



**TUGAS AKHIR SM 141501**

**PERANCANGAN PROTOTIPE PERANGKAT  
LUNAK UNTUK PENEMPATAN PEGAWAI  
DENGAN MODEL PILIHAN DARI PERSPEKTIF  
DUA ARAH BERBASIS ALGORITMA GENETIKA**

**MOHAMAD MUHTAROMI  
NRP 1210 100 026**

**Dosen Pembimbing  
Prof. Dr. Mohammad Isa Irawan, MT**

**Jurusan Matematika  
Fakultas Matematika dan Ilmu Pengetahuan Alam  
Institut Teknologi Sepuluh Nopember  
Surabaya 2015**



FINAL PROJECT SM 141501

***DESIGN OF SOFTWARE PROTOTYPE FOR  
EMPLOYEE PLACEMENT WITH PERSPECTIVE  
OF TWO-WAY CHOICE MODEL BASED ON  
GENETIC ALGORITHM***

MOHAMAD MUHTAROMI  
NRP 1210 100 026

Supervisor  
Prof. Dr. Mohammad Isa Irawan, MT

Department of Mathematics  
Faculty of Mathematics and Sciences  
Sepuluh Nopember Institute of Technology  
Surabaya 2015

**LEMBAR PENGESAHAN**

**PERANCANGAN PROTOTIPE PERANGKAT LUNAK  
UNTUK PENEMPATAN PEGAWAI DENGAN MODEL  
PILIHAN DARI PERSPEKTIF DUA ARAH BERBASIS  
ALGORITMA GENETIKA**

***DESIGN OF SOFTWARE PROTOTYPE FOR  
EMPLOYEE PLACEMENT WITH PERSPECTIVE OF  
TWO WAY CHOICE MODEL BASED ON GENETIC  
ALGORITHM***

**TUGAS AKHIR**

Diajukan untuk memenuhi salah satu syarat  
Untuk memperoleh gelar Sarjana Sains  
Pada bidang studi Ilmu Komputer  
Program Studi S-1 Jurusan Matematika  
Fakultas Matematika dan Ilmu Pengetahuan Alam  
Institut Teknologi Sepuluh Nopember Surabaya

Oleh :

**MOHAMAD MUHTAROMI**

**NRP. 1210100 026**

Menyetujui,

Dosen Pembimbing,

Prof. Dr. Mohammad Isa Irawan, MT

NIP. 19631225198903 1 001

Mengetahui,

Ketua Jurusan Matematika  
Fakultas Matematika dan Ilmu Pengetahuan Alam ITS

Prof. Dr. Echa Apriliani, M.Si

NIP. 19660314 199102 2 001

JURUSAN Surabaya, Juli 2015

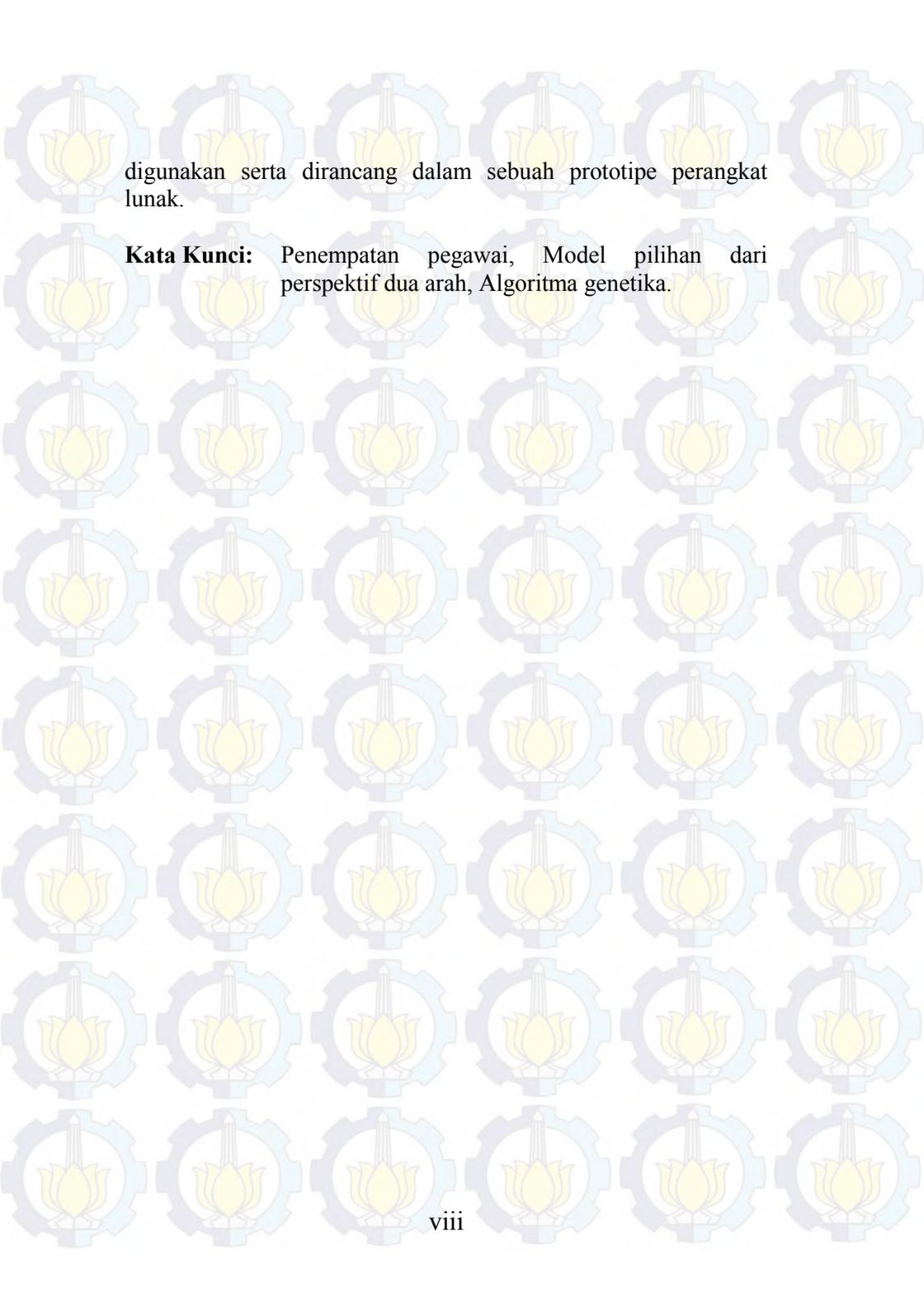


# **PERANCANGAN PROTOTIPE PERANGKAT LUNAK UNTUK PENEMPATAN PEGAWAI DENGAN MODEL PILIHAN DARI PERSPEKTIF DUA ARAH BERBASIS ALGORITMA GENETIKA**

**Nama Mahasiswa : Mohamad Muhtaromi**  
**NRP : 1210 100 026**  
**Jurusan : Matematika FMIPA-ITS**  
**Dosen Pembimbing : Prof. Dr. Mohammad Isa Irawan,  
MT**

## **Abstrak**

Dalam dunia manajemen sumber daya manusia di perusahaan, penempatan pegawai membutuhkan identifikasi dan proses seleksi yang baik untuk memilih secara tepat dari pegawai yang mempunyai keterampilan yang sesuai, pengetahuan dan pengalaman. Salah satu kasus dalam penempatan pegawai adalah setelah proses seleksi dan penerimaan pegawai telah dilakukan, pihak perusahaan memberikan kesempatan kepada pegawai untuk memilih departemen yang tersedia dalam perusahaan sebagai tujuan pribadinya dan disisi lain perusahaan yang memegang keputusan untuk menempatkan pegawai tersebut kedalam departemen yang sesuai. Namun, dalam prosesnya, sering tidak mencapai hasil yang diinginkan karena kurangnya metode ilmiah yang memenuhi kelayakan kebutuhan perusahaan. Permasalahan ini dapat menjadi suatu model yang disebut sebagai model pilihan dari perspektif dua arah. Untuk menyelesaikan model tersebut digunakanlah Algoritma Genetika sebagai algoritma pencarian yang didasarkan pada mekanisme seleksi. Dalam hal ini, Algoritma Genetika digunakan untuk mengoptimalkan parameter model tersebut dengan memaksimalkan fungsinya. Pada akhir penelitian ditunjukkan bahwa Algoritma Genetika dapat digunakan untuk menyelesaikan model pilihan dari perspektif dua arah dan didapatkan hasil formasi pegawai pada studi kasus yang



digunakan serta dirancang dalam sebuah prototipe perangkat lunak.

**Kata Kunci:** Penempatan pegawai, Model pilihan dari perspektif dua arah, Algoritma genetika.

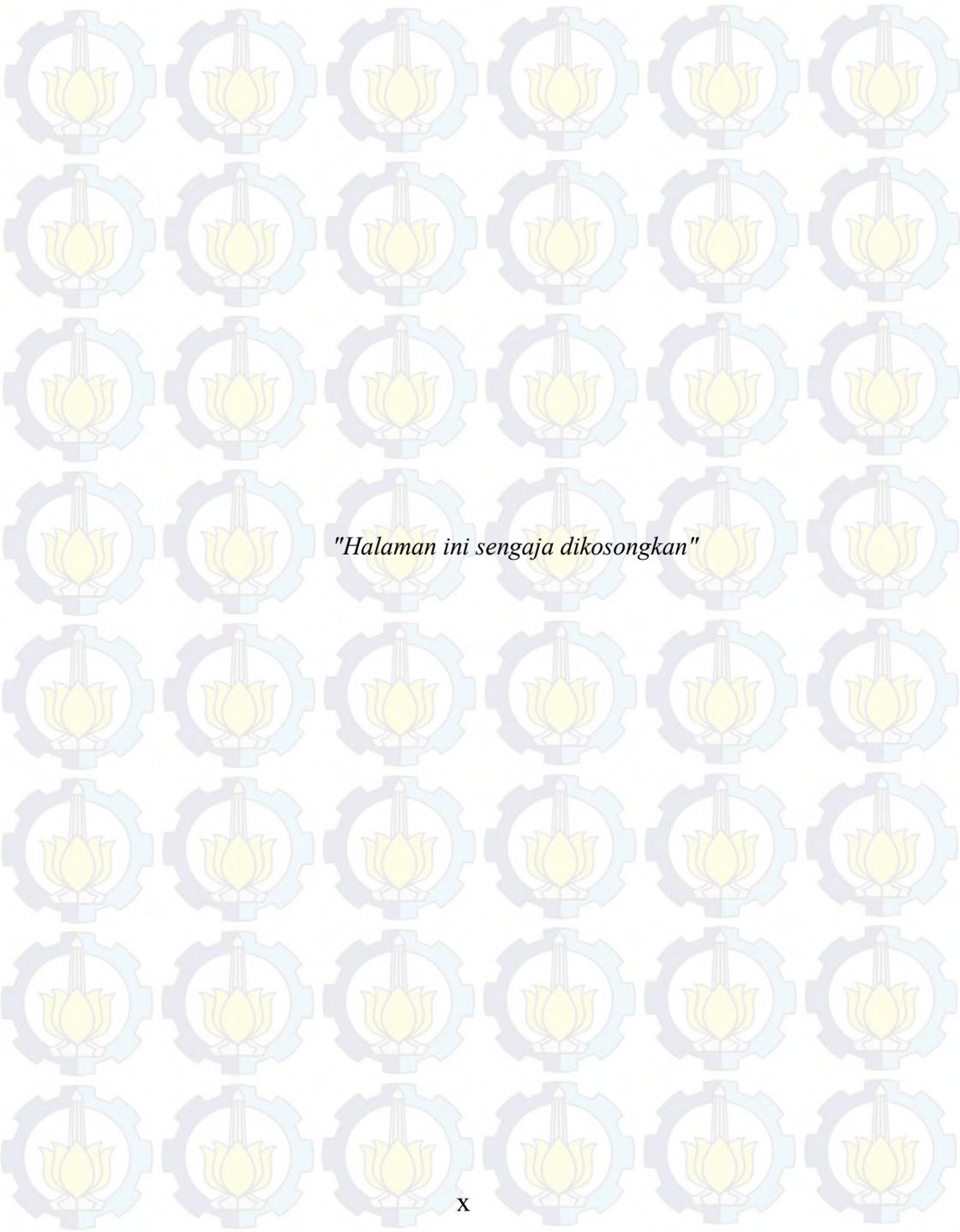
# DESIGN OF SOFTWARE PROTOTYPE FOR EMPLOYEE PLACEMENT WITH PERSPECTIVE OF TWO-WAY CHOICE MODEL BASED ON GENETIC ALGORITHM

**Name** : Mohamad Muhtaromi  
**NRP** : 1210 100 026  
**Department** : Mathematics FMIPA-ITS  
**Counsellor** : Prof. Dr. Mohammad Isa Irawan,  
MT

## *Abstract*

*Disposition refer to human resource management in a company, assigning the employee requires the identification and selection process to choose the appropriate of employees in according with the skills, knowledge and experience. A case in assigning the employee is after the selection and recruitment process has been carried out, the company provides the opportunity for employees to choose an available department within the company as their own personal goals and on the other hand, the company determine to puts the employee into the appropriate department. However, in the process, the company often does not achieve the desired results due to the lack of scientific methods that meet the eligibility requirements of the company. This problem can be a model called as Perspective of Two-way Choice Model. To complete the model, it is used the genetic algorithm as searching algorithm based on the mechanism of natural selection and genetics. In this case, the Genetic Algorithm is one of optimization techniques used to optimize the model parameters by maximizing function. In the end of the study, it is indicated that Genetic Algorithm can be used to complete the selection of models from the two-way perspective, showed the formation of employees used in the case studies and designed in a software prototype.*

**Keywords** : *Employee Assignment, Perspective of Two-way Choice Model, Genetic Algorithm.*



*"Halaman ini sengaja dikosongkan"*

## KATA PENGANTAR

Alhamdulillah Robbil ‘aalamiin, segala puji dan syukur senantiasa terpanjatkan atas kehadiran Allah SWT, yang telah melimpahkan selalu rahmat-Nya, sehingga penyusunan Tugas Akhir yang berjudul: **“Perancangan Prototipe Perangkat Lunak untuk Penempatan Pegawai dengan Model Pilihan dari Perspektif Dua-Arah Berbasis Algoritma Genetika”**, dapat terselesaikan dengan baik.

Salah satu tujuan dari disusunnya Tugas Akhir ini adalah untuk memenuhi sebagian persyaratan dalam mencapai jenjang Sarjana Sains dari Jurusan Matematika FMIPA ITS. Untuk itu penulis mengucapkan terima kasih kepada:

1. Bapak Prof. Dr. Mohammad Isa Irawan, MT selaku Dosen Pembimbing.
2. Ibu Dian Winda Setyawati S.Si, M.Si selaku Dosen Wali.
3. Prof. Dr. Erna Apriliani, M.Si selaku ketua Jurusan Matematika ITS.
4. Bapak Ibu Dosen Penguji, Alvida Mustika Rukmi S.Si, M.Si, Dr. Imam Mukhlash, S.Si, MT, Dr. Budi Setyono, S.Si, MT, dan Dr. Dwi Ratna Sulistyaningrum, S.Si, MT.
5. Dr. Chairul Imron, MI.Komp. selaku koordinator Tugas Akhir.
6. Bapak Ibu Dosen serta seluruh staf Tata Usaha Jurusan Matematika ITS.
7. Seluruh teman-teman mahasiswa Jurusan Matematika ITS.

Penulis menyadari bahwa dalam penyusunan Tugas Akhir ini masih jauh dari sempurna. Oleh karena itu, penulis sangat mengharapkan saran dan kritik dari pembaca. Harapan penulis semoga Tugas Akhir ini bermanfaat bagi semua pihak yang berkepentingan.

Surabaya, Juli 2015

**Penulis**

### *Special thanks to*

Selama proses pembuatan tugas akhir ini, banyak pihak yang telah memberikan bantuan dan dukungan untuk penulis. Untuk itu penulis sungguh ingin mengucapkan terima kasih kepada:

1. Bapak dan Ibu tercinta. Bapak Shohibul Ilmi dan Ibu Suci Hati Ilmiah yang senantiasa mendoakan dan menasehati penulis.
2. Kakak penulis, Mau'idhatul Chasanah yang selalu cerewet dan kritis terhadap segala sesuatu yang dilakukan penulis.
3. A. Sirojuddin "Tux/Totot" Lutfi dan Danang W. "DFC" Wicaksono, duo totot yang memberikan bantuan terbesar bagi penulis. Juga M. Athoillah yang selalu bersaing soal Bola dengan penulis dan juga bantuannya dalam memahami Matlab.
4. Sahabat-sahabat baik penulis, Indira "Gembel" Anggriani yang sealiran soal musik dan selalu mendukung penulis, Firdha Dwi S. "Sephia" Zainal yang selalu aneh dan gak jelas tingkahnya namun selalu peduli, M. Ridho "Otong" Bintang yang selalu membutuhkan penulis sebagai TB, dan tentu saja Arifatul Masruroh dengan segala dukungan dan perhatiannya.
5. Teman-teman seperjuangan di Lane, Zaki, Ridho, Andi, Ahmad, Khoir, Omdon, Suripin, Bang Sat, Supandi, Mas Angga, Polo, Reza, dll. yang telah menanamkan motto legendaris "Jangan sampai TA mu mengganggu DoTA mu".
6. Kabinet Harmonis Himatika ITS serta anak-anakku Dept. PSDM kepengurusan periode 2012/2013 yang telah memberikan motivasi, semangat belajar dan kesungguhan.
7. Semua yang ada didalam Himatika ITS yang telah mengajarkan arti sebuah keluarga.

Tentu saja masih banyak pihak lain yang turut andil dalam penyelesaian tugas akhir ini yang tidak bisa penulis sebutkan satu persatu. Semoga Allah membalas dengan balasan yang lebih baik bagi semua pihak yang telah membantu penulis. *Aamiin yaa rabbal 'aalamiin.*

## DAFTAR ISI

	Halaman
<b>HALAMAN JUDUL</b> .....	i
<b>LEMBAR PENGESAHAN</b> .....	v
<b>ABSTRAK</b> .....	vii
<b>ABSTRACT</b> .....	ix
<b>KATA PENGANTAR</b> .....	xi
<b>DAFTAR ISI</b> .....	xiii
<b>DAFTAR GAMBAR</b> .....	xv
<b>DAFTAR TABEL</b> .....	xvii
<b>DAFTAR LAMPIRAN</b> .....	xxi
<b>BAB I. PENDAHULUAN</b>	
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah.....	2
1.4 Tujuan.....	3
1.5 Manfaat.....	3
1.6 Sistematika Penulisan Tugas Akhir.....	4
<b>BAB II. TINJAUAN PUSTAKA</b>	
2.1 Prototipe Perangkat Lunak.....	7
2.2 Penempatan Pegawai.....	8
2.3 Model Pilihan dari Perspektif Dua Arah.....	9
2.4 <i>Analytic Hierarchy Process (AHP)</i> .....	11
2.5 Algoritma Genetika (AG).....	15
2.5.1 Jenis-jenis AG.....	16
2.5.2 Komponen-Komponen AG.....	16
<b>BAB III. METODOLOGI PENELITIAN</b>	
3.1 Tahapan Penelitian.....	21
3.2 Diagram Alir Penelitian.....	23
<b>BAB IV. PERANCANGAN DAN IMPLEMENTASI SISTEM</b>	
4.1 Perhitungan Bobot Setiap Kriteria dan Alternatif-Alternatifnya.....	25

4.1.1 Mendefinisikan Struktur Hierarki .....	26
4.1.2 Membuat Matriks Perbandingan Berpasangan .....	26
4.1.3 Normalisasi Matriks .....	27
4.1.4 Menentukan Bobot Masing-Masing Kriteria .....	28
4.1.5 Menghitung Indeks Konsistensi (CI) dan Rasio Konsistensi (CR) Untuk Matriks Perbandingan Berpasangan Departemen....	29
4.1.6 Menghitung Bobot Masing-Masing Alternatif Kualifikasi Sejumlah Kriteria Departemen .....	29
4.2 Analisis Sistem .....	35
4.2.1 Deskripsi Perangkat Lunak .....	36
4.2.2 <i>Conditional Analysis</i> (Analisis Persyaratan) .....	36
4.2.3 <i>Use-case Diagram</i> .....	41
4.2.4 <i>Activity/Swimlane Diagram</i> .....	43
4.3 Perancangan Sistem .....	48
4.3.1 Perancangan Proses .....	48
4.3.2 Perancangan <i>Database</i> .....	56
4.3.3 Proses Prototyping Perangkat Lunak .....	60
<b>BAB V. UJI COBA DAN ANALISIS HASIL</b>	
5.1 Uji Tampilan Antarmuka Pengguna .....	73
5.1.1 Tampilan Antarmuka untuk User.....	73
5.1.2 Tampilan Antarmuka untuk Admin .....	79
5.2 Uji Coba Program Penempatan Pegawai .....	82
5.3 Analisis Hasil dan Uji Coba Parameter.....	87
5.4 Validasi Hasil Penempatan Pegawai.....	89
<b>BAB VI. PENUTUP</b>	
6.1 Kesimpulan .....	91
6.2 Saran .....	91
<b>DAFTAR PUSTAKA</b> .....	93
<b>LAMPIRAN</b> .....	95

## DAFTAR GAMBAR

	Halaman
Gambar 2.1 Model Prototipe .....	8
Gambar 2.2 Contoh Salah Satu Proses <i>Staffing</i> Pada Organisasi .....	9
Gambar 2.3 Diagram Alir Algoritma Genetika .....	20
Gambar 3.1 Diagram Alir Penelitian .....	23
Gambar 4.1 Hierarki Penempatan Pegawai .....	26
Gambar 4.2 <i>Use-case Diagram</i> .....	42
Gambar 4.3 <i>Activity Diagram</i> “Tambah Data Pegawai” .....	44
Gambar 4.4 <i>Activity Diagram</i> “Edit dan Hapus Data Pegawai” .....	45
Gambar 4.5 <i>Activity Diagram</i> “Program Penempatan Pegawai” .....	46
Gambar 4.6 <i>Activity Diagram</i> “Data Departemen dan Kualifikasi” .....	47
Gambar 4.7 Diagram Alir Prototipe PL .....	49
Gambar 4.8 Diagram Alir Program Penempatan Pegawai...	50
Gambar 4.9 Diagram Alir <i>Pre-Process</i> , <i>GA-Process</i> dan <i>Post-Process</i> .....	56
Gambar 4.10 <i>Conceptual Data Model</i> .....	57
Gambar 4.11 <i>Logical Data Model</i> .....	58
Gambar 4.12 <i>Physical Data Model</i> .....	59
Gambar 4.13 Design Database pada MySQL .....	60
Gambar 4.14 Halaman Utama .....	61
Gambar 4.15 Menu dan Submenu Tahap 1 .....	61
Gambar 4.16 Form Panel Perhitungan Penempatan Pegawai	62
Gambar 4.17 Panel Data Nilai A Pegawai .....	62
Gambar 4.18 Panel Data Nilai B Pegawai .....	63
Gambar 4.19 Panel Data Pilihan Pegawai .....	63
Gambar 4.20 Form Tambah Data Pegawai .....	64
Gambar 4.21 Form Login Aplikasi .....	65
Gambar 4.22 Menu dan Submenu Tahap 2 .....	66
Gambar 4.23 Tampilan Submenu View Data .....	66

Gambar 4.24	Input Parameter AG .....	67
Gambar 4.25	Menu dan Submenu Tahap 3 .....	68
Gambar 4.26	Input Parameter dan Start Proses AG Tahap 1 .....	68
Gambar 4.27	Panel Hasil Penempatan Pegawai .....	69
Gambar 4.28	Menu dan Submenu Tahap 4 .....	70
Gambar 4.29	Form Insert Data Pegawai .....	70
Gambar 4.30	Form Edit dan Delete Data Pegawai .....	71
Gambar 4.31	Input Parameter dan Start Proses AG Tahap 2 .....	71
Gambar 4.32	Panel Hasil Penempatan Pegawai.....	72
Gambar 5.1	Form Login .....	74
Gambar 5.2	Kotak Dialog Connection <i>Check</i> .....	74
Gambar 5.3	Tampilan Utama untuk User .....	75
Gambar 5.4	Submenu View Data dengan Tab Data Departemen .....	76
Gambar 5.5	Submenu View Data dengan Tab Data Pegawai .....	76
Gambar 5.6	Submenu View Data dengan Tab Data Pilihan Pegawai .....	77
Gambar 5.7	Submenu View Data dengan Tab Data A & B Pegawai .....	77
Gambar 5.8	Submenu View Data dengan Tab Data Kualifikasi .....	78
Gambar 5.9	Tampilan Insert Data Pegawai .....	78
Gambar 5.10	Tampilan Edit dan Delete Data Pegawai .....	79
Gambar 5.11	Panel Start GA Process .....	79
Gambar 5.12	Panel Hasil Penempatan Pegawai .....	80
Gambar 5.13	Tampilan Halaman Utama untuk Admin .....	80
Gambar 5.14	Submenu Edit Data .....	80
Gambar 5.15	Tampilan Edit Data Departemen .....	81
Gambar 5.16	Tampilan Edit Kualifikasi .....	81
Gambar 5.17	Hasil penempatan pegawai dengan fitness 14,281 .....	89

## DAFTAR TABEL

	Halaman
Tabel 2.1	Skala Penilaian Perbandingan Pasangan ..... 12
Tabel 2.2	Matriks Perbandingan Berpasangan ..... 13
Tabel 2.3	Daftar <i>Random Index</i> (RI) ..... 14
Tabel 2.4	Contoh Penggunaan Metode <i>Roulette-Wheel</i> Dalam Seleksi ..... 17
Tabel 4.1	Departemen dan Kualifikasi Dalam Proses Penempatan Pegawai di PT. Petrokimia Gresik 25
Tabel 4.2	Matriks Perbandingan Departemen ..... 27
Tabel 4.3	Matriks Normalisasi ..... 28
Tabel 4.4	Jumlah Baris dan Bobot Departemen ..... 28
Tabel 4.5	Bobot Kualifikasi dalam Departemen Akuntansi (D1) ..... 30
Tabel 4.6	Bobot Kualifikasi dalam Departemen Anggaran (D2) ..... 30
Tabel 4.7	Bobot Kualifikasi dalam Departemen Keuangan (D3) ..... 31
Tabel 4.8	Bobot Kualifikasi dalam Departemen Pemeliharaan I (D4) ..... 31
Tabel 4.9	Bobot Kualifikasi dalam Departemen Pemeliharaan II (D5) ..... 32
Tabel 4.10	Bobot Kualifikasi dalam Departemen Pemeliharaan III (D6) ..... 32
Tabel 4.11	Bobot Kualifikasi dalam Departemen Pengelolaan Pelabuhan (D7) ..... 33
Tabel 4.12	Bobot Kualifikasi dalam Departemen Prasarana Publik dan Kawasan (D8) ..... 33
Tabel 4.13	Bobot Kualifikasi dalam Departemen Produksi I (D9) ..... 34
Tabel 4.14	Bobot Kualifikasi dalam Departemen Produksi II A (D10) ..... 34
Tabel 4.15	Bobot Kualifikasi dalam Departemen Produksi II B (D11) ..... 35

Tabel 4.16	Alokasi Departemen Oleh Pakar Atas Bidang Yang Dipilih Pegawai .....	37
Tabel 4.17	Pilihan Pegawai .....	38
Tabel 4.18	Nilai A .....	38
Tabel 4.19	Nilai B .....	39
Tabel 4.20	Nilai Minimum Kualifikasi .....	39
Tabel 4.21	Bobot dan Kuota Departemen .....	40
Tabel 4.22	Bobot Kualifikasi Dalam Setiap Departemen ....	40
Tabel 4.23	Interaksi Antara Aktor dengan Sistem .....	42
Tabel 4.24	Contoh Matriks Random dengan Bilangan 0 dan 1 .....	52
Tabel 5.1	Hasil Uji Coba dengan Parameter PC=60%, PM=0,7% dan UP=30 .....	83
Tabel 5.2	Hasil Uji Coba dengan Parameter PC=60%, PM=0,7% dan UP=70 .....	83
Tabel 5.3	Hasil Uji Coba dengan Parameter PC=60%, PM=0,7% dan UP=80 .....	84
Tabel 5.4	Hasil Uji Coba dengan Parameter PC=60%, PM=0,8% dan UP=30 .....	84
Tabel 5.5	Hasil Uji Coba dengan Parameter PC=60%, PM=0,8% dan UP=70 .....	84
Tabel 5.6	Hasil Uji Coba dengan Parameter PC=60%, PM=0,8% dan UP=80 .....	85
Tabel 5.7	Hasil Uji Coba dengan Parameter PC=88%, PM=0,7% dan UP=30 .....	85
Tabel 5.8	Hasil Uji Coba dengan Parameter PC=88%, PM=0,7% dan UP=70 .....	85
Tabel 5.9	Hasil Uji Coba dengan Parameter PC=88%, PM=0,7% dan UP=80 .....	86
Tabel 5.10	Hasil Uji Coba dengan Parameter PC=88%, PM=0,8% dan UP=30 .....	86
Tabel 5.11	Hasil Uji Coba dengan Parameter PC=88%, PM=0,8% dan UP=70 .....	86
Tabel 5.12	Hasil Uji Coba dengan Parameter PC=88%, PM=0,8% dan UP=80 .....	87

Tabel 5.13	Kemunculan Fitness Dominan .....	87
Tabel 5.14	Hasil uji Kestabilan Nilai Fitness .....	88
Tabel 5.15	Penempatan Pegawai oleh Sistem dan Perusahaan .....	90



*“Halaman ini sengaja dikosongkan”*

# BAB I PENDAHULUAN

## 1.1 Latar Belakang Masalah

Istilah "*The right man on the right place and the right man behind the job*" menjadi pedoman dalam manajemen sumber daya manusia terutama dalam penempatan seorang pegawai yang harus didasarkan pada *job description* dan *job specification* yang telah ditentukan [1]. Istilah tersebut juga menggambarkan betapa pentingnya penempatan pegawai dengan keahlian dan kemampuan yang dimiliki sehingga sesuai dengan kualifikasi yang diberikan oleh perusahaan yang bersangkutan. Menempatkan pegawai dalam posisi yang strategis memerlukan perencanaan dan seleksi yang dilakukan secara profesional untuk meningkatkan mutu dan keuntungan perusahaan dalam persaingan global di masa sekarang dan dalam jangka waktu yang panjang.

Proses penempatan pegawai berbeda-beda sesuai kebutuhan masing-masing perusahaan. Namun inti sebenarnya dari penempatan pegawai adalah seorang manajer menempatkan pegawainya ke posisi yang bermacam-macam dibawah kendali perusahaan sehingga terjadi kombinasi antara tujuan perusahaan dengan tujuan pribadi pegawai. Penelitian telah menunjukkan bahwa produktivitas kerja dari pegawai terbaik adalah tiga kali lebih tinggi dari pegawai terburuk pada posisi yang sama. Namun, dalam sebuah perusahaan, perusahaan yang akan mengalokasikan harus memilih pegawai dengan optimal sesuai dengan sistem secara keseluruhan, dan bukan memilih pegawai terbaik ke satu departemen. Ini berarti bahwa penempatan pegawai membutuhkan identifikasi dan proses seleksi yang baik untuk memilih secara tepat dari pegawai yang mempunyai keterampilan, pengetahuan dan pengalaman yang sesuai. Pada saat yang sama penempatan pegawai menjadi pencapaian untuk optimasi sistem organisasi. Namun, dalam proses penempatan pegawai yang sebenarnya, sering tidak mencapai hasil yang diinginkan karena

kurangnya metode ilmiah dan memenuhi kelayakan kebutuhan perusahaan [2].

Berdasarkan permasalahan yang diuraikan diatas maka dalam Tugas Akhir ini akan dibahas tentang pembuatan prototipe perangkat lunak untuk penempatan pegawai berbasis Algoritma Genetika (AG) berdasarkan model pilihan perspektif dua arah. Model tersebut adalah model yang digunakan ketika disatu sisi perusahaan memilih pegawai untuk posisi yang tersedia dan disisi lain pegawai juga memilih posisi yang diinginkan. Proses penempatan pegawai yang dibahas merupakan proses pengambilan keputusan yang menggabungkan tujuan perusahaan dengan tujuan pribadi pegawai. Di satu sisi, proses penempatan pegawai harus memungkinkan perusahaan untuk mencapai tujuan organisasi. Sementara disisi lain, proses penempatan pegawai juga harus membantu pegawai untuk mencapai tujuan pribadinya. Algoritma Genetika dipilih untuk diterapkan pada model tersebut karena AG mempunyai performansi bagus dalam hal optimasi.

## **1.2 Rumusan Masalah**

Berdasarkan latar belakang diatas, rumusan masalah dalam Tugas Akhir ini adalah:

1. Bagaimana menyelesaikan model pilihan perspektif dari dua arah dengan menggunakan Algoritma Genetika.
2. Bagaimana mendapatkan hasil penempatan pegawai dalam proses penempatan pegawai dengan menggunakan Algoritma Genetika.
3. Bagaimana membuat Prototipe Perangkat Lunak untuk proses Penempatan Pegawai.

## **1.3 Batasan Masalah**

Dalam penulisan Tugas Akhir ini, penulis memberikan batasan masalah agar permasalahan yang dibahas tidak terlalu luas dan tidak menyimpang dari pokok permasalahan, yaitu:

1. Lingkup permasalahan yang dibahas dalam tugas akhir ini adalah pada proses penempatan pegawai. Tidak membahas tentang proses seleksi dan penerimaan.
2. Studi kasus yang akan dipakai adalah studi kasus dari PT. Petrokimia Gresik.
3. Data yang digunakan untuk penelitian ini adalah data dari 11 departemen, 11 pegawai dan 7 kualifikasi.
4. Perancangan Prototipe Perangkat Lunak hanya sampai pada tahap evaluasi. Tidak membahas tahap transisi ke Pengembangan Perangkat Lunak.
5. Perhitungan model permasalahan menggunakan Algoritma Genetika.
6. Prototipe perangkat lunak menggunakan Java Netbeans.

#### **1.4 Tujuan**

Tujuan penulisan Tugas Akhir ini adalah:

1. Mendapatkan penyelesaian model pilihan dari perspektif dua arah dalam salah satu kasus penempatan pegawai dengan menggunakan Algoritma Genetika.
2. Mendapatkan hasil penempatan pegawai dalam proses penempatan pegawai dengan menggunakan Algoritma Genetika.
3. Membuat Prototipe Perangkat Lunak untuk proses Penempatan Pegawai.

#### **1.5 Manfaat**

Berdasarkan informasi dan hasil yang diperoleh, penulisan Tugas akhir ini diharapkan akan memberikan manfaat-manfaat sebagai berikut:

1. Menjadi referensi tentang penempatan pegawai dengan model pilihan dari perspektif dua arah bagi Departemen Personalia PT. Petrokimia Gresik.
2. Memberikan informasi secara umum kepada penulis maupun pembaca tentang salah satu kasus proses

penempatan pegawai dengan metode yang telah disebutkan diatas.

3. Hasil penulisan Tugas Akhir ini diharapkan dapat menjadi acuan atau sebagai bahan untuk studi lanjutan yang diperlukan untuk pengembangan yang berkaitan dengan proses penempatan pegawai dengan model pilihan perspektif dua arah, dan Algoritma Genetika.

### **1.6 Sistematika Penulisan Tugas Akhir**

Sistematika penulisan dari Tugas Akhir ini adalah sebagai berikut:

#### **BAB I PENDAHULUAN**

Bab ini berisi tentang gambaran umum dari penulisan tugas akhir ini yang meliputi latar belakang masalah, perumusan masalah, batasan masalah, tujuan dan manfaat penelitian, serta sistematika penulisan.

#### **BAB II TINJAUAN PUSTAKA**

Bab ini berisi tentang dasar teori yang mendukung Tugas Akhir ini, antara lain: Prototipe perangkat lunak, Penempatan pegawai, Model pilihan dari perspektif dua arah, Pembobotan dengan *Analytic Hierarchy Process* dan Algoritma Genetika.

#### **BAB III METODOLOGI PENELITIAN**

Bab ini akan dibahas tentang metode yang digunakan untuk menyelesaikan tugas akhir ini.

#### **BAB IV PERANCANGAN DAN IMPLEMENTASI SISTEM**

Bab ini membahas tentang proses mendapatkan bobot departemen serta alternatif-alternatifnya, proses desain database, perancangan sistem, implementasi sistem menggunakan Algoritma Genetika, dan proses dalam membangun prototipe perangkat lunak.

## BAB V UJI COBA DAN ANALISIS HASIL

Bab ini membahas tentang uji coba tampilan antar muka pengguna, uji coba parameter Algoritma Genetika untuk mendapatkan hasil yang sesuai, dan analisis tentang hasil yang didapatkan.

## BAB VI KESIMPULAN

Bab ini berisi kesimpulan yang diperoleh dari pembahasan masalah sebelumnya serta saran yang diberikan untuk pengembangan selanjutnya.

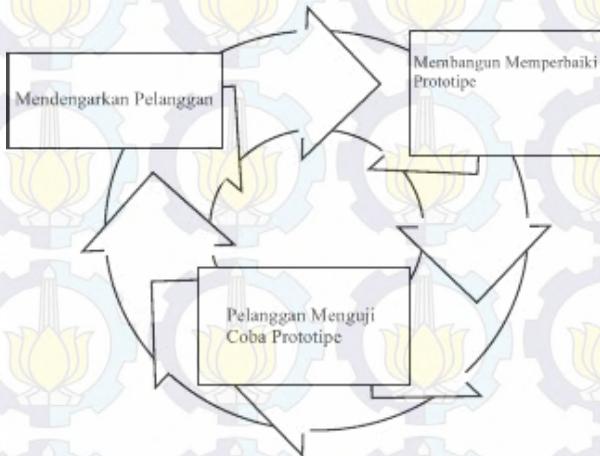


*"Halaman ini sengaja dikosongkan"*

## BAB II TINJAUAN PUSTAKA

### 2.1 Prototipe Perangkat Lunak

Prototipe merupakan metodologi pengembangan perangkat lunak yang menekankan pada pendekatan aspek desain, fungsi dan *user-interface*. Pihak pengembang dan user fokus pada *user-interface* dan bersama-sama mendefinisikan spesifikasi, fungsi, desain dan bagaimana perangkat lunak bekerja. Pengembang mengumpulkan detail dari kebutuhan dan memberikan suatu gambaran dengan cetak biru (*prototype*). Dari proses tersebut akan diketahui detail-detail yang harus dikembangkan atau ditambahkan oleh developer terhadap cetak biru, atau menghapus detail-detail yang tidak diperlukan oleh *user*. Proses akan terjadi terus menerus sehingga produk sesuai dengan keinginan dari *user*. Namun seringkali seorang user mendefinisikan serangkaian sasaran umum bagi perangkat lunak tetapi tidak mengidentifikasi kebutuhan input, pemrosesan, ataupun detail output. Pada sisi lain, pengembang mungkin tidak memiliki kepastian terhadap efisiensi algoritma, kemampuan penyesuaian dari sistem operasi, atau bentuk-bentuk yang harus dilakukan oleh interaksi manusia dan mesin [3]. Dalam situasi seperti ini model yang paling cocok untuk digunakan adalah model prototipe (*Prototype Paradigm*) yang digambarkan dalam Gambar 2.1 dibawah ini.



**Gambar 2.1.** Model Prototipe[3]

## 2.2 Penempatan Pegawai

Penempatan pegawai adalah untuk menempatkan pegawai sebagai unsur pelaksana pekerjaan pada posisi yang sesuai dengan kemampuan, kecakapan dan keahliannya [4]. Kegiatan penempatan pegawai dalam fungsi kepegawaian, dimulai setelah organisasi melaksanakan kegiatan penarikan dan seleksi, yaitu pada saat seorang calon pegawai dinyatakan diterima dan siap untuk ditempatkan pada jabatan atau unit kerja yang sesuai dengan kualifikasinya. Namun dalam kenyataannya permasalahan yang terjadi tidak sesederhana itu, karena justru keberhasilan dari keseluruhan program pengadaan tenaga kerja terletak pada ketepatan dalam menempatkan pegawai yang bersangkutan.

Secara umum proses penempatan pegawai dilatarbelakangi oleh Proses *Staffing* [5]. Salah satu contoh proses tersebut dapat dilihat pada Gambar 2.2.



**Gambar 2.2.** Contoh Salah Satu Proses *Staffing* Pada Organisasi

Dalam gambar diatas tahap seleksi dan penempatan terdapat dalam satu proses. Tahap orientasi serta pelatihan dan pengembangan dilakukan setelah proses penempatan. Namun dalam kenyataannya proses penempatan pegawai bisa dilakukan setelah tahap pelatihan. Proses *staffing* pun berbeda-beda sesuai kebutuhan masing-masing perusahaan. Inti dalam penempatan pegawai dengan perspektif dua arah adalah seorang manajer menempatkan pegawainya ke posisi yang bermacam-macam dibawah kendali perusahaan sehingga terjadi kombinasi antara tujuan perusahaan dengan tujuan pribadi pegawai [2]. Sedangkan garis merah pada gambar diatas adalah lingkup permasalahan yang dibahas oleh penulis. Yaitu hanya pada proses penempatan pegawai dan tidak membahas proses seleksi dan penerimaan.

### 2.3 Model Pilihan dari Perspektif Dua Arah [2]

Struktur kepegawaian perusahaan terdiri dari pegawai yang mempunyai kemampuan dan kualitas yang berbeda dalam departemen yang berbeda pula. Model ini disebut sebagai model pilihan dari perspektif dua arah (*Perspective of Two-way Choice Model*) karena disatu sisi pegawai ikut serta dalam memilih departemen yang tersedia dalam perusahaan sebagai tujuan

pribadinya dan disisi lain perusahaan yang memegang keputusan untuk menempatkan pegawai tersebut kedalam departemen yang sesuai. Model penempatan pegawai untuk kasus ini adalah sebagai berikut:

Fungsi tujuan:

$$\max z(x) = \sum_{i=1}^n \sum_{j=0}^m E_i \times p_{ij} \times q_{ij} \times x_{ij} \quad (2.1)$$

Dengan:

$$p_{ij} = \sum_i^n \sum_l^t \frac{\alpha_{il}(a_{jl} - d_{jl})}{d_{jl}} \quad (2.2)$$

$$q_{ij} = \sum_i^n \sum_r^s \frac{\alpha_{il}(b_{jr} - d_{jl})}{d_{jl}} \quad (2.3)$$

$$\sum_i^n \sum_l^t \frac{\alpha_{il}(a_{jl} - d_{jl})}{d_{jl}} \geq 0 \quad (2.4)$$

$$\sum_i^n \sum_r^s \frac{\alpha_{il}(b_{jr} - d_{jl})}{d_{jl}} \geq 0 \quad (2.5)$$

Dimana :

$E_i$  = tingkat kepentingan departemen ke- $i$

$\alpha_{il}$  = bobot kualifikasi ke- $l$  dari departemen ke- $i$

$a_{jl}$  = nilai kompetensi pegawai ke- $j$  pada kualifikasi ke- $l$  dari perspektif perusahaan

$b_{jl}$  = nilai kompetensi pegawai ke- $j$  pada kualifikasi ke- $l$  dari perspektif pegawai

$d_{il}$  = nilai kompetensi minimal ke- $l$  dari departemen ke- $i$

$i$  = index dari departemen

$j$  = index dari pegawai

- $n$  = jumlah departemen  
 $m$  = jumlah pegawai  
 $p_{ij}$  = nilai komprehensif dari perusahaan ketika menilai antara pegawai dengan kualifikasi departemen.  
 $q_{ij}$  = nilai komprehensif dari pegawai ketika menilai antar departemen dengan kualifikasi pegawai.

Dan variabel keputusan:

$$x_{ij} = \begin{cases} 1, & \text{jika pegawai ke } j \text{ memilih departemen ke } i \\ 0, & \text{untuk } x \text{ yang lain} \end{cases}$$

## 2.4 Analytic Hierarchy Process (AHP)

Untuk mendapatkan bobot pada variabel yang dibutuhkan dalam fungsi diatas digunakan *Analytic Hierarchy Process* (AHP). Kriteria-kriteria dibandingkan dalam bentuk perbandingan berpasangan, untuk membentuk suatu matriks preferensi, demikian pula halnya dengan alternatif-alternatif.

Hirarki didefinisikan sebagai suatu representasi dari sebuah permasalahan yang kompleks dalam suatu struktur multi level dimana level pertama adalah tujuan, yang diikuti level faktor, kriteria, sub kriteria, dan seterusnya ke bawah hingga level terakhir dari alternatif. Dengan hirarki, suatu masalah yang kompleks dapat diuraikan ke dalam kelompok-kelompoknya yang kemudian diatur menjadi suatu bentuk hirarki sehingga permasalahan akan tampak lebih terstruktur dan sistematis [6].

Tahap-tahap atau prosedur AHP dalam menentukan bobot kriteria meliputi hal-hal sebagai berikut :

1. Mendefinisikan struktur hierarki masalah
2. Penilaian kriteria dan alternatif.

Pengisian nilai tabel perbandingan berpasangan dilakukan berdasarkan kebijakan pembuat keputusan dengan melihat tingkat kepentingan antar satu elemen dengan elemen yang lainnya.

**Tabel 2.1.** Skala Penilaian Perbandingan Pasangan

Tingkat kepentingan	Definisi	Keterangan
1	Kedua elemen sama pentingnya	Kedua elemen seimbang sama besar pada sifat tersebut
3	Elemen yang satu sedikit lebih penting daripada elemen lainnya	Pengalaman menyatakan sedikit memihak pada satu elemen
5	Elemen yang satu lebih penting daripada elemen lainnya	Pengalaman menunjukkan secara kuat memihak pada satu elemen
7	Satu elemen jelas lebih mutlak penting daripada elemen lainnya	Pengalaman menunjukkan secara kuat disukai dan didominasi satu elemen yang sangat jelas lebih penting
9	Satu elemen mutlak penting daripada elemen lainnya	Pengalaman menunjukkan satu elemen sangat jelas lebih penting
2,4,6,8	Nilai tengah diantara dua penilaian yang berdampingan	Nilai ini diberikan jika diperlukan kompromi
Kebalikan		Bila elemen ke-ij pada faktor i mendapat nilai nilai x maka elemen ke-ji pada faktor ke-j mendapat nilai $1/x$

3. Membuat matriks perbandingan berpasangan kriteria terhadap kriteria.

Dari penilaian kriteria dan alternatif, dibuat suatu matriks perbandingan berpasangan dengan cara membandingkan masing-masing kriteria dengan kriteria lainnya dan alternatif dengan alternatif lainnya sebanyak sejumlah kriteria. Maka susunan elemen-elemen yang dibandingkan tersebut akan tampak seperti pada tabel di bawah ini:

**Tabel 2.2.** Matriks Perbandingan Berpasangan

	$A_1$	$A_2$	...	$A_n$
$A_1$	$a_{1,1}$	$a_{1,2}$	...	$a_{1,n}$
$A_2$	$a_{2,1}$	$a_{2,2}$	...	$a_{2,n}$
...	...	...	...	...
$A_n$	$a_{n,1}$	$a_{n,2}$	...	$a_{n,n}$

Jika kuisisioner diinputkan oleh beberapa pakar, akan terjadi perbedaan beberapa pendapat. Maka pendapat dari masing-masing pakar disatukan dengan menggunakan persamaan rata-rata geometri [7], yaitu:

$$GM = \sqrt[i]{(x_1)(x_2) \dots (x_i)} \quad (2.6)$$

$GM$  = Geometric Mean

$x$  = Pakar

$i$  = Banyaknya Pakar

4. Menjumlahkan matriks kolom
5. Menghitung matriks normaliasi dengan cara membagi setiap nilai elemen kolom dengan jumlah matrik kolom
6. Menentukan prioritas atau bobot kriteria yang dilakukan dengan cara jumlah baris dibagi dengan banyaknya kriteria
7. Menghitung prioritas alternatif dengan membuat matrik berpasangan alternatif terhadap alternatif sebanyak jumlah kriteria.
8. Menghitung Indeks Konsistensi (CI)

$$CI = \frac{(\lambda_{max} - n)}{(n - 1)} \quad (2.7)$$

$$\lambda_{max} = \Sigma t_i b_i$$

Dengan:

$CI$  = Consistency Index atau rasio penyimpangan konsistensi

$\lambda_{max}$  = Nilai eigen terbesar dari matriks ber-ordo  $n$

14

$t_i$  = Jumlah kolom ke- $i$  matriks perbandingan berpasangan

$b_i$  = Bobot kriteria ke- $i$

$n$  = Ordo matriks

9. *Consistency Ratio* (CR), merupakan pernyataan yang menyatakan seberapa besar derajat *Inconsistency* dari penetapan nilai perbandingan antar kriteria yang telah dibuat, yaitu

$$CR = \frac{CI}{RI} \quad (2.8)$$

Dengan:

$CR$  = *Consistency Ratio*

$CI$  = *Consistency Index*

$RI$  = *Random Index*

**Tabel 2.3.** Daftar *Random Index* (RI)

Ukuran Matriks	Nilai RI
1,2	0,00
3	0,58
4	0,90
5	1,12
6	1,24
7	1,32
8	1,41
9	1,45
10	1,49
11	1,51
12	1,58

Bila matriks matriks perbandingan berpasangan dengan nilai  $CR < 0,1$  maka ketidakkonsistenan pendapat dari pakar masih dapat diterima jika tidak maka penilaian perlu diulang.

## 2.5 Algoritma Genetika (AG)

Algoritma Genetika adalah algoritma pencarian yang didasarkan pada mekanisme seleksi alamiah dan genetika alamiah. Sejak algoritma genetika (AG) pertama kali dirintis oleh John Holland dari Universitas Michigan pada tahun 1960-an, AG telah diaplikasikan secara luas pada berbagai bidang. AG banyak digunakan untuk memecahkan masalah optimasi, walaupun pada kenyataannya juga memiliki kemampuan yang baik untuk masalah- masalah selain optimasi.

Pada algoritma genetika, teknik pencarian dilakukan sekaligus atas sejumlah solusi yang dikenal dengan istilah populasi. Individu yang terdapat dalam satu populasi disebut dengan istilah 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 dari kromosom dalam 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 cara 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.

Ada tiga keunggulan dari aplikasi Algoritma Genetika dalam proses optimasi, yaitu: (a) Algoritma Genetika tidak terlalu banyak memerlukan persyaratan matematika dalam penyelesaian proses optimasi. Algoritma Genetika dapat diaplikasikan pada

beberapa jenis fungsi obyektif dengan beberapa fungsi pembatas baik berbentuk linier maupun non-linier; (b) Operasi evolusi dari Algoritma Genetika sangat efektif untuk mengobservasi posisi global secara acak; dan (c) Algoritma Genetika mempunyai fleksibilitas untuk diimplementasikan secara efisien pada problematika tertentu. menyeleksi nilai fitness dari kromosom induk (*parent*) dan nilai fitness.

### 2.5.1 Jenis-jenis AG

Terdapat dua jenis variasi AG, yaitu:

#### 1. *Steady State*

Pada AG jenis ini, proses *replacement* dilakukan setiap kali dihasilkan dua *offspring* hasil *crossover*. *Offspring* menggantikan kromosom yang nilai *fitness*-nya paling kecil. Dengan demikian, populasi baru yang dihasilkan selalu memiliki individu-individu yang lebih baik dibandingkan populasi lama.

#### 2. *Generational Replacement*

AG jenis ini, proses *replacement* dilakukan sekaligus ketika dihasilkan satu populasi baru. Untuk mempertahankan individu terbaik pada suatu generasi, diperlukan suatu komponen yang disebut *Elitisme*, yakni pengkopian individu terbaik untuk dimasukkan sebagai anggota populasi pada generasi berikutnya. Dengan adanya komponen ini, maka populasi baru yang dihasilkan selalu memiliki satu individu terbaik yang kualitasnya sama baiknya atau bahkan lebih baik dibandingkan populasi lama.

### 2.5.2 Komponen-komponen AG [8]

AG terdiri dari delapan komponen yaitu: Skema Pengkodean, Nilai *Fitness*, Seleksi Orang Tua, Pindah Silang (*crossover*), Mutasi, Elitisme (untuk AG yang berjenis *Generational Replacement*), Penggantian Populasi, dan Kriteria Penghentian.

#### • **Skema Pengkodean**

Untuk dapat diproses menggunakan AG, suatu permasalahan harus dikonversi dahulu ke bentuk individu yang diawali oleh

satu atau lebih kromosom dengan kode tertentu. AG merepresentasikan gen buatan, secara umum sebagai bilangan real, desimal atau biner.

- **Nilai *Fitness***

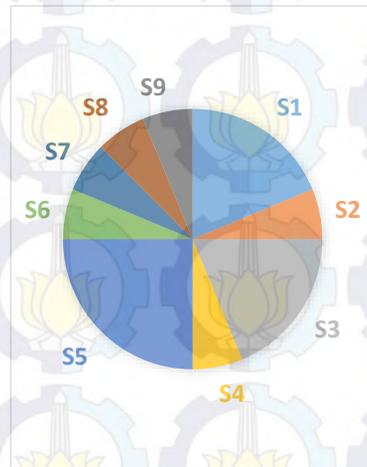
Pada evolusi di dunia nyata, individu bernilai *fitness* tinggi akan bertahan hidup. Sedangkan individu bernilai *fitness* rendah akan mati. Pada AG suatu individu dievaluasi berdasarkan fungsi tertentu yang pada Tugas Akhir ini menggunakan fungsi di atas sebagai ukuran *fitness*-nya karena permasalahan pada optimasi ini adalah memaksimalkan sebuah fungsi.

- **Seleksi Induk**

Proses pemilihan dua individu sebagai induk biasanya dilakukan secara proporsional berdasarkan nilai-nilai *fitness*-nya. Metode seleksi yang akan digunakan dalam Tugas Akhir ini adalah *roulette-wheel*. Sesuai dengan namanya, metode ini menirukan permainan *roulette-wheel* dimana masing-masing individu menempati potongan lingkaran pada roda *roulette* secara proporsional sesuai dengan nilai *fitness*-nya. Berikut ni adalah ilustrasi penggunaan metode *roulette-wheel*.

**Tabel 2.4.** Contoh penggunaan metode *roulette-wheel* dalam seleksi

String	Nilai Fitness
S1	0.6
S2	0.2
S3	0.6
S4	0.2
S5	0.8
S6	0.2
S7	0.2
S8	0.2
S9	0.2



Berdasarkan gambar diatas individu S5 yang bernilai *fitness* paling besar menempati seperempat lingkaran roda *roulette*. Dengan demikian, S5 memiliki peluang sebesar 0,25 untuk terpilih sebagai orang tua.

- **Pindah Silang**

Di dunia nyata, tidak mungkin ada dua individu yang sama. Hal ini disebabkan adanya evolusi yang didalamnya terdapat proses pindah silang atau *crossover*. Pada proses pindah silang terjadi kombinasi pewarisan gen-gen dari induknya, gen-gen dari kedua induk dapat bercampur sehingga dihasilkan susunan kromosom yang baru. Setelah didapatkan dua individu orang tua, selanjutnya ditentukan titik pindah silang secara acak. Jika diasumsikan  $L$  adalah panjang kromosom, maka titik pindah silang berada antara 1 sampai  $L - 1$ . Kemudian beberapa bagian dari dua kromosom ditukar pada titik pindah silang yang dipilih. Pertukaran tersebut akan menghasilkan dua individu anak. Namun operasi pindah silang tidak selamanya berhasil. Peluang keberhasilan operasi pindah silang dinyatakan dengan probabilitas pindah silang atau  $P_c$ .

- **Mutasi**

Mutasi diperlukan untuk mengembalikan informasi bit yang hilang akibat *crossover*. Mutasi diterapkan dengan probabilitas yang sangat kecil. Jika mutasi dilakukan terlalu sering, maka akan menghasilkan individu yang lemah karena konfigurasi gen pada individu yang unggul akan dirusak. Berdasarkan bagian yang termutasi, proses mutasi dapat dibedakan atas 3 bagian, yaitu:

1. Mutasi pada tingkat kromosom, yaitu semua gen dalam satu kromosom berubah.
2. Mutasi pada tingkat gen, yaitu semua bit dalam satu gen akan berubah.
3. Mutasi pada tingkat bit, yaitu hanya satu bit yang berubah.

Untuk semua gen yang ada, jika bilangan random yang dibangkitkan kurang dari probabilitas mutasi  $P_{mut}$  yang ditentukan maka gen tersebut menjadi nilai kebalikannya.

Biasanya  $P_{mut}$  diset sebagai  $\frac{1}{n}$ , dimana  $n$  adalah jumlah gen dalam kromosom.

- **Elitisme**

Karena seleksi dilakukan secara acak, maka tidak ada jaminan bahwa suatu individu bernilai *fitness* tertinggi akan selalu terpilih. Kalaupun individu bernilai *fitness* tertinggi terpilih, mungkin saja individu tersebut akan rusak karena proses *crossover*. Untuk menjaga agar individu bernilai *fitness* tertinggi tersebut tidak hilang selama evolusi, perlu dibuat satu atau dua kopinya. Prosedur ini disebut sebagai elitisme.

- **Penggantian Populasi**

Pada AG berjenis *Generational Replacement*,  $N$  individu pada suatu generasi digantikan sekaligus oleh  $N$  individu baru hasil *crossover* dan mutasi. Dan untuk mempertahankan individu terbaik dilakukan prosedur elitisme seperti yang telah dijelaskan diatas. Sedangkan pada AG berjenis *Steady-State* urutan prosedur yang digunakan adalah:

1. Selalu mengganti individu yang memiliki nilai *fitness* terkecil.
2. Selalu mengganti individu yang paling tua.
3. Membandingkan anak dengan orang tua. Apabila anak memiliki nilai *fitness* yang lebih baik dari pada salah satu atau kedua orang tua, maka anak menggantikan orang tua yang memiliki nilai *fitness* terendah.

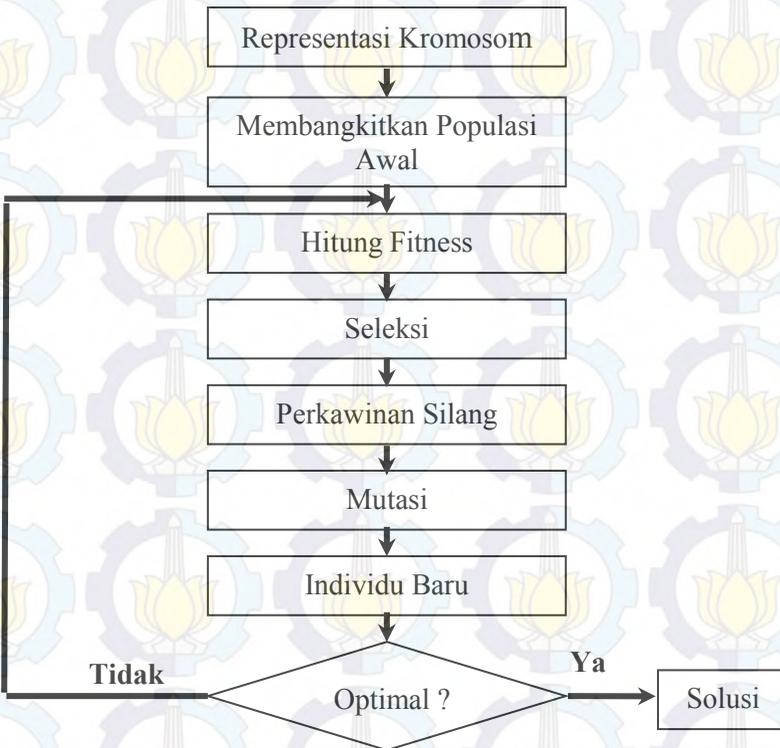
- **Kriteria Penghentian**

Terdapat berbagai macam kriteria penghentian proses AG, diantaranya:

1. Memberikan batasan jumlah iterasi. Apabila batas iterasi tersebut dicapai, iterasi dihentikan dan memilih individu yang mempunyai nilai *fitness* tertinggi sebagai solusi terbaik.
2. Memberikan batasan waktu AG. Kriteria ini digunakan pada sistem-sistem waktu yang nyata real time system, dimana solusi harus ditemukan dalam waktu yang ditentukan. Dengan demikian AG bisa dihentikan ketika proses sudah berlangsung selama waktu yang telah ditentukan tersebut.

- Menghitung kegagalan penggantian anggota populasi yang terjadi secara berurutan sampai jumlah tertentu. Misalkan, setelah iterasi ke 100 tidak ada penggantian individu dalam populasi karena individu anak yang dihasilkan selalu memiliki nilai *fitness* yang lebih rendah daripada orang tuanya. Maka AG dapat dihentikan.

Secara umum diagram alir dari Algoritma Genetika dapat digambarkan sebagai berikut:



**Gambar 2.3.** Diagram Alir Algoritma Genetika

## BAB III METODOLOGI PENELITIAN

### 3.1 Tahapan Penelitian

Tahapan penelitian yang akan digunakan untuk penyelesaian permasalahan dalam Tugas Akhir ini adalah sebagai berikut:

#### 1. Studi Tentang AHP, Penempatan Pegawai dan Algoritma Genetika

Tahap ini adalah tahap melakukan identifikasi permasalahan dengan cara mencari referensi yang menunjang penelitian. Referensi dapat berupa Tugas Akhir, jurnal, buku maupun artikel terkait prototipe perangkat lunak, penempatan pegawai, model pilihan dari perspektif dua arah, *Analytic Hierarchy Process*, dan Algoritma Genetika.

#### 2. Survey Lapangan dan Pengambilan Data

Tahap ini dilakukan wawancara dan diskusi kepada pihak Personalia untuk mengetahui kondisi lapangan dan sistem kepegawaian di PT. Petrokimia Gresik. Kemudian dilakukan pengambilan data dengan penyebaran kuisioner kepada pakar.

#### 3. Perhitungan Bobot Kriteria dan Alternatif dengan AHP

Tahap ini adalah tahap mendapatkan bobot masing-masing departemen dan kualifikasinya menggunakan AHP berdasarkan kuisioner yang telah diisi oleh pakar.

#### 4. Perancangan Prototipe Perangkat Lunak

Terdapat langkah-langkah dalam perancangan perangkat lunak dengan model prototipe, antara lain:

- **Pengumpulan Kebutuhan**

Pengumpulan Kebutuhan dilakukan untuk mengetahui kebutuhan pengguna atas prototipe perangkat lunak yang akan dibuat. Pengguna dan pengembang bersama-sama mendefinisikan format seluruh perangkat lunak, mengidentifikasi semua kebutuhan, dan garis besar sistem.

- **Perancangan Prototipe**

Pengembangan perangkat lunak dengan model prototipe juga memerlukan *Business Process* untuk mendefinisikan

keseluruhan sistem. Maka pada tahap ini dibuat desain prototipe perangkat lunak yang akan dibangun.

- **Membangun Prototipe**

Prototipe dibangun berdasarkan perancangan yang dibuat. Tahap ini merupakan hasil dari analisis kebutuhan awal dan berfokus pada penyajian kepada pengguna.

- **Algoritma Genetika**

Tahap ini adalah tahap pembuatan program penempatan pegawai dengan implementasi Algoritma Genetika.

- **Evaluasi Pengguna**

Evaluasi ini dilakukan oleh pengguna, apakah prototipe yang sudah dibangun sudah sesuai dengan keinginan pengguna atau belum. Jika sudah sesuai, maka metode prototipe berhenti dan langkah selanjutnya adalah transisi pembangunan perangkat lunak. Jika tidak sesuai, prototipe yang telah dibuat akan memasuki tahap revisi dan perbaikan.

- **Revisi dan Perbaikan Prototipe**

Pada langkah ini dilakukan revisi dan perbaikan prototipe sesuai dengan keinginan pengguna yang telah mengevaluasi prototipe yang dibuat.

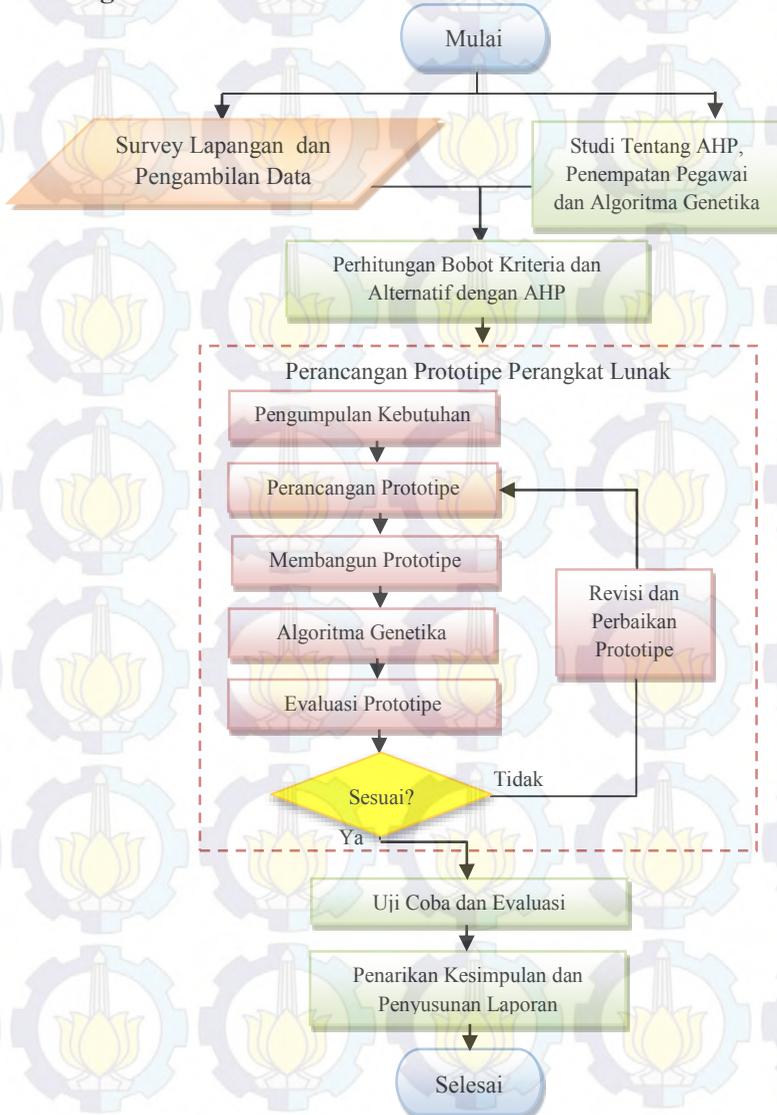
## 5. Uji Coba dan Evaluasi

Uji coba dan evaluasi dilakukan pada program penempatan pegawai yang telah dibuat dengan tujuan untuk mengetahui berapa nilai parameter AG yang paling optimal.

## 6. Penarikan Kesimpulan dan Penyusunan Laporan

Penarikan kesimpulan dilakukan setelah dilakukan evaluasi dan didasarkan atas hasil yang didapat. Sehingga akan diketahui informasi yang merupakan jawaban dari rumusan masalah yang dikemukakan serta merujuk pada tujuan dalam pembuatan Tugas Akhir ini. Sedangkan Penyusunan laporan adalah tahap akhir dalam pengerjaan Tugas Akhir yang merupakan sebuah bentuk tulisan atas penelitian yang telah dilakukan.

### 3.2 Diagram Alir Penelitian



**Gambar 3.1.** Diagram Alir Penelitian



*"Halaman ini sengaja dikosongkan"*

## BAB IV PERANCANGAN DAN IMPLEMENTASI SISTEM

Bab ini membahas tentang proses mendapatkan bobot departemen serta alternatif-alternatifnya, proses desain *database*, perancangan sistem, implementasi sistem menggunakan Algoritma Genetika, dan proses dalam membangun prototipe perangkat lunak.

### 4.1 Perhitungan Bobot Setiap Kriteria dan Alternatif-Alternatifnya.

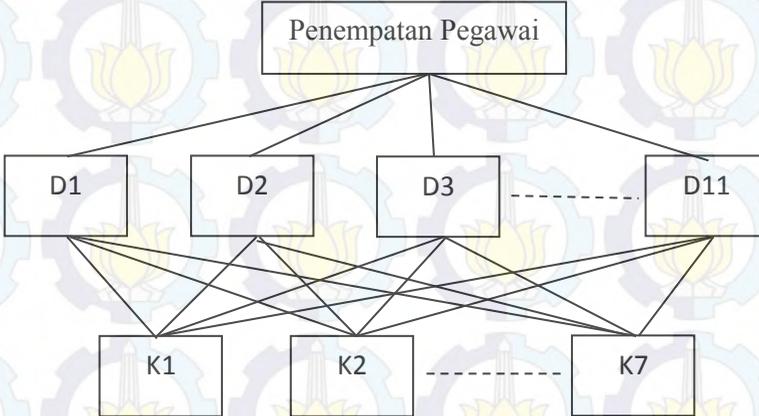
Berdasarkan wawancara (dokumentasi wawancara terlampir) di PT. Petrokimia Gresik, diketahui terdapat 11 Departemen sebagai kriteria utama dan 7 Kualifikasi sebagai alternatifnya pada proses penempatan pegawai tahun 2014 dengan lulusan pegawai D3. Yang ditunjukkan pada tabel berikut:

**Tabel 4.1.** Departemen dan Kualifikasi dalam proses penempatan pegawai di PT. Petrokimia Gresik

Departemen	Kualifikasi
Akuntansi (D1)	Penampilan (K1)
Anggaran (D2)	Pengetahuan Khusus (K2)
Kuangan (D3)	Kemampuan Menyatakan Pendapat (K3)
Pemeliharaan I (D4)	Kematangan Kepribadian (K4)
Pemeliharaan II (D5)	Ambisi (K5)
Pemeliharaan III (D6)	Potensi untuk Berkembang (K6)
Pengelolaan Pelabuhan (D7)	Kegiatan Organisasi (K7)
Prasarana Publik dan Kawasan (D8)	
Produksi I (D9)	
Produksi II A (D10)	
Produksi II B (D11)	

#### 4.1.1 Mendefinisikan Struktur Hierarki

Model hierarki untuk penempatan pegawai dalam studi kasus ini adalah sebagai berikut:



**Gambar 4.1.** Hierarki Penempatan Pegawai

#### 4.1.2 Membuat Matriks Perbandingan Berpasangan

Penilaian kriteria dan alternatif dilakukan oleh 3 orang pakar di Departemen Personalia PT. Petrokimia Gresik (bentuk kuisisioner terlampir) dan masing-masing kuisisioner dibuat matriks perbandingannya (terlampir). Kemudian dari 3 matriks perbandingan tersebut disatukan menggunakan Persamaan Rata-rata Geometri (2.6). Untuk Rata-rata Geometri Departemen dilakukan perhitungan sebagai berikut:

$$GM_D =$$

$$\begin{bmatrix} \sqrt[3]{(x_{1(1,1)})(x_{2(1,1)})(x_{3(1,1)})} & \dots & \sqrt[3]{(x_{1(1,11)})(x_{2(1,11)})(x_{3(1,11)})} \\ \sqrt[3]{(x_{1(2,1)})(x_{2(2,1)})(x_{3(2,1)})} & \dots & \sqrt[3]{(x_{1(2,11)})(x_{2(2,11)})(x_{3(2,11)})} \\ \vdots & \dots & \vdots \\ \sqrt[3]{(x_{1(11,1)})(x_{2(11,1)})(x_{3(11,1)})} & \dots & \sqrt[3]{(x_{1(11,11)})(x_{2(11,11)})(x_{3(11,11)})} \end{bmatrix}$$

$$GM_D = \begin{bmatrix} \sqrt[3]{(1)(1)(1)} & \dots & \sqrt[3]{\left(\frac{1}{6}\right)\left(\frac{1}{8}\right)\left(\frac{1}{6}\right)} \\ \sqrt[3]{(1)(1)(1)} & \dots & \sqrt[3]{\left(\frac{1}{6}\right)\left(\frac{1}{8}\right)\left(\frac{1}{6}\right)} \\ \vdots & \dots & \vdots \\ \vdots & \dots & \vdots \\ \sqrt[3]{(6)(8)(6)} & \dots & \sqrt[3]{(1)(1)(1)} \end{bmatrix}$$

Sehingga diperoleh matriks perbandingan hasil Rata-rata Geometri untuk Departemen dengan jumlah setiap kolomnya sebagai berikut:

**Tabel 4.2.** Matriks Perbandingan Departemen

	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11
D1	1,000	1,000	1,000	0,161	0,161	0,161	0,171	1,260	0,151	0,151	0,151
D2	1,000	1,000	1,000	0,171	0,171	0,171	0,212	0,550	0,151	0,151	0,151
D3	1,000	1,000	1,000	0,572	0,572	0,572	0,523	1,442	0,500	0,500	0,500
D4	6,214	5,848	1,747	1,000	1,000	1,000	3,271	5,013	0,794	0,415	0,415
D5	6,214	5,848	1,747	1,000	1,000	1,000	1,913	4,718	0,415	0,794	0,794
D6	6,214	5,848	1,747	1,000	1,000	1,000	3,271	4,718	0,415	0,415	0,415
D7	5,848	4,718	1,913	0,306	0,523	0,306	1,000	3,826	0,199	0,199	0,199
D8	0,794	1,817	0,693	0,199	0,212	0,212	0,261	1,000	0,151	0,161	0,151
D9	6,604	6,604	2,000	1,260	2,410	2,410	5,013	6,604	1,000	0,585	0,585
D10	6,604	6,604	2,000	2,410	1,260	2,410	5,013	6,214	1,710	1,000	1,000
D11	6,604	6,604	2,000	2,410	1,260	2,410	5,013	6,604	1,710	1,000	1,000
TOTAL	48,097	46,890	16,848	10,490	9,569	11,652	25,662	41,949	7,197	5,372	5,362

#### 4.1.3 Normalisasi Matriks

Setelah diperoleh matriks perbandingan berpasangan diatas, akan dicari matriks normalisasinya dengan cara membagi setiap nilai elemen kolom dengan jumlah matriks kolom. Sehingga matriks normalisasinya adalah sebagai berikut:

**Tabel 4.3.** Matriks Normalisasi

	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11
D1	0,021	0,021	0,059	0,015	0,017	0,014	0,007	0,030	0,021	0,028	0,028
D2	0,021	0,021	0,059	0,016	0,018	0,015	0,008	0,013	0,021	0,028	0,028
D3	0,021	0,021	0,059	0,055	0,060	0,049	0,020	0,034	0,069	0,093	0,093
D4	0,129	0,125	0,101	0,095	0,105	0,086	0,127	0,120	0,110	0,077	0,077
D5	0,129	0,125	0,104	0,095	0,105	0,086	0,075	0,112	0,058	0,148	0,148
D6	0,129	0,125	0,104	0,095	0,105	0,086	0,127	0,112	0,058	0,077	0,077
D7	0,122	0,103	0,111	0,029	0,055	0,026	0,039	0,091	0,028	0,037	0,037
D8	0,017	0,039	0,041	0,019	0,022	0,018	0,010	0,021	0,030	0,028	0,028
D9	0,137	0,141	0,119	0,120	0,252	0,207	0,195	0,157	0,139	0,109	0,109
D10	0,137	0,141	0,119	0,230	0,132	0,207	0,195	0,148	0,238	0,180	0,186
D11	0,137	0,141	0,119	0,230	0,132	0,207	0,195	0,157	0,238	0,180	0,186

#### 4.1.4 Menentukan Bobot Masing-Masing Kriteria

Untuk mendapatkan bobot masing-masing kriteria dilakukan dengan cara membagi jumlah baris matriks normalisasi dengan 11 (banyaknya kriteria). Sehingga diperoleh jumlah baris dan bobot masing-masing departemen sebagai berikut:

**Tabel 4.4.** Jumlah Baris dan Bobot Departemen

Baris	Jumlah Baris	Departemen	Bobot Departemen ( $E_i$ )
1	0,262	D1	0,024
2	0,249	D2	0,023
3	0,576	D3	0,052
4	1,155	D4	0,105
5	1,184	D5	0,108
6	1,095	D6	0,100
7	0,678	D7	0,062
8	0,269	D8	0,024
9	1,685	D9	0,153
10	1,919	D10	0,174
11	1,928	D11	0,175

#### 4.1.5 Menghitung Indeks Konsistensi (CI) dan Rasio Konsistensi (CR) Untuk Matriks Perbandingan Berpasangan Departemen

Akan dicari  $\lambda_{max}$  terlebih dahulu, yaitu:

$$\begin{aligned}\lambda_{max} &= \sum t_i b_i \\ &= t_1 b_1 + t_2 b_2 + t_3 b_3 + \dots + t_i b_i \\ &= 48,097 * 0,024 + 46,890 * 0,023 + 16,848 * 0,052 + \dots + 5,362 * 0,175 \\ &= 11,966\end{aligned}$$

Maka didapatkan Index Konsistensi, yaitu:

$$\begin{aligned}CI &= \frac{(\lambda_{max} - n)}{(n-1)} \\ &= \frac{(11,966 - 11)}{(11-1)} \\ &= 0,0966\end{aligned}$$

Karena terdapat 11 kriteria maka *Random Index* nya adalah 1,51. Sehingga Rasio Konsistensinya adalah sebagai berikut:

$$\begin{aligned}CR &= \frac{0,0966}{1,51} \\ CR &= 0,064\end{aligned}$$

Jadi matriks perbandingan berpasangan untuk kriteria departemen dapat diterima karena nilai  $CR < 0,1$ .

#### 4.1.6 Menghitung Bobot Masing-Masing Alternatif Kualifikasi Sejumlah Kriteria Departemen

Dengan cara yang sama diperoleh bobot masing-masing Kualifikasi dalam setiap Departemen beserta Rasio Konsistensinya sebagai berikut:

1. Bobot dan Konsistensi Rasio Kualifikasi dalam Departemen Akuntansi (D1)

**Tabel 4.5.** Bobot Kualifikasi dalam Departemen Akuntansi (D1)

Kualifikasi	Bobot
K1	0,050
K2	0,189
K3	0,090
K4	0,257
K5	0,084
K6	0,244
K7	0,086

Karena terdapat 7 Kualifikasi maka RI-nya adalah 1,32. Maka CR-nya adalah:

$$CR = \frac{0,13136}{1,32}$$

$$CR = 0,0995$$

2. Bobot dan Konsistensi Rasio Kualifikasi dalam Departemen Anggaran (D2)

**Tabel 4.6.** Bobot Kualifikasi dalam Departemen Anggaran (D2)

Kualifikasi	Bobot
K1	0,032
K2	0,188
K3	0,138
K4	0,218
K5	0,093
K6	0,193
K7	0,137

Dengan nilai CR:

$$CR = \frac{0,12003}{1,32}$$

$$CR = 0,0909$$

### 3. Bobot dan Konsistensi Rasio Kualifikasi dalam Departemen Keuangan (D3)

**Tabel 4.7.** Bobot Kualifikasi dalam Departemen Keuangan (D3)

Kualifikasi	Bobot
K1	0,043
K2	0,170
K3	0,121
K4	0,284
K5	0,083
K6	0,195
K7	0,103

Dengan nilai CR:

$$CR = \frac{0,1312}{1,32}$$

$$CR = 0,0994$$

### 4. Bobot dan Konsistensi Rasio Kualifikasi dalam Departemen Pemeliharaan I (D4)

**Tabel 4.8.** Bobot Kualifikasi dalam Departemen Pemeliharaan I (D4)

Kualifikasi	Bobot
K1	0,032
K2	0,289
K3	0,111
K4	0,243
K5	0,098
K6	0,139
K7	0,087

Dengan nilai CR:

$$CR = \frac{0,1255}{1,32}$$

$$CR = 0,0951$$

5. Bobot dan Konsistensi Rasio Kualifikasi dalam Departemen Pemeliharaan II (D5)

**Tabel 4.9.** Bobot Kualifikasi dalam Departemen Pemeliharaan II (D5)

Kualifikasi	Bobot
K1	0,032
K2	0,267
K3	0,130
K4	0,289
K5	0,101
K6	0,120
K7	0,061

Dengan nilai CR:

$$CR = \frac{0,0769}{1,32}$$

$$CR = 0,0583$$

6. Bobot dan Konsistensi Rasio Kualifikasi dalam Departemen Pemeliharaan III (D6)

**Tabel 4.10.** Bobot Kualifikasi dalam Departemen Pemeliharaan III (D6)

Kualifikasi	Bobot
K1	0,034
K2	0,308
K3	0,138
K4	0,207
K5	0,106
K6	0,122
K7	0,085

Dengan nilai CR:

$$CR = \frac{0,0686}{1,32}$$

$$CR = 0,0520$$

7. Bobot dan Konsistensi Rasio Kualifikasi dalam Departemen Pengelolaan Pelabuhan (D7)

**Tabel 4.11.** Bobot Kualifikasi dalam Departemen Pengelolaan Pelabuhan (D7)

Kualifikasi	Bobot
K1	0,030
K2	0,214
K3	0,173
K4	0,274
K5	0,094
K6	0,133
K7	0,082

Dengan nilai CR:

$$CR = \frac{0,1294}{1,32}$$

$$CR = 0,0980$$

8. Bobot dan Konsistensi Rasio Kualifikasi dalam Departemen Prasarana Publik dan Kawasan (D8)

**Tabel 4.12.** Bobot Kualifikasi dalam Departemen Prasarana Publik dan Kawasan (D8)

Kualifikasi	Bobot
K1	0,037
K2	0,284
K3	0,140
K4	0,284
K5	0,081
K6	0,088
K7	0,086

Dengan nilai CR:

$$CR = \frac{0,0842}{1,32}$$

$$CR = 0,0638$$

9. Bobot dan Konsistensi Rasio Kualifikasi dalam Departemen Produksi I (D9)

**Tabel 4.13.** Bobot Kualifikasi dalam Departemen Produksi I (D9)

Kualifikasi	Bobot
K1	0,035
K2	0,283
K3	0,147
K4	0,274
K5	0,094
K6	0,109
K7	0,059

Dengan nilai CR:

$$CR = \frac{0,0765}{1,32}$$

$$CR = 0,058$$

10. Bobot dan Konsistensi Rasio Kualifikasi dalam Departemen Produksi II A (D10)

**Tabel 4.14.** Bobot Kualifikasi dalam Departemen Produksi II A (D10)

Kualifikasi	Bobot
K1	0,034
K2	0,295
K3	0,112
K4	0,284
K5	0,107
K6	0,112
K7	0,057

Dengan nilai CR:

$$CR = \frac{0,0637}{1,32}$$

$$CR = 0,0483$$

## 11. Bobot dan Konsistensi Rasio Kualifikasi dalam Departemen Produksi II B (D11)

**Tabel 4.15.** Bobot Kualifikasi dalam Departemen Produksi II B (D11)

Kualifikasi	Bobot
K1	0,035
K2	0,302
K3	0,104
K4	0,253
K5	0,134
K6	0,111
K7	0,062

Dengan nilai CR:

$$CR = \frac{0,0582}{1,32}$$

$$CR = 0,0441$$

### 4.2 Analisis Sistem

Seperti yang telah dijelaskan sebelumnya pada batasan masalah, bahwa ruang lingkup Tugas Akhir ini adalah pada proses penempatan pegawai. Yang berarti bahwa pegawai telah diterima oleh perusahaan setelah proses seleksi dan penerimaan. Maka sistem yang dibangun tidak mengatasi tentang proses seleksi dan penerimaan.

Perangkat lunak penempatan pegawai ini dirancang untuk membantu perusahaan dalam proses penempatan pegawainya secara matematis dengan data yang telah didapatkan sesuai dengan kebutuhan sistem. Salah satu data yang penting dalam sistem ini adalah data pilihan pegawai atas departemen yang diminati. Dalam Tugas Akhir ini data tersebut digunakan untuk membatasi perhitungan agar hasil yang didapatkan sesuai dengan harapan pegawai tersebut. Namun tidak mengabaikan kebutuhan perusahaan akan ketepatan pegawai yang menempati departemennya. Yaitu dengan memasukkan bidang keahlian calon pegawai dalam pilihan departemennya. Sehingga harapan

pegawai dan kebutuhan perusahaan dapat terpenuhi. Jadi dapat dikatakan bahwa perangkat lunak ini merupakan pendukung keputusan perusahaan dalam menempatkan pegawainya. Karena hasil penempatan pegawai pada perangkat lunak ini merupakan perkiraan dari hasil sebenarnya yang sepenuhnya menggunakan persepsi manusia dalam prosesnya.

#### 4.2.1 Deskripsi Perangkat Lunak

Permasalahan dalam proses penempatan pegawai adalah kurangnya metode ilmiah dan memenuhi kelayakan kebutuhan masing-masing perusahaan. Sehingga sering tidak mencapai hasil yang diinginkan. Oleh karena itu pada Tugas Akhir ini akan dibahas bagaimana menemukan aturan-aturan dalam Algoritma Genetika sebagai metode ilmiah dalam proses penempatan pegawai yang akan dibahas pada subbab 4.4. Disamping itu pada Tugas Akhir ini juga akan dilakukan pengembangan prototipe perangkat lunak yang mampu melakukan proses penempatan pegawai dengan memperhatikan aturan-aturan yang berlaku sehingga sesuai dengan harapan pegawai dan kebutuhan perusahaan.

#### 4.2.2 Conditional Analysis (Analisis Persyaratan)

Untuk membangun sebuah *database* berdasarkan analisis sistem diatas, dibutuhkan langkah-langkah agar menjadi *database* yang baik. Langkah pertama dalam membangun sebuah *database* adalah mengetahui dan memahami keinginan *user* serta sistem yang akan dibangun. Pada Tugas Akhir ini terdapat data awal dan analisis kebutuhan *user*.

Data awal adalah data yang dibutuhkan sistem untuk menjalankan program Algoritma Genetika. Data tersebut adalah sebagai berikut:

##### 1. Data Pilihan Pegawai

Data pilihan pegawai didapatkan dari bidang yang diinginkan oleh pegawai saat proses penerimaan pegawai dan diterjemahkan oleh pakar kedalam departemen yang sesuai. Pada proses

pengambilan data, didapatkan 11 data pegawai sebagai data training. Berikut adalah tabel yang menunjukkan alokasi departemen oleh pakar atas bidang yang dipilih pegawai dan tabel daftar pilihan pegawai setelah dirubah ke bentuk biner. Angka 1 menunjukkan bahwa pegawai memilih departemen yang tersebut dan angka 0 menunjukkan sebaliknya.

**Tabel 4.16.** Alokasi Departemen Oleh Pakar Atas Bidang Yang Dipilih Pegawai

<b>Pegawai</b>	<b>Bidang yang Diinginkan</b>	<b>Alokasi bidang ke Departemen</b>
P1	Akuntansi/keuangan	Akuntansi, Anggaran, Keuangan
P2	Teknis/Engineering	Pemeliharaan I, Pemeliharaan II, Pemeliharaan III, PPK
P3	Engineer	PPK
P4	instrumentasi kontrol	PPK, Prod I, Prod IIA, Prod IIB
P5	Berhubungan dengan pengukuran, instrumentasi, sistem integrasi	Pemeliharaan I, Pemeliharaan II, Pemeliharaan III, PPK, Peng. Pelabuhan, Prod I, Prod IIA, Prod IIB.
P6	Maintenance/System Engineer/HSE	Pemeliharaan I, Pemeliharaan II, Pemeliharaan III, PPK
P7	Kontrol	Prod I, Prod IIA, Prod IIB.
P8	IT	Peng. Pelabuhan
P9	Operator/Technic	Prod I, Prod IIA, Prod IIB, Pemeliharaan I, Pemeliharaan II, Pemeliharaan III
P10	Operator	Prod I, Prod IIA, Prod IIB,
P11	Maintenance	Pemeliharaan I, Pemeliharaan II, Pemeliharaan III

**Tabel 4.17.** Pilihan Pegawai

	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11
P1	1	1	1	0	0	0	0	0	0	0	0
P2	0	0	0	1	1	1	0	1	0	0	0
P3	0	0	0	0	0	0	0	1	0	0	0
P4	0	0	0	0	0	0	0	1	1	1	1
P5	0	0	0	1	1	1	1	1	1	1	1
P6	0	0	0	1	1	1	0	1	0	0	0
P7	0	0	0	0	0	0	0	0	1	1	1
P8	0	0	0	0	0	0	1	0	0	0	0
P9	0	0	0	1	1	1	0	0	1	1	1
P10	0	0	0	0	0	0	0	0	1	1	1
P11	0	0	0	1	1	1	0	0	0	0	0

## 2. Data Nilai Pegawai

Terdapat dua jenis data nilai pegawai dikarenakan menyesuaikan model dua arah. Yaitu data nilai pegawai dari penilaian perusahaan (nilai A) dan nilai pegawai dari penilaian dirinya sendiri (nilai B).

**Tabel 4.18.** Nilai A

	K1	K2	K3	K4	K5	K6	K7
P1	85,0	87,0	87,0	92,0	88,0	87,0	78,0
P2	93,0	93,0	88,0	88,0	89,0	94,0	83,0
P3	88,0	87,0	87,0	87,0	80,0	85,0	85,0
P4	93,0	90,0	93,0	93,0	91,0	91,0	90,0
P5	92,0	92,0	93,0	91,0	92,0	91,0	90,0
P6	93,0	92,0	89,0	89,0	90,0	90,0	92,0
P7	92,0	92,0	92,0	91,0	92,0	91,0	90,0
P8	90,0	90,0	89,0	87,0	88,0	88,0	85,0
P9	82,5	85,0	85,0	82,5	85,0	82,5	85,0
P10	82,5	85,0	82,5	87,5	85,0	80,0	85,0
P11	85,0	76,0	88,5	80,0	77,5	82,5	75,5

Tabel 4.19. Nilai B

	K1	K2	K3	K4	K5	K6	K7
P1	85,0	90,0	90,0	90,0	100,0	90,0	95,0
P2	80,0	80,0	85,0	90,0	95,0	95,0	95,0
P3	80,0	80,0	80,0	80,0	80,0	80,0	80,0
P4	90,0	80,0	85,0	70,0	100,0	100,0	95,0
P5	80,0	80,0	80,0	85,0	85,0	80,0	85,0
P6	80,0	85,0	82,0	86,0	84,0	87,0	83,0
P7	90,0	90,0	90,0	85,0	100,0	85,0	90,0
P8	80,0	89,0	75,0	80,0	90,0	87,0	65,0
P9	95,0	90,0	90,0	85,0	95,0	90,0	85,0
P10	85,0	85,0	90,0	85,0	90,0	90,0	90,0
P11	90,0	80,0	80,0	80,0	90,0	85,0	80,0

Disamping kedua nilai tersebut terdapat juga nilai minimum yang menjadi batasan dalam penerimaan pegawai pada tahap wawancara. Yaitu sebagai berikut:

Tabel 4.20. Nilai Minimum Kualifikasi

	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11
K1	60	60	60	40	40	40	40	40	40	40	40
K2	70	70	70	70	70	70	70	70	70	70	70
K3	60	60	60	60	60	60	60	60	60	60	60
K4	70	70	70	70	70	70	70	70	70	70	70
K5	60	60	60	60	60	60	60	60	60	60	60
K6	70	70	70	70	70	70	70	70	70	70	70
K7	60	60	60	50	50	50	50	50	50	50	50

### 3. Bobot dan Kuota Departemen Serta Kualifikasi

Bobot departemen dan kualifikasi didapatkan dari perhitungan AHP pada subbab 4.1. Selain itu terdapat kuota pegawai dalam setiap departemen yaitu sebagai berikut:

**Tabel 4.21.** Bobot dan Kuota Departemen

Departemen	Bobot	Kuota
D1	0,024	3 pegawai
D2	0,023	1 pegawai
D3	0,052	9 pegawai
D4	0,105	3 pegawai
D5	0,108	6 pegawai
D6	0,100	4 pegawai
D7	0,062	3 pegawai
D8	0,024	2 pegawai
D9	0,153	1 pegawai
D10	0,174	1 pegawai
D11	0,175	2 pegawai

**Tabel 4.22.** Bobot Kualifikasi Dalam Setiap Departemen

	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11
K1	0,050	0,032	0,043	0,032	0,032	0,034	0,030	0,037	0,035	0,034	0,035
K2	0,189	0,188	0,170	0,289	0,267	0,308	0,214	0,284	0,283	0,295	0,302
K3	0,090	0,138	0,121	0,111	0,130	0,138	0,173	0,140	0,147	0,112	0,104
K4	0,257	0,218	0,284	0,243	0,289	0,207	0,274	0,284	0,274	0,284	0,253
K5	0,084	0,093	0,083	0,098	0,101	0,106	0,094	0,081	0,094	0,107	0,134
K6	0,244	0,193	0,195	0,139	0,120	0,122	0,133	0,088	0,109	0,112	0,111
K7	0,086	0,137	0,103	0,087	0,061	0,085	0,082	0,086	0,059	0,057	0,062

Sedangkan kebutuhan *user* terhadap perangkat lunak yang akan dibuat sesuai hasil wawancara antara lain:

1. Menampilkan data.
2. Dapat mengelola data nilai pegawai, diantaranya menambahkan data pegawai baru dan edit data pegawai.
3. Melakukan proses perhitungan sehingga diperoleh hasil penempatan pegawai.
4. Menampilkan hasil perhitungan.
5. Menampilkan hasil penempatan pegawai ke dalam bentuk yang mudah dipahami dan interaktif.

Setelah melakukan Analisis Persyaratan, langkah selanjutnya adalah memodelkan interaksi antara aktor dengan sistem. Berdasarkan analisis sistem yang telah dilakukan, terdapat dua aktor sebagai pengguna perangkat lunak ini. Yaitu penulis sebagai Admin dan satu *user* dari departemen Personalia PT. Petrokimia Gresik. Aktor Admin dalam perangkat lunak ini juga berfungsi untuk membatasi hak akses *user*. Misalkan *user* bisa mengakses data departemen dan kualifikasi namun tidak bisa melakukan edit atau pun delete. Karena data departemen dan kualifikasi masuk dalam perhitungan AG sehingga jika dilakukan perubahan, akan mengacaukan perhitungan. Data tersebut adalah bobot departemen dan bobot kualifikasi dalam setiap departemen. Untuk memodelkan interaksi antara aktor dengan sistem tersebut, maka digambarkan dalam *Use-case Diagram*.

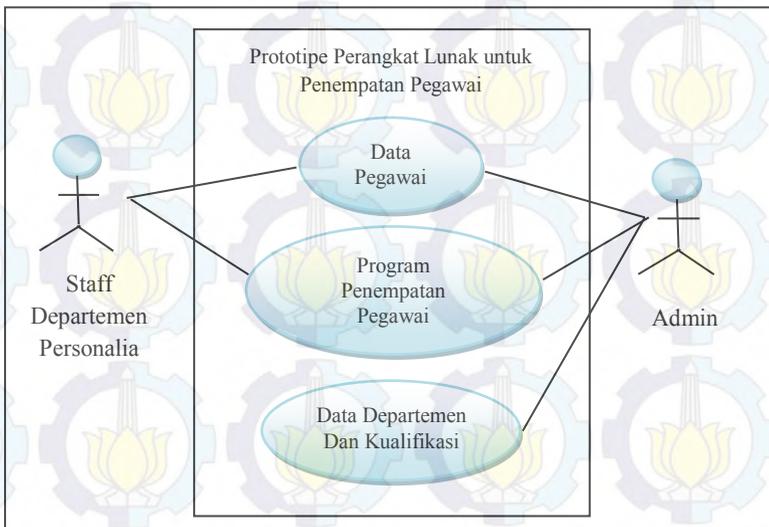
#### 4.2.3 *Use-case Diagram*

*Use-case diagram* merupakan model diagram UML yang digunakan untuk menggambarkan kebutuhan fungsional yang diharapkan dari sebuah sistem. *Use-case diagram* menekankan pada “siapa” melakukan “apa” dalam lingkungan sistem perangkat lunak yang akan dibangun. Berdasarkan uraian diatas, *use-case* dan interaksinya dengan aktor ditunjukkan dalam Tabel 4.23 berikut:

**Tabel 4.23.** Interaksi Antara Aktor dengan Sistem

No	Use Case	Deskripsi	Aktor
1	Data Pegawai	Use Case ini menggambarkan kegiatan memeriksa data nilai pegawai untuk melakukan edit, delete dan tambah data pegawai.	Staff Personalia, Admin
2	Program Penempatan Pegawai	Use Case ini menggambarkan kegiatan <i>input</i> parameter dalam Algoritma Genetika serta melihat hasil penempatan pegawainya.	Staff Personalia, Admin
3	Data Departemen dan Kualifikasi	Use Case ini menggambarkan kegiatan melihat data Departemen dan Kualifikasi untuk melakukan edit.	Admin

Sehingga *Use-case diagram* yang menggambarkan interaksi diatas ditunjukkan dalam Gambar 4.6 sebagai berikut:

**Gambar 4.2.** Use-case Diagram

Pada saat awal perangkat lunak dijalankan, akan muncul form login. Setelah id dan password dimasukkan, maka pengguna akan dibawa ke halaman utama perangkat lunak. Pada halaman tersebut terdapat tiga menu utama yaitu Menu, Data, dan Process. Pada menu dengan nama Menu terdapat tiga submenu, yaitu submenu Home, Help dan Exit. Submenu Home digunakan untuk kembali ke halaman utama, submenu Help digunakan sebagai panduan singkat penggunaan perangkat lunak untuk *user* dan terdapat juga informasi tentang pembuat perangkat lunak yang berisi biodata singkat penulis dan pembimbing penulis pada Tugas Akhir ini. Pada menu Data terdapat tiga submenu, yaitu Data Pegawai, Data Departemen, Data Kualifikasi, dan Insert Data Pegawai. Sedangkan pada menu Process terdapat program penempatan pegawai dengan Algoritma Genetika.

Selanjutnya untuk menggunakan program penempatan pegawai, *user* harus menginputkan terlebih dahulu parameter AG yang meliputi, *Probabilitas Crossover*, *Probabilitas Mutasi*, Jumlah Populasi dan Jumlah Iterasi. Kemudian proses dijalankan dan sistem akan menampilkan hasil penempatan pegawai jika proses telah selesai.

Untuk melihat pegawai yang akan ditempatkan, *user* dapat melihat di submenu Data Pegawai pada menu Data. Pada submenu tersebut terdapat juga tombol untuk melakukan edit data pegawai. Jika *user* ingin menambahkan data pegawai baru, *user* dapat melakukannya dalam submenu Insert Data Pegawai dan sistem akan menampilkan form untuk diisi oleh *user*. Untuk memodelkan aktivitas *user* pada penggunaan perangkat lunak sesuai deskripsi dan *use-case* diatas, maka digambarkan dalam bentuk *Activity/Swimlane Diagram*.

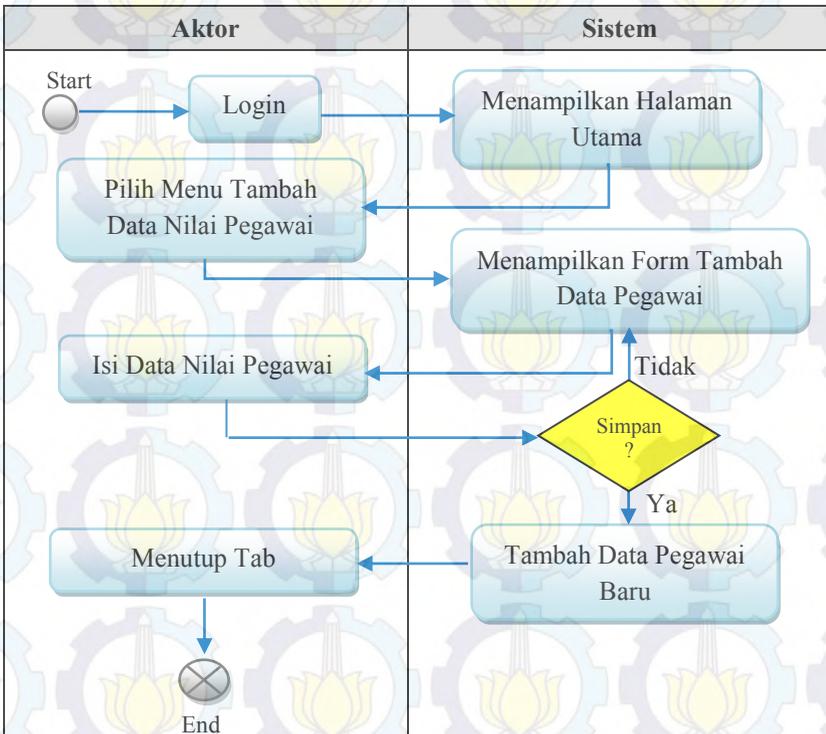
#### 4.2.4 Activity/Swimlane Diagram

*Activity/Swimlane Diagram* adalah sebuah cara untuk memodelkan alur kerja (*workflow*) dari *use-case* dalam bentuk grafik. Terdapat tiga *Case* dalam *use-case* diagram diatas. Sehingga *Activity Diagram* yang menggambarkan proses tersebut

juga terdapat tiga aktivitas. Yaitu aktivitas Data Pegawai, Program Penempatan Pegawai, Data Departemen dan Kualifikasi. Untuk *use-case* “Data Pegawai” terdapat dua Diagram Aktivitas. Antara lain, aktivitas “Tambah Data Pegawai”, “Edit dan Hapus Data Pegawai”.

### 1. *Activity Diagram* “Tambah Data Pegawai”

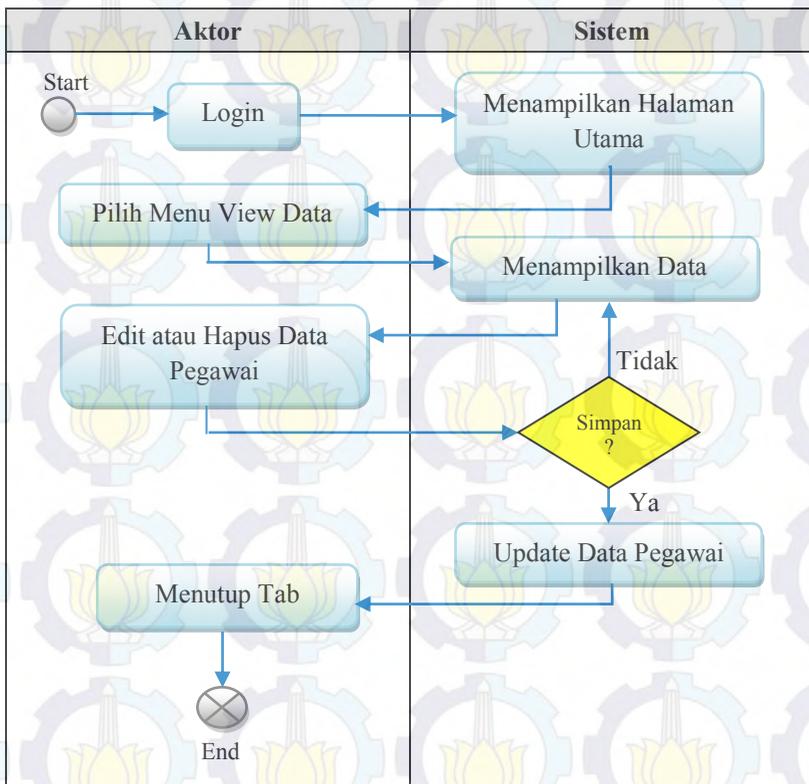
Diagram aktivitas ini menjelaskan bagaimana *user* melakukan penambahan data pegawai. Aktor menginputkan data yang diperlukan dan sistem menyimpan data tersebut. Diagram aktivitas “Tambah Data Pegawai” ditunjukkan sebagai berikut:



**Gambar 4.3.** *Activity Diagram* “Tambah Data Pegawai”

## 2. Activity Diagram “Edit dan Hapus Data Pegawai”

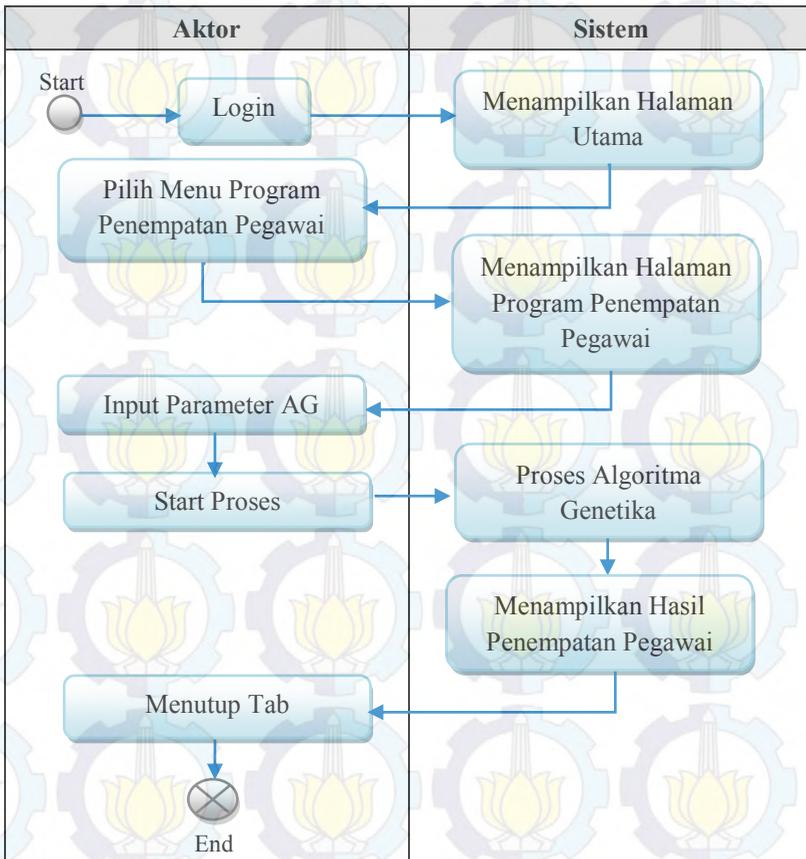
Diagram aktivitas ini menjelaskan bagaimana *user* melakukan perubahan data nilai pegawai dan menghapus data pegawai. Sistem menampilkan halaman data pegawai yang sudah tersimpan. Kemudian aktor mengubah atau menghapus data nilai yang diinginkan dan sistem menyimpan data tersebut. Diagram aktivitas “Edit dan Hapus Data Pegawai” ditunjukkan dengan Gambar 4.8 berikut.



**Gambar 4.4.** Activity Diagram “Edit dan Hapus Data Pegawai”

### 3. *Activity Diagram* dari *use-case* “Program Penempatan Pegawai”

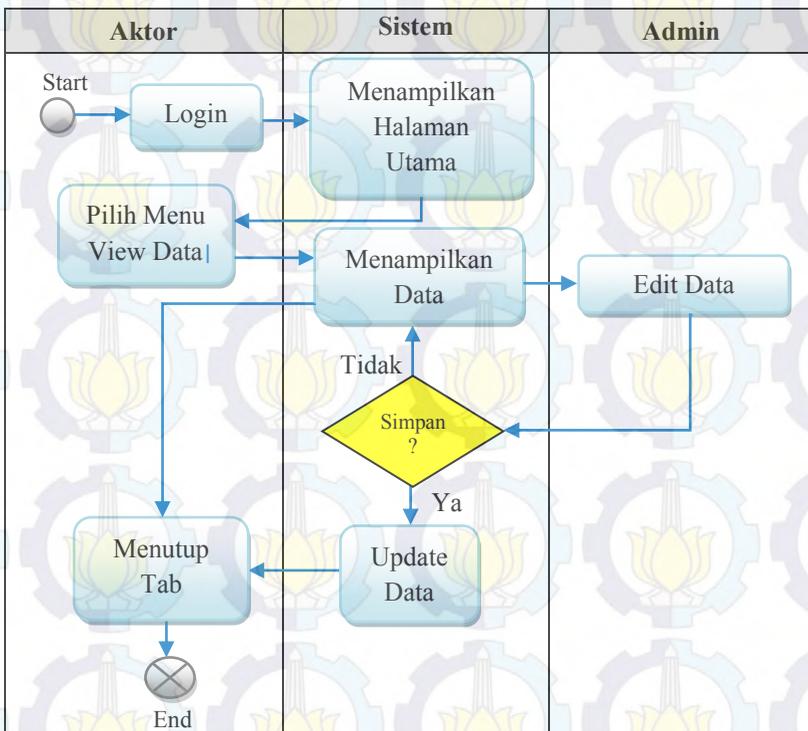
Diagram aktivitas dibawah ini menjelaskan bagaimana aktor menginputkan parameter AG kemudian sistem melakukan penghitungan. Sistem menampilkan formasi penempatan pegawai sebagai hasilnya. Diagram aktivitas “Program Penempatan Pegawai” ditunjukkan dengan Gambar 4.9 berikut.



**Gambar 4.5.** *Activity Diagram* “Program Penempatan Pegawai”

#### 4. *Activity Diagram* dari *use-case* “Data Departemen dan Kualifikasi”

Diagram aktivitas dibawah ini menjelaskan bagaimana *user* melihat dan admin melakukan edit pada data departemen dan kualifikasi. Sistem menampilkan halaman data departemen dan kualifikasi. Kemudian aktor memeriksa data tersebut sedangkan admin dapat mengubah data dan sistem menyimpannya. Diagram aktivitas “Data Departemen dan Kualifikasi” ditunjukkan dengan Gambar 4.10 berikut.



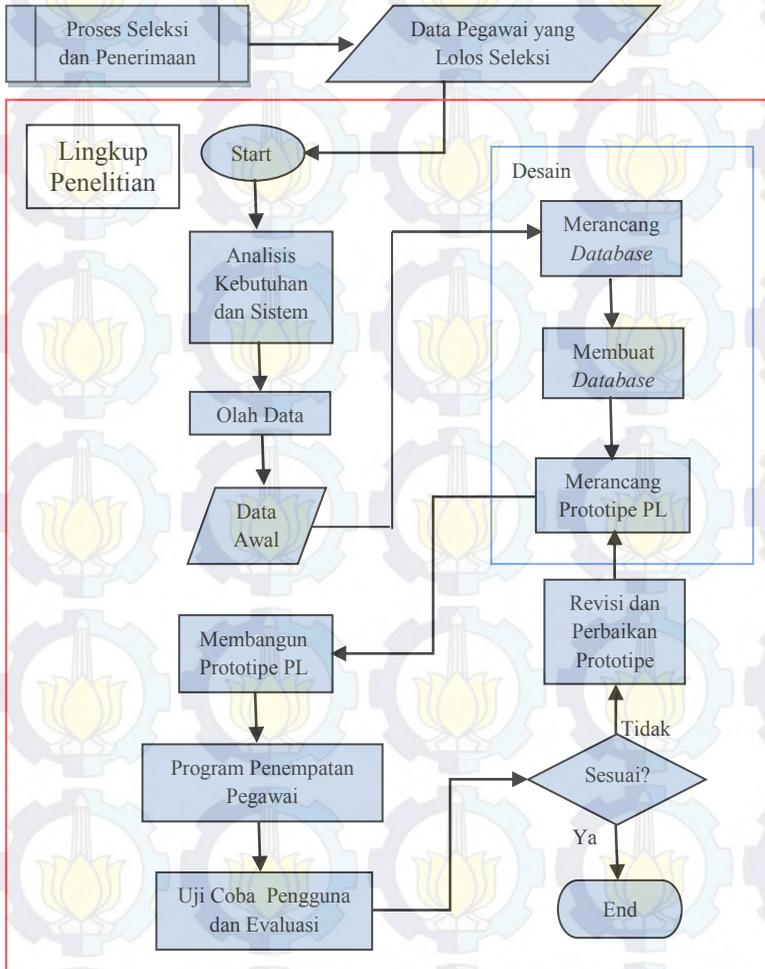
**Gambar 4.6.** *Activity Diagram* “Data Departemen dan Kualifikasi”

### 4.3 Perancangan Sistem

Perancangan sistem dalam perangkat lunak penempatan pegawai ini meliputi Perancangan Proses, Perancangan *Database*, dan Proses Prototyping Perangkat Lunak.

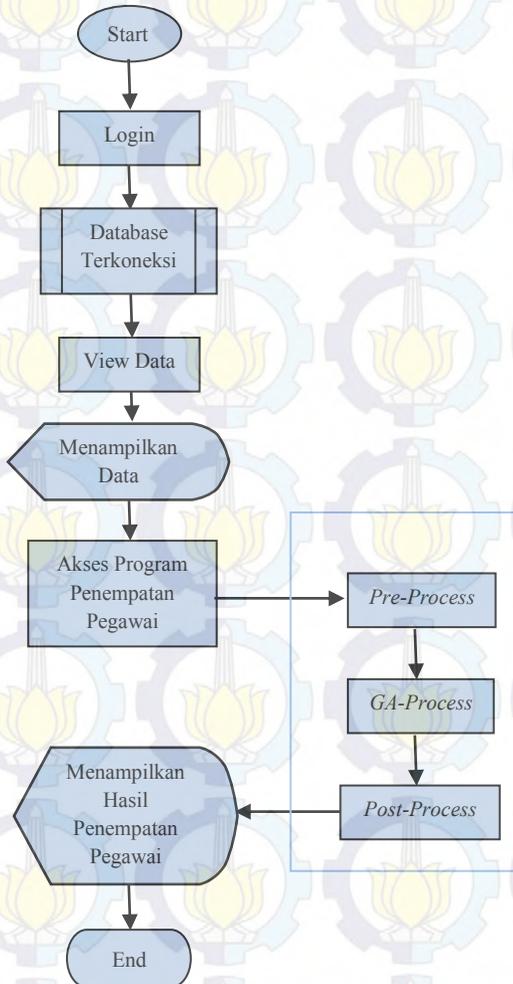
#### 4.3.1 Perancangan Proses

Perancangan proses ini meliputi Perancangan Prototipe Perangkat Lunak dan Perancangan Program Penempatan Pegawai. Dibawah ini merupakan diagram alir yang menggambarkan proses pembuatan prototipe perangkat lunak penempatan pegawai. Metode prototyping dimulai pada tahap desain, yang meliputi: perancangan prototipe, membangun prototipe PL sesuai perancangan, mengujikan prototipe kepada *user* dan *user* memberikan evaluasinya, jika masih tidak sesuai dengan keinginan *user*, maka dilakukan revisi dan perbaikan prototipe. Kemudian terdapat garis merah yang menandakan batas lingkup penelitian yang dilakukan oleh penulis.



**Gambar 4.7.** Diagram Alir Prototipe PL

Pada gambar diatas, tahap Program Penempatan Pegawai dapat ditunjukkan dengan Diagram Alir berikut:



**Gambar 4.8.** Diagram Alir Program Penempatan Pegawai

Dari Diagram Alir yang dipaparkan diatas, dibawah ini dipaparkan bagaimana menemukan aturan-aturan dalam Algoritma Genetika dalam proses penempatan pegawai. Program penempatan pegawai dalam perangkat lunak ini mempunyai tiga

tahap, yaitu tahap *Pre-Process*, tahap *GA-Process* dan tahap *Post-Process*.

### 1. Tahap *Pre-Process*

Tahap ini merupakan tahap persiapan untuk menjalankan proses Algoritma Genetika. Seperti yang telah dijelaskan diatas, sebelum tahap ini dijalankan, *user* harus menginputkan parameter AG terlebih dahulu. Yaitu *Probabilitas Crossover*, *Probabilitas Mutasi*, Jumlah Populasi dan Jumlah Iterasi. Kemudian *user* melakukan perintah untuk menjalankan proses AG.

Tahap *GA-Process* dan tahap *Post-Process* membutuhkan data-data dalam *database* yang telah disebutkan pada subab 4.2. oleh karena itu pada tahap ini dilakukan pemanggilan data dari *database*. Data tersebut antara lain:

#### 1. Data Pilihan Pegawai ( $x_{ij}$ ).

Data pilihan pegawai digunakan untuk mengetahui departemen apa saja yang dipilih oleh pegawai sehingga pada saat pembangkitan kromosom, random posisi tidak melebar ke departemen yang bukan menjadi pilihan pegawai tersebut.

#### 2. Nilai-A Pegawai ( $a_{jl}$ ), Nilai-B Pegawai ( $b_{jr}$ ) dan Nilai Minimum ( $d_{jl}$ ).

Ketiga nilai ini digunakan untuk menentukan nilai  $p_{ij}$  dan  $q_{ij}$  dalam fungsi fitness. Selain itu Nilai-A Pegawai dan Nilai Minimum digunakan juga untuk menentukan nilai akhir pegawai dalam tahap *Post-Process*.

#### 3. Bobot Departemen ( $E_i$ ).

#### 4. Bobot Kualifikasi ( $\alpha_{il}$ ).

#### 5. Kuota Departemen.

Kuota departemen adalah batas jumlah pegawai yang boleh memasuki departemen. Data ini digunakan untuk melihat apakah terdapat departemen yang melebihi kuota sebagai hasil dari tahap *GA-Process*.

Setelah semua data dipanggil, maka langkah selanjutnya adalah membangkitkan kromosom. Pembangkitan kromosom dilakukan dengan *men-generate* matriks *be-rordo*

pegawai  $x$  departemen yang berisi bilangan random 0 dan 1 dengan aturan setiap baris matriks hanya diisi satu angka 1 dan memperhatikan pilihan departemen yang diminati pegawai ( $x_{ij}$ ) sesuai dengan Tabel 4.16.

**Tabel 4.24.** Contoh Matriks Random dengan Bilangan 0 dan 1

Pegawai	Departemen										
	1	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	1	0	0	0	0	0	0
	0	0	0	0	0	0	0	1	0	0	0
	0	0	0	0	0	0	0	0	0	1	0
	0	0	0	0	0	0	0	0	1	0	0
	0	0	0	0	0	0	0	1	0	0	0
	0	0	0	0	0	0	0	0	1	0	0
	0	0	0	0	0	0	1	0	0	0	0
	0	0	0	0	0	0	0	0	1	0	0
	0	0	0	0	0	0	0	0	1	0	0
	0	0	0	1	0	0	0	0	0	0	0

Karena satu kromosom merepresentasikan satu solusi, maka matriks tersebut diubah kedalam bentuk  $1 \times n$  dengan  $n$  perkalian banyaknya pegawai dengan banyaknya departemen seperti dibawah ini:

P1                      P2                      ...                      P10                      P11  
 [1000000000 | 0000100000 | ... | 00000000100 | 0001000000]

Kemudian di-generate lagi sebanyak populasi. Sehingga diperoleh populasi awal yang merepresentasikan solusi.

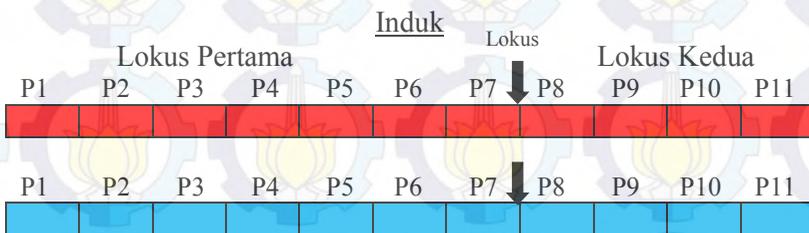
## 2. Tahap GA-Process

Tahap ini merupakan tahap untuk mendapatkan formasi pegawai yang optimal dengan nilai fitness paling tinggi. Setelah generasi pertama didapatkan, nilai fitness tiap kromosom dihitung. Fungsi fitness yang digunakan adalah fungsi objektif (2.1) yang memaksimalkan formasi pegawai dalam departemen ( $x_{ij}$ ) dengan memperhatikan tingkat kepentingan setiap

departemen ( $E_i$ ) serta nilai-nilai pegawai dari pandangan perusahaan ( $p_{ij}$ ) dan dari pandangan pegawai sendiri ( $q_{ij}$ ).

Langkah selanjutnya adalah seleksi induk dengan metode seleksi yang digunakan adalah *roulette-wheel*. Dimana masing-masing individu menempati potongan lingkaran pada roda *roulette* secara proporsional sesuai dengan nilai fitness yang telah dihitung. Kemudian setiap kromosom dihitung nilai proporsi dan nilai proporsi kumulatif. Semakin tinggi nilai fitnessnya, semakin tinggi pula nilai proporsi kumulatif kromosom tersebut. Sehingga peluang untuk terpilih menjadi induk juga semakin tinggi.

Setelah induk terpilih, maka dilakukan pindah silang atau *crossover*. *Crossover* dilakukan dengan menggunakan metode *single-point* yaitu menentukan lokus dari setiap dua *parent* yang terpilih. Lokus kedua individu pertama ditukar dengan lokus kedua individu kedua. Untuk tetap menjaga formasi pegawai yang telah disusun, penentuan letak lokus dilakukan secara random sesuai dengan kelipatan jumlah departemen. Hasil dari *crossover* ini adalah 2 individu baru sebanyak populasi. Sehingga dihasilkan populasi baru dengan jumlah yang sama. Selanjutnya populasi yang lama dengan populasi yang baru disatukan dan disorting berdasarkan nilai fitness tertinggi yang akan dipilih sejumlah populasi awal. Namun individu yang akan di *crossover* ditentukan dengan parameter *probabilitas crossover* ( $P_c$ ). Dibawah ini merupakan ilustrasi contoh proses *crossover*. Terdapat dua individu yang disimbolkan dengan warna merah dan biru dan tanda panah yang menunjukkan letak lokus.





Alur kerja atau algoritma proses ini adalah sebagai berikut:

### **Proses Crossover**

Random angka desimal dengan range 0-1;

Jika hasil Random  $< P_c$

1. Pilih induk dari proses seleksi;
2. Random posisi lokus;
3. Tukar posisi lokus;
4. Sortir berdasarkan fitness tertinggi untuk menjadi populasi di iterasi berikutnya;

Mutasi dilakukan pada tingkat bit dengan cara menukarkan bit pada gen pegawai yang sama dalam satu kromosom. Mutasi juga ditentukan dengan parameter probabilitas mutasi ( $P_m$ ). Alur kerja atau algoritma proses ini adalah sebagai berikut:

### **Proses Mutasi**

Random angka desimal dengan range 0-1;

Jika hasil Random  $< P_m$

1. Cari atau pindai angka 1 secara random dalam satu kromosom;
2. Tentukan pegawai mana yang sesuai dengan hasil pemindaian angka 1;
3. Cek pilihan pegawai tersebut;
4. Pilih posisi pegawai tersebut secara acak;
5. Tukar atau mutasikan sesuai posisi pegawai yang terpilih dengan mengganti angka 0 dengan 1;

Untuk penghentian proses iterasi, dilakukan dengan memberi batasan jumlah iterasi dan memberikan batasan nilai konvergen sebesar 90% kepada nilai fitness. Tujuan pengambilan nilai

konvergen sebesar 90% adalah untuk menghindari konvergen prematur jika konvergennya dibawah 90%.

### 3. Tahap Post-Process

Setelah tahap *GA-Process* selesai dan didapatkan formasi penempatan pegawai, tahap *Post-Process* dibutuhkan untuk mengatasi departemen yang mengalami over kuota. Langkah yang dibutuhkan adalah yang pertama ambil hasil formasi penempatan pegawai dari tahap *GA-Process*. Kemudian hitung nilai akhir pegawai dengan rumus berikut:

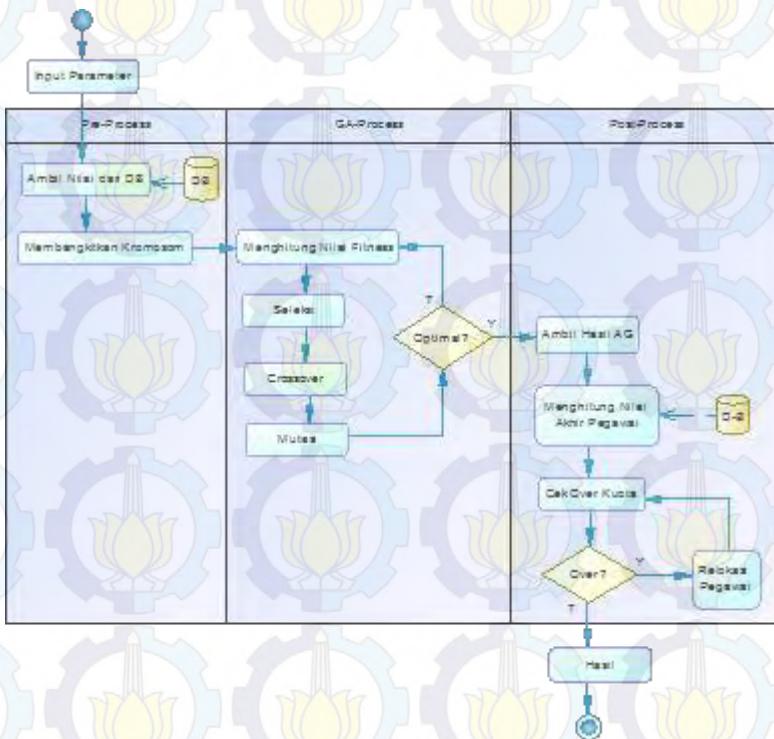
$$\text{nilai akhir} = \frac{\sum a_j}{k} - \frac{\sum d_l}{n} \quad (4.1)$$

Dimana  $a_j$  adalah nilai A dari pegawai,  $k$  adalah jumlah pegawai,  $d_l$  adalah nilai minimum tiap kualifikasi, dan  $n$  adalah jumlah departemen. Kemudian cek apakah terdapat departemen yang mengalami over kuota. Jika iya relokasi pegawai yang ada di departemen tersebut ke departemen lain sesuai pilihannya. Alur kerja atau algoritma proses ini adalah sebagai berikut:

#### **Proses Relokasi Pegawai**

1. Ambil hasil formasi pegawai dari *GA-Process*;
  2. Hitung nilai akhir tiap pegawai;
  3. Cek over kuota dalam departemen;
- Jika over kuota
4. Relokasi pegawai;
  5. Ulangi langkah 3 sampai tidak ada departemen yang over kuota;

Program penempatan pegawai yang mempunyai tiga tahap, yaitu tahap *Pre-Process*, tahap *GA-Process* dan tahap *Post-Process* ini dapat digambarkan pada diagram alir berikut:



**Gambar 4.9.** Diagram Alir *Pre-Process*, *GA-Process* dan *Post-Process*

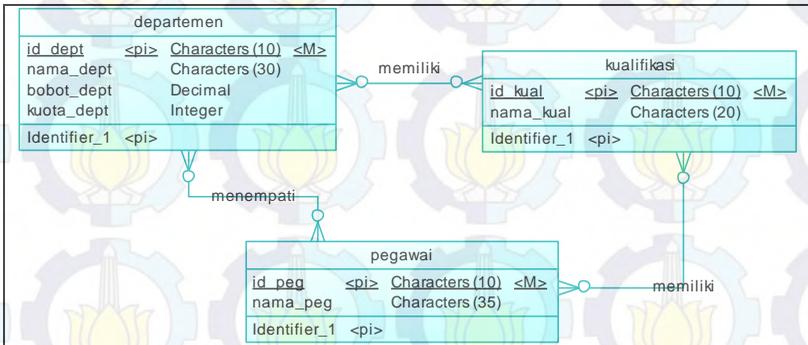
#### 4.3.2 Perancangan Database

Berdasarkan deskripsi sistem yang telah dipaparkan dan data yang telah didapatkan, dituangkan dalam rancangan database yang meliputi Desain Konsep *Database* atau *Conceptual Database Design*, *Logical Database Design*, dan *Physical Database Design*, serta implementasinya dalam DBMS MySQL.

##### 1. *Conceptual Database Design*

*Conceptual Database Design* adalah proses membangun model data yang digunakan didalam suatu perusahaan, bersifat independent dari semua pertimbangan fisik [9]. Pada tahap ini

informasi-informasi yang terdapat dalam tahap *Conditional Analysis* digunakan untuk mengembangkan deskripsi data yang disimpan dalam *database*. Langkah ini sering diproyeksikan dengan menggunakan model ER (*Entity Relational*). *Conceptual Data Model (CDM)* dalam permasalahan ini adalah sebagai berikut:

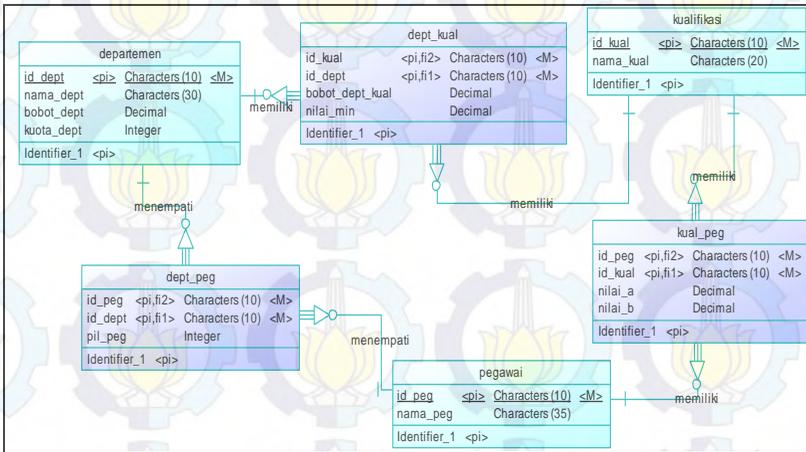


**Gambar 4.10.** *Conceptual Data Model*

Pada gambar diatas dapat dilihat bahwa CDM tersebut mempunyai 3 entitas, yaitu entitas departemen yang mempunyai entitas departemen, kualifikasi, dan pegawai. Ketiga entitas tersebut dihubungkan dengan relasi yang memiliki kardinalitas *many-to-many*.

## 2. *Logical Database Design*

*Logical Database Design* adalah proses membangun model data yang digunakan didalam suatu perusahaan, bersifat independen terhadap DBMS tertentu dan segala pertimbangan fisik lainnya [9]. Pada langkah ketiga ini CDM atau ER yang telah dibuat diubah menjadi sebuah skema *database* relasional dari DBMS yang digunakan. *Logical Data Model* dari CDM diatas menjadi seperti berikut:

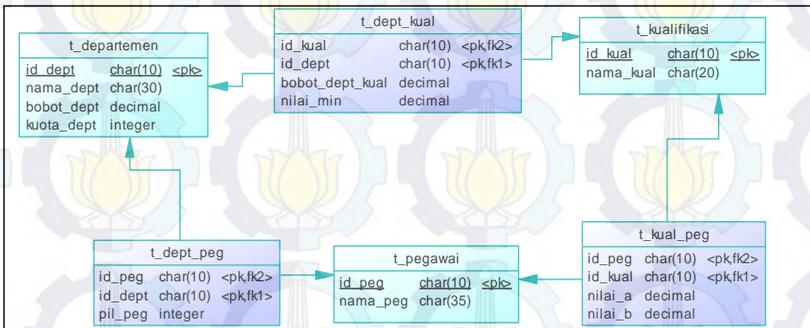


**Gambar 4.11. Logical Data Model**

Pada gambar *Logical Data Model* diatas, relasi antar entitas mempunyai atribut. Untuk relasi antara entitas departemen dengan entitas kualifikasi terdapat atribut “bobot\_dept\_kual” yang merepresentasikan Tabel 4.22 dan atribut “nilai\_min” yang merepresentasikan Tabel 4.20. Untuk relasi antara entitas departemen dengan entitas pegawai terdapat atribut “pil\_peg” yang merepresentasikan Tabel 4.17. Dan untuk relasi antara pegawai dengan kualifikasi terdapat atribut “nilai\_a” dan “nilai\_b” yang merepresentasikan Tabel 4.18 Dan Tabel 4.19.

### 3. *Physical Database Design*

*Physical Database Design* merupakan model data yang memperlihatkan akibat dari relasi antar entitas yang ada. Entitas akan menjadi sebuah tabel dan atribut akan menjadi sebuah field. Sedangkan relasi yang mempunyai kardinalitas *many-to-many* akan menjadi sebuah tabel baru. *Physical Data Model* dari CDM dan LDM diatas adalah sebagai berikut:



**Gambar 4.12.** *Physical Data Model*

Pada gambar diatas dapat ditunjukkan bahwa PDM tersebut mempunyai 3 tabel master, yaitu tabel *t\_departemen* dengan *primary key* *id\_dept*, tabel *t\_kualifikasi* dengan *primary key* *id\_kual* dan tabel *t\_pegawai* dengan *primary key* *id\_peg*.

#### 4. *Generate Database*

Proses generate *database* menghasilkan file teks SQL. File teks tersebut digunakan pada *database* MySQL sebagai query untuk membuat *database* baru beserta tabel dan *field*-nya. Sehingga design pada *database* MySQL menjadi seperti berikut:



**Gambar 4.13.** Design Database pada MySQL

Pada gambar design *database* diatas terdapat tambahan tabel *t\_user* yang digunakan untuk hak akses pengguna yang dibedakan atas admin perangkat lunak dan *user* yang dalam hal ini adalah pihak dari Departemen Personalia PT. Petrokimia Gresik.

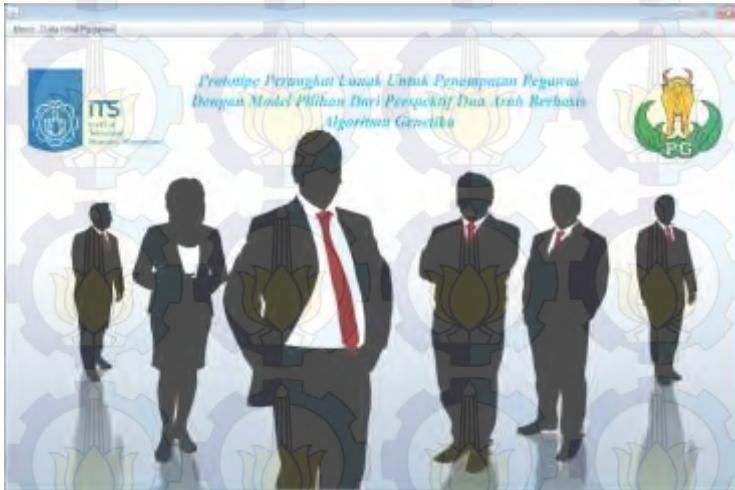
#### 4.3.3 Proses Prototyping Perangkat Lunak

Sesuai definisi prototipe PL di bab 2.1, yaitu Prototipe merupakan metodologi pengembangan perangkat lunak yang menekankan pada pendekatan aspek desain, fungsi dan *user-interface*. Pihak pengembang dan *user*, fokus pada *user-interface* dan bersama-sama mendefinisikan spesifikasi, fungsi, desain dan bagaimana perangkat lunak bekerja, maka pada subbab 4.6 ini akan dipaparkan dokumentasi tentang interaksi antara pihak pengembang yaitu penulis dengan pihak *user*. Interaksi tersebut meliputi pembuatan *user-interface*, evaluasi *user*, dan perbaikan prototipe sampai prototipe perangkat lunak ini sesuai dengan keinginan *user*. Interaksi yang dilakukan penulis dengan *user* dapat dipaparkan sebagai berikut:

1. Tampilan muka pertama

- *User-interface.*

Pada tampilan muka pertama dengan pihak departemen personalia, penulis menunjukkan interface pertama sesuai dengan hasil analisis kebutuhan pengguna. Yaitu sebagai berikut:



**Gambar 4.14.** Halaman Utama



**Gambar 4.15.** Menu dan Submenu Tahap 1

**Algoritma Genetika**

Menu: Data Nilai Pegawai

Parameter Algoritma Genetika:

Jumlah Individu:

Arah Waktu:

Fungsi Fitness:

Populasi Awal:

Populasi Akhir:

Evolusi:

Maksimum Generasi:

Pindah:

Aksi:

Jeda  Data  Clear

**Gambar 4.16.** Form Panel Perhitungan Penempatan Pegawai

**Data Nilai Pegawai (A)**

Menu: Data Nilai Pegawai

No. Pegawai	Nama Pegawai	Mata Pelajaran	Nilai	Kategori	Rata-rata
<input type="text"/>					

Mata Pelajaran:

Nilai:

Simpan

**Gambar 4.17.** Panel Data Nilai A Pegawai

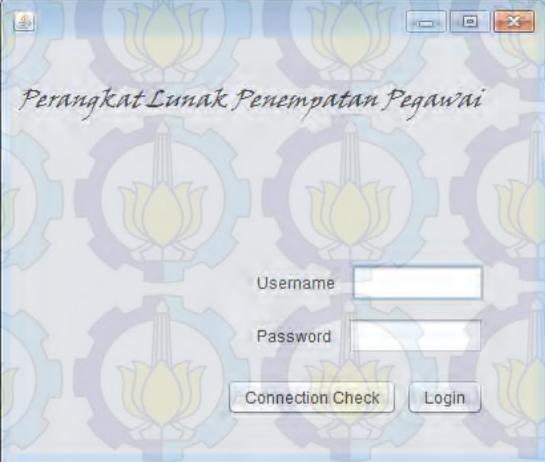




2. Tatap muka kedua

- *User-interface.*

Berdasarkan evaluasi dari tatap muka pertama, maka *User-interface* tahap kedua adalah sebagai berikut:



The image shows a screenshot of a Windows-style application window titled "Perangkat Lunak Penempatan Pegawai". The window has a standard title bar with minimize, maximize, and close buttons. The main content area is light gray and contains the following elements:

- The title "Perangkat Lunak Penempatan Pegawai" in a stylized, italicized font.
- A "Username" label followed by a text input field.
- A "Password" label followed by a password input field.
- Two buttons at the bottom: "Connection Check" and "Login".

**Gambar 4.21.** Form Login Aplikasi

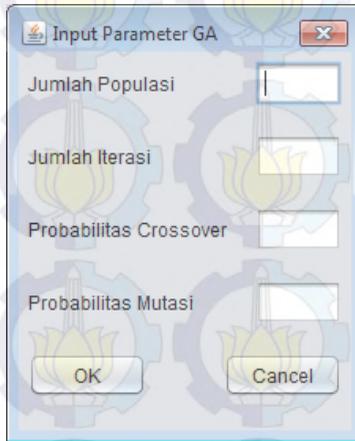


Gambar 4.22. Menu dan Submenu Tahap 2

The screenshot shows the 'View Data' submenu, which displays a data table with the following columns: 'No', 'Nama Pegawai', 'Gaji Pokok Pegawai', 'Jumlah Pegawai', 'Jumlah Pegawai', and 'Status'. The table contains 11 rows of data:

No	Nama Pegawai	Gaji Pokok Pegawai	Jumlah Pegawai	Jumlah Pegawai	Status
1	Andrianto	5.211	1	1	1
2	Anggoro	6.422	1	1	1
3	Anggoro	6.422	1	1	1
4	Anggoro	6.422	1	1	1
5	Anggoro	6.422	1	1	1
6	Anggoro	6.422	1	1	1
7	Anggoro	6.422	1	1	1
8	Anggoro	6.422	1	1	1
9	Anggoro	6.422	1	1	1
10	Anggoro	6.422	1	1	1
11	Anggoro	6.422	1	1	1

Gambar 4.23. Tampilan Submenu View Data



**Gambar 4.24.** Input Parameter AG

- Evaluasi *user*.  
Didapatkan evaluasi *user* yang kedua, yaitu submenu Input Parameter, dan Start GA Process serta menu Result pada program penempatan pegawai kurang sinkron secara penggunaan.
  - Revisi dan perbaikan.  
Revisi dan perbaikan pada tahap kedua ini adalah Start GA proses dilakukan pada input parameter kemudian setelah proses selesai, hasil akan otomatis ditampilkan. Sehingga menu Result dihapuskan.
3. Tatap muka ketiga
- *User-interface*.  
Berdasarkan evaluasi *user* pada tahap sebelumnya, maka *user-interface* tahap ketiga ini adalah sebagai berikut

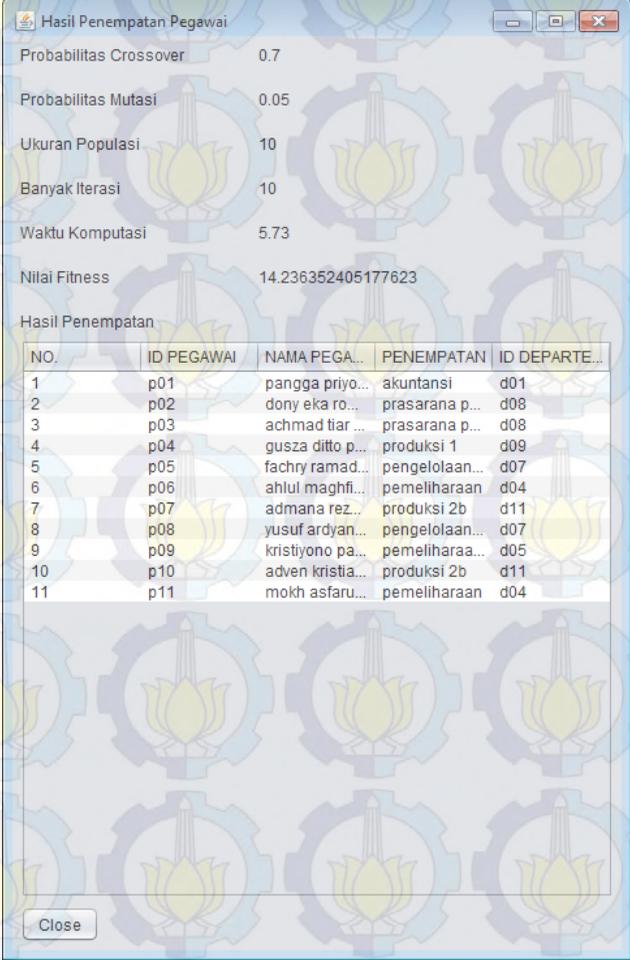


**Gambar 4.25.** Menu dan Submenu Tahap 3

The screenshot shows a dialog box titled "Input Parameter GA" with a close button (X) in the top right corner. It contains the following fields and controls:

- Jumlah Populasi: 10
- Jumlah Iterasi: 1000
- Probabilitas Crossover: 0.7
- Probabilitas Mutasi: 0.05
- A progress bar showing 0% completion.
- Time: 0 s
- Buttons: "Start GA" and "Cancel"

**Gambar 4.26.** Input Parameter dan Start Proses AG Tahap 1



The screenshot shows a window titled "Hasil Penempatan Pegawai" with the following parameters:

- Probabilitas Crossover: 0.7
- Probabilitas Mutasi: 0.05
- Ukuran Populasi: 10
- Banyak Iterasi: 10
- Waktu Komputasi: 5.73
- Nilai Fitness: 14.236352405177623

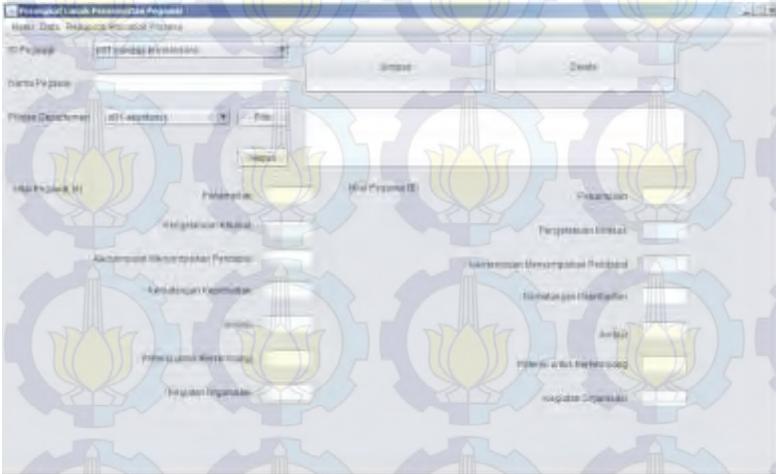
Below the parameters is a table titled "Hasil Penempatan" with the following data:

NO.	ID PEGAWAI	NAMA PEGA...	PENEMPATAN	ID DEPARTE...
1	p01	pangga priyo...	akuntansi	d01
2	p02	dony eka ro...	prasarana p...	d08
3	p03	achmad tiar...	prasarana p...	d08
4	p04	gusza ditto p...	produksi 1	d09
5	p05	fachry ramad...	pengelolaan...	d07
6	p06	ahlul maghfi...	pemeliharaan	d04
7	p07	admana rez...	produksi 2b	d11
8	p08	yusuf ardyan...	pengelolaan...	d07
9	p09	kristiyono pa...	pemeliharaa...	d05
10	p10	adven kristia...	produksi 2b	d11
11	p11	mokh asfaru...	pemeliharaan	d04

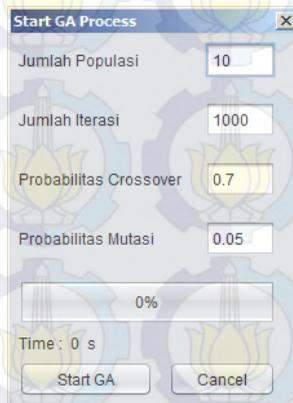
**Gambar 4.27.** Panel Hasil Penempatan Pegawai

- Evaluasi *user*.  
 Pada tatap muka ketiga ini *user* telah memberikan pernyataan bahwa *user*-interface sudah sesuai dengan keinginan. Namun title pada panel tertentu agak sulit dipahami.





**Gambar 4.30.** Form Edit dan Delete Data Pegawai



**Gambar 4.31.** Input Parameter dan Start Proses AG Tahap 2



The screenshot shows a window titled "Hasil Penempatan Pegawai" with the following parameters:

- Probabilitas Crossover: 0.7
- Probabilitas Mutasi: 0.05
- Ukuran Populasi: 10
- Banyak Iterasi: 10
- Waktu Komputasi: 2.72
- Nilai Fitness: 14.034520950222749

Below the parameters is a table titled "Hasil Penempatan" with the following data:

NO.	ID PEGAWAI	NAMA PEGA	PENEMPATAN	ID DEPAR.
1	p01	pangga priyo	akuntansi	d01
2	p02	dony eka ro	pemeliharaan	d04
3	p03	achmad tiar	prasarana pub	d08
4	p04	gusza difo p	produksi 1	d09
5	p05	fachry ramad	pemeliharaan 2	d05
6	p06	ahul maghf	pemeliharaan 2	d05
7	p07	admana rez	produksi 2a	d10
8	p08	yuaf aryan	pengelolaan p	d07
9	p09	kristyono pa	pemeliharaan 3	d06
10	p10	adren krista	produksi 2b	d11
11	p11	mdkh asfaru	pemeliharaan	d04

A "Close" button is visible at the bottom left of the window.

Gambar 4.32. Panel Hasil Penempatan Pegawai

## BAB V

### UJI COBA DAN ANALISIS HASIL

Pada bab ini akan dilakukan uji coba perangkat lunak yang telah dibuat. Pengujian ini meliputi pengujian tampilan antarmuka, pengujian parameter AG dalam program penempatan pegawai, analisis hasil uji coba, dan validasi hasil penempatan pegawai.

#### 5.1 Uji Tampilan Antarmuka Pengguna

Pada subbab ini dibahas tentang pengujian tampilan perangkat lunak. Sesuai perancangan sistem yang dibahas pada bab 4, terdapat dua aktor pada perangkat lunak ini. Yaitu admin dan *user* dari departemen personalia PT. Petrokimia Gresik. Maka pada perangkat lunak ini dibedakan hak akses untuk *user* dan admin. Perbedaannya adalah *user* hanya dapat melihat data departemen dan kualifikasi, tidak dapat melakukan edit. Sedangkan admin dapat melakukannya, karena data departemen dan kualifikasi merupakan bagian dari sistem dan tidak bisa diubah dengan sesuka hati. Berikut ini ditunjukkan screenshot tampilan antar muka untuk *user* dan admin.

##### 5.1.1 Tampilan Antarmuka untuk User

Tampilan antarmuka untuk *user* dimulai dengan login ke aplikasi. Pada form login terdapat *textfield* untuk memasukkan *username* dan *password*. Selain itu terdapat juga tombol *connection check* untuk memeriksa koneksi aplikasi dengan *database* dan tombol *login*.



The screenshot shows a Windows-style application window titled "Peningkat Lunak Penempatan Pegawai". The window contains a login form with the following elements:

- A title bar with standard Windows window controls (minimize, maximize, close).
- A title "Peningkat Lunak Penempatan Pegawai" in a stylized font.
- A "Username" label followed by a text input field.
- A "Password" label followed by a password input field.
- Two buttons at the bottom: "Connection Check" and "Login".

**Gambar 5.1.** Form Login

Jika *database* belum terkoneksi, maka muncul kotak dialog disebelah kiri berikut. Sedangkan jika *database* sudah terkoneksi, maka muncul kotak dialog disebelah kanan berikut ini.



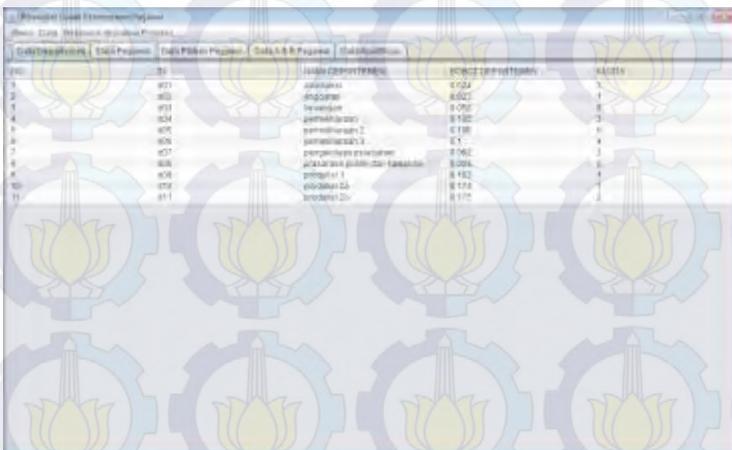
**Gambar 5.2.** Kotak Dialog Connection Check

Kemudian jika login sukses, maka ditampilkan halaman utama seperti berikut ini:



**Gambar 5.3.** Tampilan Utama untuk User

View data pada menu Data berisikan Data Departemen, Data Pegawai, Data Pilihan Pegawai, Data A & B Pegawai, dan Data Kualifikasi yang berbentuk tab.



**Gambar 5.4.** Submenu View Data dengan Tab Data Departemen

No	Jenis	Pegawai
1	001	Andi Nur Hafidza
2	002	Andi Nur Hafidza
3	003	Andi Nur Hafidza
4	004	Andi Nur Hafidza
5	005	Andi Nur Hafidza
6	006	Andi Nur Hafidza
7	007	Andi Nur Hafidza
8	008	Andi Nur Hafidza
9	009	Andi Nur Hafidza
10	010	Andi Nur Hafidza
11	011	Andi Nur Hafidza

**Gambar 5.5.** Submenu View Data dengan Tab Data Pegawai

No	Kategori	Jenis	Tipe	Lama	Jumlah	Kategori	Tipe	Jenis	Lama	Jumlah	Kategori	Tipe	Jenis
1	001	1	1	1	1	1	1	1	1	1	1	1	1
2	002	1	1	1	1	1	1	1	1	1	1	1	1
3	003	1	1	1	1	1	1	1	1	1	1	1	1
4	004	1	1	1	1	1	1	1	1	1	1	1	1
5	005	1	1	1	1	1	1	1	1	1	1	1	1
6	006	1	1	1	1	1	1	1	1	1	1	1	1
7	007	1	1	1	1	1	1	1	1	1	1	1	1
8	008	1	1	1	1	1	1	1	1	1	1	1	1
9	009	1	1	1	1	1	1	1	1	1	1	1	1
10	010	1	1	1	1	1	1	1	1	1	1	1	1
11	011	1	1	1	1	1	1	1	1	1	1	1	1

**Gambar 5.6.** Submenu View Data dengan Tab Data Pilihan Pegawai

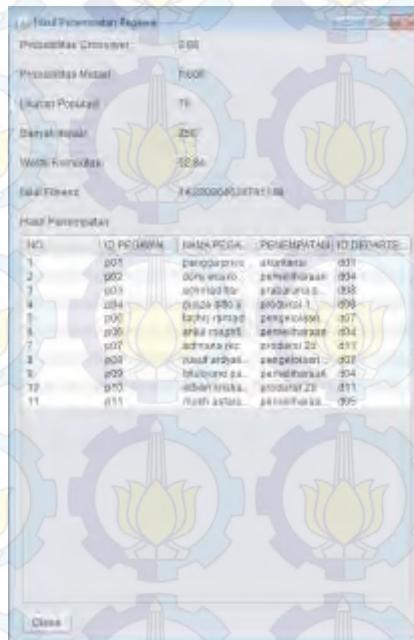




mengetahui berapa lama proses AG yang berjalan. Terdapat juga tombol Start GA untuk memulai proses dan cancel untuk membatalkan proses. Ketika proses selesai, hasil akan ditampilkan yang meliputi parameter yang diinputkan oleh *user*, waktu komputasi, nilai fitness dan hasil penempatan pegawai.



**Gambar 5.11.** Panel Start GA Process



**Gambar 5.12.** Panel Hasil Penempatan Pegawai

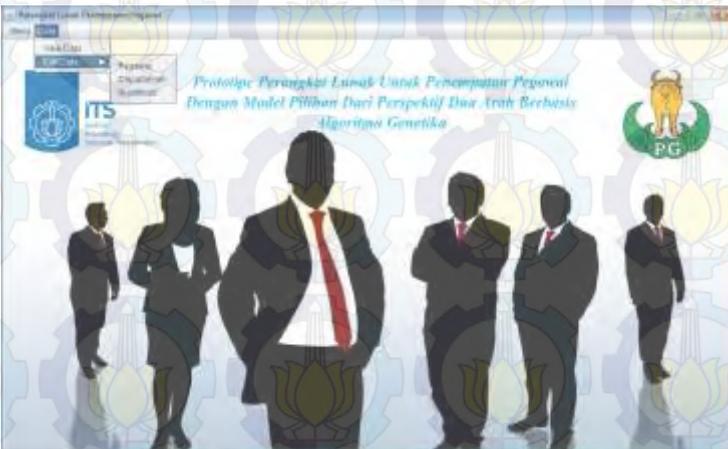
### 5.1.2 Tampilan Antarmuka untuk Admin

Tampilan antarmuka awal untuk admin sama dengan *user*. Setelah login pengguna akan dibawa kehalaman utama seperti berikut:



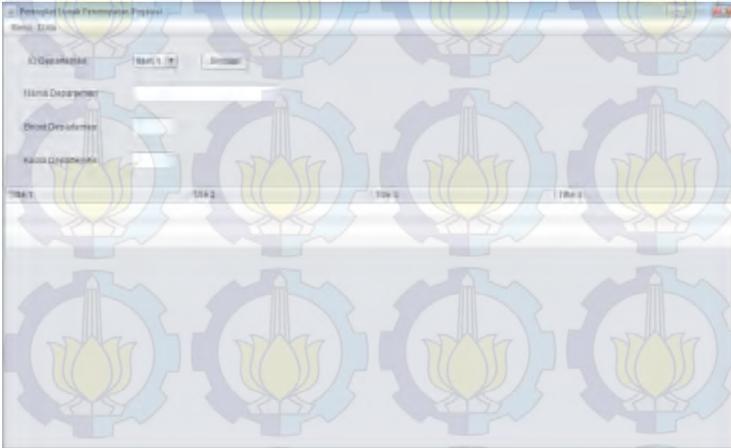
**Gambar 5.13.** Tampilan Halaman Utama untuk Admin

Terdapat menu Data dengan submenu View Data dan Edit Data. Isi dari submenu View Data sama dengan tampilan untuk *user*. Sedangkan submenu Edit Data terdapat tiga pilihan. Yaitu Pegawai, Departemen, dan Kualifikasi.

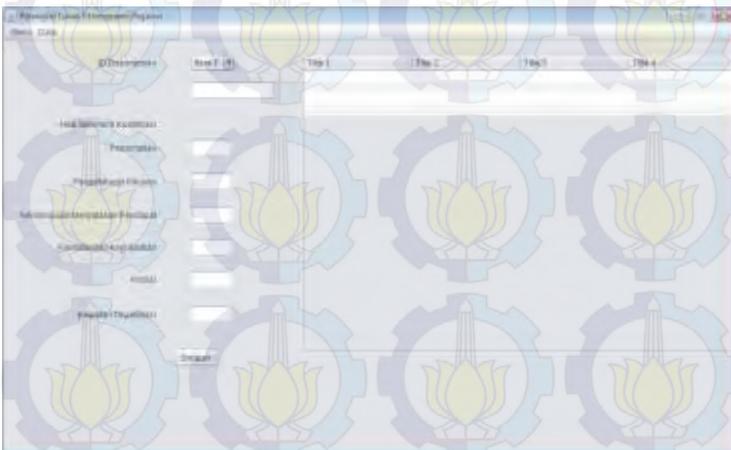


**Gambar 5.14.** Submenu Edit Data

Submenu Edit Data dengan pilihan Pegawai sama dengan tampilan untuk *user*. Sedangkan untuk Departemen dan Kualifikasi, tampilan antarmuka pengguna dapat dilihat pada Gambar 5.15 dan 5.16 berikut ini.



**Gambar 5.15.** Tampilan Edit Data Departemen



**Gambar 5.16.** Tampilan Edit Kualifikasi

## 5.2 Uji Coba Program Penempatan Pegawai

Uji coba pada subbab ini adalah uji coba parameter AG yang meliputi Probabilitas Crossover, Probabilitas Mutasi, dan Ukuran Populasi untuk mengetahui kombinasi parameter yang tepat dalam menghasilkan formasi penempatan pegawai yang stabil. Selain parameter yang telah disebutkan terdapat juga parameter untuk menentukan hasil AG yang optimal, yaitu nilai fitness. Pada umumnya nilai fitness yang tertinggi untuk permasalahan optimalisasi maksimum adalah yang terbaik. Pada permasalahan ini nilai fitness pada formasi penempatan pegawai merupakan acuan untuk setiap hasil yang keluar. Perbedaan nilai fitness pada setiap uji coba, mengakibatkan perbedaan pula formasi penempatan pegawai yang dihasilkan. Sehingga uji coba yang akan dilakukan adalah untuk mendapatkan parameter AG yang menghasilkan nilai fitness yang stabil.

Parameter AG yang pertama adalah Probabilitas Crossover. Hasil penelitian yang sudah pernah dilakukan oleh praktisi AG menunjukkan bahwa angka probabilitas crossover sebaiknya cukup tinggi, yaitu antara 80% sampai 95% untuk memberikan hasil yang baik. Namun untuk beberapa masalah tertentu, probabilitas crossover 60% memberikan hasil yang lebih baik [10]. Untuk itu pada uji coba ini, akan diambil probabilitas crossover sebesar 60% dan nilai pertengahan antara 80% dan 95% yaitu 88%.

Parameter AG yang kedua adalah Probabilitas Mutasi. Probabilitas mutasi dalam AG diberi nilai yang kecil. Umumnya probabilitas mutasi diset untuk mendapatkan rata-rata satu mutasi per kromosom yaitu  $1/(\text{panjang kromosom})$ . Pada permasalahan ini panjang kromosom adalah 121(perkalian antara jumlah pegawai dengan jumlah departemen). Maka probabilitas mutasi dapat diambil angka 0,0082 atau 0,82%. Kemudian hasil penelitian yang telah dipublikasikan menunjukkan bahwa angka probabilitas terbaik adalah antara 0,5% sampai 1% [10]. Maka pada uji coba ini akan diambil dua angka pertengahan probabilitas mutasi yang direkomendasikan yaitu 0,7%, dan 0,8%.

Parameter AG ketiga yang juga merupakan penentu efisiensi kinerja AG adalah ukuran populasi. Ukuran populasi yang sering digunakan oleh peneliti adalah 20 sampai 30. Namun terkadang ukuran populasi sebesar 50 sampai 100 dilaporkan baik [10]. Oleh karena itu pada uji coba ini ukuran populasi yang diambil adalah 30, 70, dan 80.

Tabel-tabel dibawah ini menunjukkan hasil uji coba AG pada program penempatan pegawai. PC adalah Probabilitas Crossover, PM adalah Probabilitas Mutasi, UP adalah Ukuran Populasi, dan WK adalah Waktu Komputasi dalam satuan detik. Iterasi diambil sebanyak 250 kali untuk semua uji coba. Sedangkan percobaan dilakukan 5 kali setiap kombinasi parameter.

**Tabel 5.1.** Hasil Uji Coba dengan Parameter PC=60%, PM=0,7% dan UP=30

Percobaan ke-	PC	PM	UP	WK	Fitness
1	60%	0,7%	30	16,42	14,153
2	60%	0,7%	30	20,95	14,281
3	60%	0,7%	30	19,91	14,445
4	60%	0,7%	30	29,22	14,765
5	60%	0,7%	30	17,27	14,871

**Tabel 5.2.** Hasil Uji Coba dengan Parameter PC=60%, PM=0,7% dan UP=70

Percobaan ke-	PC	PM	UP	WK	Fitness
1	60%	0,7%	70	67,72	14,632
2	60%	0,7%	70	76,19	14,281
3	60%	0,7%	70	60,02	14,281
4	60%	0,7%	70	43,59	14,575
5	60%	0,7%	70	57,68	14,643

**Tabel 5.3.** Hasil Uji Coba dengan Parameter PC=60%, PM=0,7% dan UP=80

Percobaan ke-	PC	PM	UP	WK	Fitness
1	60%	0,7%	80	51,99	14,281
2	60%	0,7%	80	67,52	14,517
3	60%	0,7%	80	69,71	14,281
4	60%	0,7%	80	53,29	14,281
5	60%	0,7%	80	48,27	14,281

**Tabel 5.4.** Hasil Uji Coba dengan Parameter PC=60%, PM=0,8% dan UP=30

Percobaan ke-	PC	PM	UP	WK	Fitness
1	60%	0,8%	30	17,08	14,611
2	60%	0,8%	30	23,84	13,986
3	60%	0,8%	30	13,17	14,816
4	60%	0,8%	30	29,36	14,214
5	60%	0,8%	30	24,52	14,826

**Tabel 5.5.** Hasil Uji Coba dengan Parameter PC=60%, PM=0,8% dan UP=70

Percobaan ke-	PC	PM	UP	WK	Fitness
1	60%	0,8%	70	55,16	14,575
2	60%	0,8%	70	70,63	14,281
3	60%	0,8%	70	39,86	14,593
4	60%	0,8%	70	88,29	14,956
5	60%	0,8%	70	70,42	14,281

**Tabel 5.6.** Hasil Uji Coba dengan Parameter PC=60%, PM=0,8% dan UP=80

Percobaan ke-	PC	PM	UP	WK	Fitness
1	60%	0,8%	80	58,46	14,624
2	60%	0,8%	80	86,58	14,643
3	60%	0,8%	80	42,5	15,293
4	60%	0,8%	80	78,21	14,281
5	60%	0,8%	80	69,54	14,27

**Tabel 5.7.** Hasil Uji Coba dengan Parameter PC=88%, PM=0,7% dan UP=30

Percobaan ke-	PC	PM	UP	WK	Fitness
1	88%	0,7%	30	28,48	14,643
2	88%	0,7%	30	16,86	14,956
3	88%	0,7%	30	15,28	13,46
4	88%	0,7%	30	26,41	14,643
5	88%	0,7%	30	25,1	14,575

**Tabel 5.8.** Hasil Uji Coba dengan Parameter PC=88%, PM=0,7% dan UP=70

Percobaan ke-	PC	PM	UP	WK	Fitness
1	88%	0,7%	70	45,5	14,281
2	88%	0,7%	70	41,64	14,281
3	88%	0,7%	70	40,68	14,624
4	88%	0,7%	70	44,7	14,593
5	88%	0,7%	70	47,13	14,281

**Tabel 5.9.** Hasil Uji Coba dengan Parameter PC=88%, PM=0,7% dan UP=80

Percobaan ke-	PC	PM	UP	WK	Fitness
1	88%	0,7%	80	69,61	14,27
2	88%	0,7%	80	88,75	14,281
3	88%	0,7%	80	73,02	14,281
4	88%	0,7%	80	64,85	14,281
5	88%	0,7%	80	59,16	14,281

**Tabel 5.10.** Hasil Uji Coba dengan Parameter PC=88%, PM=0,8% dan UP=30

Percobaan ke-	PC	PM	UP	WK	Fitness
1	88%	0,8%	30	28,32	13,81
2	88%	0,8%	30	27,15	14,509
3	88%	0,8%	30	16,73	15,606
4	88%	0,8%	30	14,77	14,281
5	88%	0,8%	30	18,71	12,956

**Tabel 5.11.** Hasil Uji Coba dengan Parameter PC=88%, PM=0,8% dan UP=70

Percobaan ke-	PC	PM	UP	WK	Fitness
1	88%	0,8%	70	63,25	14,643
2	88%	0,8%	70	43,84	14,281
3	88%	0,8%	70	65,19	14,281
4	88%	0,8%	70	38,67	14,281
5	88%	0,8%	70	53,89	14,281

**Tabel 5.12.** Hasil Uji Coba dengan Parameter PC=88%, PM=0,8% dan UP=80

Percobaan ke-	PC	PM	UP	WK	Fitness
1	88%	0,8%	80	57,76	14,281
2	88%	0,8%	80	88,2	14,281
3	88%	0,8%	80	67,28	14,575
4	88%	0,8%	80	72,67	14,281
5	88%	0,8%	80	53,78	14,27

### 5.3 Analisis Hasil Uji Coba Parameter

Data hasil uji coba program penempatan pegawai diatas membutuhkan analisis agar diketahui kombinasi parameter terbaik. Data uji coba diatas menunjukkan bahwa terdapat nilai fitness tertinggi yaitu sebesar 15,606 pada kombinasi parameter Tabel 5.10 dan nilai fitness terendah yaitu sebesar 12,956 pada tabel yang sama. Sedangkan hasil fitness yang lain berkisar pada angka 14. Namun hasil fitness yang keluar paling banyak adalah fitness sebesar 14,281 yang keluar sebanyak 21 kali dengan rincian sebagai berikut:

**Tabel 5.13.** Kemunculan Fitness Dominan

Nama Tabel	Kemunculan
Tabel 5.1	1 kali
Tabel 5.2	2 kali
Tabel 5.3	4 kali
Tabel 5.4	Tidak muncul
Tabel 5.5	2 kali
Tabel 5.6	1 kali
Tabel 5.7	Tidak muncul
Tabel 5.8	3 kali
Tabel 5.9	4 kali
Tabel 5.10	1 kali
Tabel 5.11	4 kali
Tabel 5.12	3 kali
<b>Total</b>	<b>21 Kali</b>

Dari rincian tabel diatas diketahui bahwa kemunculan fitness dominan terbanyak berada pada tabel 5.3, tabel 5.9 dan tabel 5.11. Maka diambil kombinaasi parameter pada ketiga tabel untuk diuji kestabilan nilai fitnessnya. Diambil 10 kali uji coba dengan iterasi yang sama. Tabel 5.14 dibawah ini menunjukkan hasil pengujian kestabilan nilai fitness pada ketiga kombinasi parameter.

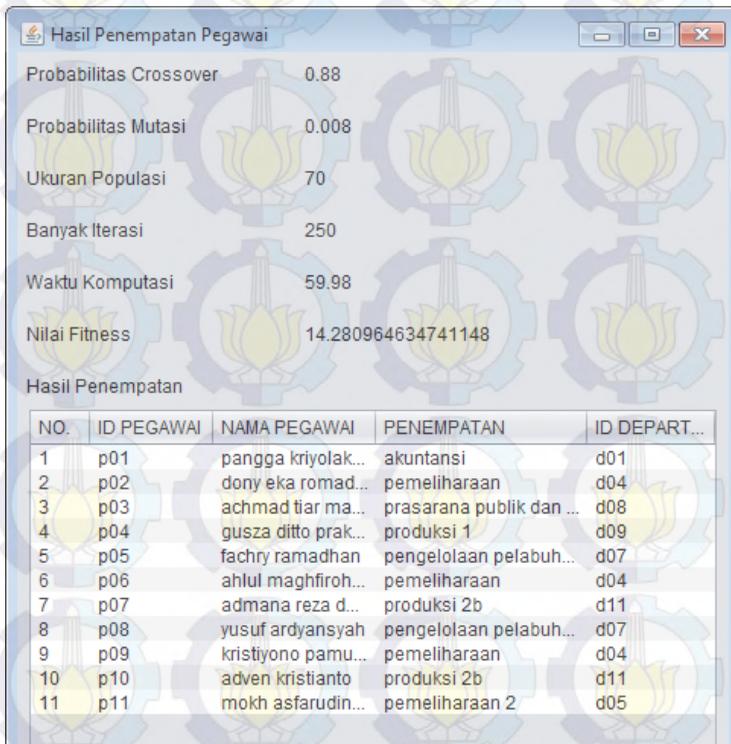
**Tabel 5.14.** Hasil uji Kestabilan Nilai Fitness

	PC	PM	UP	PC	PM	UP	PC	PM	UP			
	60%	0,7%	80	88%	0,7%	80	88%	0,8%	70			
	WK		Fitness		WK		Fitness		WK		Fitness	
1	40,42	14,624	46,38	14,281	64,23	14,281						
2	60,59	14,643	57,56	14,281	58,06	14,281						
3	93,68	14,281	76,36	14,575	53,81	14,281						
4	58,05	14,593	66,7	14,281	53,71	14,281						
5	48,73	14,27	65,48	14,613	71,14	14,281						
6	87,63	14,281	85,1	14,281	56,5	14,281						
7	83,91	14,575	49,16	14,281	62,19	14,281						
8	69,54	14,281	71,03	14,281	64,13	14,281						
9	47,22	14,27	63,2	14,445	45,21	14,281						
10	63,36	14,36	54,92	14,281	41,42	14,624						

Dapat dilihat bahwa dari 10 kali uji coba kestabilan yang dilakukan, kombinasi parameter AG yang pertama menghasilkan fitness dominan sebanyak 3 kali, kombinasi parameter AG yang kedua menghasilkan fitness dominan sebanyak 7 kali, sedangkan kombinasi parameter AG yang ketiga menghasilkan fitness dominan sebanyak 9 kali. Sehingga berdasarkan data hasil uji kestabilan diatas, dapat diambil kesimpulan bahwa nilai parameter AG dengan PC=88%, PM=0,8%, UP=70 adalah yang paling stabil. Sehingga parameter ini dapat digunakan dalam proses AG selanjutnya.

### 5.4 Validasi Hasil Penempatan Pegawai

Pada pembahasan uji coba parameter AG diatas, diketahui bahwa nilai fitness yang paling dominan adalah sebesar 14,281 yang dihasilkan oleh formasi penempatan pegawai seperti gambar berikut:



The screenshot shows a window titled "Hasil Penempatan Pegawai" with the following parameters:

- Probabilitas Crossover: 0.88
- Probabilitas Mutasi: 0.008
- Ukuran Populasi: 70
- Banyak Iterasi: 250
- Waktu Komputasi: 59.98
- Nilai Fitness: 14.280964634741148

Below the parameters is a table titled "Hasil Penempatan" with the following data:

NO.	ID PEGAWAI	NAMA PEGAWAI	PENEMPATAN	ID DEPART...
1	p01	pangga kriyolak...	akuntansi	d01
2	p02	dony eka romad...	pemeliharaan	d04
3	p03	achmad tiar ma...	prasarana publik dan ...	d08
4	p04	gusza ditto prak...	produksi 1	d09
5	p05	fachry ramadhan	pengelolaan pelabuh...	d07
6	p06	ahlul maghfiroh...	pemeliharaan	d04
7	p07	admana reza d...	produksi 2b	d11
8	p08	yusuf ardyansyah	pengelolaan pelabuh...	d07
9	p09	kristiyono pamu...	pemeliharaan	d04
10	p10	adven kristianto	produksi 2b	d11
11	p11	mokh asfarudin...	pemeliharaan 2	d05

**Gambar 5.17.** Hasil penempatan pegawai dengan fitness 14,281

Validasi yang dilakukan adalah dengan membandingkan hasil penempatan pegawai oleh sistem dengan hasil penempatan oleh perusahaan. Apabila terjadi ketidaksamaan dalam penempatan yang dilakukan, maka penulis melanjutkan validasi dengan mendapatkan respon dari pegawai yang bersangkutan.

Jika pegawai yang bersangkutan ditempatkan pada departemen yang berbeda dari yang ditempatkan oleh perusahaan, bagaimana tanggapan atau responnya dalam menanggapi hal ini. Penempatan pegawai yang dilakukan oleh sistem dikatakan valid jika sesuai dengan penempatan oleh perusahaan dan hasil positif dari respon pegawai. Tabel berikut adalah tabel yang menunjukkan penempatan pegawai oleh sistem dan perusahaan.

**Tabel. 5.17.** Penempatan Pegawai oleh Sistem dan Perusahaan

<b>ID Pegawai</b>	<b>Penempatan oleh Sistem</b>	<b>Penempatan oleh Perusahaan</b>
P01	Akuntansi	Anggaran
P02	Pemeliharaan I	Pemeliharaan II
P03	PPK	Pemeliharaan III
P04	Produksi I	Pemeliharaan I
P05	Pengelolaan Pelabuhan	Pemeliharaan II
P06	Pemeliharaan I	Pemeliharaan II
P07	Produksi II B	Pemeliharaan III
P08	Pengelolaan Pelabuhan	Pengelolaan Pelabuhan
P09	Pemeliharaan I	Produksi I
P10	Produksi II B	Produksi II B
P11	Pemeliharaan II	Pemeliharaan II

Dapat dilihat pada tabel diatas, bahwa terdapat 3 pegawai yang ditempatkan pada departemen yang sama oleh sistem maupun perusahaan, yaitu pegawai dengan ID pegawai P08, P10, dan P11. Hasil validasi untuk pegawai yang lain dapat ditunjukkan di Lampiran B.

## BAB VI PENUTUP

Pada Bab ini diberikan kesimpulan dari pembahasan pada bab sebelumnya yang telah dilakukan. Selain itu, diberikan pula saran yang dapat dilakukan sebagai pengembangan Tugas Akhir ini.

### 6.1 Kesimpulan

Dari pembahasan pada bab yang sebelumnya maka dapat diambil beberapa kesimpulan, diantaranya adalah pendekatan AG sebagai salah satu metode dalam optimalisasi dapat diimplementasikan dalam permasalahan pada Tugas Akhir ini. Pada uji coba yang telah dilakukan, didapatkan nilai fitness yang dominan yang menjadi acuan kestabilan dalam uji coba parameter AG yaitu sebesar 14,281. Sedangkan pada uji parameter AG itu sendiri didapatkan pula nilai yang stabil pada 9 kali percobaan dari 10 kali percobaan. Yaitu 88% untuk Probabilitas Crossover, 0,8% untuk Probabilitas Mutasi dan 70 untuk Ukuran Populasi. Dari nilai fitness dominan yang telah didapatkan, diketahui pula hasil formasi penempatan pegawai oleh sistem yang telah dibuat dengan validitas yang dibahas pada subbab 5.4. Proses prototyping yang dibahas pada subbab 4.3 ditunjukkan sebagai salah satu metode dalam perancangan perangkat lunak dilakukan sebanyak 4 kali pertemuan dengan *user* untuk mendapatkan *user-interface*, spesifikasi dan fungsi yang diinginkan.

### 6.2 Saran

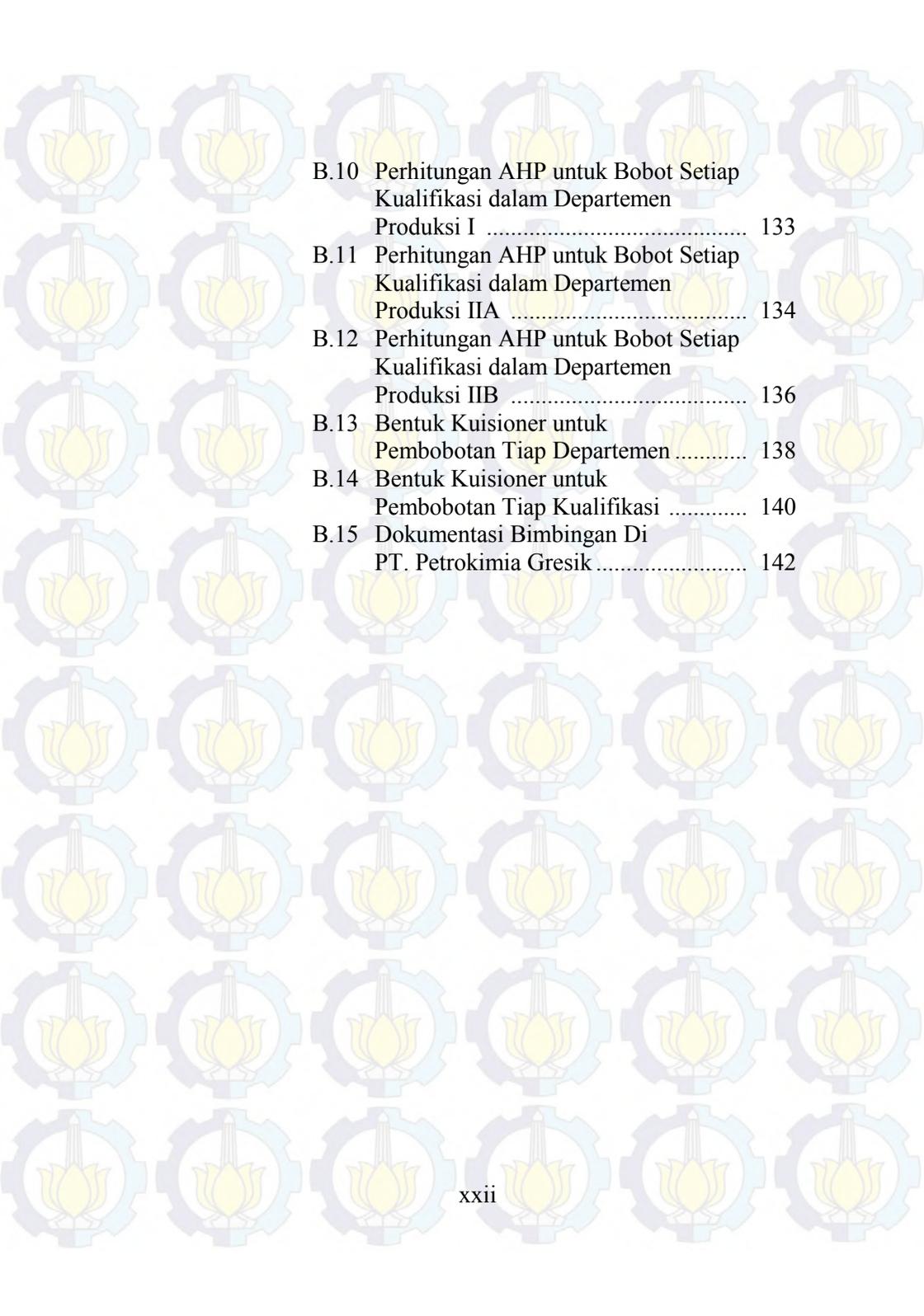
Saran yang diajukan dari Tugas Akhir ini untuk penelitian selanjutnya adalah sebagai berikut :

1. Hasil penelitian ini dapat dilanjutkan dengan pembahasan transisi dari Prototipe Perangkat Lunak kedalam Perangkat Lunak yang dirancang sesuai dengan kebutuhan pengguna.
2. Jumlah departemen, jumlah kualifikasi, bobot kriteria, bobot alternatif, dan jumlah kuota pegawai dalam setiap departemen dibuat lebih dinamis sebagai sistem informasi.

3. Model permasalahan yang dikasji dalam kasus penempatan pegawai dapat berbeda sesuai bentuk permasalahan sehingga Perangkat Lunak yang akan dibuat dapat dipakai dalam studi kasus yang lain.

## DAFTAR LAMPIRAN

	Halaman
LAMPIRAN A .....	95
A.1 <i>Source Code</i> Tahap <i>Pre-Process</i> .....	95
A.2 <i>Source Code</i> Tahap <i>GA-Process</i> .....	102
A.3 <i>Source Code</i> Tahap <i>Post-Process</i> .....	107
LAMPIRAN B .....	119
B.1 Perhitungan AHP untuk Bobot Setiap Departemen .....	119
B.2 Perhitungan AHP untuk Bobot Setiap Kualifikasi dalam Departemen Akuntansi .....	120
B.3 Perhitungan AHP untuk Bobot Setiap Kualifikasi dalam Departemen Anggaran .....	122
B.4 Perhitungan AHP untuk Bobot Setiap Kualifikasi dalam Departemen Keuangan .....	123
B.5 Perhitungan AHP untuk Bobot Setiap Kualifikasi dalam Departemen Pemeliharaan I .....	125
B.6 Perhitungan AHP untuk Bobot Setiap Kualifikasi dalam Departemen Pemeliharaan II .....	126
B.7 Perhitungan AHP untuk Bobot Setiap Kualifikasi dalam Departemen Pemeliharaan III .....	128
B.8 Perhitungan AHP untuk Bobot Setiap Kualifikasi dalam Departemen Pengelolaan Pelabuhan .....	130
B.9 Perhitungan AHP untuk Bobot Setiap Kualifikasi dalam Departemen PPK ...	131



B.10	Perhitungan AHP untuk Bobot Setiap Kualifikasi dalam Departemen Produksi I .....	133
B.11	Perhitungan AHP untuk Bobot Setiap Kualifikasi dalam Departemen Produksi IIA .....	134
B.12	Perhitungan AHP untuk Bobot Setiap Kualifikasi dalam Departemen Produksi IIB .....	136
B.13	Bentuk Kuisisioner untuk Pembobotan Tiap Departemen .....	138
B.14	Bentuk Kuisisioner untuk Pembobotan Tiap Kualifikasi .....	140
B.15	Dokumentasi Bimbingan Di PT. Petrokimia Gresik .....	142

## LAMPIRAN A

### A.1 Source Code Tahap Pre-Process

```
public void preProsesGA() {
    status = "Reading Data...";
    jProgressBar1.setString(status);
    try {
        connect.Connect();
        con = connect.getConnection();
        stm = con.createStatement();

        ResultSet jum =
            stm.executeQuery("select count(*) from
            t_pegawai");
        jum.next();
        jumlahPeg = jum.getInt(1);

        jum = stm.executeQuery("select count(*)
        from t_departemen");
        jum.next();
        jumlahDep = jum.getInt(1);

        total = jumlahPeg * jumlahDep;
        ResultSet peg =
            stm.executeQuery("select id_peg from
            t_pegawai order by id_peg asc");
        ArrayList listPeg = new
        ArrayList();//daftar pegawai

        oke = new int[jumlahPeg][jumlahDep+1];
        int i = 0, j = 0;
        while (peg.next()) {
            listPeg.add(peg.getString(1));
        }
        Iterator iterator = listPeg.iterator();
        while (iterator.hasNext()) {
            String id = iterator.next().toString();
```

## LAMPIRAN A (LANJUTAN)

```

ResultSet pilPeg =
stm.executeQuery("SELECT
id_dept,pilihan_peg FROM
`t_dept_peg` WHERE id_peg='" + id +
"' order by id_dept asc");
j = 0;
int temp = 0, tempx = 0;
while (pilPeg.next()) {
temp++;
if (pilPeg.getInt(2) == 1) {
oke[i][j++] = temp;
tempx++;
}
}
oke[i][jumlahDep] = tempx;
i++;
}

```

```

//membangkitkan kromosom
Random random = new Random();
int A[][];
kromosom = new double[(int) param[0]][(int)
total + 3];
for (int pop = 0; pop < param[0]; pop++) {
A = new int[jumlahPeg][jumlahDep];
for (int staff = 0; staff < jumlahPeg;
staff++) {
int pos =
random.nextInt(oke[staff][jumlahDe
p]);
A[staff][oke[staff][pos] - 1] = 1;
}
}
//jadikan matriks baris
int temp = 0;
for (int x = 0; x < jumlahPeg; x++) {

```

## LAMPIRAN A (LANJUTAN)

```
        for (int y = 0; y < jumlahDep;
            y++)
        {
            kromosom[pop][temp++] =
                A[x][y];
        }
    }
}

//ambil nilai dept_kual
ResultSet kual = stm.executeQuery("select
id_kual from t_kualifikasi order by id_kual
asc");
ArrayList listKual = new ArrayList();
//daftar kualifikasi
while (kual.next()) {
    listKual.add(kual.getString(1));
}
alpha11 = new
double[listKual.size()][jumlahDep];
int temp = 0, tempx = 0;
int jumlahKual = listKual.size();

iterator = listKual.iterator();
while (iterator.hasNext()) {
    String id =
        iterator.next().toString();
    ResultSet deptKual =
        stm.executeQuery("SELECT
bobot_dept_kual FROM t_dept_kual WHERE
id_kual='" + id + "' order by id_dept
asc,id_kual asc");
    while (deptKual.next()) {
        if (tempx == jumlahDep) {
            tempx = 0;
            temp++;
        }
    }
}
```

## LAMPIRAN A (LANJUTAN)

```
        if (tempx != jumlahDep) {
            alphail[temp][tempx++] =
                Double.valueOf(deptKual.getString(
                    1));
        }
    }

    //ambil nilai peg_kual
    dijl = new double[jumlahPeg][jumlahKual];
    iterator = listPeg.iterator();
    temp = 0;
    tempx = 0;

    while (iterator.hasNext()) {
        String id =
            iterator.next().toString();
        ResultSet pegKual = stm.executeQuery
            ("SELECT nilai_a FROM t_kual_peg where
            id_peg='" + id + "' order by id_kual
            asc");
        while (pegKual.next()) {
            if (tempx == jumlahKual) {
                tempx = 0;
                temp++;
            }
            if (tempx != jumlahKual) {
                dijl[temp][tempx++] =
                    Double.valueOf(pegKual.getStr
                    ing(1));
            }
        }
    }

    //ambil nilai peg_kual
    bijr = new double[jumlahPeg][jumlahKual];
    iterator = listPeg.iterator();
```

## LAMPIRAN A (LANJUTAN)

```
temp = 0;
tempx = 0;

while (iterator.hasNext()) {
    String id =
        iterator.next().toString();
    ResultSet pegKual =
        stm.executeQuery("SELECT nilai_b FROM
t_kual_peg where id_peg='" + id + "'
order by id_kual asc");
    while (pegKual.next()) {
        if (tempx == jumlahKual) {
            tempx = 0;
            temp++;
        }
        if (tempx != jumlahKual) {
            bijr[temp][tempx++] =
                Double.valueOf(pegKual.getSt
                    ring(1));
        }
    }
}

//ambil nilai dept_kual
dilnol = new double[jumlahKual][jumlahDep];
iterator = listKual.iterator();
temp = 0;
tempx = 0;
while (iterator.hasNext()) {
    String id =
        iterator.next().toString();
    ResultSet deptKual =
        stm.executeQuery("SELECT nilai_min
FROM t_dept_kual WHERE id_kual='" + id
+ "' order by id_dept asc,id_kual
asc");
    while (deptKual.next()) {
```

## LAMPIRAN A (LANJUTAN)

```
        if (tempx == jumlahDep) {
            tempx = 0;
            tempx++;
        }
        if (tempx != jumlahDep) {
            dilnol[temp][tempx++] =
                Double.valueOf(deptKual.getSt
                    ring(1));
        }
    }

//ambil nilai dept_kual
e = new double[jumlahDep];
ResultSet botDept =
    stm.executeQuery("SELECT bobot_dept FROM
    t_departemen order by id_dept asc");
temp = 0;
while (botDept.next()) {
    e[temp++] =
        Double.valueOf(botDept.getString(1));
}

//hitung p
double tempz = 0;
p = new double[jumlahDep][jumlahPeg];
for (i = 0; i < jumlahDep; i++) {
    for (j = 0; j < jumlahPeg; j++) {
        tempz = 0;
        for (int n = 0; n < jumlahDep;
            n++) {
            for (int t = 0; t <
                jumlahKual; t++) {
                tempz = tempz +
                    ((alpha11[t][n] *
                    (dij1[j][t] -
```

## LAMPIRAN A (LANJUTAN)

```
        dilnol[t][n])) /
        dilnol[t][n]);
    }
}
p[i][j] = tempz;
}
}

q = new double[jumlahDep][jumlahPeg];
for (i = 0; i < jumlahDep; i++) {
    for (j = 0; j < jumlahPeg; j++) {
        tempz = 0;
        for (int n = 0; n < jumlahDep;
            n++) {
            for (int t = 0; t <
                jumlahKual; t++) {
                tempz = tempz +
                    ((alpha1l[t][n] *
                    (bijr[j][t] -
                    dilnol[t][n])) /
                    dilnol[t][n]);
            }
        }
        q[i][j] = tempz;
    }
}
}
}
catch (SQLException ex) {
    Logger.getLogger(FMain.class.getName()).log
    (Level.SEVERE, null, ex);
}
}
```

## LAMPIRAN A (LANJUTAN)

### A.2 Source Code Tahap GA-Process

```
public void prosesGA() {
    //mulai GA
    timer=0;
    timex.start();
    status = "Running GA...";
    statsKon="ko";
    for (int iterasi = 0; iterasi < param[1];
    iterasi++) { //param[1]
        if(stats.equals("ok")&&statsKon.equals("ko"
        )){
            proses = ((iterasi + 1) / param[1]) *
            100;
            jProgressBar1.setValue((int)
            Math.floor(proses));
            jProgressBar1.setString(status + " " +
            (int) Math.floor(proses) + "%");
            double totalfit = 0;
            //hitung seluruh fitness kromosom
            for (int populasi = 0; populasi <
            param[0]; populasi++) {
                kromosom[(int) populasi][(int)
                total] =
                fitness(kromosom[populasi]);
                totalfit = totalfit +
                kromosom[populasi][total];
            }
            //hitung proporsi kromosoom
            double proporsikum = 0;
            for (int populasi = 0; populasi <
            param[0]; populasi++) {
                double proporsi =
                kromosom[populasi][total] /
                totalfit;
                kromosom[populasi][total + 1] =
                proporsi;
            }
        }
    }
}
```

## LAMPIRAN A (LANJUTAN)

```
        proporsikum += proporsi;
        kromosom[populasi][total + 2] =
        proporsikum;
    }
    System.out.println("Kromosom:");
    for (int i = 0; i < kromosom.length;
    i++) {
        System.out.println("Kromosom ke" +
        i);
        for (int j = 0; j < total + 3;
        j++) {
            System.out.print(kromosom[i][
            j] + " ");
        }
        System.out.println();
    }

    double kromosomx[][] = new double[2 *
    (int) param[0]][total + 3]; //tampung
    kromosom dan hasil crossover
    for (int i = 0; i < kromosom.length;
    i++) {
        kromosomx[i] =
        kromosom[i].clone();
    }
    tempPopulasi = kromosom.clone();
    nextPopulasi = tempPopulasi.clone();
    for (int j = 0; j < kromosom.length; j
    += 2) {
        int IP1 = rouletteWheel(kromosom);
        int IP2 = rouletteWheel(kromosom);
        System.out.println(IP1 + " " +
        IP2);
        double RN = Math.random();
        nextPopulasi[j] =
        kromosom[IP1].clone();
```

## LAMPIRAN A (LANJUTAN)

```
nextPopulasi[j + 1] =
kromosom[IP2].clone();
if (RN < param[2]) {
    crossOver(nextPopulasi[j],
nextPopulasi[j + 1], j, j +
1);
}
}
System.out.println("sebelum mutasi:");
for (int i = 0; i <
nextPopulasi.length; i++) {
    for (int j = 0; j < total + 3;
j++) {
        System.out.print(nextPopulasi
[i][j] + " ");
    }
    System.out.println();
}
//mutasi
Random rand = new Random();
for (int a = 0; a < oke.length; a++) {
    for (int b = 0; b < oke[0].length;
b++) {
        System.out.print(oke[a][b] +
" ");
    }
    System.out.println();
}
for (int i = 0; i <
nextPopulasi.length; i++) {
    double RN = Math.random();
    if (RN < param[3]) {
        int staffMutasi =
rand.nextInt(total);
```

## LAMPIRAN A (LANJUTAN)

```
while (nextPopulasi
[i][staffMutasi] != 1) {
    staffMutasi =
    rand.nextInt(total);
}
int cekPosisi = (int)
Math.floor(staffMutasi /
(jumlahDep * 1.0));
int posisiMutasi =
rand.nextInt(oke[cekPosisi][j
umlahDep]);
nextPopulasi[i][staffMutasi]
= 0;
int index_awal =
oke[cekPosisi][posisiMutasi];
System.out.println("staff ke-
" + (cekPosisi + 1) + "
pilihan-" + index_awal);
nextPopulasi[i][(cekPosisi *
jumlahDep) + (index_awal -
1)] = 1;
}
}
System.out.println("setelah mutasi:");
for (int i = 0; i <
nextPopulasi.length; i++) {
    for (int j = 0; j < total + 3;
j++) {
        System.out.print(nextPopulasi
[i][j] + " ");
    }
    System.out.println();
}
for (int i = kromosom.length; i < 2 *
kromosom.length; i++) {
    kromosomx[i] = nextPopulasi[i %
kromosom.length].clone();
}
```

## LAMPIRAN A (LANJUTAN)

```
}
System.out.println("sebelum sort:");
for (int i = 0; i < kromosomx.length;
i++) {
    for (int j = 0; j < total + 3;
j++) {
        System.out.print(kromosomx[i]
[j] + " ");
    }
    System.out.println();
}
Arrays.sort(kromosomx, new
Comparator<double[]>() {
    @Override
    public int compare(double[] t,
double[] t1) {
        return
        Double.compare(t1[total],
t[total]);
    }
});
System.out.println("setelah sort:");
for (int i = 0; i < kromosomx.length;
i++) {
    for (int j = 0; j < total + 3;
j++) {
        System.out.print(kromosomx[i]
[j] + " ");
    }
    System.out.println();
}
//ambil populasi untuk iterasi
berikutnya
for (int i = 0; i < param[0]; i++) {
    kromosom[i] =
    kromosomx[i].clone();
}
```

## LAMPIRAN A (LANJUTAN)

```

    }
    else{
        break;
    }
    //cek konvergen di fitness
    int mandeg=(int) (0.9*param[0]); //80% dari
    jumlah populasi
    double fit=kromosom[0][total]; //ambil
    fitness tertinggi
    int counter=0;
    for(int i=1;i<kromosom.length;i++){
        if(kromosom[i][total]==fit){
            counter++;
        }
        if(counter==mandeg){
            statsKon="ok";
            System.out.println("konvergen
            90%");
            break;
        }
    }
}
}

```

### A.3 Source Code Tahap Post-Process

```

public void postGA() {
    if(stats.equals("ok")){
        status = "Post Process...";
        jProgressBar1.setString(status);
        double[] hasil = kromosom[0]; //ambil hasil GA
        dengan fitness teratas
        hasilx = new double[jumlahPeg][jumlahDep];
        System.out.println("Post GA:");
        hasil[total] = fitness(kromosom[0]);
        System.out.println("Fitness awal = " +
        (hasil[total]));
    }
}

```

## LAMPIRAN A (LANJUTAN)

```
hasil[total + 1] = 0;
hasil[total + 2] = 0;
int counter = 0;
int counterx = 0;
for (int i = 0; i < total; i++) { //kembalikan
ke dalam bentuk matriks jumlahPeg x jumlahDep
    if (counter == 11) {
        counter = 0;
        counterx++;
    }
    hasilx[counterx][counter++] = hasil[i];
}
System.out.println();
for (int i = 0; i < jumlahPeg; i++) {
    for (int j = 0; j < jumlahDep; j++) {
        System.out.print(hasilx[i][j] + " ");
    }
    System.out.println();
}
System.out.println();
//menghitung nilai akhir
try {
    connect.Connect();
    con = connect.getConnection();
    stm = con.createStatement();
    ResultSet peg = stm.executeQuery("select
id_peg from t_pegawai order by id_peg
asc");
    ArrayList listPeg = new ArrayList();
    //daftar pegawai
    while (peg.next()) {
        listPeg.add(peg.getString(1));
    }
    Iterator iterator = listPeg.iterator();
    double avgKualPeg[] = new double
[listPeg.size()];
    counter = 0;
```

**LAMPIRAN A (LANJUTAN)**

```
while (iterator.hasNext()) {
    String id = iterator.next().toString();
    ResultSet avgPeg =
    stm.executeQuery("SELECT avg(nilai_a)
    FROM `t_kual_peg` WHERE id_peg='" + id +
    "'");
    if (avgPeg.next()) {
        avgKualPeg[counter++] =
        avgPeg.getDouble(1);
    }
}
ResultSet kual = stm.executeQuery("select
id_kual from t_kualifikasi order by id_kual
asc");
ArrayList listKual = new ArrayList();
//daftar kualifikasi
counter = 0;
double totalmin = 0;
while (kual.next()) {
    listKual.add(kual.getString(1));
}
iterator = listKual.iterator();
while (iterator.hasNext()) {
    String id = iterator.next().toString();
    ResultSet avgKual =
    stm.executeQuery("SELECT avg(nilai_min)
    FROM `t_dept_kual` WHERE id_kual='" + id
    + "'");
    if (avgKual.next()) {
        totalmin += avgKual.getDouble(1);
    }
}
double rata2min = totalmin /
listKual.size();
```

## LAMPIRAN A (LANJUTAN)

```
//nilai akhir
double nilaiakhir[][] = new double
[listPeg.size()][2];
for (int i = 0; i < nilaiakhir.length; i++)
{
    nilaiakhir[i][0] = i;
    nilaiakhir[i][1] = avgKualPeg[i] -
    rata2min;
}
Arrays.sort(nilaiakhir, new Comparator
<double[]>() {
    @Override
    public int compare(double[] t, double[]
    t1) {
        return Double.compare(t[1], t1[1]);
    }
});

//ambil kuota
ResultSet kuot = stm.executeQuery("SELECT
kuota_dept FROM `t_departemen` order by
id_dept");
double kuotaDep[] = new double[jumlahDep];
counter = 0;
while (kuot.next()) {
    kuotaDep[counter++] = kuot.getDouble(1);
}
//ambil nilai min
ResultSet min = stm.executeQuery("SELECT
bobot_dept FROM `t_departemen` order by
id_dept");
double nilaiMinDep[][] = new double
[jumlahDep][2];
counter = 0;
while (min.next()) {
    nilaiMinDep[counter][0] = counter;
```

## LAMPIRAN A (LANJUTAN)

```
        nilaiMinDep[counter][1] =
            min.getDouble(1);
        counter++;
    }
    double nilaiMaxDep[][] =
        nilaiMinDep.clone();
    Arrays.sort(nilaiMinDep, new Comparator
    <double[]>() {
        @Override
        public int compare(double[] t, double[]
        t1) {
            return Double.compare(t1[1],
            t[1]);
        }
    });
    Arrays.sort(nilaiMaxDep, new Comparator
    <double[]>() {
        @Override
        public int compare(double[] t, double[]
        t1) {
            return Double.compare(t[1],
            t1[1]);
        }
    });

    //cek over kuota
    ArrayList listDepOK;
    ArrayList listDepOKi;
    ArrayList listPegPindahFix = null;
    int dep = 0;
    int oke_temp[][]=oke.clone();
    do {
        listDepOK = new ArrayList();
        listDepOKi = new ArrayList();
        //cek jumlah pegawai setiap departemen
        for (int j = 0; j < jumlahDep; j++) {
            counter = 0;
```

## LAMPIRAN A (LANJUTAN)

```
for (int i = 0; i < jumlahPeg; i++)
{
    if (hasilx[i][j] == 1) {
        counter++;
    }
}
System.out.println("jumlah peg pada dep-"
+ (j+1) + " = " + counter + " dengan
kuota = " + kuotaDep[j]);
if (counter > kuotaDep[j]) {
    listDepOK.add(j);
    listDepOKi.add(j+1);
}
}
System.out.println("dep yang over kuota = "
+ listDepOKi);
if (listDepOK.isEmpty()) {
    break;
}

//jika ada departemen yang over kuota
listPegPindahFix = new ArrayList();
for (int i = 0; i < listDepOK.size(); i++)
{
    ArrayList listPegPindah = new
    ArrayList();
    ArrayList listPegPindahi = new
    ArrayList();
    dep = (int) listDepOK.get(i);
    for (int k = 0; k < jumlahPeg; k++) {
        if (hasilx[k][dep] == 1) {
            listPegPindah.add(k);
            listPegPindahi.add(k+1);
        }
    }
}
```

## LAMPIRAN A (LANJUTAN)

```
System.out.println("pegawai yang ada di
dep = " + listPegPindahi);
String selesai = "\n";
for (int a = 0; a < nilaiakhir.length;
a++) {
    for (int b = 0; b <
listPegPindah.size(); b++) {
        ArrayList listPilihanPeg = new
ArrayList();
        ArrayList listPilihanPegi = new
ArrayList();
        //jika pegawai yang akan dipindah
mempunyai nilai akhir yang paling
kecil
        if (nilaiakhir[a][0] == (int)
listPegPindah.get(b)) {
            listPegPindahFix.add(listPegPindah
.get(b));
            System.out.println("pegawai yang
dipindah = "+int)
listPegPindahi.get(b));
            int indexPegawai = (int)
listPegPindah.get(b);
            for (int c = 0; c <
oke_temp.length; c++) {
                for (int d = 0; d <
oke_temp[0].length; d++) {
                    System.out.print(oke_temp[c
][d] + " ");
                }
                System.out.println();
            }
        }
        for (int c = 0; c < jumlahDep;
c++) {
            int depx =
oke_temp[indexPegawai][c];
```

## LAMPIRAN A (LANJUTAN)

```
if (depX - 1 != dep && depX != 0) {
    listPilihanPeg.add(depX - 1);
    listPilihanPegi.add(depX);
}
if (depX-1==dep) {
    for (int f = 0; f < oke_temp[0].length; f++) {
        if (oke_temp[indexPegawai][f] - 1 == dep) {
            oke_temp[indexPegawai][f] = 0; //hapus pilihannya
            break;
        }
    }
}
System.out.println("dipindah ke = " + listPilihanPegi);
String selesaix = "\n";
if (listPilihanPeg.isEmpty()) {
    //pilihan pegawai sudah habis
    //status pegawai dipindah secara paksa
    System.out.println("pegawai " + (int) listPegPindahi.get(b) + " pilihannya sudah habis");
    System.out.println("peg yang dipaksa dipindahkan");
    ArrayList listDepKosong = new ArrayList();
    ArrayList listDepKosongi = new ArrayList();
    for (int c = 0; c < jumlahDep; c++) {
        counter = 0;
```

## LAMPIRAN A (LANJUTAN)

```
for (int d= 0; d <
jumlahPeg; d++) {
    if (hasilx[c][d] == 1) {
        counter++;
    }
}
System.out.println("jumlah
peg pada dep-" + (c+1) + "
= " + counter + " dengan
kuota = " + kuotaDep[c]);
if (counter < kuotaDep[c])
{
    listDepKosong.add(c);
    //list departemen yang
    masih ada kuota
    listDepKosongi.add(c+1);
}
}
System.out.println("dep yang
ada kuota = " +
listDepKosongi);
for (int d = 0; d <
nilaiMaxDep.length; d++) {
    for (int e = 0; e <
listDepKosong.size(); e++)
    {
        if (nilaiMaxDep[d][0] ==
(int)
listDepKosong.get(e)) {
            hasilx[indexPegawai][d
ep] = 0;
            hasilx[indexPegawai][ (
(int)
listDepKosong.get(e) ]
= 1;
            System.out.println("su
dah dipindah ke dep "
```

## LAMPIRAN A (LANJUTAN)

```
+ (int)
listDepKosong.get(e)+1
);
selesaix = "y";
break;
}
}
if (selesaix.equals("y")) {
break;
}
selesai = "y";
}
else{
for (int d = 0; d <
nilaiMinDep.length; d++) {
for (int e = 0; e <
listPilihanPeg.size(); e++)
{
if (nilaiMinDep[d][0] ==
(int)
listPilihanPeg.get(e)) {
hasilx[indexPegawai][d
ep] = 0;
hasilx[indexPegawai][
(int)listPilihanPeg.get
(e)] = 1;
System.out.println("su
dah dipindah ke dep
"+(int)listPilihanPegi
.get(e));
for(int
f=0;f<oke_temp[0].leng
th;f++){
if(oke_temp[indexPeg
awai][f]-
```

## LAMPIRAN A (LANJUTAN)

```
1==(int)listPilihanP
eg.get(e)){
    oke_temp[indexPeg
awai][f]=0;
    //hapus
    pilihannya
    break;
}
}
selesaix = "y";
break;
}
}
if (selesaix.equals("y")) {
    break;
}
}
listPegPindah.remove(b);
if ((listPegPindah.size() -
kuotaDep[dep]) == 0) {
    selesai = "y";
}
break;
}
}
}
if (selesai.equals("y")) {
    break;
}
}
}
if(!listPegPindahFix.isEmpty())
System.out.println("peg yang fix
dipindahkan = " + listPegPindahFix);
}
While
(!listDepOK.isEmpty());
```

## LAMPIRAN A (LANJUTAN)

```
System.out.println();
for (int i = 0; i < jumlahPeg; i++) {
    for (int j = 0; j < jumlahDep; j++) {
        System.out.print(hasilx[i][j]+" ");
    }
    System.out.println();
}
int temp = 0;
hasilakhir = new double[1][total];
for (int x = 0; x < jumlahPeg; x++) {
    for (int y = 0; y < jumlahDep; y++) {
        hasilakhir[0][temp++] = hasilx[x][y];
    }
}
finalFitness = fitness(hasilakhir[0]);
System.out.println("Fitness akhir = " +
    finalFitness);
}
catch (SQLException ex) {
    Logger.getLogger(FMain.class.getName()).log(
        Level.SEVERE, null, ex);
}
finally {
    timex.stop();
    connect.Close();
    this.dispose();
}
}
```

## LAMPIRAN B

## B.1 Perhitungan AHP untuk Bobot Setiap Departemen

Matriks Perbandingan Kuisisioner 1

	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11
D1	1,000	1,000	1,000	0,200	0,200	0,200	0,200	2,000	0,167	0,167	0,167
D2	1,000	1,000	1,000	0,200	0,200	0,200	0,200	3,000	0,167	0,167	0,167
D3	1,000	1,000	1,000	0,250	0,250	0,250	0,250	3,000	0,167	0,167	0,167
D4	5,000	5,000	4,000	1,000	1,000	1,000	1,000	3,000	0,500	0,500	0,500
D5	5,000	5,000	4,000	1,000	1,000	1,000	1,000	3,000	0,500	0,500	0,500
D6	5,000	5,000	4,000	1,000	1,000	1,000	1,000	3,000	0,500	0,500	0,500
D7	0,500	0,333	0,333	0,333	0,333	0,333	0,333	0,500	1,000	0,167	0,167
D8	6,000	6,000	6,000	2,000	2,000	2,000	3,000	6,000	1,000	1,000	1,000
D9	6,000	6,000	6,000	2,000	2,000	2,000	3,000	6,000	1,000	1,000	1,000
D10	6,000	6,000	6,000	2,000	2,000	2,000	3,000	6,000	1,000	1,000	1,000
D11	6,000	6,000	6,000	2,000	2,000	2,000	3,000	6,000	1,000	1,000	1,000

Matriks Perbandingan Kuisisioner 2

	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11
D1	1,000	1,000	1,000	0,125	0,125	0,125	0,125	0,167	0,125	0,125	0,125
D2	1,000	1,000	1,000	0,125	0,125	0,125	0,143	0,167	0,125	0,125	0,125
D3	1,000	1,000	1,000	0,125	0,125	0,125	0,143	0,167	0,125	0,125	0,125
D4	8,000	8,000	8,000	1,000	1,000	1,000	7,000	7,000	1,000	0,143	0,143
D5	8,000	8,000	8,000	1,000	1,000	1,000	7,000	7,000	1,000	0,143	1,000
D6	8,000	8,000	8,000	1,000	1,000	1,000	7,000	7,000	1,000	0,143	0,143
D7	8,000	7,000	7,000	0,143	0,143	0,143	1,000	7,000	0,143	0,143	0,143
D8	6,000	6,000	6,000	0,143	0,143	0,143	0,143	1,000	0,125	0,125	0,125
D9	8,000	8,000	8,000	1,000	7,000	7,000	7,000	8,000	1,000	1,000	1,000
D10	8,000	8,000	8,000	7,000	1,000	7,000	7,000	8,000	1,000	1,000	1,000
D11	8,000	8,000	8,000	7,000	1,000	7,000	7,000	8,000	1,000	1,000	1,000

Matriks Perbandingan Kuisisioner 3

	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11
D1	1,000	1,000	1,000	0,167	0,167	0,167	0,200	6,000	0,167	0,167	0,167
D2	1,000	1,000	1,000	0,200	0,200	0,200	0,333	0,333	0,167	0,167	0,167
D3	1,000	1,000	1,000	6,000	6,000	6,000	4,000	6,000	6,000	6,000	6,000
D4	6,000	5,000	0,167	1,000	1,000	1,000	5,000	6,000	1,000	1,000	1,000
D5	6,000	5,000	0,167	1,000	1,000	1,000	5,000	5,000	1,000	1,000	1,000
D6	6,000	5,000	0,167	1,000	1,000	1,000	5,000	5,000	1,000	1,000	1,000
D7	5,000	3,000	0,250	0,200	1,000	0,200	1,000	4,000	0,167	0,167	0,167
D8	0,167	3,000	0,167	0,167	0,200	0,200	0,250	1,000	0,167	0,200	0,167
D9	6,000	6,000	0,167	1,000	1,000	1,000	6,000	6,000	1,000	0,200	0,200
D10	6,000	6,000	0,167	1,000	1,000	1,000	6,000	5,000	5,000	1,000	1,000
D11	6,000	6,000	0,167	1,000	1,000	1,000	6,000	6,000	5,000	1,000	1,000

LAMPIRAN B (LANJUTAN)

**HASIL RATA2 MATRIKS PERBANDINGAN KUISIONER**

	K1	K2	K3	K4	K5	K6	K7	K8	K9	K10	K11
K1	1,000	1,000	1,000	0,161	0,161	0,161	0,171	1,260	0,151	0,151	0,151
K2	1,000	1,000	1,000	0,171	0,171	0,171	0,212	0,550	0,151	0,151	0,151
K3	1,000	1,000	1,000	0,572	0,572	0,572	0,523	1,442	0,500	0,500	0,500
K4	6,214	5,848	1,747	1,000	1,000	1,000	3,271	5,013	0,794	0,415	0,415
K5	6,214	5,848	1,747	1,000	1,000	1,000	1,913	4,718	0,415	0,794	0,794
K6	6,214	5,848	1,747	1,000	1,000	1,000	3,271	4,718	0,415	0,415	0,415
K7	5,848	4,718	1,913	0,306	0,523	0,306	1,000	3,826	0,199	0,199	0,199
K8	0,794	1,817	0,693	0,199	0,212	0,212	0,261	1,000	0,151	0,151	0,151
K9	6,604	6,604	2,000	1,260	2,410	2,410	5,013	6,604	1,000	0,585	0,585
K10	6,604	6,604	2,000	2,410	1,260	2,410	5,013	6,214	1,710	1,000	1,000
K11	6,604	6,604	2,000	2,410	1,260	2,410	5,013	6,604	1,710	1,000	1,000
TOTAL	48,097	46,890	16,848	10,490	9,569	11,652	29,662	41,949	7,197	8,372	8,362

**MATRIKS NORMALISASI**

	K1	K2	K3	K4	K5	K6	K7	K8	K9	K10	K11
K1	0,021	0,021	0,059	0,017	0,017	0,014	0,007	0,030	0,021	0,028	0,028
K2	0,021	0,021	0,059	0,018	0,018	0,015	0,008	0,031	0,021	0,028	0,028
K3	0,021	0,021	0,059	0,055	0,055	0,046	0,020	0,054	0,069	0,093	0,093
K4	0,129	0,125	0,104	0,093	0,103	0,086	0,127	0,120	0,110	0,077	0,077
K5	0,129	0,125	0,104	0,093	0,105	0,086	0,075	0,112	0,078	0,148	0,148
K6	0,129	0,125	0,104	0,093	0,105	0,086	0,127	0,112	0,078	0,077	0,077
K7	0,122	0,101	0,114	0,028	0,055	0,026	0,039	0,091	0,020	0,037	0,037
K8	0,017	0,019	0,041	0,018	0,022	0,018	0,010	0,024	0,021	0,010	0,028
K9	0,137	0,141	0,119	0,120	0,252	0,207	0,155	0,157	0,110	0,109	0,109
K10	0,137	0,141	0,119	0,230	0,132	0,207	0,155	0,148	0,238	0,186	0,186
K11	0,137	0,141	0,119	0,230	0,132	0,207	0,155	0,157	0,238	0,186	0,186
TOTAL	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000

**B.2 Perhitungan AHP untuk Bobot Setiap Kualifikasi dalam Departemen Akuntansi**

**MATRIKS PERBANDINGAN KUISIONER 1**

	K1	K2	K3	K4	K5	K6	K7
K1	1,000	0,500	0,500	0,250	0,500	0,500	2,000
K2	2,000	1,000	3,000	0,333	4,000	4,000	2,000
K3	2,000	0,333	1,000	0,250	0,500	0,500	2,000
K4	4,000	3,000	4,000	1,000	4,000	3,000	2,000
K5	2,000	0,250	2,000	0,250	1,000	2,000	0,250
K6	2,000	0,250	2,000	0,333	0,500	1,000	1,000
K7	0,500	0,500	0,500	0,200	4,000	1,000	1,000

## LAMPIRAN B (LANJUTAN)

Matriks Perbandingan Kuisisioner 2

	K1	K2	K3	K4	K5	K6	K7
K1	1,000	0,200	0,200	0,200	0,200	0,200	0,200
K2	5,000	1,000	7,000	0,143	7,000	0,111	0,125
K3	5,000	0,143	1,000	0,125	8,000	0,143	0,125
K4	5,000	7,000	8,000	1,000	8,000	0,125	6,000
K5	5,000	0,143	0,125	0,125	1,000	0,111	0,111
K6	5,000	9,000	7,000	8,000	9,000	1,000	8,000
K7	5,000	8,000	8,000	0,167	9,000	0,125	1,000

Matriks Perbandingan Kuisisioner 3

	K1	K2	K3	K4	K5	K6	K7
K1	1,000	0,200	0,333	0,333	0,200	0,111	3,000
K2	5,000	1,000	6,000	4,000	3,000	0,167	6,000
K3	3,000	0,167	1,000	1,000	3,000	0,250	3,000
K4	3,000	0,250	1,000	1,000	1,000	5,000	5,000
K5	5,000	0,333	0,333	1,000	1,000	1,000	4,000
K6	9,000	6,000	4,000	0,200	1,000	1,000	5,000
K7	0,333	0,167	0,333	0,200	0,250	0,200	1,000

Hasil Rata2 Matriks Perbandingan

	K1	K2	K3	K4	K5	K6	K7
K1	1,000	0,271	0,322	0,255	0,271	0,223	1,063
K2	3,684	1,000	5,013	0,575	4,380	0,420	1,145
K3	3,107	0,199	1,000	0,315	2,289	0,261	0,909
K4	3,915	1,738	3,175	1,000	3,175	1,233	3,915
K5	3,684	0,228	0,437	0,315	1,000	0,606	0,481
K6	4,481	2,381	3,826	0,811	1,651	1,000	3,420
K7	0,941	0,874	1,101	0,188	2,080	0,292	1,000
TOTAL	20,813	6,692	14,873	3,460	14,846	4,036	11,932

## LAMPIRAN B (LANJUTAN)

MATEMATIKA NORMALISASI

	K1	K2	K3	K4	K5	K6	K7	Imj	Wangs	Bobot
K1	0,048	0,041	0,022	0,074	0,018	0,055	0,089	0,347	0,050	
K2	0,177	0,149	0,337	0,166	0,295	0,104	0,096	1,325	0,189	
K3	0,149	0,030	0,067	0,091	0,154	0,065	0,076	0,632	0,090	
K4	0,188	0,260	0,213	0,289	0,214	0,306	0,328	1,798	0,257	
K5	0,177	0,034	0,029	0,091	0,067	0,150	0,040	0,589	0,084	
K6	0,215	0,356	0,257	0,234	0,111	0,248	0,287	1,708	0,244	
K7	0,045	0,131	0,074	0,054	0,140	0,072	0,084	0,601	0,086	
	1,000	1,000	1,000	1,000	1,000	1,000	1,000	7,000	1,000	

## B.3 Perhitungan AHP untuk Bobot Setiap Kualifikasi dalam Departemen Anggaran

MATEMATIKA PERBANDINGAN KUISIONER 1

	K1	K2	K3	K4	K5	K6	K7
K1	1,000	0,250	0,250	0,200	0,250	0,333	1,000
K2	4,000	1,000	5,000	0,200	5,000	4,000	3,000
K3	4,000	0,200	1,000	0,250	0,500	3,000	3,000
K4	5,000	5,000	4,000	1,000	0,250	0,250	0,250
K5	4,000	0,200	2,000	4,000	1,000	0,500	0,500
K6	3,000	0,250	0,333	4,000	2,000	1,000	0,500
K7	1,000	0,333	0,333	4,000	2,000	2,000	1,000

MATEMATIKA PERBANDINGAN KUISIONER 3

	K1	K2	K3	K4	K5	K6	K7
K1	1,000	0,200	0,250	0,200	0,200	0,250	0,167
K2	5,000	1,000	0,200	5,000	1,000	4,000	5,000
K3	4,000	5,000	1,000	1,000	1,000	4,000	7,000
K4	5,000	0,200	1,000	1,000	1,000	4,000	6,000
K5	5,000	1,000	1,000	1,000	1,000	0,250	5,000
K6	4,000	0,250	0,250	0,250	0,200	1,000	3,000
K7	6,000	0,200	0,143	0,167	5,000	0,333	1,000

## LAMPIRAN B (LANJUTAN)

HASIL RATA2 MATRIKS PERBANDINGAN

	K1	K2	K3	K4	K5	K6	K7
K1	1,000	0,193	0,207	0,179	0,193	0,228	0,322
K2	5,192	1,000	1,913	0,523	3,271	1,211	1,233
K3	4,820	0,523	1,000	0,315	1,587	1,197	1,442
K4	5,593	1,913	3,175	1,000	1,260	0,500	2,080
K5	5,192	0,306	0,630	0,794	1,000	0,240	0,652
K6	4,380	0,825	0,836	2,000	1,533	1,000	2,289
K7	3,107	0,811	0,693	0,481	4,481	0,437	1,000
TOTAL	29,285	5,570	8,454	5,291	13,325	4,814	9,019

MATRIKS NORMALISASI

	K1	K2	K3	K4	K5	K6	K7	Jumlah	Bobot	Ranking
K1	0,034	0,035	0,025	0,034	0,014	0,047	0,036	0,225	0,032	
K2	0,177	0,180	0,226	0,099	0,245	0,252	0,137	1,316	0,188	
K3	0,165	0,094	0,118	0,060	0,119	0,249	0,160	0,964	0,138	
K4	0,191	0,343	0,376	0,189	0,095	0,104	0,231	1,528	0,218	
K5	0,177	0,055	0,075	0,150	0,075	0,050	0,072	0,654	0,093	
K6	0,150	0,148	0,099	0,378	0,115	0,208	0,254	1,351	0,193	
K7	0,106	0,146	0,082	0,091	0,336	0,091	0,111	0,962	0,137	
	1,000	1,000	1,000	1,000	1,000	1,000	1,000	7,000	1,000	

## B.4 Perhitungan AHP untuk Bobot Setiap Kualifikasi dalam Departemen Keuangan

MATRIKS PERBANDINGAN KUISIONER 1

	K1	K2	K3	K4	K5	K6	K7
K1	1,000	2,000	0,250	0,167	0,250	0,200	1,000
K2	0,500	1,000	3,000	0,333	2,000	8,000	4,000
K3	4,000	0,333	1,000	0,333	3,000	3,000	3,000
K4	6,000	3,000	3,000	1,000	4,000	4,000	4,000
K5	4,000	0,500	0,333	0,250	1,000	0,333	0,333
K6	5,000	0,125	0,333	0,250	3,000	1,000	0,500
K7	1,000	0,250	0,333	0,250	3,000	2,000	1,000

## LAMPIRAN B (LANJUTAN)

Matriks Perbandingan Kuisisioner 2

	K1	K2	K3	K4	K5	K6	K7
K1	1,000	0,167	0,167	0,143	0,167	0,167	0,200
K2	6,000	1,000	7,000	0,143	5,000	0,111	0,125
K3	6,000	0,143	1,000	0,250	4,000	0,143	0,125
K4	7,000	7,000	4,000	1,000	8,000	0,125	6,000
K5	6,000	0,200	0,250	0,125	1,000	0,111	0,111
K6	6,000	9,000	7,000	8,000	9,000	1,000	0,125
K7	5,000	0,125	8,000	0,167	9,000	8,000	1,000

Matriks Perbandingan Kuisisioner 3

	K1	K2	K3	K4	K5	K6	K7
K1	1,000	0,111	0,333	0,167	0,200	0,333	3,000
K2	9,000	1,000	3,000	3,000	1,000	0,333	4,000
K3	3,000	0,333	1,000	0,333	1,000	3,000	3,000
K4	6,000	0,333	3,000	1,000	3,000	0,333	3,000
K5	5,000	1,000	1,000	0,333	1,000	3,000	3,000
K6	3,000	3,000	0,333	3,000	0,333	1,000	7,000
K7	0,333	0,250	0,333	0,333	0,333	0,143	1,000

Hasil Rata2 Matriks Perbandingan

	K1	K2	K3	K4	K5	K6	K7
K1	1,000	0,333	0,240	0,158	0,203	0,223	0,843
K2	3,000	1,000	3,979	0,523	2,154	0,667	1,260
K3	4,160	0,251	1,000	0,303	2,289	1,087	1,040
K4	6,316	1,913	3,302	1,000	4,579	0,550	4,160
K5	4,932	0,464	0,437	0,218	1,000	0,481	0,481
K6	4,481	1,500	0,920	1,817	2,080	1,000	0,759
K7	1,186	0,198	0,961	0,240	2,080	1,317	1,000
TOTAL	25,076	5,660	10,839	4,260	14,386	5,326	9,543

## LAMPIRAN B (LANJUTAN)

## B.5 Perhitungan AHP untuk Bobot Setiap Kualifikasi dalam Departemen Pemeliharaan I

## Matriks Perbandingan Kuisisioner 1

	K1	K2	K3	K4	K5	K6	K7
K1	1,000	0,167	0,167	0,200	0,250	0,250	0,250
K2	6,000	1,000	4,000	0,200	3,000	2,000	3,000
K3	6,000	0,250	1,000	0,200	0,250	0,500	2,000
K4	5,000	5,000	5,000	1,000	4,000	4,000	5,000
K5	4,000	0,333	4,000	0,250	1,000	0,500	0,500
K6	4,000	0,500	2,000	0,250	2,000	1,000	0,500
K7	4,000	0,333	0,500	0,200	2,000	2,000	1,000

## Matriks Perbandingan Kuisisioner 2

	K1	K2	K3	K4	K5	K6	K7
K1	1,000	0,143	0,143	0,167	0,200	0,200	0,200
K2	7,000	1,000	5,000	0,167	4,000	3,000	4,000
K3	7,000	0,200	1,000	0,200	0,250	0,500	2,000
K4	6,000	0,167	5,000	1,000	4,000	4,000	5,000
K5	5,000	0,250	4,000	0,250	1,000	0,333	0,333
K6	5,000	0,333	2,000	0,250	3,000	1,000	0,333
K7	5,000	0,250	0,500	0,200	3,000	3,000	1,000

## Matriks Perbandingan Kuisisioner 3

	K1	K2	K3	K4	K5	K6	K7
K1	1,000	0,200	0,167	0,167	0,200	0,200	0,500
K2	5,000	1,000	5,000	5,000	5,000	5,000	5,000
K3	6,000	0,200	1,000	1,000	1,000	1,000	7,000
K4	6,000	0,200	1,000	1,000	3,000	0,250	4,000
K5	5,000	0,200	1,000	0,333	1,000	0,250	3,000
K6	5,000	0,200	1,000	4,000	4,000	1,000	3,000
K7	2,000	0,200	0,143	0,250	0,333	0,333	1,000

## LAMPIRAN B (LANJUTAN)

## HASIL RATA2 MATRIKS PERBANDINGAN

	K1	K2	K3	K4	K5	K6	K7
K1	1,000	0,168	0,158	0,177	0,215	0,215	0,292
K2	5,944	1,000	4,642	0,550	3,915	3,107	3,915
K3	6,316	0,215	1,000	0,342	0,397	0,630	3,037
K4	5,646	0,550	2,924	1,000	3,634	1,587	4,642
K5	4,642	0,255	2,520	0,275	1,000	0,347	0,794
K6	4,642	0,322	1,587	0,630	2,884	1,000	0,794
K7	3,420	0,255	0,329	0,215	1,260	1,260	1,000
TOTAL	31,610	2,767	13,160	3,190	13,306	8,147	14,473

## MATRIKS NORMALISASI

	K1	K2	K3	K4	K5	K6	K7	Jml Huru	Bobot
K1	0,032	0,061	0,012	0,056	0,016	0,026	0,020	0,223	0,032
K2	0,188	0,361	0,353	0,173	0,294	0,381	0,270	2,021	0,289
K3	0,200	0,078	0,076	0,107	0,030	0,077	0,210	0,778	0,111
K4	0,179	0,199	0,222	0,313	0,273	0,195	0,321	1,702	0,243
K5	0,147	0,092	0,191	0,086	0,075	0,043	0,055	0,689	0,098
K6	0,147	0,116	0,121	0,197	0,217	0,123	0,055	0,976	0,139
K7	0,108	0,092	0,025	0,068	0,095	0,155	0,069	0,612	0,087
	1,000	1,000	1,000	1,000	1,000	1,000	1,000	7,000	1,000

## B.6 Perhitungan AHP untuk Bobot Setiap Kualifikasi dalam Departemen Pemeliharaan II

## MATRIKS PERBANDINGAN KUISIONER I

	K1	K2	K3	K4	K5	K6	K7
K1	1,000	0,200	0,200	0,200	0,250	0,333	0,333
K2	5,000	1,000	3,000	0,250	0,500	3,000	3,000
K3	5,000	0,333	1,000	0,200	3,000	3,000	4,000
K4	5,000	4,000	5,000	1,000	5,000	5,000	5,000
K5	4,000	2,000	0,333	0,200	1,000	3,000	3,000
K6	3,000	0,333	0,333	0,200	0,333	1,000	3,000
K7	3,000	0,333	0,250	0,200	0,333	0,333	1,000

## LAMPIRAN B (LANJUTAN)

MATEMATIKA PERBANDINGAN KUISIONER 2

	K1	K2	K3	K4	K5	K6	K7
K1	1,000	0,143	0,143	0,167	0,200	0,200	0,200
K2	7,000	1,000	5,000	0,167	4,000	3,000	4,000
K3	7,000	0,200	1,000	0,200	0,250	0,500	2,000
K4	6,000	6,000	5,000	1,000	4,000	4,000	5,000
K5	5,000	0,250	4,000	0,250	1,000	0,333	0,333
K6	5,000	0,333	2,000	0,250	3,000	1,000	0,333
K7	5,000	0,250	0,500	0,200	3,000	3,000	1,000

MATEMATIKA PERBANDINGAN KUISIONER 3

	K1	K2	K3	K4	K5	K6	K7
K1	1,000	0,167	0,167	0,143	0,200	0,167	3,000
K2	6,000	1,000	5,000	5,000	5,000	5,000	5,000
K3	6,000	0,200	1,000	1,000	1,000	1,000	7,000
K4	7,000	0,200	1,000	1,000	3,000	0,250	3,000
K5	5,000	0,200	1,000	0,333	1,000	0,250	3,000
K6	6,000	0,200	1,000	4,000	4,000	1,000	3,000
K7	0,333	0,200	0,143	0,333	0,333	0,333	1,000

HASIL RATA2 MATEMATIKA PERBANDINGAN

	K1	K2	K3	K4	K5	K6	K7
K1	1,000	0,168	0,168	0,168	0,215	0,223	0,585
K2	5,944	1,000	4,217	0,593	2,154	3,557	3,915
K3	5,944	0,237	1,000	0,342	0,909	1,145	3,826
K4	5,944	1,687	2,924	1,000	3,915	1,710	4,217
K5	4,642	0,464	1,101	0,255	1,000	0,630	1,442
K6	4,481	0,281	0,874	0,585	1,587	1,000	1,442
K7	1,710	0,255	0,261	0,237	0,693	0,693	1,000
TOTAL	29,665	4,093	10,545	3,180	10,474	8,958	16,427

## LAMPIRAN B (LANJUTAN)

Matriks Normalisasi									
	K1	K2	K3	K4	K5	K6	K7	Jml Baris	Bobot
K1	0,034	0,041	0,016	0,053	0,021	0,025	0,036	0,225	0,032
K2	0,200	0,244	0,400	0,186	0,206	0,397	0,238	1,872	0,267
K3	0,200	0,058	0,095	0,108	0,087	0,128	0,233	0,908	0,130
K4	0,200	0,412	0,277	0,314	0,374	0,191	0,257	2,026	0,289
K5	0,156	0,113	0,104	0,080	0,095	0,070	0,088	0,708	0,101
K6	0,151	0,069	0,083	0,184	0,152	0,112	0,088	0,837	0,120
K7	0,058	0,062	0,025	0,075	0,066	0,077	0,061	0,424	0,061
	1,000	1,000	1,000	1,000	1,000	1,000	1,000	7,000	1,000

### B.7 Perhitungan AHP untuk Bobot Setiap Kualifikasi dalam Departemen Pemeliharaan III

#### Matriks Perbandingan Kuisisioner 1

	K1	K2	K3	K4	K5	K6	K7
K1	1,000	0,200	0,200	0,200	0,250	0,333	0,333
K2	5,000	1,000	0,500	3,000	3,000	3,000	3,000
K3	5,000	0,500	1,000	0,250	2,000	3,000	3,000
K4	5,000	0,333	4,000	1,000	3,000	4,000	4,000
K5	4,000	0,333	0,500	0,333	1,000	1,000	0,333
K6	3,000	0,333	0,333	0,250	1,000	1,000	0,333
K7	3,000	0,333	0,333	0,250	3,000	3,000	1,000

#### Matriks Perbandingan Kuisisioner 2

	K1	K2	K3	K4	K5	K6	K7
K1	1,000	0,143	0,143	0,143	0,167	0,200	0,200
K2	7,000	1,000	5,000	0,167	4,000	3,000	4,000
K3	7,000	0,200	1,000	0,200	0,250	0,500	2,000
K4	7,000	6,000	5,000	1,000	4,000	4,000	5,000
K5	6,000	0,250	4,000	0,250	1,000	0,333	0,333
K6	5,000	0,333	2,000	0,250	3,000	1,000	0,333
K7	5,000	0,250	0,500	0,200	3,000	3,000	1,000

## LAMPIRAN B (LANJUTAN)

Matriks Perbandingan Kuisisioner 3

	K1	K2	K3	K4	K5	K6	K7
K1	1,000	0,167	0,167	0,143	0,200	0,167	3,000
K2	6,000	1,000	5,000	5,000	5,000	5,000	5,000
K3	6,000	0,200	1,000	1,000	1,000	1,000	7,000
K4	7,000	0,200	1,000	1,000	0,250	0,250	0,333
K5	5,000	0,200	1,000	4,000	1,000	0,250	3,000
K6	6,000	0,200	1,000	4,000	4,000	1,000	3,000
K7	0,333	0,200	0,143	0,333	0,333	0,333	1,000

HASIL RATA2 Matriks Perbandingan

	K1	K2	K3	K4	K5	K6	K7
K1	1,000	0,168	0,168	0,160	0,203	0,223	0,585
K2	5,944	1,000	2,321	1,357	3,915	3,557	3,915
K3	5,944	0,271	1,000	0,368	0,794	1,145	3,476
K4	6,257	0,737	2,714	1,000	1,442	1,587	1,882
K5	4,932	0,255	1,260	0,693	1,000	0,437	0,693
K6	4,481	0,281	0,874	0,630	2,289	1,000	0,693
K7	1,710	0,255	0,288	0,255	1,442	1,442	1,000
TOTAL	30,269	2,969	8,625	4,464	11,085	9,391	12,244

Matriks Normalisasi

	K1	K2	K3	K4	K5	K6	K7	Total	Bobot
K1	0,033	0,057	0,020	0,036	0,018	0,024	0,048	0,235	0,034
K2	0,196	0,337	0,269	0,304	0,353	0,379	0,320	2,158	0,308
K3	0,196	0,091	0,116	0,083	0,072	0,122	0,284	0,964	0,138
K4	0,207	0,248	0,315	0,224	0,130	0,169	0,154	1,447	0,207
K5	0,163	0,086	0,146	0,155	0,090	0,047	0,057	0,744	0,106
K6	0,148	0,095	0,101	0,141	0,207	0,106	0,057	0,855	0,122
K7	0,056	0,086	0,033	0,057	0,130	0,154	0,082	0,598	0,085
TOTAL	1,000	1,000	1,000	1,000	1,000	1,000	1,000	7,000	1,000

## LAMPIRAN B (LANJUTAN)

## B.8 Perhitungan AHP untuk Bobot Setiap Kualifikasi dalam Departemen Pengelolaan Pelabuhan

Matriks Perbandingan Kuisisioner 1

	K1	K2	K3	K4	K5	K6	K7
K1	1,000	0,167	0,167	0,167	0,200	0,250	0,333
K2	6,000	1,000	4,000	0,250	4,000	0,333	3,000
K3	6,000	0,250	1,000	0,250	2,000	2,000	3,000
K4	6,000	4,000	4,000	1,000	3,000	7,000	7,000
K5	5,000	0,250	0,500	0,333	1,000	3,000	0,333
K6	4,000	3,000	0,500	0,143	0,333	1,000	0,333
K7	3,000	0,333	0,333	0,143	3,000	3,000	1,000

Matriks Perbandingan Kuisisioner 2

	K1	K2	K3	K4	K5	K6	K7
K1	1,000	0,143	0,143	0,143	0,250	0,200	0,167
K2	7,000	1,000	3,000	0,200	5,000	0,250	4,000
K3	7,000	0,333	1,000	0,200	4,000	3,000	4,000
K4	7,000	5,000	5,000	1,000	4,000	5,000	5,000
K5	4,000	0,200	0,250	0,250	1,000	4,000	0,250
K6	5,000	4,000	0,333	0,200	0,250	1,000	0,250
K7	6,000	0,250	0,250	0,200	4,000	4,000	1,000

Matriks Perbandingan Kuisisioner 3

	K1	K2	K3	K4	K5	K6	K7
K1	1,000	0,167	0,250	0,143	0,200	0,167	3,000
K2	6,000	1,000	2,000	4,000	2,000	2,000	2,000
K3	4,000	0,500	1,000	3,000	1,000	1,000	7,000
K4	7,000	0,250	0,333	1,000	3,000	0,250	3,000
K5	5,000	0,500	1,000	0,333	1,000	0,250	3,000
K6	6,000	0,500	1,000	4,000	4,000	1,000	5,000
K7	0,333	0,500	0,143	0,333	0,333	0,200	1,000

## LAMPIRAN B (LANJUTAN)

## HASIL RATA2 MATRIKS PERBANDINGAN

	K1	K2	K3	K4	K5	K6	K7
K1	1,000	0,158	0,181	0,150	0,215	0,203	0,550
K2	6,316	1,000	2,884	0,585	3,420	0,550	2,884
K3	5,518	0,347	1,000	0,531	2,000	1,817	4,380
K4	6,649	1,710	1,882	1,000	3,302	2,061	4,718
K5	4,642	0,292	0,500	0,303	1,000	1,442	0,630
K6	4,932	1,817	0,550	0,485	0,693	1,000	0,747
K7	1,817	0,347	0,228	0,212	1,587	1,339	1,000
TOTAL	30,875	5,671	7,226	3,267	12,218	8,412	14,909

## MATRIKS NORMALISASI

	K1	K2	K3	K4	K5	K6	K7	Jml Hari	Robot
K1	0,032	0,028	0,025	0,046	0,018	0,024	0,037	0,210	0,030
K2	0,205	0,176	0,399	0,179	0,280	0,065	0,193	1,498	0,214
K3	0,179	0,061	0,138	0,163	0,164	0,216	0,294	1,214	0,173
K4	0,215	0,302	0,260	0,306	0,270	0,245	0,316	1,915	0,274
K5	0,150	0,052	0,069	0,093	0,082	0,171	0,042	0,659	0,094
K6	0,160	0,320	0,076	0,149	0,057	0,119	0,050	0,931	0,133
K7	0,059	0,061	0,032	0,065	0,130	0,159	0,067	0,573	0,082
	1,000	1,000	1,000	1,000	1,000	1,000	1,000	7,000	1,000

## B.9 Perhitungan AHP untuk Bobot Setiap Kualifikasi dalam Departemen PPK

## MATRIKS PERBANDINGAN KUISIONER 1

	K1	K2	K3	K4	K5	K6	K7
K1	1,000	0,250	0,250	0,250	0,333	0,500	0,250
K2	4,000	1,000	3,000	0,250	3,000	4,000	4,000
K3	4,000	0,333	1,000	0,250	3,000	3,000	4,000
K4	4,000	4,000	4,000	1,000	4,000	4,000	4,000
K5	3,000	0,333	0,333	0,250	1,000	2,000	0,333
K6	2,000	0,250	0,333	0,250	0,500	1,000	0,333
K7	4,000	0,250	0,250	0,250	3,000	3,000	1,000

## LAMPIRAN B (LANJUTAN)

## Matriks Perbandingan Kuisiner 2

	K1	K2	K3	K4	K5	K6	K7
K1	1,000	0,200	0,200	0,200	0,250	0,333	0,200
K2	5,000	1,000	4,000	0,200	4,000	5,000	5,000
K3	5,000	0,250	1,000	0,200	4,000	4,000	5,000
K4	5,000	5,000	5,000	1,000	5,000	5,000	5,000
K5	4,000	0,250	0,250	0,200	1,000	3,000	0,250
K6	3,000	0,200	0,250	0,200	0,333	1,000	0,250
K7	5,000	0,200	0,200	0,200	4,000	4,000	1,000

## Matriks Perbandingan Kuisiner 3

	K1	K2	K3	K4	K5	K6	K7
K1	1,000	0,167	0,167	0,143	0,200	0,167	3,000
K2	6,000	1,000	4,000	4,000	4,000	4,000	4,000
K3	6,000	0,250	1,000	1,000	1,000	1,000	0,250
K4	7,000	0,250	1,000	1,000	3,000	0,250	3,000
K5	5,000	0,250	1,000	0,333	1,000	0,250	3,000
K6	6,000	0,250	1,000	4,000	4,000	1,000	3,000
K7	0,333	0,250	4,000	0,333	0,333	0,333	1,000

## HASIL RATA2 Matriks Perbandingan

	K1	K2	K3	K4	K5	K6	K7
K1	1,000	0,203	0,203	0,193	0,255	0,303	0,531
K2	4,932	1,000	3,634	0,585	3,634	4,309	4,309
K3	4,932	0,275	1,000	0,368	2,289	2,289	1,710
K4	5,192	1,710	2,714	1,000	3,915	1,710	3,915
K5	3,915	0,275	0,437	0,255	1,000	1,145	0,630
K6	3,302	0,232	0,437	0,585	0,874	1,000	0,630
K7	1,882	0,232	0,585	0,255	1,587	1,587	1,000
TOTAL	25,156	3,927	9,010	3,241	13,555	12,343	12,725

## LAMPIRAN B (LANJUTAN)

## B.10 Perhitungan AHP untuk Bobot Setiap Kualifikasi dalam Departemen Produksi I

Matriks Perbandingan Kuisisioner 1

	K1	K2	K3	K4	K5	K6	K7
K1	1,000	0,200	0,200	0,200	0,250	0,333	0,500
K2	5,000	1,000	3,000	0,250	2,000	3,000	3,000
K3	5,000	0,333	1,000	0,250	3,000	2,000	2,000
K4	5,000	4,000	4,000	1,000	4,000	4,000	4,000
K5	4,000	0,500	0,333	0,250	1,000	2,000	1,000
K6	3,000	0,333	0,500	0,250	0,500	1,000	0,500
K7	2,000	0,333	0,500	0,250	1,000	2,000	1,000

Matriks Perbandingan Kuisisioner 2

	K1	K2	K3	K4	K5	K6	K7
K1	1,000	0,167	0,167	0,167	0,200	0,250	0,333
K2	6,000	1,000	4,000	0,200	3,000	4,000	4,000
K3	6,000	0,250	1,000	0,200	4,000	2,000	2,000
K4	6,000	5,000	5,000	1,000	4,000	4,000	4,000
K5	5,000	0,333	0,250	0,250	1,000	2,000	1,000
K6	4,000	0,250	0,500	0,250	0,500	1,000	3,000
K7	3,000	0,250	0,500	0,250	1,000	0,333	1,000

Matriks Perbandingan Kuisisioner 3

	K1	K2	K3	K4	K5	K6	K7
K1	1,000	0,167	0,167	0,143	0,200	0,167	3,000
K2	6,000	1,000	5,000	5,000	5,000	5,000	5,000
K3	6,000	0,200	1,000	1,000	1,000	1,000	7,000
K4	7,000	0,200	1,000	1,000	3,000	0,250	3,000
K5	5,000	0,200	1,000	0,333	1,000	0,250	3,000
K6	6,000	0,200	1,000	4,000	4,000	1,000	3,000
K7	0,333	0,200	0,143	0,333	0,333	0,333	1,000

## LAMPIRAN B (LANJUTAN)

## HASIL RATA2 MATRIKS PERBANDINGAN

	K1	K2	K3	K4	K5	K6	K7
K1	1,000	0,177	0,177	0,168	0,215	0,240	0,794
K2	5,646	1,000	3,915	0,630	3,107	3,915	3,915
K3	5,646	0,255	1,000	0,368	2,289	1,587	3,037
K4	5,944	1,587	2,714	1,000	3,634	1,587	3,634
K5	4,642	0,322	0,437	0,275	1,000	1,000	1,442
K6	4,160	0,255	0,630	0,630	1,000	1,000	1,651
K7	1,260	0,255	0,329	0,275	0,693	0,606	1,000
TOTAL	28,298	3,853	9,202	3,347	11,940	9,936	15,473

## MATRIKS NORMALISASI

	K1	K2	K3	K4	K5	K6	K7	Inti. Bantu	Robot
K1	0,035	0,046	0,019	0,050	0,018	0,024	0,051	0,244	0,035
K2	0,200	0,260	0,425	0,188	0,260	0,394	0,253	1,980	0,283
K3	0,200	0,066	0,109	0,110	0,192	0,160	0,196	1,032	0,147
K4	0,210	0,412	0,295	0,299	0,304	0,160	0,235	1,915	0,274
K5	0,164	0,084	0,047	0,082	0,084	0,101	0,093	0,655	0,094
K6	0,147	0,066	0,068	0,188	0,084	0,101	0,107	0,761	0,109
K7	0,045	0,066	0,036	0,082	0,058	0,061	0,065	0,412	0,059
	1,000	1,000	1,000	1,000	1,000	1,000	1,000	7,000	1,000

## B.11 Perhitungan AHP untuk Bobot Setiap Kualifikasi dalam Departemen Produksi IIA

## MATRIKS PERBANDINGAN KUISIONER 1

	K1	K2	K3	K4	K5	K6	K7
K1	1,000	0,200	0,200	0,200	0,250	0,333	0,333
K2	5,000	1,000	4,000	0,250	3,000	4,000	3,000
K3	5,000	0,250	1,000	0,200	3,000	3,000	3,000
K4	5,000	4,000	5,000	1,000	4,000	5,000	5,000
K5	4,000	0,333	0,333	0,250	1,000	3,000	2,000
K6	3,000	0,250	0,333	0,200	0,333	1,000	1,000
K7	3,000	0,333	0,333	0,200	0,500	1,000	1,000

## LAMPIRAN B (LANJUTAN)

MATEMATIKA PERBANDINGAN KUISIONER 2

	K1	K2	K3	K4	K5	K6	K7
K1	1,000	0,167	0,167	0,167	0,200	0,250	0,333
K2	6,000	1,000	4,000	0,200	3,000	4,000	4,000
K3	6,000	0,250	1,000	0,200	0,250	0,500	0,500
K4	6,000	5,000	5,000	1,000	4,000	4,000	4,000
K5	5,000	0,333	4,000	0,250	1,000	2,000	1,000
K6	4,000	0,250	2,000	0,250	0,500	1,000	3,000
K7	3,000	0,250	2,000	0,250	1,000	0,333	1,000

MATEMATIKA PERBANDINGAN KUISIONER 3

	K1	K2	K3	K4	K5	K6	K7
K1	1,000	0,167	0,167	0,143	0,200	0,167	3,000
K2	6,000	1,000	5,000	5,000	5,000	5,000	5,000
K3	6,000	0,200	1,000	1,000	1,000	1,000	6,000
K4	7,000	0,200	1,000	1,000	3,000	0,250	4,000
K5	5,000	0,200	1,000	0,333	1,000	0,250	3,000
K6	6,000	0,200	1,000	4,000	4,000	1,000	3,000
K7	0,333	0,200	0,167	0,250	0,333	0,333	1,000

HASIL RATA2 MATEMATIKA PERBANDINGAN

	K1	K2	K3	K4	K5	K6	K7
K1	1,000	0,177	0,177	0,168	0,215	0,240	0,693
K2	5,646	1,000	4,309	0,630	3,557	4,309	3,915
K3	5,646	0,232	1,000	0,342	0,909	1,145	2,080
K4	5,944	1,587	2,924	1,000	3,634	1,710	4,309
K5	4,642	0,281	1,101	0,275	1,000	1,145	1,817
K6	4,160	0,232	0,874	0,585	0,874	1,000	2,080
K7	1,442	0,255	0,481	0,232	0,550	0,481	1,000
TOTAL	28,480	3,765	10,865	3,232	10,739	10,029	15,894

## LAMPIRAN B (LANJUTAN)

MATEMATIKA NORMALISASI									
	K1	K2	K3	K4	K5	K6	K7	Nilai Rangsang	Bobot
K1	0,035	0,047	0,016	0,052	0,020	0,024	0,044	0,238	0,034
K2	0,198	0,266	0,397	0,195	0,331	0,430	0,246	2,062	0,295
K3	0,198	0,062	0,092	0,106	0,085	0,114	0,131	0,787	0,112
K4	0,209	0,422	0,269	0,309	0,338	0,170	0,271	1,989	0,284
K5	0,163	0,075	0,101	0,085	0,093	0,114	0,114	0,746	0,107
K6	0,146	0,062	0,080	0,181	0,081	0,100	0,131	0,781	0,112
K7	0,051	0,068	0,044	0,072	0,051	0,048	0,063	0,397	0,057
Σ	1,000	1,000	1,000	1,000	1,000	1,000	1,000	7,000	1,000

### B.12 Perhitungan AHP untuk Bobot Setiap Kualifikasi dalam Departemen Produksi IIB

#### MATEMATIKA PERBANDINGAN KUISIONER 1

	K1	K2	K3	K4	K5	K6	K7
K1	1,000	0,200	0,200	0,200	0,250	0,333	0,333
K2	4,000	1,000	4,000	0,250	3,000	3,000	4,000
K3	4,000	0,250	1,000	0,250	0,500	2,000	2,000
K4	4,000	4,000	4,000	1,000	5,000	5,000	5,000
K5	4,000	0,333	2,000	0,200	1,000	2,000	3,000
K6	3,000	0,333	0,500	0,200	0,500	1,000	0,250
K7	3,000	0,250	0,500	0,200	0,333	4,000	1,000

#### MATEMATIKA PERBANDINGAN KUISIONER 2

	K1	K2	K3	K4	K5	K6	K7
K1	1,000	0,167	0,167	0,167	0,200	0,250	0,333
K2	6,000	1,000	4,000	0,200	3,000	4,000	4,000
K3	6,000	0,250	1,000	0,200	0,250	0,500	0,500
K4	6,000	5,000	5,000	1,000	4,000	4,000	4,000
K5	5,000	0,333	4,000	0,250	1,000	2,000	1,000
K6	4,000	0,250	2,000	0,250	0,500	1,000	3,000
K7	3,000	0,250	2,000	0,250	1,000	0,333	1,000

## LAMPIRAN B (LANJUTAN)

Matriks Perbandingan Kuisisioner 3

	K1	K2	K3	K4	K5	K6	K7
K1	1,000	0,167	0,167	0,143	0,200	0,167	3,000
K2	6,000	1,000	5,000	5,000	5,000	5,000	5,000
K3	6,000	0,200	1,000	1,000	1,000	1,000	7,000
K4	7,000	0,200	1,000	1,000	0,250	0,250	3,000
K5	5,000	0,200	1,000	4,000	1,000	0,250	3,000
K6	6,000	0,200	1,000	4,000	4,000	1,000	3,000
K7	0,333	0,200	0,143	0,333	0,333	0,333	1,000

Hasil Rata2 Matriks Perbandingan

	K1	K2	K3	K4	K5	K6	K7
K1	1,000	0,177	0,177	0,168	0,215	0,240	0,693
K2	5,241	1,000	4,309	0,630	3,557	3,915	4,309
K3	5,241	0,232	1,000	0,368	0,500	1,000	1,913
K4	5,518	1,587	2,714	1,000	1,710	1,710	3,915
K5	4,642	0,281	2,000	0,585	1,000	1,000	2,080
K6	4,160	0,255	1,000	0,585	1,000	1,000	1,310
K7	1,442	0,232	0,523	0,255	0,481	0,763	1,000
TOTAL	27,245	3,765	11,723	3,592	8,463	9,628	15,220

Matriks Normalisasi

	K1	K2	K3	K4	K5	K6	K7	Jml. Hamb	Hambig
K1	0,037	0,047	0,015	0,047	0,025	0,025	0,046	0,242	0,035
K2	0,192	0,266	0,368	0,175	0,420	0,407	0,283	2,111	0,302
K3	0,192	0,062	0,085	0,103	0,059	0,104	0,126	0,731	0,104
K4	0,203	0,422	0,232	0,278	0,202	0,178	0,257	1,771	0,253
K5	0,170	0,075	0,171	0,163	0,118	0,104	0,137	0,937	0,134
K6	0,153	0,068	0,085	0,163	0,118	0,104	0,086	0,777	0,111
K7	0,053	0,062	0,045	0,071	0,057	0,079	0,066	0,432	0,062
TOTAL	1,000	1,000	1,000	1,000	1,000	1,000	1,000	7,000	1,000

## LAMPIRAN B (LANJUTAN)

## B.13 Bentuk Kuisioner untuk Pembobotan Tiap Departemen

Nama :  
Departemen :

## Kuisioner Tingkat Kepentingan Departemen

## I. Departemen

1. Akuntansi (D1)
2. Anggaran (D2)
3. Keuangan (D3)
4. Pemeliharaan I (D4)
5. Pemeliharaan II (D5)
6. Pemeliharaan III (D6)
7. Pengelolaan Pelabuhan (D7)
8. Prasarana Publik dan Kawasan (D8)
9. Produksi I (D9)
10. Produksi II A (D10)
11. Produksi II B (D11)

## Keterangan Tingkat Kepentingan

Tingkat Kepentingan	Definisi
1	Kedua kriteria <b>sama penting</b> (Equal)
3	Kriteria 1 <b>sedikit lebih penting</b> dari kriteria 2 (Moderate)
5	Kriteria 1 <b>lebih penting</b> dari kriteria 2 (Strong)
7	Kriteria 1 <b>jelas lebih penting</b> dari kriteria 2 (Very Strong)
9	Kriteria 1 <b>mutlak lebih penting</b> dari kriteria 2 (Very Strong)
2, 4, 6, 8	Nilai-nilai diantara dua pertimbangan yang berdekatan

Contoh pengisian kuisioner.

Jika anda cenderung memilih Departemen di sisi kiri karena Departemen tersebut **lebih penting** dari Departemen di sisi kanan, maka lingkari angka sesuai dengan keterangan tingkat kepentingan diatas yaitu angka 5 di sisi kiri.

Departemen	Tingkat Kepentingan																	Departemen
Akuntansi	9	8	7	6	5	4	3	2	1	2	3	4	5	6	7	8	9	Anggaran

Jika anda cenderung memilih Departemen di sisi kanan karena Departemen tersebut **sedikit lebih penting** dari Departemen di sisi kiri, maka lingkari angka sesuai dengan keterangan tingkat kepentingan diatas yaitu angka 3 di sisi kanan.

Departemen	Tingkat Kepentingan																	Departemen
Akuntansi	9	8	7	6	5	4	3	2	1	2	3	4	5	6	7	8	9	Keuangan

## LAMPIRAN B (LANJUTAN)

Jika persepsi anda mengartikan Departemen di sisi kanan **sama penting** dengan Departemen di sisi kiri, maka lingkari angka sesuai dengan keterangan tingkat kepentingan diatas yaitu angka 1 di tengah.

Departemen	Tingkat Kepentingan																	Departemen
Akuntansi	9	8	7	6	5	4	3	2	1	2	3	4	5	6	7	8	9	Pemeliharaan I

Tingkat Kepentingan antar Departemen.

Lingkari angka pada kolom Tingkat Kepentingan sesuai dengan persepsi anda tentang kedua Departemen.

- I. Bagaimana perbandingan tingkat kepentingan antara departemen disisi kiri dengan departemen disisi kanan.

Departemen	Tingkat Kepentingan																	Departemen
Akuntansi	9	8	7	6	5	4	3	2	1	2	3	4	5	6	7	8	9	Anggaran
Akuntansi	9	8	7	6	5	4	3	2	1	2	3	4	5	6	7	8	9	Keuangan
Akuntansi	9	8	7	6	5	4	3	2	1	2	3	4	5	6	7	8	9	Pemeliharaan I
Akuntansi	9	8	7	6	5	4	3	2	1	2	3	4	5	6	7	8	9	Pemeliharaan II
Akuntansi	9	8	7	6	5	4	3	2	1	2	3	4	5	6	7	8	9	Pemeliharaan III
Akuntansi	9	8	7	6	5	4	3	2	1	2	3	4	5	6	7	8	9	Pengelolaan Pelabuhan
Akuntansi	9	8	7	6	5	4	3	2	1	2	3	4	5	6	7	8	9	PPK
Akuntansi	9	8	7	6	5	4	3	2	1	2	3	4	5	6	7	8	9	Produksi I
Akuntansi	9	8	7	6	5	4	3	2	1	2	3	4	5	6	7	8	9	Produksi II A
Akuntansi	9	8	7	6	5	4	3	2	1	2	3	4	5	6	7	8	9	Produksi II B
Anggaran	9	8	7	6	5	4	3	2	1	2	3	4	5	6	7	8	9	Keuangan
Anggaran	9	8	7	6	5	4	3	2	1	2	3	4	5	6	7	8	9	Pemeliharaan I
Anggaran	9	8	7	6	5	4	3	2	1	2	3	4	5	6	7	8	9	Pemeliharaan II
Anggaran	9	8	7	6	5	4	3	2	1	2	3	4	5	6	7	8	9	Pemeliharaan III
Anggaran	9	8	7	6	5	4	3	2	1	2	3	4	5	6	7	8	9	Pengelolaan Pelabuhan
Anggaran	9	8	7	6	5	4	3	2	1	2	3	4	5	6	7	8	9	PPK
Anggaran	9	8	7	6	5	4	3	2	1	2	3	4	5	6	7	8	9	Produksi I
Anggaran	9	8	7	6	5	4	3	2	1	2	3	4	5	6	7	8	9	Produksi II A
Anggaran	9	8	7	6	5	4	3	2	1	2	3	4	5	6	7	8	9	Produksi II B

## LAMPIRAN B (LANJUTAN)

## B.14 Bentuk Kuisioner untuk Pembobotan Tiap Kualifikasi

Nama : \_\_\_\_\_  
Departemen : \_\_\_\_\_

## Kuisioner Tingkat Kepentingan Kualifikasi dalam Setiap Departemen

## II. Kualifikasi

1. Penampilan (K1)
2. Pengetahuan Khusus / Spesialisasi dalam pekerjaan (K2)
3. Kemampuan Menyatakan Pendapat (KMP) (K3)
4. Kematangan Kepribadian (K4)
5. Ambisi / Dorongan untuk maju (K5)
6. Potensi untuk Berkembang (PB) (K6)
7. Kegiatan Organisasi (K7)

## Keterangan Tingkat Kepentingan

Tingkat Kepentingan	Definisi
1	Kedua kriteria <b>sama penting</b> (Equal)
3	Kriteria 1 <b>sedikit lebih penting</b> dari kriteria 2 (Moderate)
5	Kriteria 1 <b>lebih penting</b> dari kriteria 2 (Strong)
7	Kriteria 1 <b>jelas lebih penting</b> dari kriteria 2 (Very Strong)
9	Kriteria 1 <b>mullak lebih penting</b> dari kriteria 2 (Very Strong)
2, 4, 6, 8	Nilai-nilai di antara dua perimbangan yang berdekatan.

Contoh pengisian kuisioner.

Jika anda cenderung memilih Kualifikasi di sisi kiri karena Kualifikasi tersebut **lebih penting** dari Kualifikasi di sisi kanan, maka lingkari angka sesuai dengan keterangan tingkat kepentingan diatas yaitu angka 5 di sisi kiri.

Kualifikasi	Tingkat Kepentingan																	Kualifikasi
Penampilan	9	8	7	6	5	4	3	2	1	2	3	4	5	6	7	8	9	Pengetahuan Khusus

Jika anda cenderung memilih Kualifikasi di sisi kanan karena Kualifikasi tersebut **sedikit lebih penting** dari Kualifikasi di sisi kiri, maka lingkari angka sesuai dengan keterangan tingkat kepentingan diatas yaitu angka 3 di sisi kanan.

Kualifikasi	Tingkat Kepentingan																	Kualifikasi
Penampilan	9	8	7	6	5	4	3	2	1	2	3	4	5	6	7	8	9	KMP

Jika persepsi anda mengatakan Kualifikasi di sisi kanan **sama penting** dengan Kualifikasi di sisi kiri, maka lingkari angka sesuai dengan keterangan tingkat kepentingan diatas yaitu angka 1 di tengah.

Kualifikasi	Tingkat Kepentingan																	Kualifikasi
Penampilan	9	8	7	6	5	4	3	2	1	2	3	4	5	6	7	8	9	Kematangan Kepribadian

## LAMPIRAN B (LANJUTAN)

Tingkat **Kepentingan** antar Kualifikasi dalam Departemen.

Lingkari angka pada kolom Tingkat Kepentingan sesuai dengan persepsi anda tentang kedua Kualifikasi.

- I. Bagaimana perbandingan tingkat kepentingan antara kualifikasi disisi kiri dengan kualifikasi disisi kanan dalam Departemen Akutansi.

Kualifikasi	Tingkat Kepentingan																		Kualifikasi
Penampilan	9	8	7	6	5	4	3	2	1	2	3	4	5	6	7	8	9	Pengetahuan Khusus	
Penampilan	9	8	7	6	5	4	3	2	1	2	3	4	5	6	7	8	9	KMP	
Penampilan	9	8	7	6	5	4	3	2	1	2	3	4	5	6	7	8	9	Kematangan Kepribadian	
Penampilan	9	8	7	6	5	4	3	2	1	2	3	4	5	6	7	8	9	Ambisi	
Penampilan	9	8	7	6	5	4	3	2	1	2	3	4	5	6	7	8	9	PB	
Penampilan	9	8	7	6	5	4	3	2	1	2	3	4	5	6	7	8	9	Kegiatan Organisasi	
Pengetahuan Khusus	9	8	7	6	5	4	3	2	1	2	3	4	5	6	7	8	9	KMP	
Pengetahuan Khusus	9	8	7	6	5	4	3	2	1	2	3	4	5	6	7	8	9	Kematangan Kepribadian	
Pengetahuan Khusus	9	8	7	6	5	4	3	2	1	2	3	4	5	6	7	8	9	Ambisi	
Pengetahuan Khusus	9	8	7	6	5	4	3	2	1	2	3	4	5	6	7	8	9	PB	
Pengetahuan Khusus	9	8	7	6	5	4	3	2	1	2	3	4	5	6	7	8	9	Kegiatan Organisasi	
KMP	9	8	7	6	5	4	3	2	1	2	3	4	5	6	7	8	9	Kematangan Kepribadian	
KMP	9	8	7	6	5	4	3	2	1	2	3	4	5	6	7	8	9	Ambisi	
KMP	9	8	7	6	5	4	3	2	1	2	3	4	5	6	7	8	9	PB	
KMP	9	8	7	6	5	4	3	2	1	2	3	4	5	6	7	8	9	Kegiatan Organisasi	
Kematangan Kepribadian	9	8	7	6	5	4	3	2	1	2	3	4	5	6	7	8	9	Ambisi	
Kematangan Kepribadian	9	8	7	6	5	4	3	2	1	2	3	4	5	6	7	8	9	PB	
Kematangan Kepribadian	9	8	7	6	5	4	3	2	1	2	3	4	5	6	7	8	9	Kegiatan Organisasi	
Ambisi	9	8	7	6	5	4	3	2	1	2	3	4	5	6	7	8	9	PB	
Ambisi	9	8	7	6	5	4	3	2	1	2	3	4	5	6	7	8	9	Kegiatan Organisasi	

**LAMPIRAN B (LANJUTAN)****B.15 Dokumentasi Bimbingan Di PT. Petrokimia Gresik**

## 1. Pertemuan 1 – Perkenalan dan Diskusi

## Pembahasan:

- Perkenalan antara penulis dengan pembimbing dari Departemen Personalia PT. Petrokimia Gresik yang bernama Bapak Widiarto Dwi Pracoyo, S.Psi sebagai Staff muda Pengembangan SDM.
- Diskusi tentang permasalahan yang dibahas dalam Tugas Akhir Penulis.

## 2. Pertemuan 2 – Wawancara 1

## Pembahasan:

- Diskusi tentang sistem seleksi pegawai di PT. Petrokimia Gresik.

## Hasil Pertemuan:

- Diketahui bahwa ada tiga bentuk seleksi pegawai di PT. Petrokimia Gresik, yaitu seleksi lulusan S1, lulusan D3, lulusan SMA/SMK. Terdapat juga program CSR yaitu seleksi Lolapil dengan pembinaan selama 6 bulan.
- Didapatkan detail dari seleksi lulusan D3 tahun 2014. Diantaranya adalah departemen yang akan dimasuki calon pegawai baru beserta kuotanya, yaitu departemen Akuntansi(3) ,Anggaran(1), Keuangan(9), Pemeliharaan I(3), Pemeliharaan II(6), Pemeliharaan III(4), Pengelolaan Pelabuhan(3), Prasarana Publik dan Kawasan(2), Produksi I(1), Produksi II A(1), dan Produksi II B(2). Kemudian didapat juga kualifikasi dalam tahap wawancara, yaitu Penampilan, Pengetahuan Khusus, Kemampuan Menyatakan Pendapat, Kematangan Kepribadian, Ambisi, Potensi untuk Berkembang, dan Kegiatan Organisasi.

## 3. Pertemuan 3 - Wawancara 2

## Pembahasan:

- Observasi data pegawai yang diterima pada seleksi lulusan D3 tahun 2014.

## LAMPIRAN B (LANJUTAN)

Hasil Pertemuan:

- Didapatkan data pegawai yang bersangkutan. Diantaranya adalah nilai kualifikasi dalam tahap wawancara, bidang yang diinginkan pegawai saat mendaftar, dan penerjemahan bidang oleh pembimbing ke dalam departemen yang sesuai.

### 4. Pertemuan 4 - Pengajuan Kuisisioner

Pembahasan:

- Pengajuan kuisisioner tentang tingkat kepentingan antar departemen dan antar kualifikasi.
- Penjelasan tentang tujuan dan maksud kuisisioner serta cara pengisian kuisisioner kepada pakar.

Hasil Pertemuan:

- Didapatkan tiga pakar dari departemen personalia yang bersedia untuk menjadi responden.

### 5. Pertemuan 5 - Pengambilan Kuisisioner

Pembahasan:

- Pengambilan kuisisioner dari pakar.

### 6. Pertemuan 6 - Validasi Pengisian Kuisisioner

Pembahasan:

- Melakukan validasi kepada pakar tentang pengisian kuisisioner yang telah dilakukan. Karena terdapat beberapa hasil yang tidak konsisten pada matriks perbandingan berpasangan.

Hasil Pertemuan:

- Revisi pengisian kuisisioner oleh pakar dengan tidak merubah kepentingan yang dibandingkan, namun hanya merubah pada nilai tingkat kepentingannya.

### 7. Pertemuan 7 - Analisa Kebutuhan User

Pembahasan:

- Analisa kebutuhan user atas perangkat lunak yang akan dibuat.

Hasil Pertemuan:

**LAMPIRAN B (LANJUTAN)**

- Didapatkan daftar awal kebutuhan user, diantaranya menampilkan data, dapat mengelola data nilai pegawai, diantaranya menambahkan data pegawai baru dan edit data pegawai, melakukan proses perhitungan sehingga diperoleh hasil penempatan pegawai, menampilkan hasil perhitungan, dan menampilkan hasil penempatan pegawai ke dalam bentuk yang mudah dipahami dan interaktif.
8. Pertemuan 8 - Pengujian Prototipe PL 1
- Pembahasan:
- Pengujian prototipe awal kepada user dan penjelasan atas prototipe yang telah dibuat.
- Hasil Pertemuan:
- Didapatkan evaluasi user atas prototipe yang telah di coba. Diantaranya adalah tidak adanya form login sebelum halaman awal ditampilkan, tidak terdapat tombol Help sebagai panduan user, hanya menampilkan data pegawai, dan terlalu banyak proses yang dilakukan user dalam menggunakan program penempatan pegawai.
9. Pertemuan 9 - Pengujian Prototipe PL 2
- Pembahasan:
- Pengujian kedua prototipe berdasarkan hasil revisi dan perbaikan dari evaluasi user pada pertemuan sebelumnya.
- Hasil Pertemuan:
- Didapatkan evaluasi user yang kedua, yaitu submenu Input Parameter, dan Start GA Process serta menu Result pada program penempatan pegawai kurang sinkron secara penggunaan.
10. Pertemuan 10 - Pengujian Prototipe PL 3
- Pembahasan:
- Hasil Pertemuan:
- Pengujian ketiga prototipe berdasarkan hasil revisi dan perbaikan dari evaluasi user pada pertemuan sebelumnya.
- Hasil Pertemuan:

## LAMPIRAN B (LANJUTAN)

- User telah mengatakan sesuai namun dengan revisi title pada tiap panel diperbaiki agar lebih mudah dipahami oleh user.

### 11. Pertemuan 11 - Laporan Penelitian.

#### Pembahasan:

- Data yang telah diambil ditunjukkan kepada pembimbing dan presentasi 80% hasil penelitian.

#### Hasil Pertemuan:

- Didapatkan surat keterangan selesai melakukan kegiatan penelitian tugas akhir sebagai berikut:



**LAMPIRAN B (LANJUTAN)****12. Pertemuan 12 - Validasi Hasil Penelitian Pembahasan:**

- Interaksi dengan pegawai yang ada didalam hasil penelitian untuk mendapatkan validitas hasil penelitian.

Hasil:

- Pegawai dengan ID P01 merespon dengan baik saat diskusi tentang penempatannya oleh sistem. Karena departemen yang dirokemendasikan oleh sistem tersebut jobdesc-nya serupa dengan penempatan oleh perusahaan dan sesuai dengan pilihan pegawai tersebut.
- Pegawai dengan ID P02 menginginkan tetap pada departemen Pemeliharaan II. Dengan alasan resiko kerja pada Pabrik I lebih tinggi karena *basic* kerjanya pada produksi pupuk nitrogen sehingga bahan pembuatan seperti ammonia mudah terbakar dan meledak. Sedangkan Pabrik II dan III *basic* kerjanya pada bahan sulfur dan fosfat. Alasan yang sama juga dikemukakan oleh pegawai dengan ID P06.
- Pegawai dengan ID P03 merespon dengan baik. Karena menurut pegawai ini lebih baik ditempatkan pada departemen PPK dari pada di tempatkan di Pabrik. Alasan yang sama juga dikemukakan oleh pegawai dengan ID P05.
- Pegawai dengan ID P04 merespon dengan baik. Karena bagi pegawai ini departemen Produksi ataupun Pemeliharaan sama. Disamping itu memang kedua departemen ini sesuai dengan bidang yang dipilih yaitu Instrumentasi dan Kontrol. Alasan yang sama juga dikemukakan oleh pegawai dengan ID P07 dan P09.

## DAFTAR PUSTAKA

- [1] Hasibuan, Malayu S.P. (2005). “Manajemen Sumber Daya Manusia”. Jakarta, Bumi Aksara.
- [2] Zhang, Yi. (2010). “Research on Human Resource Allocation Optimizatoin Based on Genetic Algorithm From the Perspective of Two-way Choice Model”. **International Conference on Educational and Information Technology**.
- [3] Pressman, R.S. (2001). “Software Engineering: A Practitioner’s Approach 5th Edition”. New York, McGraw-Hill.
- [4] Sastrohadiwiryo, Siswanto, B. (2002). “Manajemen Tenaga Kerja Indonesia”. Jakarta, Bumi Aksara.
- [5] Manullang, M. (2008). “Dasar-Dasar Manajemen”. Yogyakarta, Ghalia Indonesia
- [6] Saaty, T.L. (1990). “Multicriteria Decision Making: The Analytic Hierarchy process-Planing Priority Setting, Resource Allocation”. New York, McGraww-Hill.
- [7] Ishizaka, A., Lusti, M. (2006). “How to derive priorities in AHP: A comparative study”. **Central European Journal of Operations Research**, Springer.
- [8] Suyanto, (2011). “Artificial Intelligence: Searching–Reasoning-Planning-Learning”. Bandung, Informatika.
- [9] Connolly, T. & Begg, C. (2002). “Database System: A Practical Approach in Design, Implementation, and Management”. England, Addison Wesley.
- [10] Orbitko, Marek. (1998). “Introduction to Genetic Algorithm with Java Applets”. Czech, Hochschule fur Technik and Wirtschaft Dresden.



*“Halaman ini sengaja dikosongkan”*

## BIODATA PENULIS



Penulis memiliki nama Mohamad Muhtaromi. Biasa dipanggil dengan “Omi” atau “Romi”. Lahir di Kota Gresik pada tanggal 14 Desember 1991. Anak kedua dan terakhir dari pasangan Shohibul Ilmi dan Suci Hati Ilmiah ini telah menempuh pendidikan di TK Muslimat 05 Sukomulyo, MI Al-Ma’arif Sukomulyo, SMP N 1 Manyar, SMA N 1 Manyar, dan di Jurusan Matematika ITS. Selama

menjalani masa perkuliahan penulis aktif di beberapa organisasi mahasiswa, diantaranya adalah di BEM ITS sebagai staff Kementerian Dalam Negeri (DAGRI) dan di HIMATIKA ITS sebagai Kepala Departemen Pengembangan Sumber Daya Mahasiswa (PSDM) pada kepengurusan periode 2012/2013.

Penulis yang mempunyai hobi bermain gitar ini, bercita-cita ingin membahagiakan dan membanggakan orang tua serta pergi ke museum Sherlock Holmes di London. Selain itu penulis juga mempunyai hobi bermain Dota, nonton film, membaca novel dan buku komik, jalan-jalan dan sebagainya.

Untuk mendapatkan informasi atau memberikan saran dan kritik yang berhubungan dengan penulis ataupun Tugas Akhir ini, penulis dapat dihubungi melalui e-mail: [m.muhtaromi@gmail.com](mailto:m.muhtaromi@gmail.com). Semoga Bermanfaat.