

**IMPLEMENTASI METODE *THE DISTANCE TO THE IDEAL*
ALTERNATIVE PADA SISTEM INFORMASI
SELEKSI PEGAWAI DI PT. XYZ**

TUGAS AKHIR



ADDINA NURIYANTI RAHMI

1112001012

**PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNIK DAN ILMU KOMPUTER
UNIVERSITAS BAKRIE
JAKARTA
2016**

**IMPLEMENTASI METODE *THE DISTANCE TO THE IDEAL*
ALTERNATIVE PADA SISTEM INFORMASI
SELEKSI PEGAWAI DI PT. XYZ**

TUGAS AKHIR

**Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana
Komputer**



ADDINA NURIYANTI RAHMI

1112001012

**PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNIK DAN ILMU KOMPUTER
UNIVERSITAS BAKRIE
JAKARTA
2016**

HALAMAN PERNYATAAN ORISINALITAS

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

Nama : Addina Nuriyanti Rahmi

NIM : 1112001012

Tanda Tangan :

Tanggal : 13 September 2016

HALAMAN PENGESAHAN

Tugas Akhir ini diajukan oleh:

Nama : Addina Nuriyanti Rahmi
NIM : 1112001012
Program Studi : Informatika
Fakultas : Teknik dan Ilmu Komputer
Judul Skripsi : Implementasi Metode *The Distance To The Ideal Alternative* Pada Sistem Informasi Seleksi Pegawai di PT.XYZ

Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai persyaratan yang diperlukan untuk memperoleh gelar Sarjana Komputer (S.Kom) pada Program Studi Informatika Fakultas Teknik dan Ilmu Komputer, Universitas Bakrie.

DEWAN PENGUJI

Pembimbing : Yusuf Lestanto, S.T., M.Sc. ()

Penguji 1 : Boy Pasaribu, S. Kom, G.D.B.S., M.I.T., M.I.S. ()

Penguji 2 : Dr. Siti Rohajawati, S.Kom., M.Kom. ()

Ditetapkan di : Jakarta

Tanggal : 13 September 2016

UNGKAPAN TERIMA KASIH

Alhamdulillahirabbil'alamin, puji syukur kehadiran Allah SWT karena hanya berkat rahmat serta karunia-Nya penyusunan Tugas Akhir ini dapat diselesaikan. Penulisan Tugas Akhir ini dilakukan dalam rangka memenuhi salah satu syarat untuk mencapai gelar Sarjana Komputer Program Studi Informatika Fakultas Teknik dan Ilmu Komputer Universitas Bakrie.

Dalam proses penyelesaian Tugas Akhir ini terdapat banyak pihak yang bersedia meluangkan waktunya untuk membantu serta memberikan dukungan moril untuk penulis. Oleh karena itu, pada kesempatan ini penulis ingin mengungkapkan terima kasih dan penghargaan kepada :

1. Bapak Prof. Dr. Hoga Saragih, S.T., M.T., selaku Kepala Program Studi Informatika, atas segala saran serta dukungan yang diberikan dalam pelaksanaan penelitian ini;
2. Bapak Yusuf Lestanto, S.T., M.Sc., selaku dosen pembimbing, atas segala waktu yang diluangkan serta kesabaran dan pengertian dalam memberikan bimbingan dan arahan untuk penulis. Terima kasih karena telah memberikan kesempatan dan membimbing saya dengan baik sehingga mampu menyelesaikan tugas akhir ini.
3. Ibu Dr. Siti Rohajawati, S.Kom., M.Kom. dan Bapak Boy Pasaribu, S.Kom, G.D.B.S., M.I.T., M.I.S., selaku dosen penguji, atas segala saran dan koreksi yang membangun untuk membantu meningkatkan kualitas laporan tugas akhir ini. Terima kasih atas ilmu yang diberikan.
4. Kedua orang tua, Mila Rusmila dan Rahmat Danafi, yang begitu pengertian dan sabar menunggu. Atas perhatian dan kasih sayang, kerja keras dan keringat, dukungan, dan doa yang telah mengantarkan penulis hingga titik ini. Terima kasih karena telah percaya. Pencapaian ini dipersembahkan untuk kalian.
5. Kedua adik, Ama dan Syaka, telah menemani, menjadi teman pengisi hari dengan penuh keseruan, riang tawa, dan canda. Untuk Mbah Tati dan Te Nanah yang meminjamkan “ruang kerja” untuk penggarapan tugas akhir

penulis. Atas segala perhatian, dukungan, dan dorongan dari keluarga besar, terima kasih.

6. Ayyu Andhysa, Syafira Puji Virginia, Ana Ainul Syamsi, Atikah Chairunisa, Khairunnisah, atas waktu, kebersamaan, canda tawa, bantuan, dukungan, dan doa yang diberikan selama ini. Terima kasih atas kehadiran kalian.
7. Teman-teman TIF seperjuangan yang telah membantu penulis dengan memberikan dukungan moril, doa, dan informasi untuk penyelesaian tugas akhir ini. Terima kasih atas kebaikan kalian selama lebih dari 4 tahun ini.
8. Teman sepermainan, Liana, Karnelia, Audia, Puti, Daniyah, Santi, Nana, Iqbal, Irham, Raga, Kak Dian. Terima kasih atas keberadaan kalian, atas waktu, kebersamaan, kebaikan, dukungan, canda tawa, dan doa yang kalian berikan selama ini.
9. Seluruh pihak yang telah memberikan dukungan dalam penyelesaian tugas akhir ini yang tidak dapat penulis sebutkan satu-persatu.

Akhir kata, penulis berharap semoga Allah SWT melimpahkan berkah dan nikmat-Nya serta membalas kebaikan semua pihak yang telah membantu penulis dalam penyusunan tugas akhir ini. Semoga tugas akhir ini dapat membantu dan berguna bagi orang yang membacanya.

Jakarta, 13 September 2016

Addina Nuriyanti Rahmi

HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI

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

Nama : Addina Nuriyanti Rahmi
NIM : 1112001012
Program Studi : Informatika
Fakultas : Teknik Dan Ilmu Komputer
Jenis Tugas Akhir : Implementasi

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

**Implementasi Metode *The Distance to The Ideal Alternative* Pada Sistem
Informasi Seleksi Pegawai di PT. XYZ**

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

Demikian pernyataan ini saya buat sebenarnya.

Dibuat di : Jakarta
Pada tanggal : 13 September 2016

Yang Menyatakan

(Addina Nuriyanti Rahmi)

IMPLEMENTASI METODE *THE DISTANCE TO THE IDEAL ALTERNATIVE* PADA SISTEM INFORMASI SELEKSI PEGAWAI DI PT. XYZ

Addina Nuriyanti Rahmi

ABSTRAK

Rekrutmen merupakan sebuah proses penting yang dilakukan perusahaan untuk melakukan pencarian serta pengadaan Sumber Daya Manusia (SDM). Dalam proses ini terdapat beberapa situasi yang membutuhkan pengambilan keputusan, contohnya saat rekrutmen. Pada tahap awal, pengambilan keputusan dilakukan saat penyortiran *Curriculum Vitae* (CV) dengan membandingkan kualifikasi pelamar dengan spesifikasi posisi. *The Distance to The Ideal Alternative* (DiA) merupakan salah satu metode penunjang keputusan. Metode ini digunakan untuk memecahkan permasalahan pemilihan berdasarkan banyak kriteria dan banyak alternatif seperti yang ditemukan dalam proses rekrutmen. Berdasarkan wawancara yang telah dilakukan di PT. XYZ, diketahui bahwa masih terdapat pegawai dengan ketidaksesuaian antara kualifikasi diri dan spesifikasi kebutuhan dari pekerjaan yang dilakukannya. Penelitian ini bertujuan untuk mengimplementasi metode DiA pada sistem informasi seleksi pegawai. DiA akan melakukan penilaian pada pelamar berdasarkan beberapa kriteria yang digunakan dalam penyaringan lamaran. Hasil dari penelitian ini berupa sistem informasi seleksi pegawai. Sistem dirancang menggunakan UML dengan pengembangan berbasis *Object-oriented*. Sistem diuji menggunakan metode pengujian *White Box* dan *Black Box*. Metode DiA berhasil diterapkan ke dalam sistem dan dapat memberikan hasil penilaian dari masing-masing pelamar. Hasil penilaian tersebut dapat membantu staf SDM untuk membuat keputusan pemilihan pelamar yang lolos ke tahap selanjutnya. Penentuan tingkat kepentingan dari masing-masing kriteria sangatlah penting untuk mendapatkan rekomendasi pelamar yang sesuai kebutuhan.

Kata Kunci: Rekrutmen, DiA, MADM, Sistem Penunjang Keputusan, Penyaringan Lamaran

**THE DISTANCE TO THE IDEAL ALTERNATIVE METHOD
IMPLEMENTATION ON EMPLOYEE SELECTION
INFORMATION SYSTEM IN PT. XYZ**

Addina Nuriyanti Rahmi

ABSTRACT

Recruitment is an important process enterprises do for searching and procuring Human Resources (HR). In this process there are some situations that need decision-making, such as when doing recruitment. In the initial step, a decision-making happens when sorting Curriculum Vitae (CV) by comparing the applicant's qualification with the job specification. The Distance to The Ideal Alternative (DiA) is one of decision-support methods. This method is used for solving problems with multi-criteria and multi-attribute characteristics which commonly found in recruitment process. Based on interview in PT.XYZ, it's known that there are still some employees with mismatch between their personal qualification and the job specification. This research intends to implement DiA method into an Employee Selection Information System. DiA will score the applicants based on some criterion that used in the application screening. Result of the Research is an Employee Selection Information System. The system designed using UML and Object-Oriented-based. White Box and Black Box testing method used to test the system. DiA method has been applied successfully and able to provide the scoring result of each applicant. The scoring result may help HR staff to establish a decision when choosing the qualified applicant for the next step. Determination of each criterion's priority acts as an important key in generating the best set of suitable applicant recommendation.

Keywords: *Recruitment, DiA, MADM, Decision Support System, Application Screening*

DAFTAR ISI

HALAMAN PERNYATAAN ORISINALITAS	iii
HALAMAN PENGESAHAN.....	iv
UNGKAPAN TERIMA KASIH	v
HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI	vii
ABSTRAK	viii
ABSTRACT	ix
DAFTAR ISI.....	x
DAFTAR GAMBAR.....	xii
DAFTAR TABEL	xiii
DAFTAR RUMUS	xiv
DAFTAR LAMPIRAN.....	xv
DAFTAR SINGKATAN.....	xvi
BAB 1 PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Identifikasi Masalah.....	4
1.3. Rumusan Masalah	5
1.4. Batasan Masalah	5
1.5. Tujuan Penelitian	5
1.6. Manfaat Penelitian	5
1.7. Sistematika Penulisan	6
BAB 2 TINJAUAN PUSTAKA.....	7
2.1. Penelitian Terdahulu	7
2.2. Perencanaan Sumber Daya Manusia.....	9
2.3. Rekrutmen.....	11
2.4. Metode <i>Distance to The Ideal Alternative</i> (DiA).....	13
2.5. <i>Rapid Application Development</i> (RAD)	18
BAB 3 METODOLOGI PENELITIAN.....	20

3.1	Kerangka Kerja Penelitian	20
3.2	Metode Pengumpulan Data.....	20
3.2.1	Studi Literatur	21
3.2.2	Wawancara.....	21
3.3	Metode Pengembangan Sistem dan Penyusunan Laporan.....	21
3.3.1	Perencanaan Kebutuhan.....	22
3.3.2	Perancangan Sistem	25
3.3.3	Pembangunan Sistem	26
3.3.4	Implementasi.....	26
3.3.5	Penyusunan Laporan	26
BAB 4 IMPLEMENTASI DAN PEMBAHASAN		27
4.1	Perancangan Sistem	27
4.1.1	<i>Use Case Diagram</i>	27
4.1.2	<i>Activity Diagram</i>	29
4.1.3	<i>Class Diagram</i>	31
4.1.4	<i>Data Model</i>	32
4.2	Implementasi	32
4.2.1	Implementasi Metode DiA.....	32
4.2.2	Implementasi Antarmuka.....	47
4.3	Pengujian Algoritma	53
4.3.1	Pengujian <i>White Box</i>	54
4.3.2	Pengujian <i>Black Box</i>	60
BAB 5 SIMPULAN DAN SARAN.....		61
5.1	Simpulan	61
5.2	Saran	62
DAFTAR PUSTAKA.....		63

DAFTAR GAMBAR

Gambar 2.1 Proses Perencanaan SDM	10
Gambar 2.2 Fase RAD	19
Gambar 3.1 Alur Pelaksanaan Penelitian.....	20
Gambar 4.1 Diagram <i>Use Case</i>	27
Gambar 4.2 Diagram Aktivitas Proses Peningkatan Pelamar.....	30
Gambar 4.3 <i>Class Diagram</i> Sistem	31
Gambar 4.4 Urutan Langkah Pengguna Untuk Melakukan Seleksi	33
Gambar 4.5 Kode Program Matriks Keputusan X.....	35
Gambar 4.6 Kode Program Matriks R	37
Gambar 4.7 Kode Program Matriks V	39
Gambar 4.8 Kode Program Solusi Ideal Positif dan Negatif	41
Gambar 4.9 Kode Program Jarak Manhattan dan PIA.....	43
Gambar 4.10 Kode Program Perhitungan RI.....	46
Gambar 4.11 Halaman Login.....	47
Gambar 4.12 Halaman Lihat Daftar Posisi	47
Gambar 4.13 <i>Pop-Up</i> Hapus Posisi	48
Gambar 4.14 Halaman Lihat <i>Job Title</i>	48
Gambar 4.15 Halaman Tambah Lowongan Pekerjaan	49
Gambar 4.16 Halaman Tambah Pelamar	50
Gambar 4.17 Halaman Awal Seleksi Resume	51
Gambar 4.18 Halaman Pemilihan Tingkat Prioritas Kriteria.....	51
Gambar 4.19 Halaman Ubah Kriteria	51
Gambar 4.20 Halaman Melihat Daftar Data Pelamar Seleksi	52
Gambar 4.21 Halaman Lihat Hasil Seleksi	53
Gambar 4.22 Tampilan <i>Pop-Up</i> Menu <i>Sign Out</i>	53

DAFTAR TABEL

Tabel 2.1 Tabel Representasi Masalah MADM.....	14
Tabel 2.2 Tabel Perbandingan Metode MADM	16
Tabel 3.1 Nilai Jenis Kelamin.....	23
Tabel 3.2 Nilai Usia	24
Tabel 3.3 Nilai Pendidikan.....	24
Tabel 3.4 Nilai Kemampuan	25
Tabel 3.5 Bobot Kriteria	25
Tabel 4.1 Deskripsi <i>Use Case</i> Modul Rekrutmen	28
Tabel 4.2 Data Kualifikasi Alternatif.....	33
Tabel 4.3 Tabel Perbandingan Kualifikasi Pelamar dan Posisi	34
Tabel 4.4 Data Kualifikasi Alternatif Setelah Perbandingan dan Konversi	35
Tabel 4.5 Hasil Perhitungan Normalisasi Matriks X	36
Tabel 4.6 Hasil Perhitungan Matriks V Terbobot.....	38
Tabel 4.7 Solusi Ideal Positif dan Negatif	40
Tabel 4.8 Jarak Manhattan	42
Tabel 4.9 Jarak Alternatif ke PIA	44
Tabel 4.10 Data Kualifikasi Pelamar Setelah Pemeringkatan	45
Tabel 4.11 Hasil Pengujian Fungsi matriksX().....	54
Tabel 4.12 Hasil Pengujian Fungsi matriksR()	55
Tabel 4.13 Hasil Pengujian Fungsi matriksV().....	56
Tabel 4.14 Hasil Pengujian Fungsi solusiIdeal()	57
Tabel 4.15 Hasil Pengujian Fungsi <i>manhattan</i> ()	58
Tabel 4.16 Hasil Pengujian Fungsi jarakPIA()	59

DAFTAR RUMUS

Rumus 2.1 Matriks Keputusan X	14
Rumus 2.2 Matriks Keputusan R	14
Rumus 2.3 Jarak <i>Euclidean</i>	14
Rumus 2.4 Matriks Keputusan V	15
Rumus 2.5 Elemen Matriks V	15
Rumus 2.6 Solusi Ideal Positif (<i>benefit</i>)	15
Rumus 2.7 Solusi Ideal Negatif (<i>benefit</i>)	15
Rumus 2.8 Solusi Ideal Positif (<i>cost</i>)	15
Rumus 2.9 Solusi Ideal Negatif (<i>cost</i>)	15
Rumus 2.10 Jarak <i>Manhattan</i> Positif	16
Rumus 2.11 Jarak <i>Manhattan</i> Negatif	16
Rumus 2.12 <i>Positive Ideal Alternative</i> (PIA)	16
Rumus 2.13 Jarak Alternatif ke PIA	16

DAFTAR LAMPIRAN

Lampiran 1. Wawancara	65
Lampiran 2. <i>Software Requirement Specification</i> (SRS)	70
Lampiran 3. Hasil Pengujian <i>Black Box</i>	96

DAFTAR SINGKATAN

SDM	Sumber Daya Manusia
RAD	<i>Rapid Application Development</i>
KSA	<i>Knowledge, Skills, dan Abilities</i>
CV	<i>Curriculum Vitae</i>
DiA	<i>The Distance To The Ideal Alternative</i>
WP	<i>Weighted Product</i>
SAW	<i>Simple Additive Weighting</i>
TOPSIS	<i>Technique for Order Preference By Similarity To Ideal Solution</i>
MADM	<i>Multi-Attribute Decision Making</i>
HR	<i>Human Resource</i>
MPP	<i>Manpower Planning</i>
PIA	<i>Positive Ideal Alternative</i>

BAB 1 PENDAHULUAN

1.1. Latar Belakang

Pegawai, sebagai aset Sumber Daya Manusia (SDM) perusahaan, merupakan salah satu elemen krusial yang menjaga berjalannya proses bisnis perusahaan. Tanpa adanya pegawai, perusahaan belum mampu melaksanakan proses bisnis dengan baik karena ia laksana roda yang menggerakkan perusahaan. Selain sebagai penggerak perusahaan, aset SDM dapat menciptakan kompetensi inti (*core competencies*) bagi perusahaan yang membedakan dirinya dengan pesaing lainnya. Beberapa contohnya yang dipengaruhi oleh keberhasilan SDM ialah produktivitas, kualitas/pelayanan, kemampuan pegawai, serta operasi inovatif (Mathis & Jack, 2010).

Rekrutmen merupakan sebuah proses penting yang dilakukan perusahaan untuk melakukan pencarian serta pengadaan SDM. Dalam proses ini terdapat beberapa situasi yang membutuhkan pengambilan keputusan. Salah satu situasi yang melibatkan pengambilan keputusan saat rekrutmen ialah ketika melakukan seleksi pelamar. Ketika sebuah perusahaan mengumumkan pembukaan rekrutmen, sejumlah pelamar akan mengirimkan aplikasi lamaran terhadap posisi tertentu. Biasanya pelamar akan menyertakan *Curriculum Vitae* (CV) / data diri mereka pada aplikasi lamaran. Dibutuhkan pengambilan keputusan untuk menentukan siapa pelamar yang akan dipanggil untuk melanjutkan ke tahap seleksi yang lebih ketat. Pada tahap awal, pengambilan keputusan dilakukan saat penyortiran CV dengan membandingkan kualifikasi pelamar dengan spesifikasi posisi. Proses ini dilakukan dengan mempertimbangkan bahwa untuk dapat sepenuhnya mampu mendukung perusahaan, dibutuhkan SDM dengan kualitas baik dan talenta untuk mencapai tujuan perusahaan (Armstrong & Taylor, 2014). Maka dari itu, proses pencarian dan penyeleksian bibit-bibit penggerak perusahaan ini sebaiknya dilakukan dengan perencanaan serta eksekusi yang baik agar perusahaan mendapatkan orang yang tepat.

Selain untuk mendapatkan orang yang tepat, perencanaan dalam rekrutmen juga dapat memperkecil kemungkinan terjadinya kegagalan rekrutmen. Kegagalan

dalam proses rekrutmen dapat mengakibatkan masalah serius bagi perusahaan (Chack, 2013). Salah satu akibat gagalnya proses rekrutmen membuat biaya yang telah dikeluarkan oleh perusahaan untuk penyelenggaraan rekrutmen menjadi sia-sia. Perusahaan juga gagal memenuhi kebutuhan atas pegawai yang mungkin sangat mereka perlukan saat itu. Hal tersebut dapat berimbas pada penurunan produktivitas perusahaan yang akan menghambat kesuksesan bisnis (Bulmash, Chhinzer, & Speers, 2010). Selain itu, kegagalan rekrutmen juga dapat menyebabkan kesalahan sasaran pemilihan kandidat pegawai sehingga perusahaan gagal mendapatkan kandidat pegawai yang berkualifikasi sesuai standar. Tentu saja hal ini bukanlah sesuatu yang diharapkan perusahaan karena ketidakcocokan antara KSAs pekerja dengan kualifikasi pekerjaan dapat berujung pada *turnover* pegawai. *Turnover* merupakan istilah yang merujuk pada tingkat pegawai yang berhenti dan meninggalkan perusahaan (Armstrong & Taylor, 2014) (Bulmash, Chhinzer, & Speers, 2010).

Perencanaan SDM dapat menjadi solusi dalam membantu perusahaan merencanakan rekrutmen dengan baik. Perencanaan SDM juga membantu perusahaan mendapatkan SDM yang “tepat”. Tepat yang dimaksud di sini adalah jumlah orang yang tepat, dengan kemampuan yang tepat, pada waktu dan tempat yang tepat. Kegiatan ini mencakup analisa dan identifikasi ketersediaan serta kebutuhan akan SDM dalam perusahaan untuk jangka waktu panjang di masa yang akan datang (Mathis & Jack, 2010). Oleh karena itu, perencanaan SDM sangat erat hubungannya dengan rekrutmen, seleksi, serta pemeliharaan keahlian pegawai. Apabila di kemudian hari perusahaan menghadapi situasi darurat yang berhubungan dengan pengambilan keputusan terkait SDM, perusahaan dapat menghadapinya secara sigap dengan berpatokan kepada perencanaan SDM yang telah dibuat sebelumnya.

PT. XYZ merupakan sebuah perusahaan yang bergerak di bidang pembiayaan konsumen. Perusahaan ini telah memiliki sekitar 4700 pegawai bekerja pada kurang lebih 160 cabang yang tersebar di seluruh Indonesia (Lampiran 1). Berdasarkan wawancara yang dilakukan dengan Manajer SDM di perusahaan, diketahui bahwa masih terdapat pegawai dengan ketidaksesuaian antara kualifikasi diri dan spesifikasi kebutuhan dari pekerjaan yang

dilakukannya. Hal ini dikenal dengan deviasi. Sesungguhnya, deviasi merupakan hal yang sebisa mungkin dihindari oleh perusahaan dalam mempekerjakan pegawai (Armstrong & Taylor, 2014). Namun, pada kenyataannya deviasi masih ditemui dalam perusahaan. Contoh kasus deviasi adalah ketika spesifikasi kebutuhan dari posisi IT *Programmer* mengharuskan lulusan S1, sedangkan pegawai yang dipekerjakan masih berstatus lulusan SMA. Hal ini menandakan bahwa perusahaan masih kesulitan untuk mendapatkan orang yang tepat dalam proses rekrutmennya.

Perencanaan SDM yang tepat akan menghasilkan proses rekrutmen yang sukses. Dalam kasus ini, perusahaan mungkin mampu menyusun perencanaan SDM di atas kertas. Namun, untuk penerapannya terutama pada proses rekrutmen belum begitu berhasil. Hal ini ditunjukkan dengan adanya deviasi. Sebenarnya perusahaan memiliki pilihan untuk menanggulangi deviasi dengan melaksanakan program pengembangan karyawan. Tetapi, apabila perekrutan kandidat dengan *Knowledge, Skills, and Abilities* (KSA) di bawah kualifikasi pekerjaan terus terjadi, investasi waktu serta biaya yang disisihkan perusahaan untuk pengembangan karyawan akan membesar akibat akumulasi. Pada akhirnya, hasil dari pilihan tersebut hanya menyebabkan kerugian bagi perusahaan.

Di awal telah dijelaskan sebelumnya bahwa dibutuhkan SDM yang berkualitas sesuai dengan kebutuhan perusahaan untuk meningkatkan performa perusahaan. Hal ini membuat proses penyortiran CV menjadi sama pentingnya dengan proses lain dalam rekrutmen. Proses pengambilan keputusan dalam penyortiran CV akan menentukan daftar awal pelamar dengan kualifikasi yang sesuai terhadap spesifikasi kebutuhan pekerjaan dan mengeliminasi pelamar tidak sesuai kualifikasi (deviasi) (Information and Communications Technology Council, 2012). Hingga saat ini, proses penyortiran CV yang dilakukan perusahaan masih manual, yaitu dengan mengandalkan pengamatan mata untuk membandingkan kualifikasi pelamar dengan kualifikasi yang dibutuhkan posisi tersebut. Proses ini membutuhkan waktu maksimal 5 menit untuk membandingkan satu CV. Sedangkan jumlah CV yang diterima perusahaan dalam proses rekrutmen dapat mencapai rata-rata 250 berkas per bulan (Lampiran 1).

Dalam penelitian yang dilakukan oleh (Chack, 2013) menyebutkan bahwa sistem penunjang keputusan memiliki fungsi penting dalam perusahaan. Penerapan sistem ini akan meningkatkan daya guna serta budaya perusahaan. *The Distance To The Ideal Alternative* (DiA) merupakan salah satu pendekatan yang dapat diterapkan dalam sistem penunjang keputusan. Pendekatan ini telah terbukti memiliki performa yang lebih baik dari pada *Simple Additive Weighting* (SAW) (Purnamasari, 2015). Penerapan metode ini bisa menjadi jalan keluar bagi perusahaan dalam mendukung pembuatan keputusan pemilihan pelamar pada tahap penyortiran CV yang sesuai dengan spesifikasi pekerjaan. DiA akan melakukan penilaian pada pelamar berdasarkan beberapa kriteria yang digunakan dalam penyaringan lamaran. Kemudian, metode tersebut akan memberikan hasil penilaian dari masing-masing pelamar. Hasil penilaian ini dapat membantu staf SDM untuk membuat keputusan pemilihan pelamar yang lolos ke tahap selanjutnya, yaitu wawancara.

Berdasarkan penjabaran masalah di atas, diperlukan adanya sebuah sistem yang mampu menilai kecocokan kualifikasi pelamar dalam proses penyaringan lamaran. Oleh karena itu, penelitian Implementasi Metode *The Distance To The Ideal Alternative* Pada Sistem Informasi Seleksi Pegawai di PT. XYZ dilakukan.

1.2. Identifikasi Masalah

Dari uraian pada latar belakang dapat diidentifikasi beberapa masalah yang dihadapi oleh perusahaan, antara lain adalah :

- 1) Perusahaan mempekerjakan pegawai dengan kualifikasi yang tidak sesuai dengan spesifikasi pekerjaan yang telah ditentukan.
- 2) Proses penyaringan lamaran pada perusahaan masih dilakukan secara manual.
- 3) Perusahaan mengalami kesulitan memantau kesesuaian antara kualifikasi pelamar dengan kualifikasi standar pada posisi tertentu selama proses penyaringan lamaran.

1.3. Rumusan Masalah

Penelitian ini dilakukan dengan berpatokan pada pertanyaan sebagai berikut:

- 1) Bagaimana implementasi metode DiA dalam modul rekrutmen untuk memberikan rekomendasi pelamar pada tahap seleksi administratif?
- 2) Bagaimana implementasi modul rekrutmen sistem informasi seleksi pegawai dengan menerapkan model *Rapid Application Development* (RAD)?

1.4. Batasan Masalah

Adapun batasan masalah dalam pelaksanaan penelitian ini ialah :

- 1) Penyeleksian lamaran calon pegawai dengan metode DiA dilakukan berdasarkan pada kriteria umur, tingkat pendidikan, jenis kelamin, pengalaman kerja, kemampuan, dan gaji yang diharapkan.
- 2) Pengalaman kerja dan kemampuan yang dimasukkan ke dalam sistem hanya yang memiliki kaitan dengan posisi yang dilamar.
- 3) Penelitian ini dibatasi hingga tahap pengujian aplikasi.

1.5. Tujuan Penelitian

Adapun tujuan dilaksanakannya penelitian ini ialah :

- 1) Mengimplementasi metode DiA ke dalam sistem informasi seleksi pegawai untuk melakukan penilaian dan pemeringkatan kualifikasi pelamar.
- 2) Mengimplementasi modul rekrutmen ke dalam sistem informasi seleksi pegawai dengan menerapkan model RAD.

1.6. Manfaat Penelitian

Adapun manfaat dari dilaksanakannya penelitian ini ialah :

- 1) Mempermudah akses atas informasi yang dibutuhkan dalam rekrutmen melalui basis data terpusat pada sistem informasi seleksi pegawai.

- 2) Mempermudah pengelolaan informasi terkait rekrutmen, yaitu penyimpanan dan pencarian data lowongan pekerjaan, pelamar, serta aplikasi lamaran yang diajukan ke perusahaan.
- 3) Membantu pengambilan keputusan dalam proses penyaringan lamaran calon pegawai melalui implementasi metode DiA.
- 4) Penelitian ini dapat menjadi referensi bagi penulis lain yang ingin mengimplementasikan metode DiA ke dalam sistemnya.
- 5) Bagi penulis, penelitian ini menambah pengetahuan, pengalaman, serta mengasah kemampuan analisis dalam pelaksanaan penelitian hingga penyusunan laporan.

1.7. Sistematika Penulisan

Sistematika penulisan penelitian ini dibagi dalam 5 (lima) bab. Berikut adalah penjabaran dari sistematika penulisan dalam penelitian ini :

BAB I PENDAHULUAN

Latar belakang, rumusan masalah, batasan masalah, tujuan, dan manfaat dalam penelitian ini akan dijelaskan pada bab ini.

BAB II TINJAUAN PUSTAKA

Landasan teori membahas beberapa dasar-dasar teori yang mendukung penulisan skripsi ini. Penelitian terdahulu menjelaskan beberapa penelitian yang terkait dengan rekrutmen, rancang bangun sistem, dan metode DiA.

BAB III METODOLOGI PENELITIAN

Bab ini menjelaskan daftar tahapan dalam alur pelaksanaan penelitian.

BAB IV IMPLEMENTASI DAN PEMBAHASAN

Pada bab ini akan membahas penerapan model *RAD* yang digunakan dalam pengembangan sistem serta penjabaran mengenai penerapan metode DiA untuk memberikan rekomendasi pelamar pada tahap seleksi administratif penyaringan pelamar.

BAB V SIMPULAN DAN SARAN

Simpulan menjelaskan hasil dari penelitian. Saran, sesuai dengan maknanya, berisi beberapa saran yang dapat diterapkan untuk penelitian selanjutnya.

BAB 2 TINJAUAN PUSTAKA

2.1. Penelitian Terdahulu

Penelitian ini dilaksanakan dengan tujuan untuk mengimplementasi metode DiA dalam sistem informasi seleksi pegawai, khususnya modul rekrutmen. Beberapa penelitian yang bersinggungan dengan topik ini telah dilakukan, di antaranya ialah sebagai berikut :

- a) Penelitian berjudul “Penentuan Penerima Beasiswa Dengan Metode *Simple Additive Weighting* (SAW) dan Metode *The Distance To The Ideal Alternative* (DiA)” oleh Rachmah Ayu Purnamasari. Penelitian ini dimaksudkan untuk menjawab permasalahan terkait penentuan penerima beasiswa dengan mengimplementasi dua buah metode, yaitu SAW dan DiA. Penilaian penerima beasiswa dilakukan terhadap 15 mahasiswa dengan menilai 3 kriteria. Kriteria yang digunakan dalam penilaian ialah nilai IPK, jumlah penghasilan orang tua, dan jumlah saudara kandung. Pengujian dilakukan untuk mengetahui metode yang lebih sesuai dalam kasus ini. Sebuah program dibangun dengan menggunakan *software* MATLAB untuk mempermudah proses pengujian. Pengujian yang disebut uji sensitivitas menunjukkan bahwa metode DiA lebih efektif daripada metode SAW (Purnamasari, 2015).
- b) Penelitian berjudul “Uji Sensitivitas Metode WP, SAW, dan TOPSIS Dalam Menentukan Titik Lokasi *Repeater Internet Wireless*” oleh David Ahmad Effendy dan Rony Heri Irawan. Penelitian ini mengangkat masalah seringnya terjadi pendirian *repeater* pada lokasi yang kurang tepat oleh penyedia jasa jaringan internet. Penelitian mengusung tiga buah metode untuk memecahkan masalah pemilihan lokasi yang tepat untuk pendirian jaringan internet, yaitu *Weighted Product* (WP), SAW, dan *Technique for Order Preference By Similarity To Ideal Solution* (TOPSIS). Penilaian dilakukan berdasarkan 6 kriteria, yaitu jarak dengan pemancar, tingkat halangan, kepadatan pemukiman penduduk, perizinan pendirian *repeater*, kebutuhan internet, gangguan sinyal dan keamanan. Metode tersebut melewati uji sensitivitas

untuk melihat metode mana yang paling cocok. Hasil dari penelitian adalah metode WP, SAW, dan TOPSIS dapat diimplementasikan dengan cepat dan akurat. Hasil pengujian menunjukkan bahwa metode yang paling relevan dengan kasus ini ialah SAW dengan nilai perubahan terbesar dibandingkan dengan metode WP dan TOPSIS. Nilai perubahan SAW, TOPSIS, dan WP berturut-turut adalah 9%, 0.38%, dan 0,10% (Effendy & Irawan, 2015).

- c) Penelitian berjudul “Sistem Pendukung Keputusan Rekrutmen Karyawan Produksi Menggunakan Metode *Weighted Product* Pada PT. Ploss Asia Semarang” oleh Ardi Kusumaning Diah R. Penelitian dilatarbelakangi oleh pengambilan keputusan yang sering kali dipengaruhi oleh subjektivitas. Oleh karena itu, penelitian dilakukan bertujuan untuk membuat sebuah sistem dengan implementasi metode WP untuk mempermudah pemilihan karyawan sesuai dengan kriteria. Kriteria yang digunakan untuk penilaian terdiri dari penampilan (fisik dan busana), kepribadian, pengalaman kerja, motivasi kerja, ambisi dalam pekerjaan, kemampuan kerja sama, kemampuan berkomunikasi, dan potensi untuk berkembang. Metode pengembangan sistem menerapkan model *waterfall* dengan bahasa pemrograman *Microsoft Visual Basic 6.0* dan *MySQL* sebagai basis data. Sistem berbasis *desktop* tersebut mampu memberikan rekomendasi dan pertimbangan dalam memutuskan penerimaan kandidat. Metode WP berhasil di implementasi (Diah R., 2013).

Berdasarkan uraian di atas, dapat dirangkum beberapa perbedaan antara penelitian ini dengan penelitian terdahulu. Perbedaan yang ada dengan penelitian oleh (Purnamasari, 2015) ialah dari segi bidang penerapannya. Penelitian yang akan dilakukan membahas penerapan metode DiA pada penyaringan lamaran di proses rekrutmen dalam perusahaan, sedangkan (Purnamasari, 2015) menerapkan metode DiA dan SAW pada penentuan penerima beasiswa. Dengan perbedaan pada bidang penerapannya, otomatis kriteria yang digunakan oleh kedua penelitian juga berbeda. Kemudian, sistem yang dibangun oleh (Purnamasari, 2015) berbasis *desktop*, sedangkan sistem dari penelitian ini berbasis *web*.

Penelitian yang dilakukan oleh (Effendy & Irawan, 2015) membahas penerapan metode SAW, WP, dan TOPSIS untuk menentukan lokasi pendirian *repeater wireless*. Di lain sisi, penelitian yang akan dilakukan menerapkan metode DiA dalam proses rekrutmen. Kemudian, metode dan kriteria yang digunakan dalam penilaian juga berbeda.

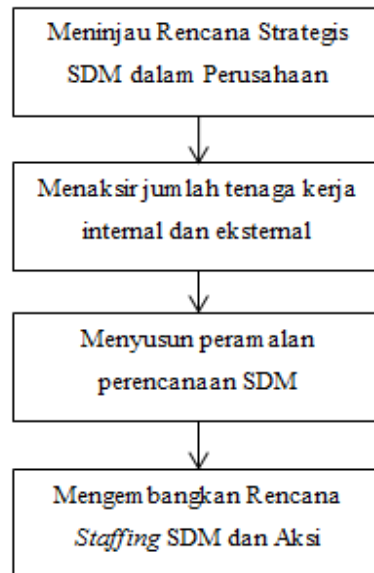
Penelitian yang dilakukan oleh (Diah R., 2013) membahas penerapan metode WP untuk penerimaan pegawai. Bidang penerapan penelitian tersebut sama dengan penelitian ini, namun metode yang digunakan berbeda. Penelitian ini menerapkan metode DiA untuk melakukan penilaian. Kemudian kriteria yang digunakan untuk penilaian juga memiliki perbedaan. Kriteria yang digunakan oleh (Diah R., 2013) terdiri dari penampilan (fisik dan busana), kepribadian, pengalaman kerja, motivasi kerja, ambisi dalam pekerjaan, kemampuan kerja sama, kemampuan berkomunikasi, dan potensi untuk berkembang. Berbeda dengan penelitian oleh (Diah R., 2013), penelitian yang akan dilakukan menggunakan kriteria umur, tingkat pendidikan, jenis kelamin, pengalaman kerja, kemampuan, dan gaji yang diharapkan.

2.2. Perencanaan Sumber Daya Manusia

Sebagai salah satu kegiatan inti dalam manajemen SDM, perencanaan SDM merupakan sebuah proses untuk menganalisis dan mengidentifikasi kebutuhan serta ketersediaan sumber daya manusia dalam sebuah perusahaan. Proses ini dilakukan dengan tujuan untuk mendapatkan tenaga kerja dengan jumlah, kemampuan, serta pada waktu dan tempat yang tepat (Armstrong & Taylor, 2014) (Mathis & Jack, 2010). Maka dari itu, Perumusan perencanaan SDM akan membantu perusahaan untuk menjawab pertanyaan seperti berapa banyak, kemampuan apa saja, kapan, dan di mana, terkait SDM dan kebutuhan serta ketersediaannya dalam perusahaan.

a) Meninjau Rencana Strategis SDM Perusahaan. Dalam pelaksanaannya, proses perencanaan SDM melalui beberapa tahapan dapat dilihat pada Gambar 2.1. Pada gambar tersebut terlihat bahwa proses perencanaan SDM dimulai dengan melakukan peninjauan terhadap rencana strategis perusahaan.

Tujuan dari rencana strategis perusahaan merupakan hal utama yang mendasari serta membentuk perencanaan SDM (Armstrong & Taylor, 2014).



Gambar 2.1 Proses Perencanaan SDM (Mathis & Jack, 2010)

- b) Menaksir Jumlah Tenaga Kerja Internal dan Eksternal.** Setelah peninjauan rencana strategis perusahaan dilakukan, penaksiran jumlah tenaga kerja internal dan eksternal akan dilaksanakan. Kegiatan ini mencakup identifikasi kondisi SDM internal dan eksternal perusahaan. Ada beberapa hal yang perlu ditinjau ketika akan melakukan identifikasi kondisi SDM internal, beberapa di antaranya ialah daftar pekerjaan yang ada saat itu, jumlah individu yang bertugas melaksanakan pekerjaan tersebut, dan KSA yang dibutuhkan untuk menyelesaikan pekerjaan tersebut. Sedangkan untuk mengidentifikasi kondisi SDM eksternal perusahaan perlu diperhatikan beberapa hal seperti faktor ekonomi, politik, dan pemerintahan.
- c) Menyusun Peramalan Perencanaan SDM.** Pada tahap ini dilakukan identifikasi kebutuhan serta ketersediaan SDM perusahaan. Berdasarkan data taksiran jumlah tenaga kerja internal dan eksternal, dapat diperoleh berapa banyak kebutuhan perusahaan akan tenaga kerja. Selain itu, ketersediaan tenaga kerja dalam perusahaan juga akan diketahui.

- d) **Mengembangkan Rencana *Staffing* SDM dan Aksi.** Selanjutnya, akan dikembangkan sebuah rencana *staffing* SDM. *Staffing* SDM merupakan kegiatan menyangkut merekrut, menyeleksi, dan mempertahankan pegawai (Mathis & Jack, 2010). Rencana ini dirancang untuk membantu mengarahkan perusahaan dalam menghadapi hal-hal tidak terduga terkait kepegawaian dalam perusahaan. Salah satu contoh dari hal yang tidak terduga adalah pengunduran diri karyawan potensial di masa kritis perusahaan.

2.3. Rekrutmen

Rekrutmen adalah sebuah proses mencari dan mengajak tenaga kerja potensial untuk melamar ke lowongan pekerjaan yang dibuka oleh perusahaan (Armstrong & Taylor, 2014) (Mathis & Jack, 2010). Rekrutmen memiliki hubungan yang erat dengan perencanaan SDM. Pada perencanaan SDM dirumuskan strategi untuk pelaksanaan rekrutmen, sehingga untuk pelaksanaan rekrutmen sendiri dapat merujuk pada formulasi perencanaan SDM.

Rekrutmen terdiri dari beberapa tingkatan, yaitu menentukan kebutuhan, menarik kandidat, menyaring lamaran, wawancara, pengujian, menilai kandidat, pengecekan referensi, pengecekan lamaran, penawaran kerja, *follow-up* (Armstrong & Taylor, 2014).

- a) **Menentukan Kebutuhan.** Pada tingkatan ini dilakukan analisis kebutuhan terkait spesifikasi pekerjaan (*job specification*). Spesifikasi pekerjaan akan dilihat berdasarkan pengetahuan, kemampuan dan keahlian, pengalaman, pelatihan, atau kebutuhan khusus jika memang diperlukan perusahaan.
- b) **Menarik Kandidat.** Sebelum bisa melakukan seleksi kandidat, tentunya diperlukan adanya sekumpulan orang yang melamar lowongan pekerjaan yang dipasang oleh perusahaan terlebih dahulu. Ada beberapa hal yang dapat dilakukan perusahaan untuk menarik kandidat, yaitu dengan menyebutkan kelebihan yang dimiliki perusahaan (contoh : penawaran keuntungan jika menjadi pegawai) dan melakukan promosi lowongan kerja melalui media *online* atau cetak.

- c) **Menyaring Lamaran.** Setelah promosi lowongan kerja, orang-orang yang tertarik akan memasukkan CV mereka untuk melamar lowongan tersebut. Dari seluruh lamaran yang masuk akan dilakukan penyaringan untuk mendapatkan daftar-pendek (*shortlisted*) calon kandidat untuk diwawancara. Lamaran disaring dengan melakukan komparasi antara informasi kandidat dan spesifikasi pekerjaan yang telah dibuat sebelumnya. Umumnya, daftar-pendek kandidat hanya terdiri dari empat hingga delapan orang. Jika penyaringan menghasilkan lebih dari delapan orang, maka sebaiknya dilakukan penyaringan ulang dengan menggunakan kriteria yang lebih ketat.
- d) **Wawancara.** Wawancara merupakan salah satu metode seleksi saat rekrutmen. Hal ini dilakukan untuk mengukur sejauh apa kualifikasi pelamar cocok dengan spesifikasi pekerjaan.
- e) **Pengujian.** Pada tahap ini, pelamar diberikan ujian yang dapat berupa ujian psikologi, kecerdasan, atau kemampuan. Dengan dilakukan pengujian akan menghasilkan informasi objektif mengenai bidang yang diujikan terhadap pelamar.
- f) **Menilai Kandidat.** Penilaian kandidat dilakukan berdasarkan pada wawancara dan pengujian yang telah dilakukan kepada pelamar.
- g) **Pengecekan Referensi.** Pengecekan referensi memiliki maksud untuk mencari tahu informasi faktual atas pegawai prospektif terkait masa kerjanya di perusahaan terdahulu. Beberapa pertanyaan yang perlu diajukan ialah lama masa kerja, apa jabatannya, pekerjaan apa yang dilaksanakan, besaran gaji, jumlah hari ketidakhadiran setahun terakhir, dan maukan perusahaan tersebut untuk mempekerjakan kembali pegawai prospektif ini.
- h) **Pengecekan Lamaran.** Survei yang dilakukan oleh *CareerBuilder* terhadap 2.811 manajer SDM dari seluruh bagian Amerika Serikat menunjukkan bahwa 58% manajer menemukan kebohongan dalam resume yang diterima (Career Builder, 2014). Maka dari itu diperlukan pengecekan lamaran untuk menghindari hal ini.
- i) **Penawaran Kerja.** Setelah didapatkan kandidat yang cocok, akan dilakukan penawaran kerja serta penandatanganan kontrak.

- j) ***Follow-Up.*** *Follow-up* dilakukan untuk memastikan bahwa pegawai yang telah direkrut mampu beradaptasi dan untuk mengecek seberapa baik kerja mereka.

Pada tingkatan penyaringan lamaran, dilakukan perbandingan kualifikasi pelamar dengan spesifikasi pekerjaan yang ditentukan pada tahapan penentuan kebutuhan. Pada tingkat ini, PT. XYZ masih mengalami masalah yaitu belum mampu mengetatkan penyaringan lamaran. Hal itu menyebabkan perusahaan mempekerjakan pegawai dengan kualifikasi yang tidak sesuai dengan spesifikasi pekerjaan. Salah satu alasan mengapa hal tersebut bisa terjadi adalah karena sulitnya mendapatkan orang yang tepat untuk posisi tersebut. Sedangkan, mempekerjakan tenaga kerja dengan kualifikasi yang tidak sesuai spesifikasi pekerjaan sebaiknya dihindari (Armstrong & Taylor, 2014).

Penelitian ini dilakukan dengan bertujuan salah satunya untuk mengimplementasikan metode DiA pada tingkatan penyaringan lamaran. Implementasi metode ini diharapkan mampu memberikan daftar pelamar yang cocok dengan spesifikasi pekerjaan.

2.4. Metode *Distance to The Ideal Alternative* (DiA)

Metode *The Distance To The Ideal Alternative* merupakan salah satu pendekatan yang dapat digunakan untuk memecahkan permasalahan pemilihan berdasarkan banyak kriteria dan banyak alternatif atau biasa disebut dengan permasalahan MADM (Tran & Boukhatem, 2008). Pendekatan metode DiA bertujuan untuk menentukan alternatif terbaik dari sekumpulan alternatif yang ada. Pada penelitian ini, alternatif ialah para pelamar. Penilaiannya dilakukan berdasarkan beberapa kriteria yang dibutuhkan. Metode ini dibangun dengan mempertimbangkan beberapa kekurangan yang dimiliki oleh metode TOPSIS, yaitu keakuratan dan abnormalitas *ranking*.

Permasalahan MADM dapat direpresentasikan ke dalam sebuah tabel untuk mempermudah pemahaman pada Tabel 2.1. Dapat dilihat pada Tabel 2.1 terdiri dari 3 elemen, yaitu A, C, dan w. Elemen A merupakan sejumlah alternatif yang merepresentasikan masing-masing pelamar. Elemen C merupakan sejumlah

atribut yang merepresentasikan kriteria penilaian penyaringan lamaran. Elemen W merupakan kepentingan relatif / bobot dari masing-masing atribut.

Tabel 2.1 Tabel Representasi Masalah MADM (Tran & Boukhatem, 2008)

	C_1 (w_1)	C_2 (w_2)	...	C_m (w_m)
A_1	x_{11}	x_{12}	...	x_{1m}
A_2	x_{21}	x_{22}	...	x_{2m}
...
A_n	x_{n1}	x_{n2}	...	x_{nm}

Berikut adalah langkah-langkah utama dalam implementasi metode DiA (Tran & Boukhatem, 2008; Purnamasari, 2015):

- a) Buat matriks keputusan X . Nilai x_{ij} didapatkan dari data para pelamar untuk setiap kriteria.

$$X = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1m} \\ x_{21} & x_{22} & \dots & x_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \dots & x_{nm} \end{bmatrix} \quad (2.1)$$

Keterangan :

x_{ij} = Nilai pelamar ke- i untuk kriteria j

n = Banyaknya pelamar

m = Banyaknya kriteria

- b) Buat matriks keputusan R ternormalisasi. Matriks ini terdiri dari elemen r_{ij} . Elemen tersebut didapatkan dengan melakukan normalisasi pada nilai x_{ij} dari matriks X .

$$R = \begin{bmatrix} r_{11} & r_{12} & \dots & r_{1m} \\ r_{21} & r_{22} & \dots & r_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ r_{n1} & r_{n2} & \dots & r_{nm} \end{bmatrix} \quad (2.2)$$

Nilai masing-masing r_{ij} bisa didapatkan dengan rumus jarak *Euclidean* :

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^n x_{ij}^2}}, i = 1, 2, \dots, n; j = 1, 2, \dots, m \quad (2.3)$$

Keterangan :

x_{ij} = Nilai pelamar ke- i untuk kriteria j

r_{ij} = Nilai ternormalisasi pelamar ke- i untuk kriteria j

n = Banyaknya pelamar

m = Banyaknya kriteria

- c) Buat matriks normalisasi keputusan V terbobot. Matriks ini terdiri dari elemen v_{ij} . Elemen tersebut didapatkan dengan mengalikan r_{ij} dari matriks R dengan nilai bobot yang ternormalisasi.

$$V = \begin{bmatrix} v_{11} & v_{12} & \dots & v_{1m} \\ v_{21} & v_{22} & \dots & v_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ v_{n1} & v_{n2} & \dots & v_{nm} \end{bmatrix} \quad (2.4)$$

Nilai masing-masing v_{ij} bisa didapatkan dengan rumus :

$$v_{ij} = r_{ij} \times w_j, \quad i = 1, 2, \dots, n; j = 1, 2, \dots, m \quad (2.5)$$

Keterangan :

v_{ij} = Nilai terbobot pelamar ke- i untuk kriteria j

r_{ij} = Nilai ternormalisasi pelamar ke- i untuk kriteria j

w_j = Nilai bobot kriteria j

n = Banyaknya pelamar

m = Banyaknya kriteria

- d) Tentukan nilai solusi ideal positif dan negatif dari tiap atribut.

- Untuk kriteria *benefit*

$$A^+ = \max_i v_{ij} = [v_1^+, v_2^+, \dots, v_j^+, \dots, v_m^+] \quad (2.6)$$

$$A^- = \min_i v_{ij} = [v_1^-, v_2^-, \dots, v_j^-, \dots, v_m^-] \quad (2.7)$$

- Untuk kriteria *cost*

$$A^+ = \min_i v_{ij} = [v_1^+, v_2^+, \dots, v_j^+, \dots, v_m^+] \quad (2.8)$$

$$A^- = \max_i v_{ij} = [v_1^-, v_2^-, \dots, v_j^-, \dots, v_m^-] \quad (2.9)$$

Keterangan :

v_j^+ = Nilai solusi ideal positif untuk kriteria j

v_j^- = Nilai solusi ideal negatif untuk kriteria j

- e) Hitung jarak *Manhattan* dari nilai atribut positif dan negatif.

$$D^+_i = \sum_{j=1}^m |v_{ij} - a^+_j| \quad (2.10)$$

$$D^-_i = \sum_{j=1}^m |v_{ij} - a^-_j| \quad (2.11)$$

Keterangan :

v_{ij} = Nilai terbobot pelamar ke- i untuk kriteria j

a^+_j = Nilai solusi ideal positif untuk kriteria j

a^-_j = Nilai solusi ideal negatif untuk kriteria j

- f) Tentukan “*Positive Ideal Alternative*” (PIA).

$$PIA = \{\min (D^+_i), \max (D^-_i)\} \quad (2.12)$$

Keterangan :

D^+_i = Nilai jarak *Manhattan* dari atribut positif untuk pelamar ke- i

D^-_i = Nilai jarak *Manhattan* dari atribut negatif untuk pelamar ke- i

- g) Hitung jarak alternatif ke PIA dengan rumus :

$$R_i = \sqrt{(D^+_i - \min (D^+_i))^2 + (D^-_i - \max (D^-_i))^2} \quad (2.13)$$

Keterangan :

D^+_i = Nilai jarak *Manhattan* dari atribut positif untuk pelamar ke- i

D^-_i = Nilai jarak *Manhattan* dari atribut negatif untuk pelamar ke- i

- h) Lakukan pemeringkatan alternatif. Semakin kecil nilai R_i menandakan alternatif tersebut lebih disarankan.

Selain metode DiA terdapat beberapa metode lain untuk menyelesaikan masalah MADM seperti, TOPSIS, SAW, dan WP. Pada Tabel 2.2 terdapat perbandingan antara metode TOPSIS, SAW, WP, dan DiA.

Tabel 2.2 Tabel Perbandingan Metode MADM

Metode	Formula Akhir	Karakteristik
TOPSIS	$C_j = \frac{S_j^-}{S_j^- + S_j^+}$ <p><i>Keterangan :</i></p> <p>S_j^+ = Nilai jarak alternatif ke-j terhadap solusi ideal positif</p> <p>S_j^- = Nilai jarak alternatif ke-j terhadap solusi ideal negatif</p>	Memiliki konsep dasar bahwa alternatif terbaik ialah yang memiliki jarak terpendek relatif ke solusi ideal (Tran & Boukhatem, 2008).

SAW	$V_i = \sum_{j=1}^n W_j r_{ij}$ <p>Keterangan :</p> <p>W_j = Nilai bobot kriteria j</p> <p>r_{ij} = Nilai ternormalisasi pelamar ke-i untuk kriteria j</p>	Memiliki konsep dasar yaitu mencari penjumlahan terbobot dari nilai atribut seluruh alternatif. Pembobotan dilakukan dengan mengalikan nilai terhadap bobot kriteria (Effendy & Irawan, 2015; Tran & Boukhatem, 2008)
WP	$S_i = \prod_{j=1}^n r_{ij}^{w_j}$ <p>Keterangan :</p> <p>W_j = Nilai bobot kriteria j</p> <p>r_{ij} = Nilai ternormalisasi pelamar ke-i untuk kriteria j</p>	Memiliki konsep dasar yaitu mencari perkalian terbobot dari nilai atribut seluruh alternatif. Pembobotan dilakukan dengan mengangkat nilai dengan bobot kriteria (Tran & Boukhatem, 2008; Effendy & Irawan, 2015).
DiA	$R_i = \sqrt{\left(D_i^+ - \min(D_i^+)\right)^2 + \left(D_i^- - \max(D_i^-)\right)^2}$ <p>Keterangan :</p> <p>D_i^+ = Nilai jarak <i>Manhattan</i> dari atribut positif untuk pelamar ke- i</p> <p>D_i^- = Nilai jarak <i>Manhattan</i> dari atribut negatif untuk pelamar ke- i</p>	Memiliki konsep dasar bahwa alternatif terbaik adalah yang memiliki jarak terdekat dengan alternatif ideal positif (Tran & Boukhatem, 2008).

Meski terdapat begitu banyak metode MADM, penelitian ini akan mengusung metode DiA untuk diterapkan dalam sistem seleksi pegawai. Pemilihan ini didasari oleh penelitian yang telah dilakukan (Tran & Boukhatem, 2008) dan (Purnamasari, 2015). Pengujian yang dilakukan oleh (Tran & Boukhatem, 2008) menunjukkan bahwa metode DiA mengalahkan TOPSIS dalam segi abnormalitas *ranking* dan memiliki akurasi yang lebih baik dari pada SAW dan WP. Dalam penelitian tersebut ditunjukkan bahwa jarak antara nilai peringkat pertama dan kedua dalam metode DiA lebih besar daripada metode SAW dan WP. Semakin besar jarak akan mempermudah penentuan alternatif terbaik,

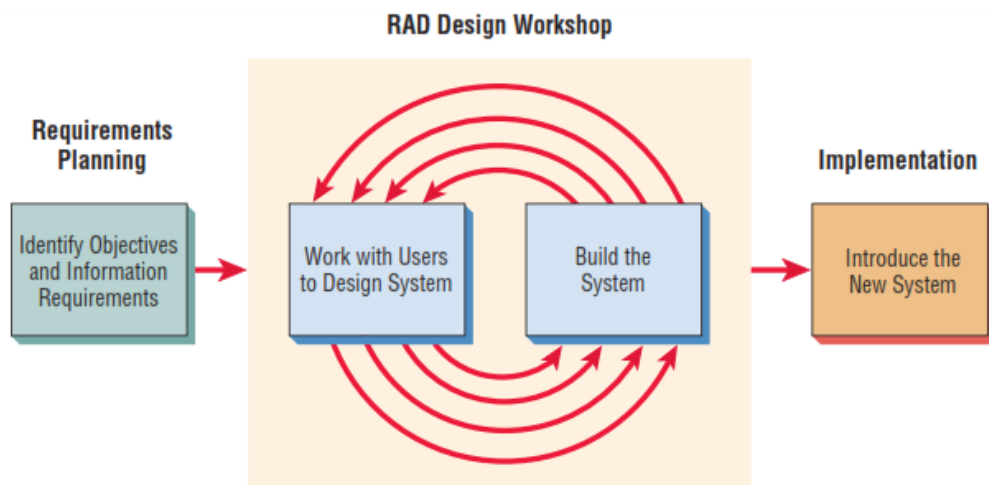
sehingga akan meningkatkan akurasi dalam penentuan peringkat. Kemudian penelitian yang dilakukan oleh (Purnamasari, 2015) menunjukkan bahwa DiA memiliki sensitivitas yang lebih tinggi dari pada SAW.

2.5. *Rapid Application Development (RAD)*

Dalam membangun sebuah sistem informasi dibutuhkan sebuah metode pengembangan sistem agar pengembangan dapat dilakukan dengan baik. Metode pengembangan sistem yang digunakan akan berbeda tergantung beberapa situasi. RAD merupakan salah satu metode pengembangan sistem yang mampu memperpendek waktu pembangunan sistem. Maka dari itu, RAD cocok diterapkan dalam sebuah proyek pengembangan sistem dengan ketersediaan waktu yang sangat terbatas (Kendall & Kendall, 2011). Model RAD mengenal beberapa fase, yaitu *Requirements Planning*, *RAD Design Workshop*, dan *Implementation* (Kendall & Kendall, 2011).

Diawali dengan *Requirements Planning* atau perencanaan kebutuhan, dilaksanakan identifikasi akan kebutuhan dari sistem informasi yang akan dibangun. Daftar kebutuhan didapatkan dengan berdasar pada tujuan dibuatnya sistem. Kegiatan tersebut dilakukan oleh sistem analis serta calon pengguna.

Setelah kebutuhan sistem informasi diketahui, dilakukan fase kedua yang dinamakan *RAD Design Workshop*. Tahap ini terdiri dari dua bagian, yaitu perancangan sistem dan pembangunan sistem. Seperti yang dapat dilihat pada Gambar 2.2, terdapat beberapa panah yang berputar berulang kali menunjuk kepada bagian perancangan sistem dan pembangunan sistem secara bergantian. Panah yang berputar berulang tersebut menandakan bahwa terdapat iterasi antara perancangan dan pembangunan sistem. Proses iterasi ini merupakan poin utama yang berperan penting dalam pemendekan waktu pembuatan sistem. Selama fase *design workshop* ini, calon pengguna akan memberikan tanggapan atas *prototype* yang telah dibangun berdasarkan kebutuhan calon pengguna. Perbaikan yang disarankan calon pengguna akan di akomodasi oleh pengembang untuk pembuatan *prototype* selanjutnya. Kegiatan ini akan terus berulang hingga terdapat persetujuan antara pihak analis dan calon pengguna.



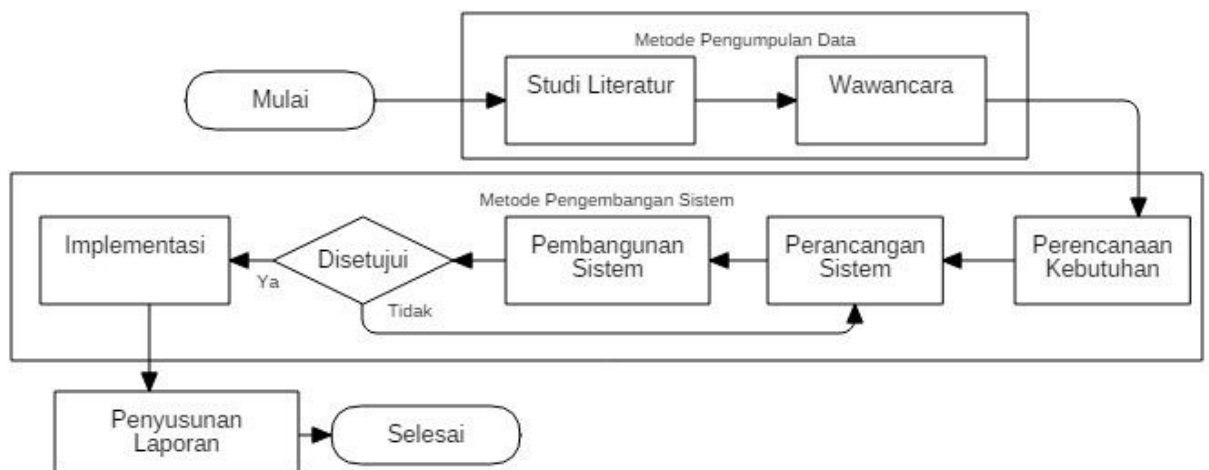
Gambar 2.2 Fase RAD (Kendall & Kendall, 2011)

Pada fase terakhir terdapat fase implementasi. Pada akhirnya dalam fase ini dibangun sebuah atau sebagian sistem yang sebenarnya, tidak lagi berbentuk *prototype*. Pengujian sistem dilakukan pada tahap ini. Setelah lolos pengujian, sistem akan diperkenalkan ke para pengguna untuk kemudian digunakan (Kendall & Kendall, 2011).

BAB 3 METODOLOGI PENELITIAN

3.1 Kerangka Kerja Penelitian

Penelitian ini akan dilakukan secara bertahap dengan alur seperti yang dapat dilihat pada Gambar 3.1.



Gambar 3.1 Alur Pelaksanaan Penelitian

Metode pengumpulan data yang diterapkan dalam penelitian adalah studi literatur dan wawancara. Deskripsi kegiatan dari studi literatur dan wawancara akan dipaparkan pada subbab Metode Pengumpulan Data tepat setelah subbab ini. Untuk pengembangan sistem, penelitian mengadopsi metode pengembangan RAD yang terdiri dari empat tahapan, yaitu perencanaan kebutuhan, perancangan sistem, pembangunan sistem, dan implementasi. Pada Gambar 3.1, tahapan metode RAD dikelompokkan menjadi satu ke dalam bagian Metode Pengembangan Sistem. Pemaparan kegiatan pada tiap tahapan RAD serta tahap penyusunan laporan akan dilakukan pada subbab berjudul Metode Pengembangan Sistem dan Penyusunan Laporan.

3.2 Metode Pengumpulan Data

Data yang akan digunakan dalam penelitian ini didapatkan melalui dua cara, yaitu studi literatur dan wawancara.

3.2.1 Studi Literatur

Dalam studi literatur dilakukan pencarian dan penyaringan referensi. Kegiatan ini menghasilkan daftar referensi yang akan digunakan untuk memperkuat analisis dalam penelitian ini. Referensi yang digunakan adalah referensi dengan topik-topik yang berhubungan dengan penelitian yang akan dilakukan. Sumber referensi dapat berupa buku, artikel dari internet, serta jurnal dan *paper* penelitian. Selain untuk mencari referensi sebagai penunjang analisis, studi literatur juga dilakukan untuk mencari penelitian-penelitian terdahulu yang telah dilakukan oleh orang lain.

3.2.2 Wawancara

Dalam wawancara dilakukan tanya-jawab dengan pihak yang memiliki pengetahuan berkaitan masalah yang sedang dialami perusahaan, yaitu Manajer *Human Capital* di PT. XYZ. Wawancara dilakukan bertujuan untuk mendapatkan gambaran terkini kondisi perusahaan terkait rekrutmen dan penyaringan lamaran. Dari wawancara diketahui bahwa pelaksanaan rekrutmen perusahaan dibedakan menjadi 3, yaitu untuk Cabang, Kantor Pusat, dan *Management Trainee*. Rata-rata jumlah orang yang melamar ke perusahaannya tiap bulan sebanyak 250 orang. Penyaringan lamaran atau seleksi administratif yang dilakukan perusahaan masih menggunakan cara manual.

Dalam tahap perancangan serta pembangunan sistem, diskusi dilakukan dengan Manajer IT serta Manajer *Human Capital* untuk mengumpulkan data yang diperlukan. Hasil wawancara dapat dilihat pada Lampiran 1, sedangkan hasil diskusi yang dilakukan pada penelitian ini tidak didokumentasikan.

3.3 Metode Pengembangan Sistem dan Penyusunan Laporan

Data yang telah dikumpulkan akan digunakan sebagai patokan dalam mengembangkan sistem. Pada penelitian ini, pengembangan sistem menerapkan pendekatan model RAD. Model RAD terdiri dari beberapa fase dengan iterasi di dalamnya. Model ini dipilih sebagai model pengembangan sistem karena karakteristiknya yang sesuai dengan keadaan di perusahaan. Keadaan memaksa perusahaan untuk tidak dapat menunggu lama hingga sistem selesai dibangun.

Subbab berikut ini akan memaparkan kegiatan dilakukan pada tiap-tiap tahapan RAD serta penyusunan laporan.

3.3.1 Perencanaan Kebutuhan

Penelitian ini dilakukan sejak awal tahun 2016, yaitu selama delapan bulan dimulai dari bulan Januari hingga Agustus. Pada tahap awal ini dilakukan diskusi untuk mencari tahu apa tujuan dibuatnya sistem serta informasi apa saja yang akan digunakan di dalamnya. Dengan merujuk pada tujuan dibuatnya sistem, dapat disusun daftar kebutuhan fungsional dan non-fungsional sistem. Kebutuhan fungsional dan non-fungsional sistem disertakan pada dokumen SRS di Lampiran 2.

Kebutuhan informasi sistem dibedakan menjadi dua, yaitu kebutuhan informasi secara umum dan kebutuhan informasi untuk Metode DiA. Metode DiA membutuhkan tiga elemen utama untuk melakukan penilaian, yaitu alternatif, kriteria, dan kepentingan relatif masing-masing atribut.

A. Kebutuhan Informasi Secara Umum

Eratnya kaitan sistem dengan proses rekrutmen mengandung arti bahwa sebagian besar informasi yang terdapat dalam sistem merupakan data yang memiliki hubungan dengan rekrutmen. Kebutuhan informasi sistem secara umum didapatkan dengan mempertimbangkan kebutuhan fungsional dari sistem yang telah disebutkan. Berikut merupakan daftar kebutuhan informasi sistem secara umum :

- *Job Title* : Informasi daftar jabatan pegawai dalam perusahaan.
- *Posisi* : Informasi daftar posisi dalam perusahaan.
- *Requirement* : Informasi kebutuhan kualifikasi tiap *Job Title* dan *Posisi*.
- *Kantor* : Informasi daftar kantor yang dimiliki perusahaan.
- *Lowongan Kerja*: Informasi daftar posisi yang sedang lowong pada kantor.
- *Pelamar* : Informasi daftar orang yang melamar ke perusahaan.

B. Kebutuhan Informasi Metode DiA

Permasalahan dalam penyaringan lamaran merupakan salah satu masalah dengan banyak kriteria dan banyak alternatif (MADM). Dalam penyaringan lamaran, sekumpulan lamaran yang diterima perusahaan akan dinilai terhadap beberapa kriteria tertentu dari perusahaan. Hasil penilaian digunakan untuk menentukan pelamar yang layak melanjutkan ke tahap berikutnya. Hingga saat ini perusahaan melakukan penyaringan lamaran dengan cara manual, yaitu mengandalkan pengamatan mata untuk membandingkan kualifikasi pelamar dengan kualifikasi posisi. Hal ini cukup memakan waktu karena proses *screening* sebuah CV membutuhkan waktu hingga 5 menit.

Penelitian ini menerapkan metode DiA ke dalam sistem untuk membantu pelaksanaan proses penyaringan lamaran. Sistem akan menggunakan model matematis dari metode DiA untuk melakukan penilaian masing-masing pelamar. Setelah nilai didapatkan, maka akan dilakukan pemeringkatan. Melalui pemeringkatan tersebut dapat dilihat daftar pelamar yang lebih unggul dari yang lain berdasarkan penilaian sistem. Mereka adalah daftar pelamar yang direkomendasikan oleh sistem untuk lanjut ke tahap berikutnya. Namun, perlu diperjelas bahwa sistem hanya sekedar memberikan rekomendasi pelamar. Pengambilan keputusan tetaplah berada di bawah kuasa pengambil keputusan.

Kebutuhan informasi dari sistem, khususnya untuk implementasi metode DiA, didapatkan melalui diskusi yang dilakukan bersama Manajer *Human Capital* perusahaan. Berikut merupakan beberapa kriteria yang dibutuhkan dalam penilaian :

- a. Jenis Kelamin (C_1) data jenis kelamin pelamar akan dibandingkan dengan kualifikasi jenis kelamin dari posisi yang dilamar. Penilaian dilakukan dengan melihat kesesuaian kualifikasi tersebut. Hasil yang sesuai akan menjadi prioritas untuk direkomendasikan. Tabel 3.1 menunjukkan konversi nilai untuk kriteria ini.

Tabel 3.1 Nilai Jenis Kelamin

Hasil Perbandingan	Nilai
Sesuai	1
Tidak Sesuai	0

- b. Usia (C_2) data usia pelamar akan dibandingkan dengan kualifikasi usia dari posisi yang dilamar. Penilaian dilakukan dengan melihat kesesuaian kualifikasi tersebut. Hasil yang sesuai akan menjadi prioritas untuk direkomendasikan. Tabel 3.2 menunjukkan konversi nilai untuk kriteria ini.

Tabel 3.2 Nilai Usia	
Hasil Perbandingan	Nilai
Sesuai	1
Selisih 1 Tahun	0,75
Lainnya	0

- c. Pendidikan (C_3) data pendidikan terakhir pelamar akan dibandingkan dengan kualifikasi pendidikan terakhir dari posisi yang dilamar. Nilai pendidikan yang lebih tinggi akan diprioritaskan. Tabel 3.3 menunjukkan konversi nilai untuk kriteria ini.

Tabel 3.3 Nilai Pendidikan	
Nilai Pendidikan	Nilai
S3	100
S2	90
S1	80
D3	70
D2	60
D1	50
SMA	40
SMP	30

- d. Pengalaman Kerja (C_4) data pengalaman kerja pelamar akan dibandingkan dengan kualifikasi pengalaman kerja dari posisi yang dilamar. Nilai pengalaman kerja yang lebih tinggi akan diprioritaskan. Karena nilai kriteria ini bersifat kuantitatif dan tidak memiliki batasan tertentu seperti pada usia, maka data akan disajikan sebagaimana adanya tanpa konversi nilai.
- e. Kemampuan (C_5) data kemampuan pelamar akan dibandingkan dengan kualifikasi kemampuan dari posisi yang dilamar. Penilaian dilakukan dengan melihat kesesuaian kualifikasi tersebut. Hasil yang sesuai akan menjadi prioritas untuk direkomendasikan. Tabel 3.4 menunjukkan konversi nilai untuk kriteria ini.

Tabel 3.4 Nilai Kemampuan

Hasil Perbandingan	Nilai
Sesuai	1
Sesuai sebanyak n	$\frac{\text{jumlah memenuhi } (n)}{\text{jumlah keseluruhan}}$

- f. Gaji yang Diharapkan (C_6) Nilai yang lebih rendah akan diprioritaskan pada kriteria ini. Karena nilai kriteria ini bersifat kuantitatif dan tidak memiliki batasan tertentu seperti pada usia, maka data akan disajikan sebagaimana adanya tanpa konversi nilai.

Masing-masing kriteria yang telah dijelaskan di atas memiliki nilai prioritas. Nilai prioritas tersebut diukur dalam skala, yaitu tidak penting, kurang penting, normal, penting, dan sangat penting. Nilai prioritas akan menentukan nilai bobot atau kepentingan relatif kriteria tersebut. Nilai prioritas kriteria dapat disesuaikan oleh pengguna untuk mengantisipasi apabila seiring berjalannya waktu terjadi perubahan tingkat kepentingan masing-masing kriteria dalam pelaksanaan penyaringan lamaran. Berikut merupakan bobot dari masing-masing kriteria berdasarkan nilai prioritas yang diterapkan :

Tabel 3.5 Bobot Kriteria

Kriteria Penilaian	Tingkat Kepentingan	Bobot
Jenis Kelamin (C_1)	Kurang Penting	2
Usia (C_2)	Sangat Penting	5
Pendidikan (C_3)	Penting	4
Pengalaman (C_4)	Normal	3
Kemampuan (C_5)	Sangat Penting	5
Gaji yang Diharapkan (C_6)	Normal	3

3.3.2 Perancangan Sistem

Setelah perencanaan kebutuhan selesai dilakukan, selanjutnya adalah perancangan sistem. Hasil yang didapatkan dari perencanaan kebutuhan akan digunakan sebagai dasar perancangan sistem. Selanjutnya akan dibuat diagram *use case*, *activity diagram*, *class diagram*, serta rancangan basis data. Hal ini dilakukan untuk mempermudah proses pemahaman akan alur kerja dan struktur dasar dari sistem. Untuk melaksanakan perancangan sistem, diskusi dilakukan bersama dengan Manajer IT. Perencanaan sistem dilakukan berulang kali. Hal ini bertujuan untuk memastikan bahwa rancangan sistem sudah cocok dengan kebutuhan pengguna.

3.3.3 Pembangunan Sistem

Dengan didasarkan pada rancangan yang telah dibuat, sebuah *prototype* dari sistem akan dibuat. *Prototype* merupakan sebuah sistem bayangan yang dibangun dengan tujuan untuk merepresentasikan sistem yang sebenarnya. Namun, yang lebih ditekankan pada *prototype* ialah dari segi fungsional di mana ia harus mampu melaksanakan fungsi utama dari sistem yang sesungguhnya. Setelah *prototype* dibangun, tahap perencanaan sistem kembali dilakukan. Hal ini bertujuan untuk memberi gambaran pada pengguna, seperti apa sistem yang akan dibangun nantinya. Pengguna melakukan evaluasi terhadap *prototype* yang telah dibangun di tahap Pembangunan Sistem. Apabila tidak ada perubahan yang diusulkan oleh calon pengguna, maka sistem yang sebenarnya dibuat dengan berdasarkan pada *prototype* yang disetujui.

3.3.4 Implementasi

Pada tahap ini dilakukan pembangunan sistem sesungguhnya berdasarkan pada *prototype* yang disetujui. Pengujian sistem juga dilakukan pada tahap ini. Pengujian sistem dilakukan dengan tujuan untuk memastikan bahwa sistem bekerja sesuai dengan yang diharapkan. Penyelidikan *error* atau *bug* pada sistem juga berpeluang besar ditemukan melalui pengujian. Pada penelitian ini, akan dilakukan dua pengujian yaitu *black-box* dan *white-box*.

3.3.5 Penyusunan Laporan

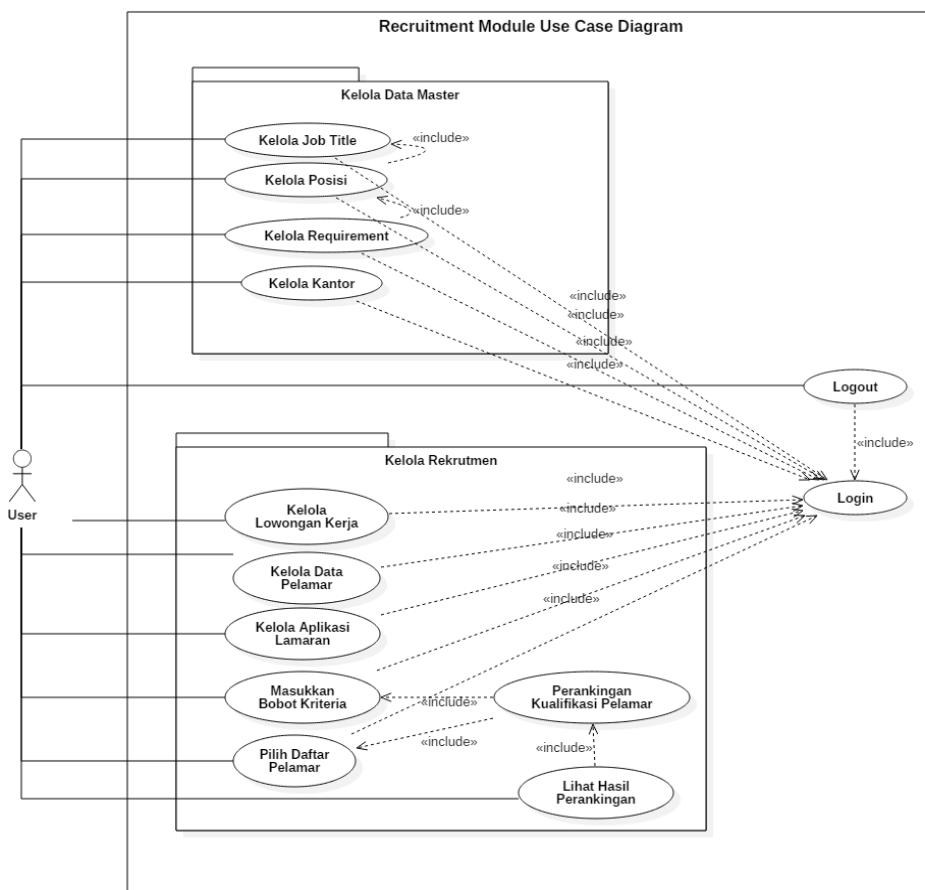
Hasil yang didapatkan dari tahap sebelumnya akan digunakan sebagai bahan penulisan laporan. Analisa akan dilakukan terhadap implementasi metode DiA pada sistem.

BAB 4 IMPLEMENTASI DAN PEMBAHASAN

4.1 Perancangan Sistem

Pada penelitian ini, rancangan sistem tersusun dari beberapa elemen seperti *use case diagram*, *activity diagram*, *class diagram*, dan rancangan basis data. Pemaparan tiap elemen dapat ditemukan pada subbab setelah ini secara berurutan.

4.1.1 Use Case Diagram



Gambar 4.1 Diagram Use Case

Di atas terdapat Gambar 4.1 yang merupakan diagram *Use Case* dari sistem. Terdapat seorang aktor, yaitu *user*. *User* merupakan seorang karyawan Departemen Rekrutmen dan Asesmen yang dapat mengelola master data seperti *jobtitle*, *posisi*, *requirement* dari *posisi*, dan *kantor*. *User* juga dapat mengelola

lowongan kerja, data pelamar, dan aplikasi lamaran serta melihat hasil pemeringkatan pelamar. Deskripsi dari *use case* dapat dilihat pada Tabel 4.1.

Tabel 4.1 Deskripsi Use Case Modul Rekrutmen

Nama Use Case	Deskripsi
Kelola <i>Job Title</i>	<i>Use case</i> ini menggambarkan kegiatan pengelolaan data <i>Job Title</i> .
Kelola Posisi	<i>Use case</i> ini menggambarkan kegiatan pengelolaan data Posisi.
Kelola <i>Requirement</i>	<i>Use case</i> ini menggambarkan kegiatan pengelolaan data <i>requirement</i> dari posisi yang ada. Terdapat dua jenis <i>requirement</i> pada sistem, yaitu <i>requirement jobtitle</i> dan <i>requirement</i> posisi.
Kelola Kantor	<i>Use case</i> ini menggambarkan kegiatan pengelolaan data kantor, seperti nama, alamat, jenis kantor.
Kelola Lowongan Kerja	<i>Use case</i> ini menggambarkan kegiatan pengelolaan informasi lowongan kerja.
Kelola Data Pelamar	<i>Use case</i> ini menggambarkan kegiatan pengelolaan data kandidat yang melamar pekerjaan.
Kelola Aplikasi Lamaran	<i>Use case</i> ini menggambarkan kegiatan pengelolaan informasi aplikasi lamaran yang menghubungkan lowongan kerja dengan pelamar.
<i>Logout</i>	<i>Use case</i> ini menggambarkan kegiatan <i>logout</i> keluar dari aplikasi.
<i>Login</i>	<i>Use case</i> ini menggambarkan kegiatan <i>login</i> atau masuk ke dalam aplikasi.
Pemeringkatan Kualifikasi Pelamar	<i>Use case</i> ini menggambarkan kegiatan pemeringkatan kualifikasi dari para pelamar dengan menggunakan metode DiA.
Masukkan Bobot Kriteria	<i>Use case</i> ini menggambarkan kegiatan <i>input</i> bobot kriteria penilaian.
Pilih Daftar Pelamar	<i>Use case</i> ini menggambarkan kegiatan pemilihan daftar pelamar untuk diseleksi.
Lihat Hasil Pemeringkatan	<i>Use case</i> ini menggambarkan kegiatan melihat hasil pemeringkatan.

Sesuai dengan deskripsi pada Tabel 4.1, *Use case* bernama pemeringkatan kualifikasi pelamar menandakan sebuah kegiatan pemeringkatan pelamar berdasarkan kualifikasinya. Pemeringkatan dilakukan oleh sistem dengan menerapkan metode DiA. Kegiatan ini dipicu oleh aksi *User* yang menekan tombol lakukan seleksi pada halaman lihat daftar pelamar. Pada subbab selanjutnya akan dijelaskan sebuah diagram aktivitas dari kegiatan pemeringkatan kualifikasi pelamar. Untuk penjelasan yang lebih detail terkait diagram *use case* sistem dapat merujuk pada Lampiran 2.

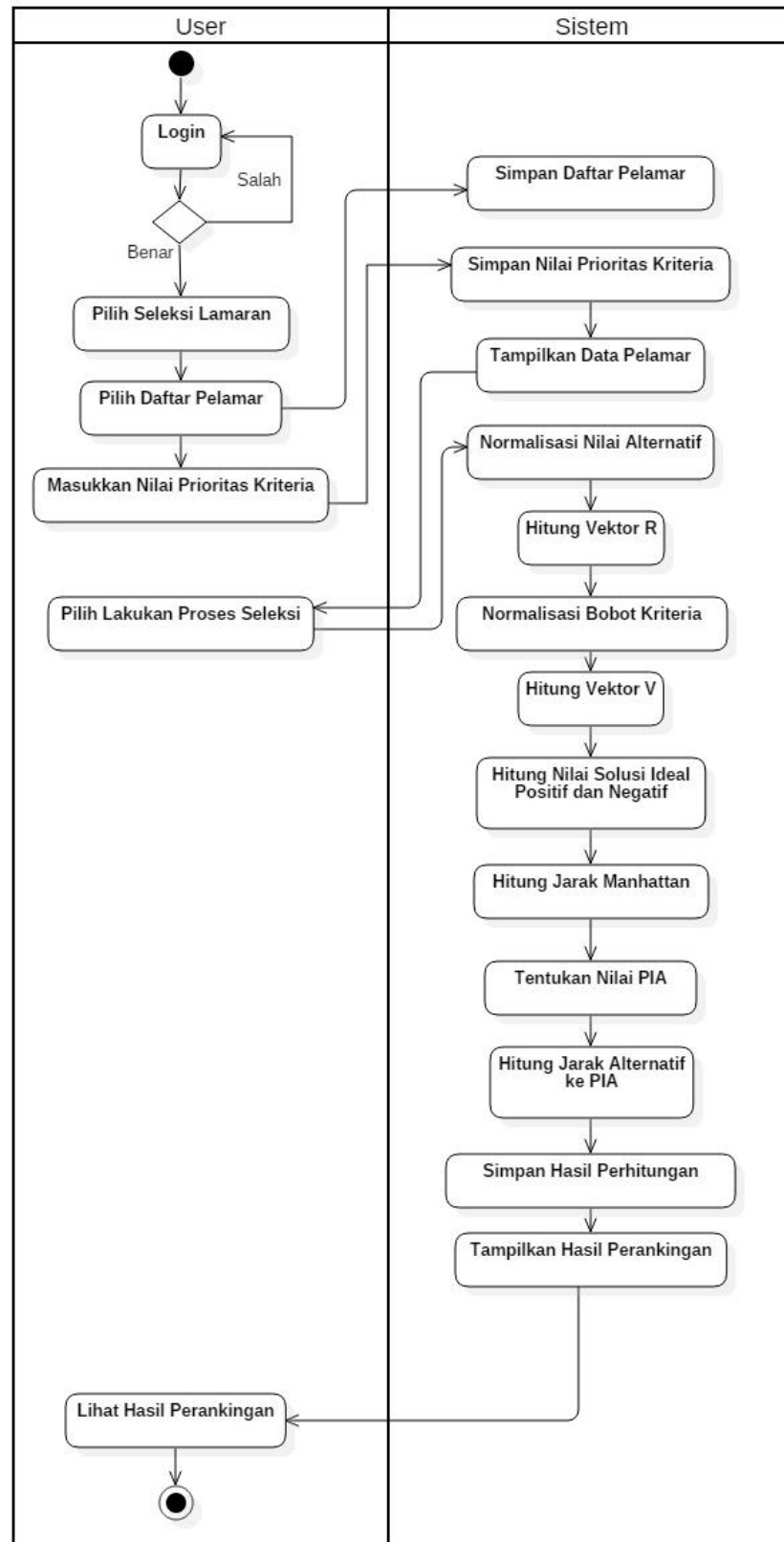
4.1.2 Activity Diagram

Pada Gambar 4.2 dapat dilihat terdapat diagram aktivitas merepresentasikan alur pemeringkatan pelamar menggunakan metode DiA. Diagram aktivitas menunjukkan interaksi yang terjadi antara *user* dengan sistem. Sebagian besar aktivitas berada pada kolom sistem. Adapun hal ini dikarenakan pemeringkatan kualifikasi pelamar merupakan sebuah proses yang dilakukan oleh sistem. *User* hanya bertugas menyediakan informasi yang dibutuhkan sistem untuk proses pemeringkatan dan memberikan perintah ke sistem untuk melakukan proses tersebut. Perintah diberikan dengan cara menekan tombol lakukan seleksi pada halaman lihat data pelamar.

Proses perhitungan dengan metode DiA dimulai dari tahap normalisasi nilai alternatif hingga penghitungan jarak alternatif ke PIA. Urutan proses tersebut merupakan urutan proses yang sama dengan langkah-langkah utama implementasi metode DiA yang diungkapkan pada Bab 2. Setelah perhitungan selesai dilakukan, hasilnya akan disimpan ke basis data. Kemudian, sistem akan menampilkan informasi hasil dari proses perhitungan. Informasi yang ditampilkan ialah nama pelamar, hasil akhir perhitungan, status kecocokan, dan deviasi. Data ditampilkan dengan pola pengurutan *ascending* berdasarkan kolom hasil perhitungan. Dalam metode DiA semakin kecil nilai hasil akhir perhitungannya maka ia lebih direkomendasikan.

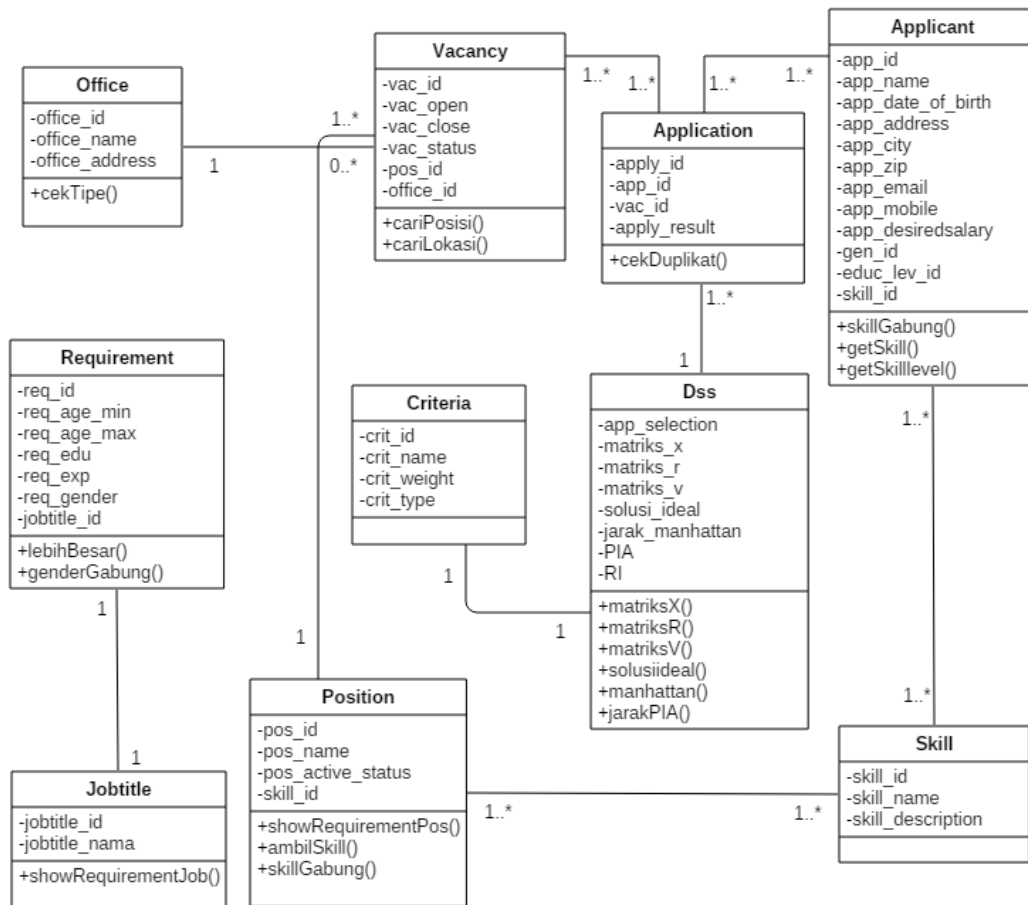
Beberapa kriteria akan digunakan sebagai dasar penilaian pelamar yang akan direkomendasikan sistem. Kriteria yang digunakan ialah umur, tingkat pendidikan, jenis kelamin, pengalaman kerja, kemampuan, dan gaji yang

diharapkan. Sedangkan alternatif merupakan daftar pelamar yang melamar pada posisi tertentu pada lowongan pekerjaan.



Gambar 4.2 Diagram Aktivitas Proses Pemeringkatan Pelamar

4.1.3 Class Diagram



Gambar 4.3 Class Diagram Sistem

Class diagram dari modul rekrutmen sistem informasi seleksi pegawai dapat dilihat pada Gambar 4.3. Berikut penjelasan ringkas dari tiap kelas :

- a) *Applicant* : Kelas ini mengelola data pelamar, yaitu informasi diri.
- b) *Jobtitle* : Kelas ini mengelola data jabatan.
- c) *Position* : Kelas ini mengelola data posisi.
- d) *Office* : Kelas ini mengelola data kantor.
- e) *Skill* : Kelas ini mengelola data kemampuan.
- f) *Requirement* : Kelas ini mengelola data *requirement* dari *jobtitle*.
- g) *Criteria* : Kelas ini mengelola data kriteria yang akan dinilai ketika melakukan pemeringkatan.
- h) *Dss* : Kelas ini mengelola proses perhitungan DiA serta pemeringkatan pelamar.
- i) *Vacancy* : Kelas ini mengelola data lowongan pekerjaan.

j) *Application* : Kelas ini mengelola data aplikasi lamaran perusahaan.

4.1.4 *Data Model*

Identifikasi rancangan basis data dari sistem dilakukan melalui tahapan, yaitu *conceptual design*, *logical design*, dan *physical design*. Rancangan dibuat dengan menggunakan notasi *Crow Foot*. Rancangan *conceptual design*, *logical design*, dan *physical design* dapat ditemukan pada Lampiran 2.

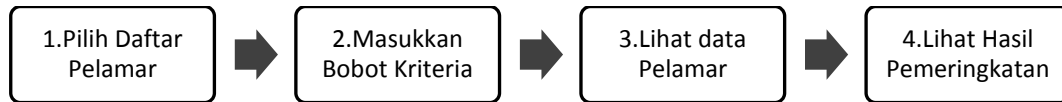
4.2 Implementasi

Pada tahap implementasi, rancangan sistem yang telah dibuat pada tahap Perancangan akan diwujudkan ke dalam bentuk sistem yang sesungguhnya. Dengan berpatokan pada hasil perancangan, dilakukan pengembangan sistem untuk membangun fitur-fitur sistem agar dapat digunakan oleh pengguna. Bagian ini akan memaparkan kegiatan yang dilakukan ketika memasuki tahap Implementasi dari proses pengembangan. Pemaparan tahap implementasi akan dibagi menjadi dua bagian, yaitu implementasi metode DiA dan implementasi sistem.

4.2.1 Implementasi Metode DiA

Seperti yang sudah disebutkan sebelumnya, metode DiA diterapkan pada subkategori bernama Seleksi Resume. Pada subkategori Seleksi Resume terdiri dari beberapa langkah yang harus dilakukan pengguna secara berurutan. Gambar 4.4 merepresentasikan urutan langkah yang perlu dilakukan oleh pengguna. Masing-masing langkah tersebut menuntun pengguna dalam menyediakan informasi yang dibutuhkan sistem untuk melakukan seleksi. Pada langkah pertama, sistem akan menyimpan data daftar pelamar yang akan diseleksi. Pada langkah kedua, sistem akan menyimpan bobot kriteria yang dimasukkan pengguna untuk selanjutnya digunakan dalam penilaian. Langkah ketiga hanya sekedar memberikan tampilan daftar pelamar yang akan diseleksi kepada pengguna. Setelah selesai melihat data pelamar, pengguna akan menekan tombol “Lakukan Seleksi”. Aksi ini akan memulai proses perhitungan dengan metode DiA. Setelah

proses perhitungan dan pemeringkatan selesai, maka hasilnya akan ditampilkan kepada pengguna.



Gambar 4.4 Urutan Langkah Pengguna Untuk Melakukan Seleksi

Untuk mendapatkan gambaran yang lebih baik mengenai penerapan metode DiA pada sistem, berikut penjelasan implementasi metode DiA langkah demi langkah dengan menggunakan sampel data pada Tabel 4.2. Data pada Tabel 4.2 merupakan daftar orang yang melamar pada posisi *Corporate Legal Officer* dengan kebutuhan kualifikasi sebagai berikut :

- Jenis Kelamin : Laki-laki / Perempuan
- Usia : 23 s/d 35 tahun
- Pendidikan : minimal S1
- Pengalaman : 2 tahun
- Kemampuan : MS Office
- Gaji : Rp3.000.000 – Rp4.000.000

Data pelamar dan kebutuhan kualifikasi posisi yang digunakan dalam penelitian ini didapatkan dari iklan lowongan kerja yang dipasang oleh perusahaan melalui *jobstreet*. Dikarenakan keterbatasan data yang disediakan *jobstreet*, maka data pelamar terkait kolom jenis kelamin, umur, dan kemampuan diisi dengan data *dummy*. Hal ini dilakukan untuk kelancaran implementasi metode DiA.

Tabel 4.2 Data Kualifikasi Alternatif

Nama (bobot)	C ₁ (2)	C ₂ (5)	C ₃ (4)	C ₄ (3)	C ₅ (5)	C ₆ (3)
Pelamar 1	P	23	S1	1	MS Office	4.000
Pelamar 2	L	30	S1	5	MS Office	4.000
Pelamar 3	L	27	S1	4	MS Office	4.000
Pelamar 4	L	29	S2	4	MS Office	7.000
Pelamar 5	L	24	S1	2	MS Office	4.000
Pelamar 6	L	35	S2	6	MS Office	5.500
Pelamar 7	P	36	S2	7	MS Office	5.000

Pelamar 8	L	22	S1	1	MS Office	4.000
Pelamar 9	L	32	S1	3	MS Office	5.000
Pelamar 10	L	33	S1	5	MS Office	4.000

Keterangan :

C_1 = Jenis Kelamin

C_2 = Usia

C_3 = Pendidikan

C_4 = Pengalaman (tahun)

C_5 = Kemampuan

C_6 = Gaji yang Diharapkan (ribuan)

A. Matriks Keputusan X

Sesuai dengan karakteristik permasalahan MADM yang tersusun dari sekumpulan data alternatif beserta kriteria, maka dalam implementasi metode DiA ke sistem banyak mengandalkan tipe data *array*. Untuk membuat matriks keputusan X, data masing-masing pelamar pada Tabel 4.2 dibandingkan dengan kebutuhan kualifikasi posisi yang dilamar dan dilakukan konversi berdasarkan tabel konversi tiap kriteria di Bab 3. Tabel 4.3 menunjukkan salah satu contoh perbandingan dan konversi dari data pelamar 1.

Tabel 4.3 Tabel Perbandingan Kualifikasi Pelamar dan Posisi

	Jenis Kelamin	Usia	Pendidikan	Pengalaman	Kemampuan	Gaji
<i>Requirement</i>		23				3.000
Posisi	L/P	s/d	S1	2	MS Office	s/d
		35				4.000
Kualifikasi Pelamar 1	P	23	S1	1	MS Office	4.000

Dengan membandingkan jenis kelamin pelamar dengan *requirement*, diketahui bahwa hasilnya sesuai. Pada Tabel 3.1, hasil perbandingan yang sesuai maka bernilai 1. Hasil perbandingan usia juga sesuai, maka bernilai 1. Hasil perbandingan pendidikan merujuk pada Tabel 3.3 menjadi bernilai 80. Untuk data pengalaman kerja disajikan apa adanya, nilai tidak berubah. Hasil perbandingan kemampuan merujuk pada Tabel 3.4 menjadi bernilai 1, karena kemampuan

pelamar memenuhi seluruh kemampuan yang disyaratkan. Sedangkan untuk gaji, data disajikan apa adanya, tidak ada perubahan. Maka hasil konversi data pelamar 1 hingga 10 dapat dilihat pada Tabel 4.4.

Tabel 4.4 Data Kualifikasi Alternatif Setelah Perbandingan dan Konversi

Nama (bobot)	C ₁ (2)	C ₂ (5)	C ₃ (4)	C ₄ (3)	C ₅ (5)	C ₆ (3)
Pelamar 1	1	1	80	1	1	4000
Pelamar 2	1	1	80	5	1	4000
Pelamar 3	1	1	80	4	1	4000
Pelamar 4	1	1	90	4	1	7000
Pelamar 5	1	1	80	2	1	4000
Pelamar 6	1	1	90	6	1	5500
Pelamar 7	1	0,75	90	7	1	5000
Pelamar 8	1	0,75	80	1	1	4000
Pelamar 9	1	1	80	3	1	5000
Pelamar 10	1	1	80	5	1	4000

```

$alternatif = array();
$counter = 1;
foreach($alts as $idx=>$alt)
{
    $this->status['stat'] = 1;
    $this->status['deviasi'] = '';

    $app=Applicant::model()->findByPk($alt->app_id);

    $alternatif[$counter] = array( 'nama'=>$app->app_name,
    1=>$this->nilaiEdu($app),
    2=>$this->nilaijk($alt),
    3=>$this->nilaiusia($alt),
    4=>$this->nilaikemampuan($alt),
    5=>$this->experienceLong($app),
    6=>$app->app_desiredsalary,
    'applyid'=>$alt->apply_id
    );

    $alternatif[$counter]['stat'] = $this->status['stat'];
    $alternatif[$counter]['deviasi'] = $this->status['deviasi'];
    $counter++;
}

```

Gambar 4.5 Kode Program Matriks Keputusan X

Gambar 4.5 merupakan kode program untuk menyusun matriks keputusan X. Data para pelamar disimpan dalam sebuah *array* dua dimensi bernama “alternatif”. Data pelamar yang tersimpan dalam *array* mencakup data hasil konversi, status kecocokan kualifikasi, dan kualifikasi yang tidak terpenuhi oleh pelamar. Fungsi `nilaiEdu()`, `nilaijk()`, `nilaiusia()`, `nilaikemampuan()`, dan `experienceLong()` adalah untuk melakukan perbandingan dan konversi nilai. Isi

array “alternatif” ialah seperti pada Tabel 4.4. Jika direpresentasikan dengan matriks ialah sebagai berikut.

$$X = \begin{bmatrix} 1 & 1 & 80 & 1 & 1 & 4000 \\ 1 & 1 & 80 & 5 & 1 & 4000 \\ 1 & 1 & 80 & 4 & 1 & 4000 \\ 1 & 1 & 90 & 4 & 1 & 7000 \\ 1 & 1 & 80 & 2 & 1 & 4000 \\ 1 & 1 & 90 & 6 & 1 & 5500 \\ 1 & 0,75 & 90 & 7 & 1 & 5000 \\ 1 & 0,75 & 80 & 1 & 1 & 4000 \\ 1 & 1 & 80 & 3 & 1 & 5000 \\ 1 & 1 & 80 & 5 & 1 & 4000 \end{bmatrix}$$

B. Matriks R Ternormalisasi

Nilai tiap-tiap elemen matriks X kemudian dinormalisasi dengan menggunakan rumus (2.3). Hasil normalisasi seluruh elemen akan membentuk matriks yang dinamakan matriks R. Salah satu contoh perhitungan normalisasi elemen x_{ij} dengan $i = 1$ (pelamar 1) dan $j = 1$ (kriteria 1) dari matriks X ialah sebagai berikut.

$$r_{11} = \frac{x_{11}}{\sqrt{\sum_{i=1}^{n=10} x_{i1}^2}} = \frac{1}{\sqrt{x_{11}^2 + x_{21}^2 + x_{31}^2 + x_{41}^2 + x_{51}^2 + x_{61}^2 + x_{71}^2 + x_{81}^2 + x_{91}^2 + x_{101}^2}} = \frac{1}{\sqrt{10}}$$

$$= 0,3162$$

Tabel 4.5 Hasil Perhitungan Normalisasi Matriks X

Nama	C₁	C₂	C₃	C₄	C₅	C₆
(bobot)	(2)	(5)	(4)	(3)	(5)	(3)
Pelamar 1	0,3162	0,331	0,3043	0,0741	0,3162	0,2665
Pelamar 2	0,3162	0,331	0,3043	0,3706	0,3162	0,2665
Pelamar 3	0,3162	0,331	0,3043	0,2965	0,3162	0,2665
Pelamar 4	0,3162	0,331	0,3424	0,2965	0,3162	0,4664
Pelamar 5	0,3162	0,331	0,3043	0,1483	0,3162	0,2665
Pelamar 6	0,3162	0,331	0,3424	0,4448	0,3162	0,3665
Pelamar 7	0,3162	0,2483	0,3424	0,5189	0,3162	0,3331
Pelamar 8	0,3162	0,2483	0,3043	0,0741	0,3162	0,2665
Pelamar 9	0,3162	0,331	0,3043	0,2224	0,3162	0,3331
Pelamar 10	0,3162	0,331	0,3043	0,3706	0,3162	0,2665

Maka akan didapatkan data hasil normalisasi para pelamar seperti pada Tabel 4.5. Data dari Tabel 4.5 akan membentuk matriks R sesuai dengan rumus (2.2) seperti berikut ini:

$$R = \begin{bmatrix} 0,3162 & 0,331 & 0,3043 & 0,0742 & 0,3162 & 0,2665 \\ 0,3162 & 0,331 & 0,3043 & 0,3706 & 0,3162 & 0,2665 \\ 0,3162 & 0,331 & 0,3043 & 0,2965 & 0,3162 & 0,2665 \\ 0,3162 & 0,331 & 0,3424 & 0,2965 & 0,3162 & 0,4664 \\ 0,3162 & 0,331 & 0,3043 & 0,1483 & 0,3162 & 0,2665 \\ 0,3162 & 0,331 & 0,3424 & 0,4448 & 0,3162 & 0,3665 \\ 0,3162 & 0,2483 & 0,3424 & 0,5189 & 0,3162 & 0,3331 \\ 0,3162 & 0,2483 & 0,3043 & 0,0741 & 0,3162 & 0,2665 \\ 0,3162 & 0,331 & 0,3043 & 0,2224 & 0,3162 & 0,3331 \\ 0,3162 & 0,331 & 0,3043 & 0,3706 & 0,3162 & 0,2665 \end{bmatrix}$$

Gambar 4.6 menunjukkan kode untuk membuat matriks R dengan menggunakan rumus (2.3). Dapat dilihat bahwa terdapat dua buah *loop*. *Loop* yang pertama bertujuan mendapatkan nilai pembagi untuk masing-masing kolom. Nilai tersebut disimpan ke dalam *array* bernama “denominator”. Kemudian, *loop* yang kedua bertujuan untuk menghitung nilai r_{ij} tiap kolom untuk masing-masing pelamar. Dapat dilihat bahwa dilakukan pembulatan hingga 4 bilangan di belakang koma. Data hasil perhitungan akan disimpan ke dalam *array* bernama “matriksR”.

```
//hitung total kuadrat dari Xij perkriteria (denominator)
for($i=1; $i<=count($crit); $i++){
    $total = 0;
    foreach($alternatif as $idxAlt => $data) {
        $total += pow($data[$i], 2);
    }
    $denominator[$i] = round(sqrt($total), 4);
}

//hitung matriks R
$counter = 1;
foreach($alternatif as $idx => $data){
    $matriksR[$counter] = array( 'nama'=>$data['nama'],
        1=>round($data[1]/$denominator[1],4),
        2=>round($data[2]/$denominator[2],4),
        3=>round($data[3]/$denominator[3],4),
        4=>round($data[4]/$denominator[4],4),
        5=>round($data[5]/$denominator[5],4),
        6=>round($data[6]/$denominator[6],4),
        'applyid'=>$data['applyid']
    );
    $counter++;
}
```

Gambar 4.6 Kode Program Matriks R

C. Matriks V Terbobot

Setelah mendapatkan nilai matriks R, berikutnya akan dihitung nilai matriks V terbobot. Perhitungan menggunakan rumus (2.5). Untuk mendapatkan

nilai matriks V, tiap elemen matriks R akan dikalikan dengan bobot ternormalisasi dari masing-masing kriteria. Bobot masing-masing kriteria yang telah ditentukan pada Bab 3 harus dinormalisasi terlebih dahulu untuk memenuhi syarat untuk $\sum w = 1$. Normalisasi dilakukan dengan cara membagi nilai bobot sebuah kriteria dengan jumlah bobot seluruh kriteria. Berikut merupakan penyelesaian normalisasi bobot kriteria jenis kelamin.

$$w_1 = \frac{2}{2 + 5 + 4 + 3 + 5 + 3} = 0,09$$

Maka didapatkan bobot ternormalisasi masing-masing kriteria sebagai berikut.

$$W = [0,09 \quad 0,23 \quad 0,18 \quad 0,14 \quad 0,23 \quad 0,14]$$

Setelah itu perhitungan matriks V dapat dilakukan. Berikut salah satu penyelesaian perhitungan elemen matriks V untuk $i = 1$ dan $j = 1$ (nilai dari pelamar 1 untuk kriteria 1).

$$v_{11} = r_{11} \times w_1 = 0,3162 \times 0,09 = 0,0285$$

Dengan begitu didapatkan nilai matriks V yang disajikan pada Tabel 4.6.

Tabel 4.6 Hasil Perhitungan Matriks V Terbobot

Nama	C₁ (0,09)	C₂ (0,23)	C₃ (0,18)	C₄ (0,14)	C₅ (0,23)	C₆ (0,14)
Pelamar 1	0,0285	0,0761	0,0548	0,0104	0,0727	0,0373
Pelamar 2	0,0285	0,0761	0,0548	0,0519	0,0727	0,0373
Pelamar 3	0,0285	0,0761	0,0548	0,0415	0,0727	0,0373
Pelamar 4	0,0285	0,0761	0,0616	0,0415	0,0727	0,0653
Pelamar 5	0,0285	0,0761	0,0548	0,0208	0,0727	0,0373
Pelamar 6	0,0285	0,0761	0,0616	0,0623	0,0727	0,0513
Pelamar 7	0,0285	0,0571	0,0616	0,0726	0,0727	0,0466
Pelamar 8	0,0285	0,0571	0,0548	0,0104	0,0727	0,0373
Pelamar 9	0,0285	0,0761	0,0548	0,0311	0,0727	0,0466
Pelamar 10	0,0285	0,0761	0,0548	0,0519	0,0727	0,0373

Penyajian Tabel 4.6 dalam bentuk matriks sesuai dengan rumus (2.4) ialah sebagai berikut.

$$V = \begin{bmatrix} 0,0285 & 0,0761 & 0,0548 & 0,0104 & 0,0727 & 0,0373 \\ 0,0285 & 0,0761 & 0,0548 & 0,0519 & 0,0727 & 0,0373 \\ 0,0285 & 0,0761 & 0,0548 & 0,0415 & 0,0727 & 0,0373 \\ 0,0285 & 0,0761 & 0,0616 & 0,0415 & 0,0727 & 0,0653 \\ 0,0285 & 0,0761 & 0,0548 & 0,0208 & 0,0727 & 0,0373 \\ 0,0285 & 0,0761 & 0,0616 & 0,0623 & 0,0727 & 0,0513 \\ 0,0285 & 0,0571 & 0,0616 & 0,0726 & 0,0727 & 0,0466 \\ 0,0285 & 0,0571 & 0,0548 & 0,0104 & 0,0727 & 0,0373 \\ 0,0285 & 0,0761 & 0,0548 & 0,0311 & 0,0727 & 0,0466 \\ 0,0285 & 0,0761 & 0,0548 & 0,0519 & 0,0727 & 0,0373 \end{bmatrix}$$

Gambar 4.7 menunjukkan kode program perhitungan matriks V. Dapat dilihat pada Gambar 4.7, terdapat tiga buah *loop*. *Foreach loop* yang di awal berguna untuk menghitung total bobot kriteria sebelum normalisasi. *Foreach loop* kedua bertujuan untuk normalisasi bobot kriteria. Hasilnya disimpan ke dalam *array* bernama “critNormalized”. Sedangkan *for loop* yang terakhir untuk menghitung matriks V dengan cara mengalikan setiap elemen matriks R dengan bobot kriteria yang sudah dinormalisasi. Kemudian dilakukan pembulatan terhadap hasil hingga 4 angka di belakang koma. Hasil perhitungan disimpan ke dalam *array* bernama “matriks”

```
//normalisasi bobot kriteria
foreach ($crit as $idx => $criteria) {
    $totalcriteria += $criteria->weight;
}
$counter = 1;
foreach ($crit as $idx => $criteria) {
    $critNormalized[$counter]['nama'] = $criteria->crit_name;
    $critNormalized[$counter]['bobot'] = round($criteria->weight / $totalcriteria, 2);
    $critNormalized[$counter]['jenis'] = $criteria->att_id;
    $counter++;
}

$matriksV = array();
for($j=1; $j<=count($matriksR); $j++){
    $matriksV[$j] = array( 'nama'=>$matriksR[$j]['nama'],
        1=>round($matriksR[$j][1]*$critNormalized[1]['bobot'],4),
        2=>round($matriksR[$j][2]*$critNormalized[2]['bobot'],4),
        3=>round($matriksR[$j][3]*$critNormalized[3]['bobot'],4),
        4=>round($matriksR[$j][4]*$critNormalized[4]['bobot'],4),
        5=>round($matriksR[$j][5]*$critNormalized[5]['bobot'],4),
        6=>round($matriksR[$j][6]*$critNormalized[6]['bobot'],4),
        'applyid'=>$matriksR[$j]['applyid']
    );
}
```

Gambar 4.7 Kode Program Matriks V

D. Nilai Solusi Ideal Positif dan Negatif

Penentuan nilai solusi ideal positif dan negatif menggunakan masing-masing sebuah rumus. Rumusnya dibedakan untuk kriteria jenis *benefit* dan *cost*. Untuk kriteria berjenis *benefit*, nilai lebih tinggi lebih diprioritaskan. Nilainya didapatkan menggunakan rumus (2.6) dan (2.7). Sedangkan, untuk kriteria berjenis *cost*, nilai lebih rendah lebih diprioritaskan. Nilainya didapatkan

menggunakan rumus (2.8) dan (2.9). Pada penelitian ini, kriteria yang berjenis *cost* ialah gaji yang diharapkan. Kriteria selain dari gaji yang diharapkan merupakan kriteria *benefit*. Sebagai contoh, nilai solusi ideal positif kriteria Jenis Kelamin yang berjenis *benefit* didapatkan dengan mencari nilai maksimum dari kolom Jenis Kelamin pada Matriks V. Sedangkan untuk nilai solusi ideal negatif kriteria Jenis Kelamin yang berjenis *benefit* didapatkan dengan mencari nilai minimum dari kolom Jenis Kelamin pada Matriks V. Tabel 4.7 menyajikan nilai solusi ideal positif dan solusi ideal negatif dari masing-masing kriteria.

Tabel 4.7 Solusi Ideal Positif dan Negatif			
Kriteria	Jenis	Solusi Ideal Positif (A^+)	Solusi Ideal Negatif (A^-)
Jenis Kelamin	<i>Benefit</i>	0,0285	0,0285
Usia	<i>Benefit</i>	0,0761	0,0571
Pendidikan	<i>Benefit</i>	0,0616	0,0548
Pengalaman	<i>Benefit</i>	0,0726	0,0104
Kemampuan	<i>Benefit</i>	0,0727	0,0727
Gaji yang Diharapkan	<i>Cost</i>	0,0373	0,0653

Dapat dilihat pada Tabel 4.7, kriteria jenis kelamin dan kemampuan masing-masing memiliki pasangan nilai solusi ideal positif dan solusi ideal negatif yang sama. Hal ini disebabkan karena dua kriteria tersebut memiliki nilai yang seragam pada matriks V. Untuk mendapatkan nilai solusi ideal positif dari kriteria berjenis *benefit* dicari nilai maksimum dari seluruh alternatif terkait kriteria tersebut. Untuk mendapatkan nilai solusi ideal negatif dari kriteria berjenis *benefit* dicari nilai minimum dari seluruh alternatif terkait kriteria tersebut. Dikarenakan nilai seluruh alternatif dalam kriteria jenis kelamin ialah seragam, yaitu 0.0285, maka nilai maksimum dan minimumnya pun 0,0285. Sama halnya dengan kriteria kemampuan.

Kriteria gaji yang diharapkan memiliki nilai solusi ideal positif yang lebih rendah daripada nilai solusi ideal negatif. Hal ini disebabkan oleh jenis kriteria *cost*. Untuk jenis kriteria ini, solusi ideal yang paling baik ialah yang memiliki

nilai paling rendah. Sedangkan untuk solusi ideal yang paling buruk ialah yang memiliki nilai tertinggi.

```
//tentukan nilai atribut maksimum dan minimum per kolom
$A = array(); //struktur : [kriteria ke-][Apos/Aneg]

for($k=1; $k<=count($critNormalized); $k++){
    $min = 1;
    $max = 0;
    foreach($matriksv as $idx=>$v){
        if($v[$k] > $max){ $max = $v[$k];}

        if($v[$k] < $min){ $min = $v[$k];}
    }
    if($critNormalized[$k]['jenis']==1){
        $A[$k]['Apos'] = $max;
        $A[$k]['Aneg'] = $min;
    }
    else if($critNormalized[$k]['jenis']==2){
        $A[$k]['Apos'] = $min;
        $A[$k]['Aneg'] = $max;
    }
}
```

Gambar 4.8 Kode Program Solusi Ideal Positif dan Negatif

Gambar 4.8 menunjukkan kode program untuk menentukan solusi ideal positif dan negatif. Program melakukan iterasi pada setiap kriteria. Kemudian menentukan nilai maksimum dan minimum dari seluruh alternatif pada kriteria tersebut dengan melakukan perbandingan. Setelah itu melakukan pengecekan jenis kriteria. Apabila kriteria berjenis *benefit*, maka simpan nilai maksimum sebagai nilai solusi ideal positif dan nilai minimum sebagai nilai solusi ideal negatif. Jika kriteria *cost*, maka lakukan sebaliknya. Hasilnya akan disimpan ke sebuah *array* bernama “A”.

E. Jarak Manhattan dan PIA

Jarak Manhattan dihitung untuk mengetahui total jarak masing-masing pelamar terhadap solusi ideal positif dan solusi ideal negatif dari seluruh kriteria. Masing-masing pelamar memiliki jarak Manhattan positif dan jarak Manhattan negatif. Jarak Manhattan positif merupakan nilai total jarak pelamar dari solusi ideal positif seluruh kriteria. Sedangkan, jarak Manhattan negatif merupakan nilai total jarak pelamar dari solusi ideal negatif seluruh kriteria. Untuk menghitung jarak Manhattan dapat menggunakan rumus (2.10) dan (2.11). Penghitungan jarak dilakukan dengan mencari selisih antara tiap-tiap nilai pelamar di matriks V dengan solusi ideal positif dan negatif masing-masing kriteria. Kemudian, jarak Manhattan positif akan didapatkan dengan menjumlahkan seluruh hasil

perhitungan selisih nilai matriks V terhadap solusi ideal positif. Jarak Manhattan negatif akan didapatkan dengan menjumlahkan seluruh hasil perhitungan selisih nilai matriks V terhadap solusi ideal negatif. Berikut salah satu penyelesaian untuk menghitung jarak Manhattan positif (D+) dan negatif (D-) dari pelamar 1 dengan $i = 1$.

$$D^+_1 = |v_{11} - a^+_1| + |v_{12} - a^+_2| + |v_{13} - a^+_3| + |v_{14} - a^+_4| + |v_{15} - a^+_5| + |v_{16} - a^+_6|$$

$$D^+_1 = |0,0285 - 0,0285| + |0,0761 - 0,0761| + |0,0548 - 0,0616| + |0,0104 - 0,0726| + |0,0727 - 0,0727| + |0,0373 - 0,0373| = 0,0690$$

$$D^-_1 = |v_{11} - a^-_1| + |v_{12} - a^-_2| + |v_{13} - a^-_3| + |v_{14} - a^-_4| + |v_{15} - a^-_5| + |v_{16} - a^-_6|$$

$$D^-_1 = |0,0285 - 0,0285| + |0,0761 - 0,0571| + |0,0548 - 0,0548| + |0,0104 - 0,0104| + |0,0727 - 0,0727| + |0,0373 - 0,0653| = 0,047$$

Dengan itu didapatkan jarak Manhattan positif dan negatif dari pelamar 1 secara berurut ialah 0,0690 dan 0,0470. Tabel 4.8 menyajikan nilai jarak Manhattan positif dan negatif dari seluruh pelamar.

Tabel 4.8 Jarak Manhattan

Nama	D+	D-
Pelamar 1	0,069	0,047
Pelamar 2	0,0275	0,0885
Pelamar 3	0,0379	0,0781
Pelamar 4	0,0591	0,0569
Pelamar 5	0,0586	0,0574
Pelamar 6	0,0243	0,0917
Pelamar 7	0,0283	0,0877
Pelamar 8	0,088	0,028
Pelamar 9	0,0576	0,0584
Pelamar 10	0,0275	0,0885

Dari Tabel 4.8, nilai D+ maksimum dimiliki oleh Pelamar 8 sebesar 0,088 dan D+ minimum dimiliki oleh Pelamar 6 sebesar 0,0243. Hal ini menyatakan

bahwa Pelamar 8 memiliki jarak paling jauh ke Solusi Ideal Positif dan Pelamar 2 memiliki jarak terdekat ke Solusi Ideal Positif.

Nilai D- maksimum dimiliki oleh Pelamar 6 sebesar 0,0917 dan D-minimum dimiliki oleh Pelamar 8 sebesar 0,028. Hal itu menyatakan bahwa Pelamar 6 memiliki jarak terjauh dengan Solusi Ideal Negatif dan Pelamar 8 memiliki jarak terdekat dengan Solusi Ideal Negatif.

Nilai PIA didapatkan dengan menggunakan rumus (2.12). Nilai PIA adalah nilai minimum D+ dan nilai maksimum D-. Maka dari itu nilai PIA ialah sebagai berikut.

$$PIA = \{0,0243, 0,0917\}$$

Gambar 4.9 merupakan kode program untuk melakukan perhitungan jarak Manhattan dan PIA. Variabel “D” menyimpan data *array* dari jarak Manhattan positif dan negatif tiap pelamar. Variabel “Dpmin” menyimpan salah satu nilai PIA, yaitu nilai minimum dari jarak Manhattan positif (D+). Variabel “Dnmax” menyimpan salah satu nilai PIA, yaitu nilai maksimum dari jarak Manhattan negatif (D-). Di dalam *for loop* dilakukan perhitungan selisih antara nilai vektor V dengan nilai solusi ideal positif dan negatif. Kemudian hasilnya akan ditotalkan dan disimpan ke dalam *array* “D”. Setelah itu akan dilakukan perbandingan untuk mendapatkan nilai PIA.

```
$D = array();
$Dpmin = 1; //PIA
$Dnmax = 0; //PIA

foreach ($matrksv as $idx => $V) {
    $jml = 0;
    $jml2 = 0;

    for($b=1; $b<=count($critNormalized); $b++){
        $jml += abs($V[$b]-$A[$b]['Apos']);
        $jml2 += abs($V[$b]-$A[$b]['Aneg']);
    }

    $D[$idx]['pos'] = $jml;
    $D[$idx]['neg'] = $jml2;
    $D[$idx]['applyid'] = $V['applyid'];
    if($D[$idx]['pos'] < $Dpmin){ $Dpmin = $D[$idx]['pos']; }
    if($D[$idx]['neg'] > $Dnmax){ $Dnmax = $D[$idx]['neg']; }
}
```

Gambar 4.9 Kode Program Jarak Manhattan dan PIA

F. Jarak Alternatif ke PIA

Setelah mendapatkan nilai PIA, jarak alternatif ke PIA dapat dihitung menggunakan rumus (2.13). Salah satu penyelesaian perhitungan R_i untuk pelamar 1 dengan $i = 1$ adalah sebagai berikut.

$$R_1 = \sqrt{(0,069 - 0,0243)^2 + (0,047 - 0,0917)^2} = \sqrt{0,0020 + 0,0020} \\ = 0,0632$$

Diketahui bahwa nilai R_i untuk pelamar 1 ialah 0,0632. Dengan menggunakan cara yang sama didapatkan nilai R_i untuk setiap pelamar. Tabel 4.9 menyajikan nilai jarak alternatif ke PIA dari masing-masing pelamar beserta peringkat mereka. Semakin kecil nilai R_i maka alternatif tersebut semakin direkomendasikan.

Tabel 4.9 Jarak Alternatif ke PIA		
Nama	R_i	Peringkat
Pelamar 1	0,0632	9
Pelamar 2	0,0045	2
Pelamar 3	0,0192	5
Pelamar 4	0,0492	8
Pelamar 5	0,0485	7
Pelamar 6	0	1
Pelamar 7	0,0057	4
Pelamar 8	0,0901	10
Pelamar 9	0,0471	6
Pelamar 10	0,0045	3

Dari Tabel 4.9, dapat dilihat 5 orang pelamar dengan peringkat teratas ialah Pelamar 6, Pelamar 2, Pelamar 10, Pelamar 7, dan Pelamar 3. Dua orang pelamar dengan posisi terendah ialah Pelamar 1 dan Pelamar 10. Tabel 4.10 menyajikan data kualifikasi alternatif setelah diurutkan sesuai peringkat. Jika diperhatikan, pelamar yang menduduki peringkat 1 hingga 3 seluruhnya memenuhi kriteria. Dari peringkat 1 hingga 3, masing-masing pelamar mendapatkan nilai maksimal pada kriteria jenis kelamin, usia, serta kemampuan.

Sehingga yang menjadi penentu ialah dari kriteria pendidikan yang memiliki tingkat prioritas Penting dan juga kriteria pengalaman. Dalam hal ini, Pelamar 6 memiliki pendidikan dan pengalaman yang lebih tinggi dari dua lainnya. Maka dari itu, Pelamar 6 lebih unggul dari yang lain. Pelamar 6 dapat menduduki peringkat 1 meskipun gaji yang diharapkannya melebihi batas. Hal ini dapat terjadi karena tingkat prioritas kriteria gaji yang diharapkan adalah normal, sehingga kekurangan nilainya dapat ditutupi oleh nilai dari kriteria pendidikan dan pengalaman.

Tabel 4.10 Data Kualifikasi Pelamar Setelah Pemeringkatan

Nama	C₁	C₂	C₃	C₄	C₅	C₆
(bobot)	(2)	(5)	(4)	(3)	(5)	(3)
Kualifikasi	L/P	23 – 35 th	S1	2 th	Ms Office	3jt – 4jt
Pelamar 6	L	35	S2	6	MS Office	5.500
Pelamar 2	L	30	S1	5	MS Office	4.000
Pelamar 10	L	33	S1	5	MS Office	4.000
Pelamar 7	P	36	S2	7	MS Office	5.000
Pelamar 3	L	27	S1	4	MS Office	4.000
Pelamar 9	L	32	S1	3	MS Office	5.000
Pelamar 5	L	24	S1	2	MS Office	4.000
Pelamar 4	L	29	S2	4	MS Office	7.000
Pelamar 1	P	23	S1	1	MS Office	4.000
Pelamar 8	L	22	S1	1	MS Office	4.000

Pada peringkat ke-empat, terdapat Pelamar 7. Pelamar 7 memiliki ketidaksesuaian pada kriteria usia, yaitu usia dari Pelamar 7 lebih besar setahun daripada yang disyaratkan. Dari kriteria usia, Pelamar 7 tidak mendapatkan nilai maksimal. Namun, kekurangan nilainya dapat terbantu dari nilai kriteria berprioritas tinggi, yaitu pendidikan serta pengalaman yang tergolong cukup besar. Pelamar 7 adalah pelamar yang memiliki pengalaman kerja terpanjang di antara yang lainnya. Terlebih lagi, pelamar 7 memiliki pendidikan yang lebih unggul daripada yang lain. Maka dari itu, pelamar 7 layak direkomendasikan.

Pelamar 4 mendapatkan nilai maksimal pada kriteria jenis kelamin, usia, dan kemampuan. Namun, Pelamar 4 berada pada posisi tiga terbawah. Kriteria yang menghalangi pelamar 4 untuk naik ke atas ialah kriteria gaji yang diharapkan. Pelamar 4 merupakan pemilik nilai kriteria C6 terbesar. Kriteria C6

berjenis *cost*; artinya adalah pelamar dengan nilai tertinggi maka kurang diprioritaskan. Meski tingkat prioritas kriteria C6 hanyalah normal, namun nilai gaji yang diharapkan pelamar 4 memiliki rentang yang terlalu besar dibanding dengan pelamar lainnya, maka ia tidak diprioritaskan.

Pelamar 1 dan 8 memiliki kualifikasi yang tidak jauh berbeda. Pelamar 1 mendapatkan nilai maksimal pada kriteria jenis kelamin, usia, pendidikan, serta kemampuan. Namun, pelamar 1 masih tertinggal jauh dengan yang lain pada kriteria pengalaman kerja. Nilai kriteria gaji yang diharapkan tidak mampu mendongkrak peringkat pelamar 1, karena mayoritas pelamar yang lain juga memiliki nilai rentang gaji yang tak jauh berbeda. Hal yang sama terjadi pada pelamar 8. Namun, pelamar 8 lebih tidak direkomendasikan daripada pelamar 1 karena nilai kriteria usianya yang tidak maksimal layaknya milik pelamar 1.

Gambar 4.10 menunjukkan kode program untuk menghitung nilai Ri. Nilai Ri atau nilai akhir masing-masing pelamar disimpan dalam variabel bernama “RI”. Dalam kode program, khususnya bagian yang menggunakan kondisi “If.. Else” berguna untuk menyimpan nilai terkait perhitungan DiA dari masing-masing pelamar ke basis data. Apabila terdapat kesalahan dalam penyimpanan, maka penyimpanan akan dibatalkan.

```

$RI = array();
$sarDIA = array();

$transaction = Yii::app()->db->beginTransaction();
try{
    foreach ($D as $idx => $d) {
        $b = $d['pos'] - $Dpmin;
        $t = $d['neg'] - $Dnmax;

        $B = pow($b, 2);
        $T = pow($t, 2);
        $S = sqrt($B+$T);

        $modelDIA = Diareresult::model()->findByAttributes(array('dia_apply_id'=>$d['applyid']));
        if(count($modelDIA)>0){ //UPDATE
            $modelDIA->dia_ri = round($S, 4);
            $modelDIA->dia_status = $alternatif[$idx]['stat'];
            $modelDIA->dia_deviasi = $alternatif[$idx]['deviasi'];
            $modelDIA->save();
        }
        else { //CREATE
            $modelDIA = new Diareresult;
            $modelDIA->dia_apply_id = $d['applyid'];
            $modelDIA->dia_ri = round($S, 4);
            $modelDIA->dia_status = $alternatif[$idx]['stat'];
            $modelDIA->dia_deviasi = $alternatif[$idx]['deviasi'];
            $modelDIA->save();
        }

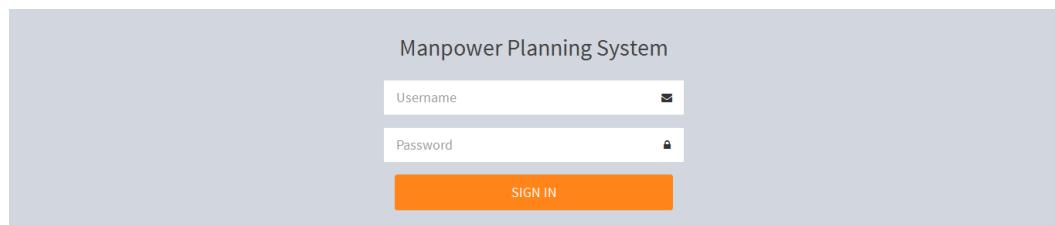
        $RI[$idx] = $modelDIA->dia_ri;
    }
    $transaction->commit();
}
catch(Exception $e){
    $transaction->rollback();
}

```

Gambar 4.10 Kode Program Perhitungan RI

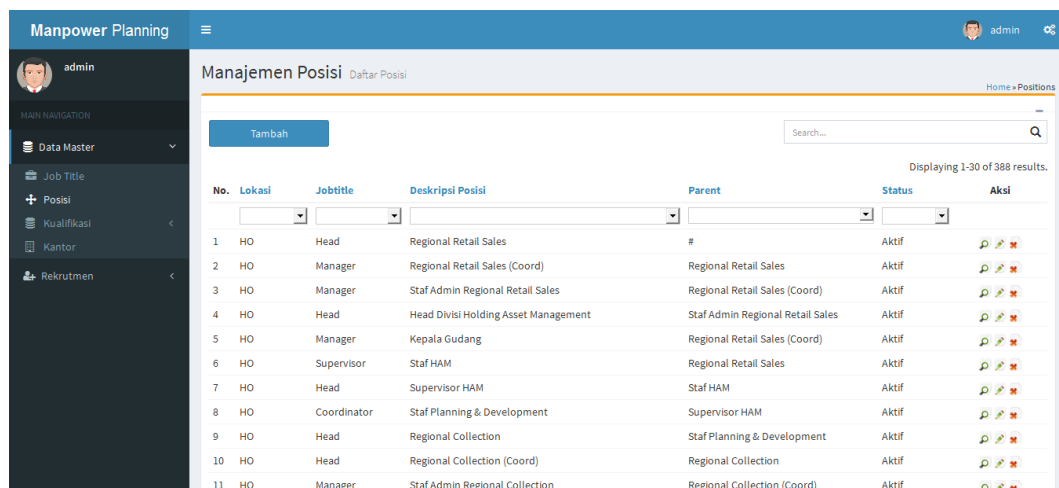
4.2.2 Implementasi Antarmuka

Implementasi antarmuka sistem dilakukan berdasarkan pada perencanaan antarmuka yang telah dilakukan. Implementasi antar muka sistem menggunakan sebuah *template* HTML responsif *open source* yang didapatkan secara bebas melalui internet. Hal ini dilakukan bertujuan untuk memangkas waktu yang diperlukan untuk implementasi antarmuka sistem. *Template* yang disebut AdminLTE 2 memiliki ketergantungan pada *Bootstrap 3*, *JQuery 1.11+*, serta *plugin* lainnya agar bisa menampilkan elemen antarmuka dengan baik. Berikut hasil implementasi antarmuka sistem.



Gambar 4.11 Halaman Login

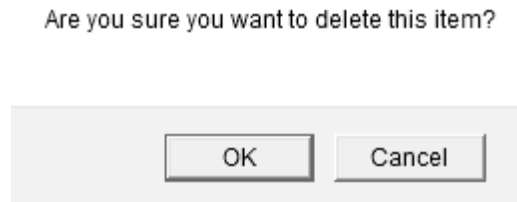
Gambar 4.11 menunjukkan halaman awal sistem, yaitu halaman *login*. Pengguna diharuskan memasukkan *username* dan *password* yang sesuai untuk bisa masuk ke dalam sistem.



Gambar 4.12 Halaman Lihat Daftar Posisi

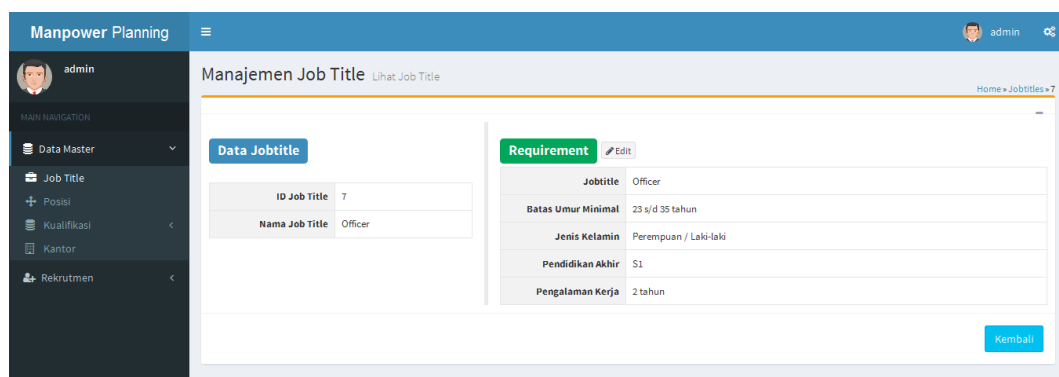
Gambar 4.12 merupakan halaman untuk melihat daftar posisi yang tersimpan dalam sistem. Kolom Aksi terdiri dari tiga buah tombol. Tombol kaca

pembesar berguna untuk melihat seluruh informasi terkait posisi pada baris tersebut. Tombol pensil berguna untuk mengubah informasi terkait posisi pada baris tersebut. Tombol silang berwarna merah berguna untuk menghapus informasi posisi pada baris tersebut.



Gambar 4.13 *Pop-Up* Hapus Posisi

Gambar 4.13 merupakan tampilan *pop-up* ketika tombol hapus ditekan.



Gambar 4.14 Halaman Lihat *Job Title*

Gambar 4.14 merupakan halaman melihat sebuah *Job Title*. Pada halaman ini terdapat informasi kualifikasi untuk *Job Title* tersebut. Tombol *edit* di samping label *Requirement* akan mengarahkan pengguna ke halaman untuk mengubah *requirement* dari *job title*.

Gambar 4.15 merupakan halaman untuk membuat lowongan pekerjaan. Label dengan *asterisk* (*) berwarna merah merupakan kolom yang wajib diisi. Tombol simpan akan menyimpan data yang telah dimasukkan ke dalam basis data.

Gambar 4.16 menunjukkan halaman untuk menambah pelamar. Informasi pelamar dibagi menjadi empat bagian, yaitu informasi dasar, informasi diri, informasi kontak, dan kolom kualifikasi. Kolom terakhir merupakan kolom untuk

memasukkan data-data yang dibutuhkan untuk pelaksanaan seleksi lamaran. Untuk dapat melakukan seleksi lamaran dengan baik, maka informasi terkait riwayat pendidikan, pengalaman, kemampuan, dan gaji yang diharapkan harus dimasukkan oleh pengguna.

Manpower Planning admin

Manajemen Lowongan Pekerjaan Tambah Lowongan Pekerjaan

Home • Vacancies • Create

Fields with * are required.

ID Lowongan Pekerjaan Auto-generated

Kantor * Kantor 1

Posisi * Regional Retail Sales

Status Pegawai * Kontrak I

Deskripsi Deskripsi Lowongan Pekerjaan

Status Lowongan Aktif

Tanggal Buka

Tanggal Tutup

Jumlah Kebutuhan Pegawai *

Kembali Simpan

Gambar 4.15 Halaman Tambah Lowongan Pekerjaan

Manpower Planning admin

Manajemen Pelamar [Tambah Pelamar](#) [Home](#) [Applicants](#) [Create](#)

Informasi Dasar

Fields with * are required.

Nama Lengkap *
Tempat Lahir *
Tanggal Lahir * format: dd-mm-yyyy

Jenis Identitas *
No. Identitas *
Validitas Identitas format: dd-mm-yyyy

Agama *
Kewarganegaraan *
Jenis Kelamin * ☐ Laki-laki ☐ Perempuan

Informasi Diri

Tinggi Badan cm
 Berat Badan kg

Status Perkawinan *
Tanggal Perkawinan

Golongan Darah * ☐ A ☐ B ☐ AB ☐ O

Informasi Kontak

Alamat *

Kota *

Kode Pos *

Nomor HP *

Telepon *

Email *

Fax

Pendidikan | Pengalaman | Kemampuan | Resume

Riwayat Pendidikan

Pengalaman Kerja

Kemampuan

Gaji yang Diharapkan *

Surat Lamaran

 Ekstensi file: *.jpg *.jpeg *.doc *.docx *.pdf

Gambar 4.16 Halaman Tambah Pelamar

Gambar 4.17 menunjukkan halaman awal subkategori seleksi resume. Pengguna diharuskan untuk memilih kolom posisi yang dilamar sebelum melakukan langkah selanjutnya. Selain itu, daftar pelamar posisi juga harus berisi minimal tiga orang pelamar untuk bisa melanjutkan ke tahap selanjutnya.

Gambar 4.17 Halaman Awal Seleksi Resume

No.	Nama Kriteria	Tingkat Kepentingan	Atribut	Aksi
1	Jenis Kelamin	2	Benefit	
2	Usia	5	Benefit	
3	Pendidikan	4	Benefit	
4	Pengalaman Kerja	3	Benefit	
5	Kemampuan	5	Benefit	
6	Gaji yang diharapkan	3	Cost	

Gambar 4.18 Halaman Pemilihan Tingkat Prioritas Kriteria

Gambar 4.18 menunjukkan halaman pemilihan tingkat prioritas kriteria. Pengguna dapat menekan tombol bergambar pensil untuk mengubah tingkat kepentingan dan jenis kriteria baris tertentu.

Gambar 4.19 Halaman Ubah Kriteria

Gambar 4.19 merupakan tampilan untuk mengubah nilai prioritas kriteria. Jenis kriteria juga dapat diubah menjadi *benefit* atau *cost*.

Manpower Planning | admin

Manajemen Data Alternatif | Lihat Nilai Alternatif

Corporate Legal Officer

Requirement

Jenis Kelamin	Usia	Pendidikan	Pengalaman	Kemampuan
Perempuan / Laki-laki	23 s/d 35 tahun	S1	2 tahun	MS Office

No.	Nama Pelamar	Jenis Kelamin	Usia	Pendidikan	Pengalaman Kerja	Kemampuan	Gaji yang diharapkan
1	Pelamar 1	Perempuan	23 tahun	S1	1 tahun	MS Office	Rp. 4.000.000
2	Pelamar 2	Laki-laki	30 tahun	S1	5 tahun	MS Office	Rp. 4.000.000
3	Pelamar 3	Laki-laki	27 tahun	S1	4 tahun	MS Office	Rp. 4.000.000
4	Pelamar 4	Laki-laki	29 tahun	S2	4 tahun	MS Office	Rp. 7.000.000
5	Pelamar 5	Laki-laki	24 tahun	S1	2 tahun	MS Office	Rp. 4.000.000
6	Pelamar 6	Laki-laki	35 tahun	S2	6 tahun	MS Office	Rp. 5.500.000
7	Pelamar 7	Perempuan	36 tahun	S2	7 tahun	MS Office	Rp. 5.000.000
8	Pelamar 8	Laki-laki	22 tahun	S1	1 tahun	MS Office	Rp. 1.000.000
9	Pelamar 9	Laki-laki	32 tahun	S1	3 tahun	MS Office	Rp. 5.000.000
10	Pelamar 10	Laki-laki	33 tahun	S1	5 tahun	MS Office	Rp. 4.000.000

[Kembali](#) [Lakukan Seleksi](#)

Gambar 4.20 Halaman Melihat Daftar Data Pelamar Seleksi

Gambar 4.20 menunjukkan halaman melihat daftar data kualifikasi pelamar sebelum diseleksi. Informasi para pelamar yang akan diseleksi serta informasi kebutuhan kualifikasi dari posisi yang dilamar ditampilkan dalam halaman ini. Apabila ternyata kebutuhan kualifikasi posisi tersebut belum ditentukan sebelumnya, maka sistem akan menghalangi pengguna melanjutkan ke tahap berikutnya dan memberikan notifikasi untuk membuat kebutuhan kualifikasi posisi terlebih dahulu. Jika tombol “Lakukan Seleksi” ditekan, maka sistem akan memulai proses perhitungan dengan metode DiA.

Gambar 4.21 merupakan halaman yang muncul setelah seleksi selesai dilakukan. Terdapat sebuah tabel hasil pemeringkatan pelamar. Tabel terdiri dari kolom nama, nilai Ri, status kecocokan, dan deviasi. Kolom deviasi menunjukkan nama kualifikasi yang tidak dipenuhi oleh pelamar dengan status kecocokan tidak sesuai. Tabel tersebut akan menampilkan lima pelamar per halamannya.

Manajemen Hasil SPK Lihat Hasil Perhitungan SPK

Home » Seleksi » nilai

Corporate Legal Officer

Requirement

Jenis Kelamin	Usia	Pendidikan	Pengalaman	Kemampuan
Perempuan / Laki-Laki	23 s/d 35 tahun	S1	2 tahun	MS Office

Hasil Perankingan

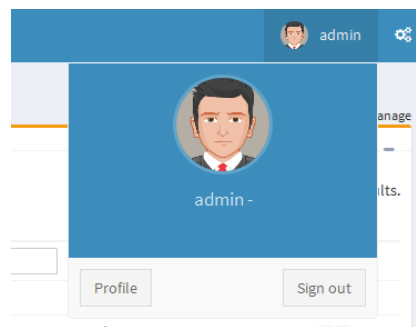
time elapsed : 0.76440095901489 detik.

Displaying 1-5 of 10 results.

No.	Nama	Nilai RI	Status Kecocokan	Deviasi
1	Pelamar 6	0.0000	Sesuai	
2	Pelamar 2	0.0038	Sesuai	
3	Pelamar 10	0.0038	Sesuai	
4	Pelamar 7	0.0055	Tidak Sesuai	Usia
5	Pelamar 3	0.0185	Sesuai	

Go to page: 1 2 Next >

Gambar 4.21 Halaman Lihat Hasil Seleksi



Gambar 4.22 Tampilan Pop-Up Menu Sign Out

Gambar 4.22 menunjukkan tampilan *pop-up* menu yang menyediakan tombol *sign out* dari sistem.

4.3 Pengujian Algoritma

Pengujian merupakan salah satu tahap penting dalam pengembangan sistem. Pada tahap ini dilakukan serangkaian pengujian terhadap sistem yang dibangun, khususnya pada implementasi algoritma di sistem. Hal ini dilakukan untuk memastikan bahwa metode DiA yang diimplementasikan ke dalam sistem telah berjalan sebagaimana mestinya. Pengujian sistem dilakukan dengan dua metode, yaitu *white box* dan *black box*. Pelaksanaan masing-masing metode yang digunakan akan dijabarkan pada subbab setelah ini.

4.3.1 Pengujian *White Box*

Pengujian *White box* diterapkan pada beberapa fungsi yang melakukan perhitungan metode DiA. Hal ini dilakukan untuk mencari tahu keberadaan *error* dalam aplikasi dengan mempertimbangkan mekanisme internal sistem (Khan, 2011). Pengujian *White box* ini dilakukan oleh penulis. Beberapa *test case* akan digunakan sebagai acuan dalam pengujian. Dalam pengujian ini juga dilakukan validasi hitungan rumus. Validasi dilakukan dengan cara membandingkan hasil perhitungan menggunakan *tools Microsoft Excel* dengan keluaran sistem. Hasil dari pengujian *White box* dapat dilihat di bawah ini.

Nama Fungsi	: matriksX()
Deskripsi	: Fungsi ini digunakan untuk membuat sebuah <i>array</i> multidimensi yang berisi nilai kualifikasi yang telah dikonversi (matriks X) serta nilai kecocokan dan deviasi masing-masing pelamar.
Input	: Sebuah <i>array</i> multidimensi dari kelas <i>Application</i> yang berisi daftar ID aplikasi lamaran yang akan diseleksi.
Output	: Sebuah <i>array</i> multidimensi berisi nilai kualifikasi yang telah dikonversi (matriks X) serta nilai kecocokan dan deviasi masing-masing pelamar.

Tabel 4.11 Hasil Pengujian Fungsi matriksX()

No.	Input	Expected Output	Actual Output	Status
1.	[0]=> [‘apply_id’]=> 22, [1]=> [‘apply_id’]=> 23	[1][‘nama’]=>Min Yoongi. [1][1]=> 1, [1][2]=>1, [1][3]=>90, [1][4]=>10, [1][5]=>1, [1][6]=>5000, [1][‘applyid’]=>22, [1][‘stat’]=>1, [1][‘deviasi’]=>’’ [2][‘nama’]=>Addina NR. [2][1]=> 1, [1][2]=>0.75,	[1][‘nama’]=>Min Yoongi. [1][1]=> 1, [1][2]=>1, [1][3]=>90, [1][4]=>10, [1][5]=>1, [1][6]=>5000, [1][‘apply_id’]=>22, [1][‘stat’]=>1, [1][‘deviasi’]=>’’ [2][‘nama’]=>Addina NR. [2][1]=> 1, [1][2]=>0.75,	<i>True</i>

		[2][3]=>50, [1][4]=>5, [2][5]=>0.5, [1][6]=>8000, [2]['applyid']=>23, [2]['stat']=>0, [2]['deviasi']=>Pendidikan Kemampuan	[2][3]=>50, [1][4]=>5, [2][5]=>0.5, [1][6]=>8000, [2]['apply_id']=>23, [2]['stat']=>0, [2]['deviasi']=>Pendidikan Kemampuan	
--	--	---	--	--

Nama Fungsi : matriksR()

Deskripsi : Fungsi ini digunakan untuk membuat sebuah *array* multidimensi yang berisi nilai dari matriks X setelah dinormalisasi menggunakan jarak *Euclidean*.

Input : Sebuah *array* multidimensi dari kelas *Criteria* yang berisi daftar objek kriteria yang digunakan untuk penilaian.

Output : Sebuah *array* multidimensi berisi nilai kualifikasi yang telah dinormalisasi (matriks R).

Tabel 4.12 Hasil Pengujian Fungsi matriksR()

No.	Input	Expected Output	Actual Output	Status
1.	[0]=> ['crit_id']=>1, [1]=> ['crit_id']=>2, [2]=> ['crit_id']=>3, [3]=> ['crit_id']=>4, [4]=> ['crit_id']=>5, [5]=> ['crit_id']=>6	[1]['nama']=> Min Yoongi [1][1]=> 0. 7071, [1][2]=>0. 8, [1][3]=> 0. 8742, [1][4]=>0. 8944, [1][5]=> 0. 8945, [1][6]=> 0. 53, [1]['applyid']=>22, [2]['nama']=> Addina NR [2][1]=> 0. 7071, [2][2]=> 0. 6, [2][3]=> 0. 4856, [2][4]=> 0. 4472, [2][5]=> 0. 4472, [2][6]=> 0. 848, [2]['applyid']=>23	[1]['nama']=>Min Yoongi [1][1]=> 0. 7071, [1][2]=>0. 8, [1][3]=> 0. 8742, [1][4]=>0. 8944, [1][5]=> 0. 8945, [1][6]=> 0. 53, [1]['apply_id']=>22, [2]['nama']=> Addina NR [2][1]=> 0. 7071, [2][2]=> 0. 6, [2][3]=> 0. 4856, [2][4]=> 0. 4472, [2][5]=> 0. 4472, [2][6]=> 0. 848, [2]['apply_id']=>23	True

Nama Fungsi : `matriksV()`

Deskripsi : Fungsi ini digunakan untuk membuat sebuah *array* multidimensi yang berisi nilai dari matriks R setelah dikalikan dengan bobot kriteria masing-masing yang telah dinormalisasi.

Input : Sebuah *array* multidimensi dari kelas *Criteria* yang berisi daftar objek kriteria yang digunakan untuk penilaian.

Output : Sebuah *array* multidimensi berisi nilai matriks R yang telah dikalikan dengan bobot masing-masing kriteria.

Tabel 4.13 Hasil Pengujian Fungsi `matriksV()`

No.	Input	Expected Output	Actual Output	Status
1.	<code>[0]=></code> <code>['crit_id']=>1,</code> <code>[1]=></code> <code>['crit_id']=>2,</code> <code>[2]=></code> <code>['crit_id']=>3,</code> <code>[3]=></code> <code>['crit_id']=>4,</code> <code>[4]=></code> <code>['crit_id']=>5,</code> <code>[5]=></code> <code>['crit_id']=>6</code>	<code>[1]['nama']=> Min Yoongi</code> <code>[1][1]=> 0. 0636,</code> <code>[1][2]=>0. 184,</code> <code>[1][3]=> 0. 1574,</code> <code>[1][4]=>0. 1252,</code> <code>[1][5]=> 0. 2057,</code> <code>[1][6]=> 0. 0742,</code> <code>[1]['applyid']=>22,</code> <code>[2]['nama']=> Addina NR</code> <code>[2][1]=> 0. 0636,</code> <code>[2][2]=> 0. 138,</code> <code>[2][3]=> 0. 0874,</code> <code>[2][4]=> 0. 0626,</code> <code>[2][5]=> 0. 1029,</code> <code>[2][6]=> 0. 1187,</code> <code>[2]['applyid']=>23</code>	<code>[1]['nama']=>Min Yoongi</code> <code>[1][1]=> 0. 0636,</code> <code>[1][2]=>0. 184,</code> <code>[1][3]=> 0. 1574,</code> <code>[1][4]=>0. 1252,</code> <code>[1][5]=> 0. 2057,</code> <code>[1][6]=> 0. 0742,</code> <code>[1]['applyid']=>22,</code> <code>[2]['nama']=> Addina NR</code> <code>[2][1]=> 0. 0636,</code> <code>[2][2]=> 0. 138,</code> <code>[2][3]=> 0. 0874,</code> <code>[2][4]=> 0. 0626,</code> <code>[2][5]=> 0. 1029,</code> <code>[2][6]=> 0. 1187,</code> <code>[2]['applyid']=>23</code>	<i>True</i>

Nama Fungsi : `solusiIdeal()`

Deskripsi : Fungsi ini digunakan untuk mencari nilai solusi ideal positif dan solusi ideal negatif dari setiap kolom di matriks V. Nilai solusi ideal positif ialah nilai terbesar yang ada di tiap-tiap kolom berjenis *benefit*. Sedangkan solusi ideal negatif ialah

nilai terkecil yang ada di tiap-tiap kolom berjenis *benefit*.
Belaku sebaliknya untuk kriteria berjenis *cost*.

Input : Sebuah *array* multidimensi yang berisi matriks V.

Output : Sebuah *array* multidimensi berisi nilai solusi ideal positif dan negatif dari setiap kriteria.

Tabel 4.14 Hasil Pengujian Fungsi solusiIdeal()

No.	Input	Expected Output	Actual Output	Status
1.	[1]['nama']=> Min Yoongi [1][1]=> 0. 0636, [1][2]=>0. 184, [1][3]=> 0. 1574, [1][4]=>0. 1252, [1][5]=> 0. 2057, [1][6]=> 0. 0742, [1]['applyid']=>22, [2]['nama']=> Addina NR [2][1]=> 0. 0636, [2][2]=> 0. 138, [2][3]=> 0. 0874, [2][4]=> 0. 0626, [2][5]=> 0. 1029, [2][6]=> 0. 1187, [2]['applyid']=>23	[1][Apos]=>0. 0636, [1][Aneg]=>0. 0636, [2][Apos] =>0. 184, [2][Aneg] =>0. 138, [3][Apos] =>0. 1574, [3][Aneg] =>0. 0874, [4][Apos] =>0. 1252, [4][Aneg] =>0. 0626, [5][Apos] =>0. 2057, [5][Aneg] =>0. 1029, [6][Apos] =>0. 0742, [6][Aneg] =>0. 1187,	[1][Apos]=>0. 0636, [1][Aneg]=>0. 0636, [2][Apos] =>0. 184, [2][Aneg] =>0. 138, [3][Apos] =>0. 1574, [3][Aneg] =>0. 0874, [4][Apos] =>0. 1252, [4][Aneg] =>0. 0626, [5][Apos] =>0. 2057, [5][Aneg] =>0. 1029, [6][Apos] =>0. 0742, [6][Aneg] =>0. 1187,	True

Nama Fungsi : *manhattan()*

Deskripsi : Fungsi ini digunakan untuk membuat sebuah *array* multidimensi yang berisi nilai dari hasil perhitungan jarak *Manhattan* terhadap matriks V.

Input : Sebuah *array* multidimensi dari matriks V serta array dari solusi ideal positif dan negatif per kriteria.

Output : Sebuah *array* multidimensi berisi nilai hasil perhitungan jarak *Manhattan* dan PIA.

Tabel 4.15 Hasil Pengujian Fungsi *manhattan()*

No.	Input	Expected Output	Actual Output	Status
1.	<p>Matriks V:</p> <p>[1]['nama']=> Min Yoongi</p> <p>[1][1]=> 0. 0636,</p> <p>[1][2]=>0. 184,</p> <p>[1][3]=> 0. 1574,</p> <p>[1][4]=>0. 1252,</p> <p>[1][5]=> 0. 2057,</p> <p>[1][6]=> 0. 0742,</p> <p>[1]['applyid']=>22,</p> <p>[2]['nama']=> Addina NR</p> <p>[2][1]=> 0. 0636,</p> <p>[2][2]=> 0. 138,</p> <p>[2][3]=> 0. 0874,</p> <p>[2][4]=> 0. 0626,</p> <p>[2][5]=> 0. 1029,</p> <p>[2][6]=> 0. 1187,</p> <p>[2]['applyid']=>23</p> <p>Array Solusi ideal:</p> <p>[1][Apos]=>0. 0636,</p> <p>[1][Aneg]=>0. 0636,</p> <p>[2][Apos] =>0. 184,</p> <p>[2][Aneg] =>0. 138,</p> <p>[3][Apos] =>0. 1574,</p> <p>[3][Aneg] =>0. 0874,</p> <p>[4][Apos] =>0. 1252,</p> <p>[4][Aneg] =>0. 0626,</p> <p>[5][Apos] =>0. 2057,</p> <p>[5][Aneg] =>0. 1029,</p> <p>[6][Apos] =>0. 0742,</p> <p>[6][Aneg] =>0. 1187,</p>	<p>Hasil perhitungan jarak <i>Manhattan</i>:</p> <p>[1][pos]=>0</p> <p>[1][neg]=> 0.3259,</p> <p>[1][applyid]=>22,</p> <p>[2][pos]=>0. 3259,</p> <p>[2][neg]=> 0,</p> <p>[2][applyid]=>23,</p> <p>Hasil perhitungan PIA:</p> <p>Dpmin = 0</p> <p>Dnmax = 0.3259</p>	<p>Hasil perhitungan jarak <i>Manhattan</i>:</p> <p>[1][pos]=>0</p> <p>[1][neg]=> 0.3259,</p> <p>[1][applyid]=>22,</p> <p>[2][pos]=>0. 3259,</p> <p>[2][neg]=> 0,</p> <p>[2][applyid]=>23,</p> <p>Hasil perhitungan PIA:</p> <p>Dpmin = 0</p> <p>Dnmax = 0.3259</p>	True

Nama Fungsi : jarakPIA()
 Deskripsi : Fungsi ini digunakan untuk membuat sebuah *array* multidimensi yang berisi nilai dari hasil perhitungan jarak relatif masing-masing alternatif ke PIA.
 Input : Sebuah *array* hasil *output* fungsi Manhattan() beserta variabel Dpmin dan Dnmax.
 Output : Sebuah *array* satu dimensi berisi nilai hasil perhitungan jarak relatif masing-masing alternatif ke PIA.

Tabel 4.16 Hasil Pengujian Fungsi jarakPIA()

No.	Input	Expected Output	Actual Output	Status
1.	Hasil perhitungan jarak Manhattan: [1][pos]=>0 [1][neg]=>0.3259, [1][applyid]=>22, [2][pos]=>0.3259, [2][neg]=>0, [2][applyid]=>23, Hasil perhitungan PIA: Dpmin = 0 Dnmax = 0.3259	[1]=>0 [2]=> 0.4609,	[1]=>0 [2]=> 0.4609,	True

Hasil pengujian *White box* pada sistem menunjukkan bahwa fungsi yang digunakan untuk melakukan perhitungan matematis model DiA telah berjalan dengan benar. Hal ini dapat diketahui dengan merujuk pada hasil pengujian *White box* di atas. Seluruh kolom *status* pada masing-masing skenario pengujian bernilai *true*. Maka dari itu, implementasi metode DiA ke dalam sistem untuk melakukan perhitungan dan pemeringkatan terhadap pelamar telah berhasil dilakukan dengan baik.

4.3.2 Pengujian *Black Box*

Pengujian *Black box* yang dilakukan dalam penelitian ini diterapkan dengan mengacu pada kebutuhan fungsional sistem yang ada di Lampiran 2. Pengujian *black box* ini dilakukan oleh *user* dari Departemen Rekrutmen dan Asesmen. Hasil pengujian *black box* disajikan pada Lampiran 4.

Berdasarkan hasil pengujian *Black Box* pada Lampiran 3, dapat disimpulkan bahwa sistem dapat berjalan sesuai dengan kebutuhan fungsional yang telah dibuat sebelumnya. Metode DiA juga telah berhasil diterapkan ke dalam sistem dan mampu memberikan rekomendasi pelamar kepada pembuat keputusan untuk dipanggil ke tahap selanjutnya.

BAB 5 SIMPULAN DAN SARAN

5.1 Simpulan

Penelitian ini dilakukan berawal dari tahap studi literatur hingga sampai ke tahap akhir ini, yaitu penulisan laporan. Setelah melalui serangkaian proses pelaksanaan penelitian, dapat disimpulkan hasil dari penelitian ini terkait tujuan dari penelitian, yaitu :

1. Implementasi metode DiA ke dalam sistem telah berhasil dilakukan dan dapat berjalan dengan baik. Implementasi dari metode DiA mampu memberikan rekomendasi pelamar pada proses penyaringan lamaran. Untuk melakukan implementasi metode DiA diperlukan beberapa informasi terkait alternatif, kriteria, serta tingkat kepentingan dari masing-masing kriteria. Pada penelitian ini, pengumpulan data yang dibutuhkan untuk implementasi metode DiA dan sistem dilakukan melalui diskusi bersama manajer divisi *Human Capital* serta manajer *IT Development* pada perusahaan. Selanjutnya diketahui bahwa perusahaan menggunakan kriteria berikut ini untuk melakukan penyaringan lamaran, yaitu jenis kelamin, usia, pendidikan, pengalaman, kemampuan, dan gaji yang diharapkan. Dengan memanfaatkan model matematis DiA, dilakukan penilaian pelamar berdasarkan kriteria-kriteria tersebut. Hasil penilaian pelamar akan digunakan untuk melakukan pemeringkatan para pelamar. Lima pelamar teratas merupakan pelamar yang direkomendasikan untuk dipanggil mengikuti tahap seleksi berikutnya. Penentuan tingkat kepentingan masing-masing kriteria menjadi faktor penting untuk mendapatkan rekomendasi pelamar yang sesuai kebutuhan. Berdasarkan pengujian *White box* yang telah dilakukan, dapat disimpulkan bahwa fungsi untuk perhitungan metode DiA telah berjalan dengan baik.

2. Implementasi modul rekrutmen telah berhasil dilakukan dan berjalan dengan baik. Implementasi sistem mengadopsi metode RAD yang dibatasi hingga pengujian. Pada tahap pertama dilakukan perencanaan kebutuhan. Tahap ini menganalisa segala kebutuhan informasi yang diperlukan dalam pembangunan sistem serta penerapan metode DiA. Hasil dari tahap ini ialah sistem memiliki kebutuhan informasi terkait *job title*, posisi, *requirement*, kantor, lowongan kerja, dan pelamar. Kebutuhan informasi metode DiA meliputi daftar konversi nilai serta tingkat kepentingan masing-masing kriteria. Kemudian, tahap perancangan sistem dilakukan. Tahap ini menghasilkan rancangan diagram *usecase*, diagram aktivitas, *class diagram*, serta data model dari sistem. Pengujian sistem yang dilakukan dengan metode *black box* menunjukkan bahwa sistem telah mampu berjalan sebagaimana yang diharapkan.

5.2 Saran

Terdapat beberapa saran yang dapat dijadikan pertimbangan untuk penelitian selanjutnya, yaitu :

1. Terdapat beberapa fitur yang dapat ditambahkan untuk melengkapi sistem rekrutmen seperti fitur manajemen *interview* untuk penjadwalan *interview*, pendataan kehadiran pelamar ke *interview*, beserta hasil *interview*. Kemudian penambahan fitur laporan rekrutmen untuk memudahkan analisis data dari rekrutmen yang telah dilakukan.
2. Fungsi pencarian pada sistem belum diterapkan dengan menyeluruh. Oleh karena itu, perlu penerapan fungsi pencarian secara menyeluruh untuk data master, lowongan kerja, pelamar, serta aplikasi lamaran pada sistem. Implementasi algoritma pencarian ke dalam sistem juga dapat dilakukan untuk memaksimalkan fungsi pencarian dalam sistem.
3. Sistem ini dapat diterapkan secara *online* sehingga penyimpanan data pelamar dapat dilakukan mandiri oleh masing-masing pelamar.

DAFTAR PUSTAKA

- Armstrong, M., & Taylor, S. (2014). *Armstrong's Handbook Of Human Resource Management Practice* (13th ed.). London: Kogan Page.
- Bulmash, J., Chhinzer, N., & Speers, E. (2010). *Strategic Planning for Human Resources* (1st ed.). Toronto: McGraw-Hill Ryerson.
- Career Builder. (2014, Agustus 7). *Fifty-eight Percent of Employers Have Caught a Lie on a Resume, According to a New CareerBuilder Survey*. Dipetik Mei 31, 2016, dari Career Builder: <http://www.careerbuilder.com/share/aboutus/pressreleasesdetail.aspx?sd=8%2F7%2F2014&id=pr837&ed=12%2F31%2F2014>
- Chack, D. K. (2013, Juli). Decision Support System For Human Resource Management Of The Organization. *International Journal of Management Research and Business Strategy (IJMRBS)*, 2.
- Diah R., A. K. (2013). Sistem Pendukung Keputusan Rekrutmen Karyawan Produksi Menggunakan Metode Wighted Product Pada PT.Ploss Asia Semarang. *Skripsi, Fakultas Ilmu Komputer*.
- Effendy, D. A., & Irawan, R. H. (2015). Uji Sensitivitas WP, SAW, dan TOPSIS Dalam Menentukan Titik Lokasi Repeater Internet Wireless. *Seminar Nasional Teknologi Informasi dan Multimedia*, 85-90.
- Information and Communications Technology Council. (2012, Juni). *Developing Tomorrow's Workforce Today*. Dipetik Agustus 25, 2016, dari Information and Communications Technology Council: http://www.ictc-ctic.ca/wp-content/uploads/2012/06/ICTC_HRTipSheetAppScreening_EN.pdf
- Kendall, K. E., & Kendall, J. E. (2011). *Systems Analysis and Design* (8th ed.). (E. Svendsen, Penyunt.) Upper Saddle River: Prentice Hall.
- Khan, M. E. (2011, Juli). Different Approaches to White Box Testing Technique for Finding Errors. *International Journal of Software Engineering and Its Applications*, 5.
- Mathis, R. L., & Jack, J. H. (2010). *Human Resource Management* (9th ed.). Amerika Serikat: Cengage Learning.

- Purnamasari, R. A. (2015, Juni). Penentuan Penerima Beasiswa Dengan Metode Simple Additive Weighting Dan Metode The Distance To The Ideal Alternative. *Skripsi, Universitas Jember*.
- Tran, P. N., & Boukhatem, N. (2008, Oktober 30). The Distance To The Ideal Alternative (DiA) Algorithm for Interface Selection in Heterogenous Wireless Network. *Mobile Computing and Networking (MobiCom)* (hal. 61-68). New York: ACM New York.

Lampiran 1 – Wawancara

Di bawah ini merupakan rangkuman wawancara yang penulis (selanjutnya disebut dengan P) lakukan dengan Bapak Vidhestira Dwimadia (selanjutnya disebut dengan VD) selaku *Human Capital Division Head* di perusahaan XYZ pada hari Jumat, 20 Mei 2016 dan Rabu, 27 Juli 2016.

- P : Berapa jumlah karyawan yang dimiliki oleh perusahaan?
- VD : Terdapat 3 jenis karyawan, yaitu karyawan kontrak, permanen, dan mitra. Total karyawan kontrak dan permanen hingga saat ini kurang lebih sekitar 3800 orang. Jumlah karyawan mitra kurang lebih sekitar 900 orang. Jadi, total karyawan keseluruhan kurang lebih 4700 orang.
- P : Berapa banyak cabang yang dimiliki oleh perusahaan?
- VD : Terdapat kurang lebih 160 cabang yang tersebar di seluruh Indonesia.
- P : Bagaimana cara untuk mengetahui apakah kebutuhan di cabang dengan *Branch model* sudah sesuai atau belum?
- VD : Caranya mengetahui sesuai atau tidaknya jumlah *Branch model* dapat dilihat dari dua sisi, yaitu kualifikasi dan pengontrolan jumlahnya. Dari segi kualifikasi, perusahaan telah memiliki standar kualifikasi minimal yang harus dimiliki oleh orang-orang di cabang. Bicara soal kualifikasi terdapat beberapa kriteria, sebagai contoh usia dan fasilitas yang dimiliki. Proses seleksi dimulai dari hal itu. Setelah seleksi dilakukan, akan dilakukan *desk call* dari kantor pusat untuk verifikasi kesesuaian kualifikasi.
- Kalau untuk pengontrolan jumlah, terdapat sistem yang membantu pelaksanaannya. Sistem memberikan informasi kebutuhan *manpower* di cabang dan juga informasi perencanaan SDM yang telah dibuat sebelumnya. Jadi, misalnya ketika cabang mengajukan seorang karyawan, sistem memberikan informasi apakah jumlahnya sudah sesuai atau belum berdasarkan perencanaan SDM yang sudah dibuat.
- P : Apa kendala yang ditemukan saat melakukan proses ini dengan bantuan sistem?

- VD : Secara sistemnya sendiri, menurut saya sudah cukup baik. Tetapi bicara lebih lanjut lagi tentang sistem juga membahas masalah kualifikasi yang tadi itu. Jadi isunya kalau buat saya ke depan adalah soal produktivitas. Jadi, bagaimana caranya dari awal bibitnya kita rekrut orang-orang yang bagus. Kemudian kita didik dia sampai nanti benar-benar orang yang kita dapat adalah orang-orang yang sesuai kualifikasi. Sehingga nanti terbentuk sistem yang memang sudah sesuai dengan *Branch model* yang sudah kita persyaratkan. Kalau sekarang masih banyak berapa kejadian, kami akui ada beberapa deviasi. Jadi artinya mungkin secara kualifikasi tidak masuk, karena cabang butuh orang jadi kita rekrut.
- P : Bagaimana alur perusahaan dalam melakukan rekrutmen?
- VD : Pelaksanaan rekrutmen dibedakan menjadi tiga, yaitu untuk cabang, kantor pusat, dan *management trainee*. Saat ini, untuk di cabang lebih simpel alurnya. Jadi hanya proses verifikasi data, *interview*, kemudian finalisasi. Jadi penandatanganan PKWT, *offering*, dan lainnya. Itu untuk yang di cabang.
- Kalau untuk kantor pusat ialah seleksi administrasi, proses psikotes, kemudian *interview* berjenjang. Ada *interview* dengan *user* dan *interview* dengan HRD. Baru tahap terakhir finalisasi.
- Kalau proyek *management trainee* lebih kompleks lagi. Ada proses *interview 1*, *interview 2*, psikotes, kemudian *leaderless group discussion*, proses *offering*, *medical check-up*, baru *onboarding*.
- P : Rata-rata berapa banyak orang yang mendaftar pada lowongan pekerjaan saat rekrutmen dilaksanakan?
- VD : Kalau pastinya berapa banyak kita berbeda-beda setiap bulan. Karena setiap bulan pasti akan ikut *jobfair*. Itu *range*-nya kalau *jobfair*, misal di Jogja ada hingga 300 orang, begitu juga di Sumatera. Tetapi ada juga yang hanya 70 orang. Untuk ODP khususnya. Kalau untuk yang lain-lain pastinya di kantor pusat, rata-rata yang kita panggil kalau di kantor pusat itu 1-15 orang per satu jabatan atau pekerjaan. Kalau sekarang misal ada 5 lowongan yang dibuka berarti ada 75 orang yang

akan dipanggil. Belum ditambah yang ODP. Di daerah beda lagi ceritanya. Kalau di daerah belum ada pemantauan khusus berapa yang dipanggil, berapa yang datang.

Rata-rata penerimaan kita untuk *field staff* itu sekitar 150 untuk sebulan, tambah yang lain-lain hampir 250 lah ya dalam satu bulan, nasional.

P : Bagaimana alur pembukaan lowongan?

VD : Terbagi menjadi dua, yaitu internal dan eksternal. Untuk internal, setiap awal tahun-kan punya *manpower plan*. Setiap divisi itu punya MPP. MPP itu jumlah-jumlah orang yang dibutuhkan perusahaan sampai 1 tahun ke depan. Dari MPP itu nanti kalau misalnya ada proses perekrutan, mereka akan mengajukan ada *form* namanya *form* pengajuan karyawan (FPK). FPK akan berisi kualifikasi akan kebutuhan orang itu apa saja, contohnya usia, pendidikan, *skill*-nya yang dibutuhkan apa. Itulah yang digunakan menjadi dasar mencari orang saat rekrutmen. Jadi kualifikasi-kualifikasinya seperti itu.

Setelah menerima *form* yang ditandatangani oleh kepala divisi sampai direksi, setelah itu baru dilakukan proses *sourcing* atau publikasi. Publikasi bisa dua cara, internal atau eksternal. Kalau internal, iklan dipasang di dalam kantor sendiri. Jadi misalkan kita ada lowongan A B C D, ada yang berminat tidak saudara atau keluarga memenuhi kualifikasi yang dimaksud.

Kalau eksternal, bisa pakai cara yang berbayar atau tidak berbayar. Kalau yang berbayar kita pakai *jobstreet*, *jobsdb*, *careerbuilder*, koran kalau diperlukan. Kalau tidak berbayar, gampangnya sekarang adalah *social media*, *facebook*, *twitter*, *instagram*, kemudian juga *website-website* yang gratis atau *free*, kita juga kerja sama dengan kampus atau SMK dan SMA. Dari situ baru ketika ada CV dari sumber tadi, baru dilakukan seleksi administrasi. Apakah kualifikasi cocok dengan yang disyaratkan. Setelah itu kita akan panggil *interview*. Pertama kali oleh HRD, lalu *user*, dan adakan psikotes. Atau *interview* dulu baru psikotes.

- P : Bagaimana alur melakukan penyaringan lamaran yang masuk? Kriteria apa saja yang dipertimbangkan dalam penilaian penyaringan lamaran?
- VD : Kriteria yang utama ada tiga, usia, pendidikan dan *skill*. Jadi yang dari kualifikasi biasa kita persyaratkan tiga tadi. Dari usia, pengalaman kalau memang perlu ada, lalu ada *skill*. Itu aja sih, ga ada yang lain. Kalau memang ada yang lain mungkin *skill-skill* khusus. Contohnya kalau IT, *skill*-nya kan butuh PHP, JAVA. Kalau yang spesifik misalnya suka ada yang minta dari kampus tertentu. Kalau itu saya akan tetap bandingkan dengan kandidat lainnya. Bukan berarti harga mati harus mengambil dari satu kampus tertentu sehingga menghilangkan peluang untuk yang lain.
- Jenis kelamin dan gaji yang diharapkan tidak termasuk kriteria. Sebenarnya kalau gaji sudah ada *grading* sebelumnya. Jadi sebenarnya tidak masuk tahapan ini. Sudah ditentukan sebelumnya. Untuk tiap-tiap golongan sudah ada *range* gaji tertentu.
- Kalau bicara soal prioritas kriteria, tergantung kebutuhan yang diminta oleh *user*. Misalnya kaya tadi IT, mencari *Programmer*, berarti mereka ingin *programming skill*-nya yang oke. Kalau akunting, ada spesifik *excel*-nya. Kalau prioritas menurut saya, kualifikasi semua saya akan usaha untuk penuhi. Kalau masalah mana yang penting dari lainnya, kalau buat saya adalah kompetensi balik lagi. Jadi kalau orang yang pada dasarnya kompeten (pernah menduduki jabatan, atau kompetensi dari atribut perilaku misal komunikatif).
- Dasarnya dari kualifikasi, awalnya pasti mulai di awal dengan usia. Kita akan *screening* di usia dulu kalau terlalu jauh di luar *range* kita langsung tidak terima. Kalau di *jobstreet* lebih gampang, ada dari usia dan *salary expectation* kalau di *jobstreet*. Kalau di CV kan tidak menuliskan gaji. Biasanya kita prioritas di usia dan *skill*-nya cocok apa tidak.

- P : Siapa yang bertanggung jawab melaksanakan tugas ini (seleksi administratif)?
- VD : Di tim saya ada bagian rekrutmen dan *assessment*. Jadi bagian rekrutmen adalah dia yang menyeleksi orang-orang berdasarkan seleksi administratif atau psikologis. Anggota rekrutmen dan *assessment* di pusat ada dua, di cabang ada 4, jadi total 6. Di cabang regional yang menangani cabang kecil di Jawa, Sumatera, dll.
- P : Berapa lama waktu yang dibutuhkan untuk melakukan proses penyaringan lamaran?
- VD : Hingga saat ini penyaringan lamaran seleksi administratif dilakukan secara manual. Kita pasang iklan hari ini, kemudian terima CV. Kalau satu CV sekitar 5 menit selesai paling lama.
- P : Apakah terdapat kendala ketika melakukan proses penyaringan lamaran melalui situs lowongan kerja yang tadi disebutkan?
- VD : Sistem pada situs lowongan kerja itu bagus bisa memberikan *rating* kecocokan. Namun, kadang saya suka liat ya yang *rejected* di sistem. Memang yang namanya sistem itu kaku. Ada yang *rejected* itu gini, kita cari umur 30, dia umur 31, tapi secara kompetensinya dia memenuhi ya saya panggil, orang bedanya setahun doang. Misal gaji, kalau misal *expected salary*-nya ga *matching* udah langsung *reject* aja. Masuk ke *rejected*. Yang *must have* bisa dikunci di sistem, usia dan gaji.

Mengetahui,

(Vidhestira Dwimadia)
Human Capital Div. Head

Lampiran 2 – *Software Requirement Specification* (SRS)

Implementasi Metode *The Distance To The Ideal Alternative* Pada Sistem Informasi Seleksi Pegawai di PT. XYZ

Software Requirement Specification

Versi 1.0

9 Agustus 2016

Addina Nuriyanti Rahmi

Software Engineer

Dipersiapkan untuk

Kelengkapan Tugas Akhir Informatika Universitas Bakrie

Pembimbing : Yusuf Lestanto, S.T., M.Sc.

1. PENDAHULUAN

Dokumen SRS berikut ini berisi penjelasan mengenai spesifikasi kebutuhan yang diperlukan dalam pengembangan modul rekrutmen dari Sistem Informasi Seleksi Pegawai berbasis *web* untuk Departemen *Recruitment* dan *Assessment* PT. XYZ. Isi dokumen ini dibagi menjadi tiga bagian besar, yaitu Pendahuluan, Gambaran Umum, dan Spesifikasi Kebutuhan.

1.1 Tujuan

Penulisan dokumen yang berisi penjelasan spesifikasi kebutuhan sistem bertujuan untuk mempermudah pembaca dalam memahami serta memberikan pengetahuan terkait sistem yang akan dibangun.

1.2 Ruang Lingkup

Dokumen SRS ini akan menjelaskan spesifikasi kebutuhan dari Modul Rekrutmen pada Sistem Informasi Seleksi Pegawai. Modul Rekrutmen dibuat untuk membantu Departemen *Recruitment and Assessment* dalam manajemen informasi Lowongan Kerja, Pelamar, serta Aplikasi Lamaran yang masuk ke perusahaan. Modul Rekrutmen juga dilengkapi fitur Seleksi Resume yang mampu membandingkan dan menilai kecocokan kualifikasi pelamar dengan spesifikasi posisi yang dilamar.

Metode DiA akan diterapkan ke dalam fitur Seleksi Resume. Metode ini akan memberikan rekomendasi pelamar yang lebih unggul dari pelamar lainnya dinilai dari beberapa kriteria. Kriteria yang digunakan ialah usia, pendidikan, jenis kelamin, kemampuan, pengalaman, dan gaji yang diharapkan.

Berdasarkan rekomendasi pelamar tersebut, sistem dapat membantu pengambil keputusan untuk menentukan pelamar yang cocok untuk dipanggil ke tahap selanjutnya dari tahap seleksi administratif.

1.3 Glosarium

Tabel 1.1 Daftar Istilah Beserta Definisinya

Istilah	Definisi
Penyaringan Pelamar	Merupakan tahap awal penyaringan lamaran untuk menentukan pelamar yang berkualifikasi untuk melanjutkan ke tahap berikutnya. (Information and Communications Technology Council, 2012)
DiA	Salah satu metode untuk memecahkan permasalahan pemilihan berdasarkan banyak kriteria dan banyak alternatif

1.4 Overview

Seperti yang telah disebutkan sebelumnya, dokumen SRS ini tersusun dari tiga bab. Pada Bab 2 akan menjelaskan informasi mengenai gambaran umum sistem yang akan dibangun. Umumnya informasi pada Bab 2 diperuntukkan bagi pengguna sistem. Pada Bab 3 akan menjelaskan secara detail spesifikasi kebutuhan dari sistem yang akan dibangun, di antaranya mencakup antarmuka pengguna dan kebutuhan fungsional sistem. Informasi pada Bab 3 ini akan lebih bermanfaat untuk pengembang sistem.

2. GAMBARAN UMUM

2.1 Perspektif Produk

Sistem Informasi Seleksi Pegawai Modul Rekrutmen merupakan sistem berbasis *web* untuk membantu manajemen data terkait lowongan kerja, pelamar, aplikasi lamaran, serta seleksi resume di PT. XYZ. Sistem ini menerapkan metode DiA untuk melakukan penilaian dengan membandingkan kualifikasi pelamar terhadap kualifikasi yang dibutuhkan sebuah posisi. Hasil penilaian masing-masing pelamar akan diurutkan untuk mendapatkan rekomendasi pelamar yang lebih unggul dari yang

lain. Pengguna dari sistem ini adalah pegawai Departemen *Recruitment* dan *Assessment* pada PT. XYZ.

2.2 Fungsi Produk

Fungsi utama dari sistem berdasarkan kebutuhan pengguna ialah sebagai berikut :

- Menyediakan fungsi untuk melakukan manajemen data terkait rekrutmen, seperti Lowongan Kerja, Pelamar, Aplikasi Lamaran, dan Spesifikasi Posisi.
- Memberikan rekomendasi pelamar yang lebih unggul dari segi kualifikasi pada tahap seleksi administratif untuk dipanggil mengikuti proses seleksi selanjutnya.

2.3 Karakteristik Pengguna

Sistem ini diperuntukkan bagi staf Departemen *Recruitment and Assessment* pada PT XYZ. Agar dapat mengoperasikan sistem dengan baik, pengguna harus memahami alur umum proses rekrutmen. Pengguna juga diharuskan memiliki pengalaman menggunakan *web browser*. *Web browser* digunakan untuk mengakses sistem melalui jaringan *intranet* perusahaan.

2.4 Batasan Umum

Berikut ini merupakan batasan umum dari sistem :

- Terlepas dari proses rekrutmen secara keseluruhan, sistem hanya mengakomodasi manajemen data lowongan kerja, pelamar, aplikasi lamaran, dan seleksi resume.
- Sistem yang dibangun berbentuk aplikasi *web* menggunakan bahasa pemrograman PHP, HTML, CSS, dan *Javascript*. Menggunakan XAMPP sebagai *web server* dan *database server*.
- Data yang diperlukan untuk melakukan seleksi resume akan dimasukkan secara manual oleh pengguna. Data yang diperlukan

termasuk *jobtitle*, posisi, kemampuan, kualifikasi, lokasi, lowongan kerja, pelamar, lamaran, serta bobot kriteria penilaian untuk seleksi.

- Metode DiA diterapkan pada bagian Seleksi Resume untuk memberikan pemeringkatan serta rekomendasi pelamar yang lebih unggul dari segi kualifikasi dibandingkan pelamar yang lainnya.
- Kriteria yang digunakan untuk penilaian ialah usia, pendidikan, jenis kelamin, kemampuan, pengalaman, serta gaji yang diharapkan. Kriteria tidak dapat ditambahkan atau dikurangkan.

2.5 Asumsi dan Ketergantungan

- Pengguna harus memiliki kemampuan untuk mengoperasikan *web browser*.
- Data posisi, lowongan kerja, pelamar, aplikasi lamaran, serta bobot kriteria tidak boleh kosong dan harus diisi dengan lengkap agar proses seleksi resume dapat dilakukan dengan baik.
- Pengalaman pelamar yang dimasukkan ke sistem hanyalah pengalaman yang sesuai dengan posisi dilamar.

3. SPESIFIKASI KEBUTUHAN

3.1 Antarmuka Pengguna

Modul Rekrutmen terdiri dari dua menu utama, yaitu Data Master dan Rekrutmen. Menu tersebut masing-masing memiliki beberapa sub-menu di bawah mereka yang tiap sub-menu memiliki beberapa halaman antarmuka. Pada Tabel 3.1 terdapat daftar sub-menu dari masing-masing menu. Kecuali untuk sub-menu Seleksi Resume, sub-menu pada Tabel 3.1 memiliki fungsi yang sama, yaitu untuk melakukan manajemen data dari nama sub-menu tersebut. Manajemen data terdiri dari operasi menampilkan daftar seluruh data (*index*), membuat data (*create*), mengubah data (*update*), menampilkan data spesifik (*view*), serta menghapus data (*delete*). Maka dari itu, sub-menu dengan fungsi manajemen data memiliki rancangan antarmuka dengan format yang sama

untuk halaman *index*, *create*, *view*, dan *update*. Sedangkan, untuk operasi *delete* tidak memiliki halaman khusus, melainkan hanya memberikan *pop-up*.

Tabel 3.1 Daftar Menu dan Sub-Menu yang dimilikinya.

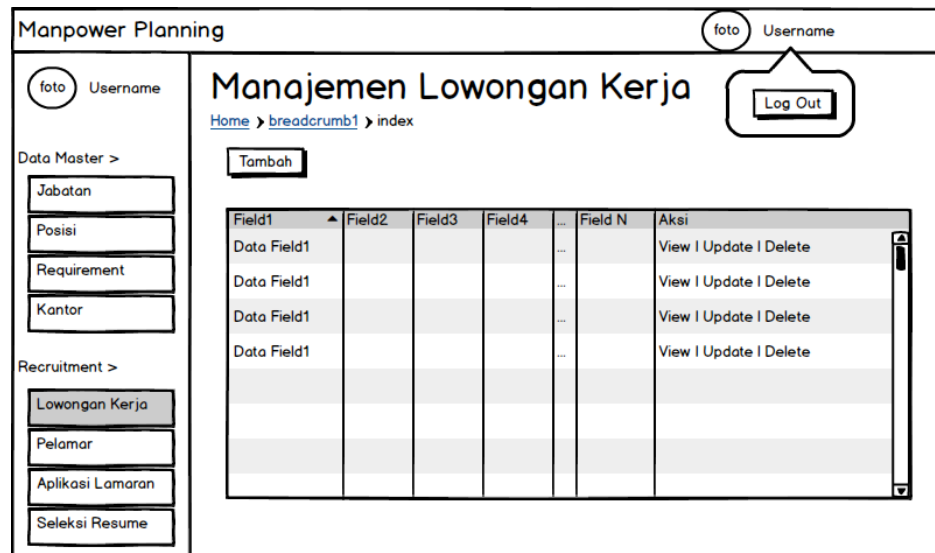
Menu	Sub-Menu yang Dimiliki
Data Master	<ul style="list-style-type: none"> • <i>Jobtitle</i> • Posisi • <i>Requirement</i> • Lokasi
Rekrutmen	<ul style="list-style-type: none"> • Lowongan Pekerjaan • Pelamar • Lamaran • Seleksi Resume

Antarmuka ditampilkan melalui *web browser* seperti *Mozilla Firefox* atau *Google Chrome* namun tidak terbatas pada yang telah disebutkan. Persyaratan antarmuka yang diharapkan pengguna ialah tampilan yang *user-friendly*.

Gambar 3.1 Rancangan Antarmuka Halaman Login

Seperti yang dapat dilihat pada Gambar 3.1, terdapat dua buah *field* untuk memasukkan data yaitu *username* dan *password* pengguna pada halaman

login. Tombol *Login* berguna untuk mengirim data hasil *input* untuk kemudian divalidasi oleh sistem. Apabila data sesuai, maka pengguna akan diarahkan ke halaman utama sistem.



Gambar 3.2 Rancangan Antarmuka Halaman Melihat Daftar Lowongan Kerja

Gambar 3.2 merupakan rancangan antarmuka halaman *index* dari sub-menu Lowongan Kerja. Pada halaman ini terdapat sebuah tombol Tambah untuk membuat lowongan kerja baru. Kemudian terdapat sebuah tabel untuk menampilkan daftar lowongan kerja beserta informasi mengenai lowongan tersebut. Tabel tersebut memiliki kolom Aksi yang berisi tiga buah tombol untuk melakukan operasi *view*, *update*, dan *delete* terhadap data pada baris tabel tertentu.

Rancangan halaman *index* ini juga berlaku untuk tampilan halaman *index* dari sub-menu lainnya yang berfungsi untuk manajemen data. Perbedaannya hanyalah pada data yang akan ditampilkan.

Gambar 3.3 merupakan rancangan antarmuka halaman *create / update* dari sub-menu Pelamar. Pada halaman ini terdapat sebuah *form* yang terdiri dari kumpulan label beserta elemen *input* untuk diisi oleh pengguna. Kemudian terdapat dua buah tombol, yaitu kembali dan simpan.

Rancangan halaman *create/update* ini juga berlaku untuk tampilan halaman *create/update* dari sub-menu lainnya yang berfungsi untuk manajemen data. Perbedaannya hanyalah pada daftar data yang perlu dilengkapi.

Manpower Planning

foto Username

foto Username

Manajemen Pelamar Tambah Pelamar

[Home](#) > [breadcrumb1](#) > create

Field with * are required.

Nama Lengkap*

Jenis Kelamin* ☐ Laki-Laki ☐ Perempuan

Tempat Lahir*

Tanggal Lahir*

Alamat*

Email*

Kembali Simpan

Data Master >

Jabatan

Posisi

Requirement

Kantor

Recruitment >

Lowongan Kerja

Pelamar

Aplikasi Lamaran

Seleksi Resume

Gambar 3.3 Rancangan Antarmuka Halaman Membuat / Mengubah Pelamar

Manpower Planning

foto Username

foto Username

Manajemen Lamaran Detail Lamaran

[Home](#) > [breadcrumb1](#) > view

ID Aplikasi :	1
Nama Posisi :	IT Developer
Kantor :	Kantor 1
Nama Pelamar :	Pelamar 1
Tanggal Melamar :	13-Agustus-2016
Hasil Akhir :	Diterima

Kembali

Data Master >

Jabatan

Posisi

Requirement

Kantor

Recruitment >

Lowongan Kerja

Pelamar

Aplikasi Lamaran

Seleksi Resume

Gambar 3.4 Rancangan Antarmuka Melihat Sebuah Aplikasi Lamaran

Gambar 3.4 merupakan rancangan antarmuka halaman *view* dari sub-menu Aplikasi Lamaran. Pada halaman ini terdapat sebuah tabel yang berisi

data-data terkait nama sub-menu tersebut. Kemudian terdapat sebuah tombol, yaitu kembali.

Rancangan halaman *view* ini juga berlaku untuk tampilan halaman *view* dari sub-menu lainnya yang berfungsi untuk manajemen data. Perbedaannya hanyalah pada daftar data yang akan ditampilkan.

Manajemen Seleksi

Home > breadcrumb1 > Pilih Pelamar

Seleksi Lowongan : - Pilih Posisi - IT Developer Search

Daftar Pelamar

No.	Nama Pelamar	Aksi
1	Pelamar 1	View Update
2	Pelamar 2	View Update
3	Pelamar 3	View Update
4	Pelamar 4	View Update

Simpan dan Pilih Kriteria

Gambar 3.5 Rancangan Antarmuka Halaman Awal Seleksi Resume

Gambar 3.5 merupakan rancangan antarmuka halaman awal dari sub-menu Seleksi Resume. Pada halaman ini terdapat sebuah *drop down list* dari daftar lowongan kerja. Sebuah tombol *Search* berfungsi untuk melakukan proses pencarian pelamar dari lowongan kerja tersebut. Di bawahnya terdapat sebuah tabel yang akan menampilkan daftar pelamar yang melamar pada lowongan kerja tersebut. Daftar nama para pelamar yang ditampilkan tersebut akan menjadi daftar alternatif yang diseleksi nantinya. Tombol *Simpan & Pilih Kriteria* akan membawa pengguna ke tahap selanjutnya dalam proses seleksi resume.

Gambar 3.6 merupakan rancangan antarmuka halaman lihat nilai alternatif dari sub-menu Seleksi Resume. Pada halaman ini terdapat sebuah tabel yang merangkum data-data para alternatif atau pelamar. Tabel akan menampilkan data nama, jenis kelamin, usia, pendidikan, pengalaman kerja, kemampuan, dan gaji yang diharapkan dari masing-masing

alternatif. Terdapat tombol kembali untuk kembali ke halaman awal seleksi resume. Sedangkan tombol lakukan seleksi akan membawa pengguna ke halaman selanjutnya pada tahap seleksi resume.

Manpower Planning foto Username foto Username Log Out

Manajemen Seleksi
[Home](#) > [breadcrumb1](#) > [Lihat Alternatif](#)

Data Master >

- Jabatan
- Posisi
- Requirement
- Kantor

Recruitment >

- Lowongan Kerja
- Pelamar
- Aplikasi Lamaran
- Seleksi Resume

IT Developer Requirement

Jenis Kelamin	Usia	Pendidikan	Pengalaman	Kemampuan
Perempuan / Laki-laki	23 s/d 27 tahun	S1	1 tahun	PHP

Daftar Pelamar

No.	Nama Pelamar	Jenis Kelamin	Usia	Pendidikan	Pengalaman	Kemampuan
1	Pelamar 1	Laki-laki	23	S1	1 tahun	PHP
2	Pelamar 2	Perempuan	25	S1	2 tahun	C++
3	Pelamar 3	Laki-laki	24	S1	2 tahun	JAVA
4	Pelamar 4	Perempuan	26	S1	3 tahun	PHP

Lakukan Seleksi

Gambar 3.6 Rancangan Antarmuka Halaman Lihat Nilai Alternatif

Manpower Planning foto Username foto Username Log Out

Manajemen Seleksi
[Home](#) > [breadcrumb1](#) > [Hasil Seleksi](#)

Data Master >

- Jabatan
- Posisi
- Requirement
- Kantor

Recruitment >

- Lowongan Kerja
- Pelamar
- Aplikasi Lamaran
- Seleksi Resume

IT Developer Requirement

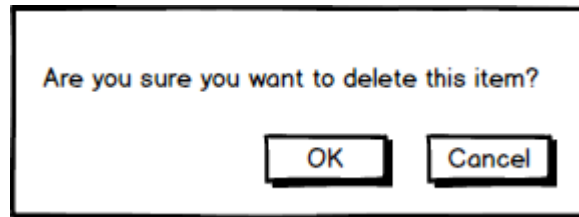
Jenis Kelamin	Usia	Pendidikan	Pengalaman	Kemampuan
Perempuan / Laki-laki	23 s/d 27 tahun	S1	1 tahun	PHP

Daftar Pelamar

Nama Pelamar	Nilai Ri	Status	Deviasi
Pelamar 4	0	Sesuai	
Pelamar 1	0.111	Sesuai	
Pelamar 2	0.123	Tidak Sesuai	Kemampuan
Pelamar 3	0.123	Tidak Sesuai	Kemampuan

Gambar 3.7 Rancangan Antarmuka Halaman Hasil Seleksi Resume

Gambar 3.7 merupakan rancangan antarmuka halaman hasil seleksi resume. Pada halaman ini terdapat informasi mengenai posisi yang dilamar oleh daftar alternatif, di antaranya ialah nama posisi dan informasi kualifikasi yang dibutuhkan oleh posisi tersebut.



Gambar 3.8 Rancangan Pop-Up Untuk Operasi *Delete*

Gambar 3.8 merupakan rancangan tampilan pop-up yang akan muncul ketika tombol *delete* ditekan.

3.2 Kebutuhan *Hardware*

Berikut merupakan informasi *hardware* yang dibutuhkan untuk membangun sistem :

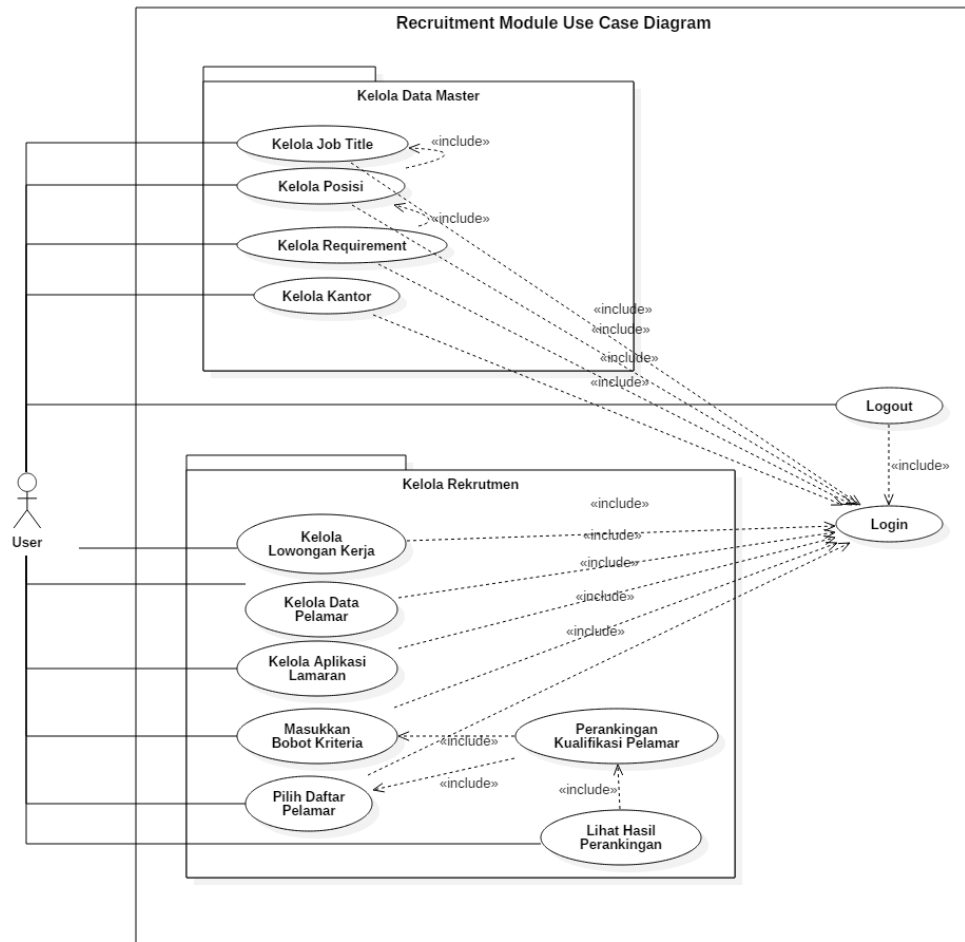
1. Sebuah komputer *laptop* atau PC digunakan untuk proses merancang, membangun, serta menjalankan sistem dengan spesifikasi sebagai berikut :
 - Prosesor Intel Pentium @2.10Ghz
 - Memori RAM 2 GB
 - 64-bit *Windows Operating System*
 - *Hard Disk* 320 GB

3.3 Kebutuhan *Software*

Berikut merupakan daftar *software* yang dibutuhkan untuk menjalankan sistem :

- XAMPP untuk menjalankan *web server* Apache dan RDBMS MySQL.
- *Web browser Mozilla Firefox*.
- Sistem Operasi *Windows 7*.
- Bahasa pemrograman PHP, HTML, CSS, Javascript.
- Yii dan JQuery sebagai *framework* PHP dan Javascript secara berurutan.
- Admin LTE 2 sebagai *template* tampilan sistem.

3.4 Kebutuhan Fungsional



Gambar 3.9 Use Case Diagram Sistem

Pada bagian ini akan dijelaskan mengenai fungsi-fungsi yang dapat dilakukan oleh sistem. Diagram *use case* pada Gambar 3.9 menunjukkan fungsi-fungsi dari sistem yang dapat dilakukan oleh pengguna.

Nama Use Case	: Login
Aktor	: User
Pre-condition	: Aktor membuka sistem melalui <i>web browser</i>
Post-condition	: Aktor berhasil <i>login</i> dan masuk ke dalam aplikasi
Deskripsi	: Use Case ini menggambarkan aktivitas ketika aktor melakukan <i>login</i> .

Tabel 3.2 Deskripsi Aksi Aktor dan Respon Sistem Untuk Use Case Login

Aktor	Sistem
1. Membuka sistem melalui <i>web browser</i>	2. Menampilkan halaman <i>login</i> serta meminta <i>username</i> dan <i>password</i>
3. Memasukkan <i>username</i> dan <i>password</i>	4. Melakukan validasi data yang dimasukkan oleh aktor.
	5. Jika validasi benar, maka Aktor dapat masuk ke sistem. Jika salah, menampilkan pesan error.

Nama *Use Case* : *Logout*

Aktor : *User*

Pre-condition : Aktor menekan tombol *logout* pada sistem

Post-condition : Aktor telah keluar dari sistem

Deskripsi : *Use Case* ini menggambarkan aktivitas ketika aktor melakukan *logout*.

Tabel 3.3 Deskripsi Aksi Aktor dan Respon Sistem Untuk Use Case Logout

Aktor	Sistem
1. Menekan tombol <i>logout</i> pada sistem	2. Mengakhiri sesi penggunaan sistem milik aktor
	3. Menampilkan halaman <i>login</i>

Nama *Use Case* : *Kelola Jobtitle*

Aktor : *User*

Pre-condition : Aktor memilih sub-menu *jobtitle*

Post-condition : Sistem menampilkan data yang telah dikelola Aktor

Deskripsi : *Use Case* ini menggambarkan aktivitas ketika aktor melakukan *Kelola Jobtitle*. Aktivitas ini terdiri dari membuat, melihat, serta menghapus data *jobtitle*.

Tabel 3.4 Deskripsi Aksi Aktor dan Respon Sistem Untuk Use Case Kelola Job Title

Aktor	Sistem
1. Memilih sub-menu <i>jobtitle</i> .	2. Menampilkan daftar seluruh <i>jobtitle</i> .

3. Menekan tombol tambah <i>jobtitle</i> .	4. Menampilkan formulir isian data <i>jobtitle</i> .
5. Mengisi data <i>jobtitle</i> baru dan menekan tombol simpan	6. Melakukan validasi data masukan. Jika tervalidasi, maka akan menyimpan data ke <i>database</i> .
	7. Menampilkan data <i>jobtitle</i> pada halaman <i>view</i> .
8. Menekan tombol kembali	9. Menampilkan seluruh daftar <i>jobtitle</i> di halaman <i>index</i>
10. Menekan tombol <i>update</i>	11. Menampilkan data <i>jobtitle</i> yang akan diubah
12. Memasukkan perubahan data <i>jobtitle</i> dan menekan tombol simpan	13. Melakukan validasi data masukan. Jika tervalidasi, maka akan menyimpan data ke <i>database</i> .
	14. Menampilkan data <i>jobtitle</i> pada halaman <i>view</i> .
15. Menekan tombol kembali	16. Menampilkan halaman <i>index jobtitle</i>
17. Menekan tombol hapus	18. Memberikan <i>pop-up</i> konfirmasi untuk menghapus data.
19. Menekan tombol yes	20. Menghapus data dari <i>database</i>

Nama *Use Case* : Kelola Posisi

Aktor : *User*

Pre-condition : Aktor memilih sub-menu posisi

Post-condition : Sistem menampilkan data yang telah dikelola Aktor

Deskripsi : *Use Case* ini menggambarkan aktivitas ketika aktor melakukan Kelola Posisi. Aktivitas ini terdiri dari membuat, melihat, serta menghapus data Posisi.

Tabel 3.5 Deskripsi Aksi Aktor dan Respon Sistem Untuk *Use Case* Kelola Posisi

Aktor	Sistem
1. Memilih sub-menu posisi.	2. Menampilkan daftar seluruh posisi.
3. Menekan tombol tambah posisi.	4. Menampilkan formulir isian data posisi.
5. Mengisi data posisi baru dan menekan tombol simpan	6. Melakukan validasi data masukan. Jika tervalidasi, maka akan menyimpan data ke <i>database</i> .

	7. Menampilkan data posisi pada halaman <i>view</i> .
8. Menekan tombol kembali	9. Menampilkan seluruh daftar posisi di halaman <i>index</i>
10. Menekan tombol <i>update</i>	11. Menampilkan data posisi yang akan diubah
12. Memasukkan perubahan data posisi dan menekan tombol simpan	13. Melakukan validasi data masukan. Jika tervalidasi, maka akan menyimpan data ke <i>database</i> .
	14. Menampilkan data posisi pada halaman <i>view</i> .
15. Menekan tombol kembali	16. Menampilkan halaman <i>index</i> posisi
17. Menekan tombol hapus	18. Memberikan <i>pop-up</i> konfirmasi untuk menghapus data.
19. Menekan tombol yes	20. Menghapus data dari <i>database</i>

Nama *Use Case* : Kelola *Requirement*

Aktor : *User*

Pre-condition : Aktor memilih sub-menu kualifikasi

Post-condition : Sistem menampilkan data yang telah dikelola Aktor

Deskripsi : *Use Case* ini menggambarkan aktivitas ketika aktor melakukan Kelola Kualifikasi. Aktivitas ini terdiri dari membuat, melihat, serta menghapus data Kualifikasi.

Tabel 3.6 Deskripsi Aksi Aktor dan Respon Sistem Untuk *Use Case* Kelola *Requirement*

Aktor	Sistem
1. Memilih sub-menu kualifikasi.	2. Menampilkan daftar seluruh kualifikasi.
3. Menekan tombol tambah kualifikasi.	4. Menampilkan formulir isian data kualifikasi.
5. Mengisi data kualifikasi baru dan menekan tombol simpan	6. Melakukan validasi data masukan. Jika tervalidasi, maka akan menyimpan data ke <i>database</i> .
	7. Menampilkan data kualifikasi pada halaman <i>view</i> .

8. Menekan tombol kembali	9. Menampilkan seluruh daftar kualifikasi di halaman <i>index</i>
10. Menekan tombol <i>update</i>	11. Menampilkan data kualifikasi yang akan diubah
12. Memasukkan perubahan data kualifikasi dan menekan tombol simpan	13. Melakukan validasi data masukan. Jika tervalidasi, maka akan menyimpan data ke <i>database</i> .
	14. Menampilkan data kualifikasi pada halaman <i>view</i> .
15. Menekan tombol kembali	16. Menampilkan halaman <i>index</i> kualifikasi
17. Menekan tombol hapus	18. Memberikan <i>pop-up</i> konfirmasi untuk menghapus data.
19. Menekan tombol yes	20. Menghapus data dari <i>database</i>

Nama *Use Case* : Kelola Kantor

Aktor : *User*

Pre-condition : Aktor memilih sub-menu Kantor

Post-condition : Sistem menampilkan data yang telah dikelola Aktor

Deskripsi : *Use Case* ini menggambarkan aktivitas ketika aktor melakukan Kelola Kantor. Aktivitas ini terdiri dari membuat, melihat, serta menghapus data Kantor.

Tabel 3.7 Deskripsi Aksi Aktor dan Respon Sistem Untuk *Use Case* Kelola Kantor

Aktor	Sistem
1. Memilih sub-menu kantor.	2. Menampilkan daftar seluruh kantor.
3. Menekan tombol tambah kantor.	4. Menampilkan formulir isian data kantor.
5. Mengisi data kantor baru dan menekan tombol simpan	6. Melakukan validasi data masukan. Jika tervalidasi, maka akan menyimpan data ke <i>database</i> .
	7. Menampilkan data kantor pada halaman <i>view</i> .
8. Menekan tombol kembali	9. Menampilkan seluruh daftar kantor di halaman <i>index</i>
10. Menekan tombol <i>update</i>	11. Menampilkan data kantor yang akan

	diubah
12. Memasukkan perubahan data kantor dan menekan tombol simpan	13. Melakukan validasi data masukan. Jika tervalidasi, maka akan menyimpan data ke <i>database</i> .
	14. Menampilkan data kantor pada halaman <i>view</i> .
15. Menekan tombol kembali	16. Menampilkan halaman <i>index</i> kantor
17. Menekan tombol hapus	18. Memberikan <i>pop-up</i> konfirmasi untuk menghapus data.
19. Menekan tombol yes	20. Menghapus data dari <i>database</i>

Nama *Use Case* : Kelola Lowongan Kerja

Aktor : *User*

Pre-condition : Aktor memilih sub-menu Lowongan Kerja

Post-condition : Sistem menampilkan data yang telah dikelola Aktor

Deskripsi : *Use Case* ini menggambarkan aktivitas ketika aktor melakukan Kelola Lowongan Kerja. Aktivitas ini terdiri dari membuat, melihat, serta menghapus data Lowongan Kerja.

Tabel 3.8 Deskripsi Aksi Aktor dan Respon Sistem Untuk *Use Case* Lowongan Kerja

Aktor	Sistem
1. Memilih sub-menu lowongan kerja.	2. Menampilkan daftar seluruh lowongan kerja.
3. Menekan tombol tambah lowongan kerja.	4. Menampilkan formulir isian data lowongan kerja.
5. Mengisi data lowongan kerja baru dan menekan tombol simpan	6. Melakukan validasi data masukan. Jika tervalidasi, maka akan menyimpan data ke <i>database</i> .
	7. Menampilkan data lowongan kerja pada halaman <i>view</i> .
8. Menekan tombol kembali	9. Menampilkan seluruh daftar lowongan kerja di halaman <i>index</i>
10. Menekan tombol <i>update</i>	11. Menampilkan data lowongan kerja

	yang akan diubah
12. Memasukkan perubahan data lowongan kerja dan menekan tombol simpan	13. Melakukan validasi data masukan. Jika tervalidasi, maka akan menyimpan data ke <i>database</i> .
	14. Menampilkan data lowongan kerja pada halaman <i>view</i> .
15. Menekan tombol kembali	16. Menampilkan halaman <i>index</i> lowongan kerja
17. Menekan tombol hapus	18. Memberikan <i>pop-up</i> konfirmasi untuk menghapus data.
19. Menekan tombol yes	20. Menghapus data dari <i>database</i>

Nama *Use Case* : Kelola Pelamar

Aktor : *User*

Pre-condition : Aktor memilih sub-menu Pelamar

Post-condition : Sistem menampilkan data yang telah dikelola Aktor

Deskripsi : *Use Case* ini menggambarkan aktivitas ketika aktor melakukan Kelola Pelamar. Aktivitas ini terdiri dari membuat, melihat, serta menghapus data Pelamar.

Tabel 3.9 Deskripsi Aksi Aktor dan Respon Sistem Untuk *Use Case* Kelola Pelamar

Aktor	Sistem
1. Memilih sub-menu pelamar.	2. Menampilkan daftar seluruh pelamar.
3. Menekan tombol tambah pelamar.	4. Menampilkan formulir isian data pelamar.
5. Mengisi data pelamar baru dan menekan tombol simpan	6. Melakukan validasi data masukan. Jika tervalidasi, maka akan menyimpan data ke <i>database</i> .
	7. Menampilkan data pelamar pada halaman <i>view</i> .
8. Menekan tombol kembali	9. Menampilkan seluruh daftar pelamar di halaman <i>index</i>
10. Menekan tombol <i>update</i>	11. Menampilkan data pelamar yang akan diubah

12. Memasukkan perubahan data pelamar dan menekan tombol simpan	13. Melakukan validasi data masukan. Jika tervalidasi, maka akan menyimpan data ke <i>database</i> .
	14. Menampilkan data pelamar pada halaman <i>view</i> .
15. Menekan tombol kembali	16. Menampilkan halaman <i>index</i> pelamar
17. Menekan tombol hapus	18. Memberikan <i>pop-up</i> konfirmasi untuk menghapus data.
19. Menekan tombol yes	20. Menghapus data dari <i>database</i>

Nama *Use Case* : Kelola Aplikasi Lamaran

Aktor : *User*

Pre-condition : Aktor memilih sub-menu Aplikasi Lamaran

Post-condition : Sistem menampilkan data yang telah dikelola Aktor

Deskripsi : *Use Case* ini menggambarkan aktivitas ketika aktor melakukan Kelola Aplikasi Lamaran. Aktivitas ini terdiri dari membuat, melihat, serta menghapus data Aplikasi Lamaran.

Tabel 3.10 Deskripsi Aksi Aktor dan Respon Sistem Untuk *Use Case* Kelola Aplikasi Lamaran

Aktor	Sistem
1. Memilih sub-menu aplikasi lamaran.	2. Menampilkan daftar seluruh aplikasi lamaran.
3. Menekan tombol tambah aplikasi lamaran.	4. Menampilkan formulir isian data aplikasi lamaran.
5. Mengisi data aplikasi lamaran baru dan menekan tombol simpan	6. Melakukan validasi data masukan. Jika tervalidasi, maka akan menyimpan data ke <i>database</i> .
	7. Menampilkan data aplikasi lamaran pada halaman <i>view</i> .
8. Menekan tombol kembali	9. Menampilkan seluruh daftar aplikasi lamaran di halaman <i>index</i>
10. Menekan tombol <i>update</i>	11. Menampilkan data aplikasi lamaran yang akan diubah

12. Memasukkan perubahan data aplikasi lamaran dan menekan tombol simpan	13. Melakukan validasi data masukan. Jika tervalidasi, maka akan menyimpan data ke <i>database</i> .
	14. Menampilkan data aplikasi lamaran pada halaman <i>view</i> .
15. Menekan tombol kembali	16. Menampilkan halaman <i>index</i> aplikasi lamaran
17. Menekan tombol hapus	18. Memberikan <i>pop-up</i> konfirmasi untuk menghapus data.
19. Menekan tombol yes	20. Menghapus data dari <i>database</i>

Nama *Use Case* : Masukkan Bobot Kriteria

Aktor : *User*

Pre-condition : Aktor telah *login* ke sistem

Post-condition : Sistem menampilkan data yang telah masukkan Aktor

Deskripsi : *Use Case* ini menggambarkan aktivitas ketika aktor memasukkan bobot kriteria.

Tabel 3.11 Deskripsi Aksi Aktor dan Respon Sistem Untuk *Use Case* Masukkan Bobot Kriteria

Aktor	Sistem
1. Menekan tombol <i>update</i> pada kolom aksi dari kriteria yang ingin diubah	2. Menampilkan formulir perubahan kriteria
3. Memasukkan bobot kriteria yang baru dan tekan tombol simpan	3. Menyimpan data yang baru dan menampilkan hasil perubahan.

Nama *Use Case* : Pilih Daftar Pelamar

Aktor : *User*

Pre-condition : Aktor berada pada halaman awal seleksi resume

Post-condition : Daftar pelamar berhasil disimpan sistem

Deskripsi : *Use Case* ini menggambarkan aktivitas ketika aktor memilih daftar pelamar untuk diseleksi.

Tabel 3.12 Deskripsi Aksi Aktor dan Respon Sistem Untuk *Use Case* Pilih Daftar Pelamar

Aktor	Sistem
1. Memilih posisi yang mana pelamarnya akan diseleksi, kemudian menekan tombol <i>search</i>	2. Menampilkan daftar pelamar dari posisi tersebut
	3. Menyimpan daftar pelamar yang dipilih

Nama *Use Case* : Perankingan Kualifikasi Pelamar

Aktor : Aktor

Pre-condition : Daftar pelamar dan tingkat prioritas kriteria sudah ditentukan

Post-condition : Hasil pemeringkatan tersimpan ke *database*

Deskripsi : *Use Case* ini menggambarkan aktivitas ketika Aktor memberikan perintah ke sistem untuk melakukan pemeringkatan kualifikasi pelamar.

Tabel 3.13 Deskripsi Aksi Aktor dan Respon Sistem Untuk *Use Case* Pemeringkatan Kualifikasi Pelamar

Aktor	Sistem
1. Menekan tombol Lakukan Seleksi pada halaman lihat data pelamar seleksi	2. Mengambil data pelamar yang akan diseleksi
	3. Melakukan proses perhitungan metode DiA
	4. Menyimpan hasil perhitungan ke <i>database</i>
	5. Menampilkan hasil pemeringkatan ke pengguna.

Nama *Use Case* : Lihat Hasil Pemeringkatan

Aktor : *User*

Pre-condition : Pemeringkatan kualifikasi pelamar telah dilakukan

Post-condition : -

Deskripsi : *Use Case* ini menggambarkan aktivitas ketika aktor melihat hasil pemeringkatan kualifikasi pelamar.

Tabel 3.14 Deskripsi Aksi Aktor dan Respon Sistem Untuk *Use Case* Lihat Hasil Pemeringkatan

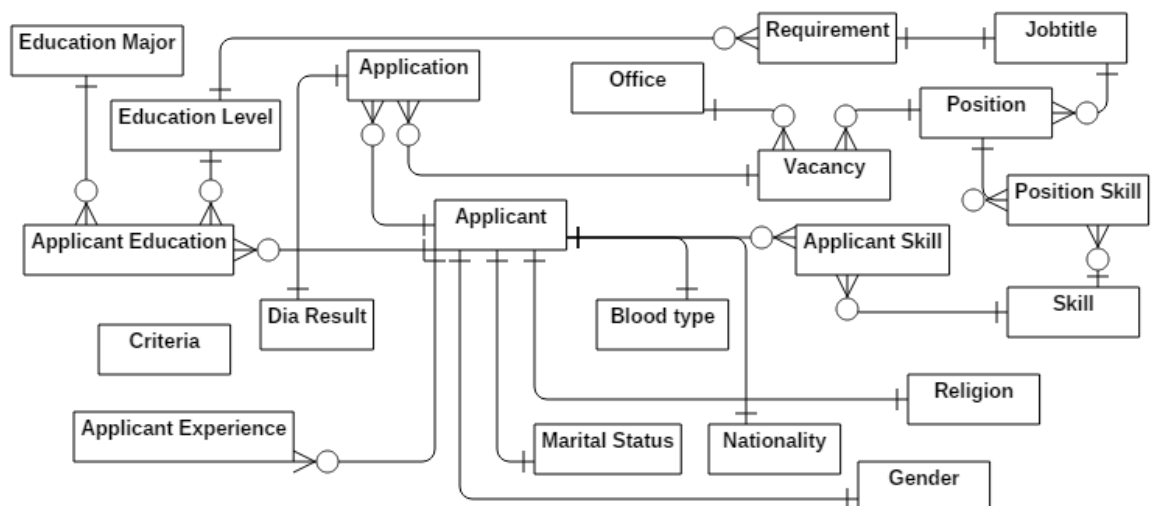
Aktor	Sistem
	1. Menampilkan hasil pemeringkatan kualifikasi pelamar
2. Melihat hasil pemeringkatan	

3.5 Kebutuhan Non-Fungsional

Kebutuhan non-fungsional dari sistem adalah sebagai berikut :

1. Sistem dapat digunakan oleh pengguna yang sudah *login*.
2. Sistem dapat digunakan pada *cross-browser*.

3.5.1 *Conceptual Database Design*



Gambar 3.10 *Conceptual Data Model*

Pada Gambar 3.10 dapat dilihat bahwa pada tahap *conceptual design*, rancangan basis data dari sistem terdiri dari 21 tabel. Nama yang melekat pada masing-masing tabel mencirikan data yang disimpan pada tabel tersebut. Terdapat beberapa tabel yang merupakan master, seperti *marital status*, *nationality*, *blood type*, *gender*, *religion*, *education major*, dan *education level*. Tabel master tidak dapat dikelola melalui sistem. Tabel *Dia Result* digunakan untuk menyimpan data hasil perhitungan menggunakan metode DiA.

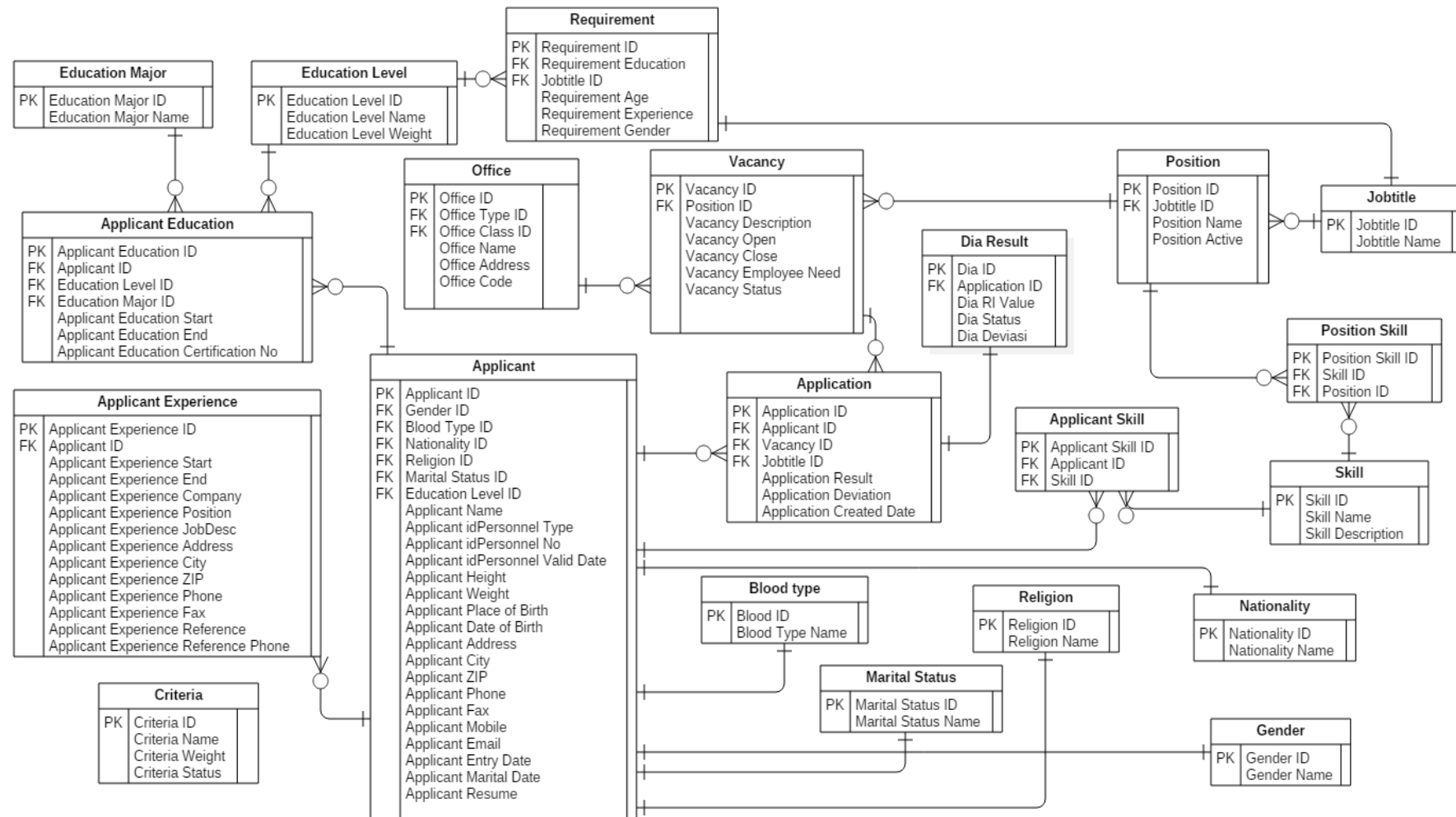
Pada rancangan basis data ini, terdapat beberapa tabel yang bersifat *many-to-many*, yaitu *applicant education* (menghubungkan tabel *applicant* dengan *education major* dan *education level*), *application* (menghubungkan tabel *applicant* dengan *vacancy*), *vacancy* (menghubungkan tabel *office* dan *position*), *position skill* (menghubungkan tabel *position* dan *skill*), *applicant skill* (menghubungkan tabel *applicant* dengan *skill*).

3.5.2 *Logical Database Design*

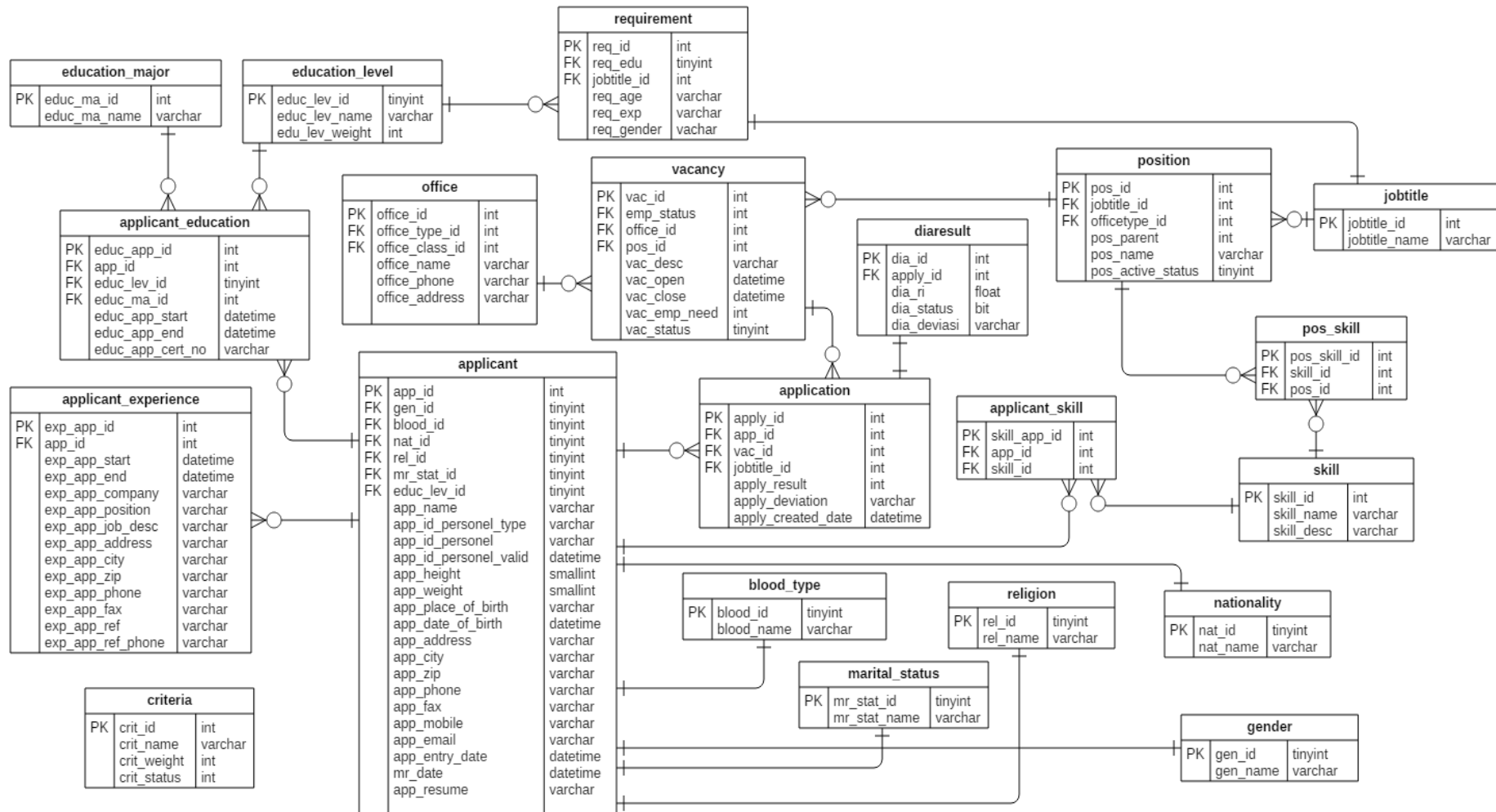
Gambar 3.11 merupakan rancangan *logical design* dari basis data modul rekrutmen pada sistem. Perbedaan *logical design* dengan *conceptual design* ialah keberadaan daftar atribut pada tiap-tiap tabel. Selain adanya daftar atribut, *primary key* serta *foreign key* pada tiap tabel juga telah diidentifikasi. Sedangkan pada rancangan *conceptual design* yang telah teridentifikasi hanya daftar tabel serta hubungan antar tabel satu dengan yang lainnya saja.

3.5.3 *Physical Database Design*

Gambar 3.12 merupakan rancangan *physical design* dari basis data modul rekrutmen pada sistem. Tidak banyak perbedaan antara *physical design* dengan *logical design*, hanya pada *physical design* masing-masing atribut pada tiap tabel telah memiliki tipe data. Dengan diidentifikasinya daftar tabel, atribut, serta tipe data atribut pada tiap tabel, pada tahap ini, maka rancangan basis data sudah selesai dilakukan. Rancangan ini kemudian akan diimplementasikan untuk mengorganisir penyimpanan data yang dibutuhkan oleh sistem nantinya.

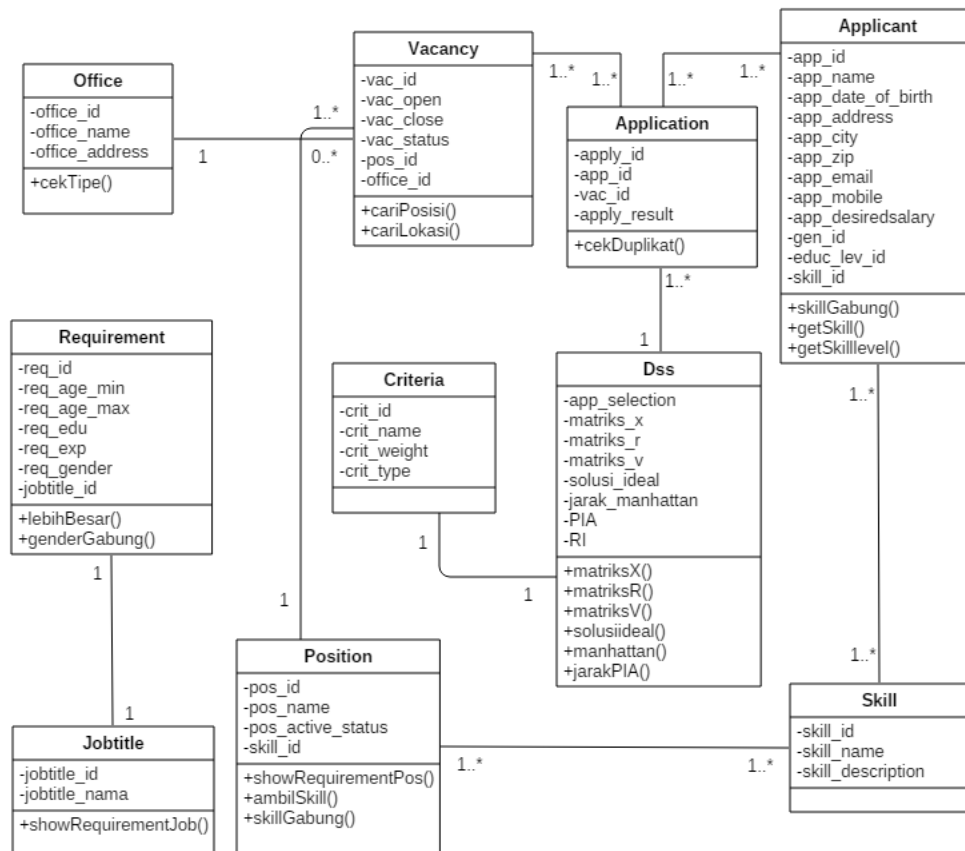


Gambar 3.11 Logical Data Model



Gambar 3.12 Physical Data Model

3.5.4 Class Diagram



Gambar 3.13 Physical Data Model

Class diagram dari modul rekrutmen sistem informasi seleksi pegawai dapat dilihat pada Gambar 4.4. Berikut penjelasan ringkas dari tiap kelas :

- a) *Applicant* : Kelas ini mengelola data pelamar, yaitu informasi diri.
- b) *Jobtitle* : Kelas ini mengelola data jabatan.
- c) *Position* : Kelas ini mengelola data posisi.
- d) *Office* : Kelas ini mengelola data kantor.
- e) *Skill* : Kelas ini mengelola data kemampuan.
- f) *Requirement* : Kelas ini mengelola data *requirement* dari *jobtitle*.
- g) *Criteria* : Kelas ini mengelola data kriteria yang akan dinilai ketika melakukan pemeringkatan.
- h) *Dss* : Kelas ini mengelola proses perhitungan DiA serta pemeringkatan pelamar.
- i) *Vacancy* : Kelas ini mengelola data lowongan pekerjaan
- j) *Application* : Kelas ini mengelola data aplikasi lamaran perusahaan.

Lampiran 3 – Hasil Pengujian *Black Box*

Nama Penguji : Rifky Hidayat Departemen : Human Capital Recruitment & Assessment			
No.	Description	Expected Result	Actual Result
1.	Penguji membuka sistem melalui <i>web browser</i> .	Sistem Menampilkan halaman <i>login</i> .	<input checked="" type="checkbox"/> Benar <input type="checkbox"/> Salah
2.	Penguji memasukkan <i>username</i> dan <i>password</i> yang salah saat <i>login</i> .	Sistem memberikan notifikasi bahwa <i>username</i> dan <i>password</i> salah. Penguji tidak dapat masuk ke dalam sistem	<input checked="" type="checkbox"/> Benar <input type="checkbox"/> Salah
3.	Penguji memasukkan <i>username</i> dan <i>password</i> yang benar saat <i>login</i> .	Sistem menampilkan halaman utama. Penguji berhasil masuk ke dalam sistem.	<input checked="" type="checkbox"/> Benar <input type="checkbox"/> Salah
4.	Penguji memilih menu Data master kemudian pilih sub menu <i>job title</i>	Sistem menampilkan halaman kelola <i>job title</i> yang berisi informasi <i>job title</i> yang ada dalam basis data. Penguji dapat menambah, melihat, mengubah, dan menghapus data <i>job title</i> dengan menekan tombol yang telah disediakan untuk tiap aksi.	<input checked="" type="checkbox"/> Benar <input type="checkbox"/> Salah
5.	Penguji memilih menu Data master kemudian pilih sub menu posisi.	Sistem menampilkan halaman kelola posisi yang berisi informasi posisi yang ada dalam basis data. Penguji dapat menambah, melihat, mengubah, dan menghapus data posisi dengan menekan tombol yang telah disediakan untuk tiap aksi.	<input checked="" type="checkbox"/> Benar <input type="checkbox"/> Salah
6.	Penguji memilih menu Data master kemudian pilih sub menu kualifikasi kemudian Kemampuan.	Sistem menampilkan halaman kelola Kemampuan yang berisi informasi kemampuan yang ada dalam basis data. Penguji dapat menambah, melihat, mengubah, dan menghapus data kemampuan dengan menekan tombol yang telah disediakan untuk tiap aksi.	<input checked="" type="checkbox"/> Benar <input type="checkbox"/> Salah

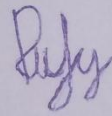
7.	Penguji memilih menu Data master kemudian pilih sub menu kualifikasi kemudian persyaratan umum.	Sistem menampilkan halaman kelola persyaratan umum yang berisi informasi persyaratan umum yang ada dalam basis data. Penguji dapat melihat, mengubah, dan menghapus data persyaratan khusus dengan menekan tombol yang telah disediakan untuk tiap aksi.	<input checked="" type="checkbox"/> Benar <input type="checkbox"/> Salah
8.	Penguji memilih menu Data master kemudian pilih sub menu kualifikasi kemudian persyaratan khusus.	Sistem menampilkan halaman kelola persyaratan khusus yang berisi informasi persyaratan khusus yang ada dalam basis data. Penguji dapat melihat, mengubah, dan menghapus data persyaratan khusus dengan menekan tombol yang telah disediakan untuk tiap aksi.	<input checked="" type="checkbox"/> Benar <input type="checkbox"/> Salah
9.	Penguji memilih menu Data master kemudian pilih sub menu kantor.	Sistem menampilkan halaman kelola kantor yang berisi informasi kantor yang ada dalam basis data. Penguji dapat menambah, melihat, mengubah, dan menghapus data kantor dengan menekan tombol yang telah disediakan untuk tiap aksi.	<input checked="" type="checkbox"/> Benar <input type="checkbox"/> Salah
10.	Penguji memilih menu Rekrutmen kemudian pilih sub menu lowongan kerja.	Sistem menampilkan halaman kelola lowongan kerja yang berisi informasi lowongan kerja yang ada dalam basis data. Penguji dapat menambah, melihat, mengubah, dan menghapus data lowongan kerja dengan menekan tombol yang telah disediakan untuk tiap aksi.	<input checked="" type="checkbox"/> Benar <input type="checkbox"/> Salah
11.	Penguji memilih menu Rekrutmen kemudian pilih sub menu pelamar.	Sistem menampilkan halaman kelola pelamar yang berisi informasi pelamar yang ada dalam basis data. Penguji dapat menambah, melihat, mengubah, dan menghapus data pelamar dengan menekan tombol yang telah disediakan untuk tiap aksi.	<input checked="" type="checkbox"/> Benar <input type="checkbox"/> Salah

12.	Penguji memilih menu Rekrutmen kemudian pilih sub menu lamaran.	Sistem menampilkan halaman kelola lamaran yang berisi informasi lamaran yang ada dalam basis data. Penguji dapat menambah, melihat, mengubah, dan menghapus data lamaran dengan menekan tombol yang telah disediakan untuk tiap aksi.	<input checked="" type="checkbox"/> Benar <input type="checkbox"/> Salah
13.	Penguji memilih menu Rekrutmen kemudian pilih sub menu Seleksi resume.	Sistem menampilkan halaman awal seleksi resume yang memiliki <i>dropdown</i> berisi daftar lowongan kerja yang dapat dipilih penguji untuk menentukan daftar pelamar untuk diseleksi	<input checked="" type="checkbox"/> Benar <input type="checkbox"/> Salah
14.	Penguji membuka halaman untuk memilih daftar pelamar yang akan diseleksi kemudian menekan tombol simpan dan pilih kriteria. Kondisi : penguji belum memilih daftar pelamar untuk diseleksi atau daftar pelamar berjumlah kurang dari tiga pelamar	Sistem akan menampilkan <i>pop-up</i> yang menghalangi pengguna untuk melanjutkan proses seleksi resume.	<input checked="" type="checkbox"/> Benar <input type="checkbox"/> Salah
15.	Penguji membuka halaman untuk memilih daftar pelamar yang akan diseleksi kemudian menekan tombol simpan dan pilih kriteria. Kondisi : penguji sudah memilih daftar pelamar untuk diseleksi sebanyak lebih dari dua pelamar	Sistem akan menampilkan halaman manajemen kriteria	<input checked="" type="checkbox"/> Benar <input type="checkbox"/> Salah
16.	Penguji membuka halaman manajemen kriteria	Sistem menampilkan halaman kelola kriteria yang berisi informasi kriteria yang ada dalam basis data. Penguji dapat melihat dan mengubah data kriteria dengan menekan tombol yang telah disediakan untuk tiap aksi.	<input checked="" type="checkbox"/> Benar <input type="checkbox"/> Salah
17.	Penguji berada pada halaman	Sistem melakukan perhitungan dan	<input checked="" type="checkbox"/> Benar

	lihat data daftar pelamar, kemudian menekan tombol lakukan seleksi	pemeringkatan pelamar kemudian menampilkan hasil pemeringkatan ke layar	<input type="checkbox"/> Salah
18.	Penguji menekan menu dengan nama pengguna pada <i>navbar</i> kemudian menekan tombol <i>sign out</i>	Sistem mengakhiri sesi penguji kemudian menampilkan halaman <i>login</i> .	<input checked="" type="checkbox"/> Benar <input type="checkbox"/> Salah

Jakarta, 31 Agustus 2016

Mengetahui,



(Rifky)