 1 | 2 | 1 | 2 |
| 150 | 1 | 1 | 3 | 1 |
| 151 | 1 | 1 | 2 | 3 |
| 152 | 1 | 1 | 2 | 2 |
| 153 | 1 | 3 | 3 | 2 |
| 154 | 1 | 3 | 3 | 2 |
| 155 | 1 | 3 | 1 | 2 |
| 156 | 1 | 2 | 1 | 2 |
| 157 | 1 | 3 | 2 | 2 |
| 158 | 1 | 2 | 1 | 1 |
| 159 | 1 | 2 | 1 | 3 |
| 160 | 1 | 1 | 3 | 2 |
| 161 | 1 | 3 | 2 | 3 |
| 162 | 1 | 3 | 2 | 3 |
| 163 | 1 | 3 | 1 | 3 |
| 164 | 1 | 3 | 2 | 1 |
| 165 | 1 | 1 | 3 | 2 |
| 166 | 1 | 3 | 1 | 2 |
| 167 | 1 | 1 | 2 | 1 |
| 168 | 1 | 1 | 1 | 2 |
| 169 | 1 | 2 | 1 | 2 |
| 170 | 1 | 1 | 3 | 1 |
| 171 | 1 | 2 | 3 | 3 |
| 172 | 1 | 3 | 3 | 2 |
| 173 | 1 | 2 | 2 | 2 |
| 174 | 1 | 2 | 3 | 1 |

Lampiran 4 : Hasil DOE

| | | | | |
|-----|---|---|---|---|
| 175 | 1 | 3 | 1 | 1 |
| 176 | 1 | 3 | 2 | 1 |
| 177 | 1 | 2 | 2 | 1 |
| 178 | 1 | 3 | 3 | 1 |
| 179 | 1 | 2 | 1 | 2 |
| 180 | 1 | 2 | 1 | 1 |
| 181 | 1 | 2 | 2 | 3 |
| 182 | 1 | 1 | 1 | 2 |
| 183 | 1 | 1 | 2 | 3 |
| 184 | 1 | 3 | 1 | 1 |
| 185 | 1 | 3 | 2 | 1 |
| 186 | 1 | 1 | 3 | 3 |
| 187 | 1 | 3 | 1 | 2 |
| 188 | 1 | 2 | 1 | 2 |
| 189 | 1 | 2 | 3 | 3 |
| 190 | 1 | 1 | 1 | 1 |
| 191 | 1 | 1 | 2 | 3 |
| 192 | 1 | 2 | 1 | 3 |
| 193 | 1 | 1 | 1 | 1 |
| 194 | 1 | 1 | 1 | 3 |
| 195 | 1 | 2 | 3 | 2 |
| 196 | 1 | 3 | 1 | 1 |
| 197 | 1 | 1 | 1 | 1 |
| 198 | 1 | 1 | 1 | 2 |
| 199 | 1 | 3 | 1 | 3 |
| 200 | 1 | 1 | 3 | 1 |
| 201 | 1 | 2 | 1 | 1 |
| 202 | 1 | 2 | 2 | 2 |
| 203 | 1 | 1 | 3 | 2 |
| 204 | 1 | 3 | 2 | 3 |
| 205 | 1 | 3 | 1 | 1 |
| 206 | 1 | 3 | 3 | 3 |
| 207 | 1 | 3 | 3 | 1 |
| 208 | 1 | 3 | 2 | 2 |
| 209 | 1 | 2 | 1 | 1 |
| 210 | 1 | 3 | 3 | 2 |
| 211 | 1 | 1 | 3 | 2 |
| 212 | 1 | 2 | 2 | 3 |
| 213 | 1 | 1 | 2 | 1 |
| 214 | 1 | 1 | 3 | 2 |
| 215 | 1 | 2 | 2 | 2 |
| 216 | 1 | 3 | 3 | 3 |
| 217 | 1 | 3 | 2 | 3 |
| 218 | 1 | 2 | 2 | 1 |
| 219 | 1 | 2 | 2 | 1 |
| 220 | 1 | 3 | 3 | 3 |
| 221 | 1 | 3 | 2 | 3 |
| 222 | 1 | 3 | 2 | 1 |
| 223 | 1 | 1 | 3 | 1 |
| 224 | 1 | 2 | 3 | 3 |
| 225 | 1 | 2 | 1 | 1 |
| 226 | 1 | 3 | 3 | 3 |
| 227 | 1 | 2 | 2 | 2 |
| 228 | 1 | 2 | 2 | 3 |
| 229 | 1 | 2 | 3 | 2 |
| 230 | 1 | 1 | 2 | 1 |
| 231 | 1 | 2 | 3 | 2 |
| 232 | 1 | 3 | 3 | 1 |
| 233 | 1 | 1 | 3 | 3 |
| 234 | 1 | 3 | 3 | 3 |
| 235 | 1 | 3 | 1 | 1 |
| 236 | 1 | 2 | 2 | 2 |
| 237 | 1 | 2 | 3 | 3 |
| 238 | 1 | 2 | 3 | 3 |
| 239 | 1 | 2 | 3 | 1 |



240 1 3 1 3
 241 1 3 3 1
 242 1 3 3 1
 243 1 1 2 3
 244 1 2 2 3
 245 1 2 3 2
 246 1 2 2 1
 247 1 1 3 2
 248 1 3 2 1
 249 1 3 1 2
 250 1 1 1 3
 251 1 2 3 1
 252 1 3 1 1
 253 1 2 1 3
 254 1 1 3 2
 255 1 1 3 1
 256 1 1 3 3
 257 1 1 2 1
 258 1 2 3 1
 259 1 1 2 2
 260 1 2 1 3
 261 1 1 2 2
 262 1 3 3 3
 263 1 3 1 3
 264 1 1 3 1
 265 1 3 3 2
 266 1 3 2 2
 267 1 1 3 3
 268 1 2 3 1
 269 1 3 1 2
 270 1 3 1 1

25/12/2009 1:31:11

Welcome to Minitab, press F1 for help.
 Retrieving project from file: 'C:\Program Files\MINITAB 14\Data\skripsi
 std
 deviasi new.MPJ'

Results for: Worksheet 2

General Linear Model: std.deviasi versus Ukpog; MaxG; Pc

| Factor | Type | Levels | Values |
|--------|-------|--------|-------------------|
| Ukpog | fixed | 3 | 30; 50; 100 |
| MaxG | fixed | 3 | 1000; 5000; 10000 |
| Pc | fixed | 3 | 0,50; 0,70; 0,95 |

Analysis of Variance for std.deviasi, using Adjusted SS for Tests

| Source | DF | Seq SS | Adj SS | Adj MS | F | P |
|---------------|-----|-----------|-----------|-----------|-------|-------|
| Ukpog | 2 | 0,0070650 | 0,0070650 | 0,0035325 | 24,52 | 0,000 |
| MaxG | 2 | 0,0006393 | 0,0006393 | 0,0003196 | 2,22 | 0,111 |
| Pc | 2 | 0,0133567 | 0,0133567 | 0,0066784 | 46,36 | 0,000 |
| Ukpog*MaxG | 4 | 0,0007121 | 0,0007121 | 0,0001780 | 1,24 | 0,296 |
| Ukpog*Pc | 4 | 0,0028930 | 0,0028930 | 0,0007233 | 5,02 | 0,001 |
| MaxG*Pc | 4 | 0,0002678 | 0,0002678 | 0,0000669 | 0,46 | 0,762 |
| Ukpog*MaxG*Pc | 8 | 0,0038566 | 0,0038566 | 0,0004821 | 3,35 | 0,001 |
| Error | 243 | 0,0350051 | 0,0350051 | 0,0001441 | | |

Lampiran 4 : Hasil DOE

Total 269 0,0637956

S = 0,0120022 R-Sq = 45,13% R-Sq(adj) = 39,26%

Unusual Observations for std.deviasi

| Obs | std.deviasi | Fit | SE Fit | Residual | St Resid |
|-----|-------------|----------|----------|-----------|----------|
| 41 | 0,484420 | 0,461113 | 0,003795 | 0,023307 | 2,05 R |
| 53 | 0,448390 | 0,475938 | 0,003795 | -0,027548 | -2,42 R |
| 59 | 0,497610 | 0,465044 | 0,003795 | 0,032566 | 2,86 R |
| 96 | 0,448390 | 0,472172 | 0,003795 | -0,023782 | -2,09 R |
| 105 | 0,444070 | 0,471234 | 0,003795 | -0,027164 | -2,39 R |
| 108 | 0,497610 | 0,472172 | 0,003795 | 0,025438 | 2,23 R |
| 120 | 0,497610 | 0,472178 | 0,003795 | 0,025432 | 2,23 R |
| 152 | 0,448390 | 0,472172 | 0,003795 | -0,023782 | -2,09 R |
| 158 | 0,458270 | 0,482442 | 0,003795 | -0,024172 | -2,12 R |
| 205 | 0,439550 | 0,463062 | 0,003795 | -0,023512 | -2,06 R |
| 214 | 0,448390 | 0,478301 | 0,003795 | -0,029911 | -2,63 R |
| 216 | 0,493590 | 0,461404 | 0,003795 | 0,032186 | 2,83 R |
| 247 | 0,501470 | 0,478301 | 0,003795 | 0,023169 | 2,03 R |
| 251 | 0,497610 | 0,471234 | 0,003795 | 0,026376 | 2,32 R |
| 254 | 0,501470 | 0,478301 | 0,003795 | 0,023169 | 2,03 R |
| 257 | 0,501470 | 0,475520 | 0,003795 | 0,025950 | 2,28 R |

R denotes an observation with a large standardized residual.

Least Squares Means for std.deviasi

| Ukpop | Mean | SE Mean |
|------------|--------|----------|
| 30 | 0,4694 | 0,001265 |
| 50 | 0,4643 | 0,001265 |
| 100 | 0,4570 | 0,001265 |
| MaxG | | |
| 1000 | 0,4649 | 0,001265 |
| 5000 | 0,4614 | 0,001265 |
| 10000 | 0,4644 | 0,001265 |
| Pc | | |
| 0,50 | 0,4721 | 0,001265 |
| 0,70 | 0,4637 | 0,001265 |
| 0,95 | 0,4549 | 0,001265 |
| Ukpop*MaxG | | |
| 30 1000 | 0,4705 | 0,002191 |
| 30 5000 | 0,4690 | 0,002191 |
| 30 10000 | 0,4689 | 0,002191 |
| 50 1000 | 0,4653 | 0,002191 |
| 50 5000 | 0,4634 | 0,002191 |
| 50 10000 | 0,4641 | 0,002191 |
| 100 1000 | 0,4589 | 0,002191 |
| 100 5000 | 0,4518 | 0,002191 |
| 100 10000 | 0,4603 | 0,002191 |
| Ukpop*Pc | | |
| 30 0,50 | 0,4779 | 0,002191 |
| 30 0,70 | 0,4742 | 0,002191 |
| 30 0,95 | 0,4562 | 0,002191 |
| 50 0,50 | 0,4755 | 0,002191 |
| 50 0,70 | 0,4624 | 0,002191 |
| 50 0,95 | 0,4549 | 0,002191 |
| 100 0,50 | 0,4629 | 0,002191 |
| 100 0,70 | 0,4545 | 0,002191 |
| 100 0,95 | 0,4536 | 0,002191 |
| MaxG*Pc | | |
| 1000 0,50 | 0,4738 | 0,002191 |

Lampiran 4 : Hasil DOE

| | | | | |
|---------------|-------|------|--------|----------|
| 1000 | 0,70 | | 0,4632 | 0,002191 |
| 1000 | 0,95 | | 0,4576 | 0,002191 |
| 5000 | 0,50 | | 0,4699 | 0,002191 |
| 5000 | 0,70 | | 0,4629 | 0,002191 |
| 5000 | 0,95 | | 0,4514 | 0,002191 |
| 10000 | 0,50 | | 0,4727 | 0,002191 |
| 10000 | 0,70 | | 0,4649 | 0,002191 |
| 10000 | 0,95 | | 0,4557 | 0,002191 |
| Ukpop*MaxG*Pc | | | | |
| 30 | 1000 | 0,50 | 0,4759 | 0,003795 |
| 30 | 1000 | 0,70 | 0,4722 | 0,003795 |
| 30 | 1000 | 0,95 | 0,4633 | 0,003795 |
| 30 | 5000 | 0,50 | 0,4755 | 0,003795 |
| 30 | 5000 | 0,70 | 0,4722 | 0,003795 |
| 30 | 5000 | 0,95 | 0,4593 | 0,003795 |
| 30 | 10000 | 0,50 | 0,4823 | 0,003795 |
| 30 | 10000 | 0,70 | 0,4783 | 0,003795 |
| 30 | 10000 | 0,95 | 0,4460 | 0,003795 |
| 50 | 1000 | 0,50 | 0,4824 | 0,003795 |
| 50 | 1000 | 0,70 | 0,4607 | 0,003795 |
| 50 | 1000 | 0,95 | 0,4527 | 0,003795 |
| 50 | 5000 | 0,50 | 0,4730 | 0,003795 |
| 50 | 5000 | 0,70 | 0,4650 | 0,003795 |
| 50 | 5000 | 0,95 | 0,4523 | 0,003795 |
| 50 | 10000 | 0,50 | 0,4712 | 0,003795 |
| 50 | 10000 | 0,70 | 0,4614 | 0,003795 |
| 50 | 10000 | 0,95 | 0,4597 | 0,003795 |
| 100 | 1000 | 0,50 | 0,4631 | 0,003795 |
| 100 | 1000 | 0,70 | 0,4568 | 0,003795 |
| 100 | 1000 | 0,95 | 0,4567 | 0,003795 |
| 100 | 5000 | 0,50 | 0,4611 | 0,003795 |
| 100 | 5000 | 0,70 | 0,4515 | 0,003795 |
| 100 | 5000 | 0,95 | 0,4426 | 0,003795 |
| 100 | 10000 | 0,50 | 0,4645 | 0,003795 |
| 100 | 10000 | 0,70 | 0,4550 | 0,003795 |
| 100 | 10000 | 0,95 | 0,4614 | 0,003795 |