



TUGAS AKHIR - SM 141501

**PERANCANGAN DAN IMPLEMENTASI
PERANGKAT LUNAK SISTEM PENDUKUNG
KEPUTUSAN MULTI KRITERIA MENGGUNAKAN
METODE AHP**

SITI MAGHIROH
NRP 1212 100 033

Dosen Pembimbing
Prof. Dr. Mohammad Isa Irawan, MT

JURUSAN MATEMATIKA
Fakultas Matematika dan Ilmu Pengetahuan Alam
Institut Teknologi Sepuluh Nopember
Surabaya 2016



FINAL PROJECT - SM 141501

**DESIGN AND IMPLEMENTATION SOFTWARE
OF MULTI CRITERIA DECISION SUPPORT
SYSTEM USING AHP METHOD**

SITI MAGHIROH
NRP 1212 100 033

Supervisor
Prof. Dr. Mohammad Isa Irawan, MT

DEPARTMENT OF MATHEMATICS
Faculty of Mathematics and Natural Science
Sepuluh Nopember Institute of Technology
Surabaya 2016

**LEMBAR PENGESAHAN
PERANCANGAN DAN IMPLEMENTASI PERANGKAT
LUNAK SISTEM PENDUKUNG KEPUTUSAN MULTI
KRITERIA MENGGUNAKAN METODE AHP**

***DESIGN AND IMPLEMENTATION SOFTWARE OF MULTI
CRITERIA DECISION SUPPORT SYSTEM USING AHP
METHOD***

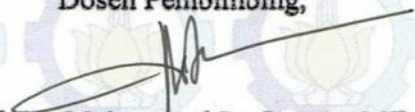
TUGAS AKHIR

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

Oleh :

**SITI MAGHFIROH
NRP. 1212 100 033**

Menyetujui,
Dosen Pembimbing,


Prof. Dr. Mohammad Isa Irawan, MT.
NIP. 19631225 198903 1 001

Mengetahui,
Ketua Jurusan Matematika,


Dr. Imam Mukhlash, S.Si, MT
NIP. 19700831 199403 1 003

Surabaya, 15 Juli 2016



PERANCANGAN DAN IMPLEMENTASI PERANGKAT LUNAK SISTEM PENDUKUNG KEPUTUSAN MULTI KRITERIA MENGGUNAKAN METODE AHP

Nama : Siti Maghfiroh
NRP : 1212 100 033
Jurusan : Matematika
Dosen Pembimbing : Prof. Dr. Mohammad Isa Irawan, MT

ABSTRAK

Sejalan dengan pesatnya perkembangan industri, pengambilan keputusan yang cepat dan akurat menjadi bagian yang penting dalam suatu instansi agar roda organisasi berjalan dengan lancar. Untuk mendukung hal tersebut, dibuat suatu sistem berbasis komputer yang dapat memudahkan proses pengambilan keputusan yaitu Sistem Pendukung Keputusan (SPK). Analytic Hierarchy Process (AHP) merupakan salah satu metode Multi Criteria Decision Making (MCDM) yang dapat digunakan dalam pengambilan keputusan. Agar dapat digunakan secara efektif, AHP perlu disusun sedemikian rupa sehingga mudah digunakan. Pada penelitian ini, dibuat perangkat lunak SPK yang dibangun dengan mengimplementasikan model AHP untuk menyelesaikan permasalahan pengambilan keputusan. Pada metode AHP, permasalahan dimodelkan dalam sebuah hirarki dan ranking alternatif ditentukan berdasarkan prioritas. Prioritas dihitung berdasarkan nilai perbandingan berpasangan yang diberikan oleh decision maker. Hasil pengujian menunjukkan bahwa perangkat lunak yang dibangun memberikan ranking alternatif yang sama dengan penelitian sebelumnya sehingga dengan adanya perangkat lunak ini decision maker tidak perlu membangun perangkat lunak baru untuk permasalahan yang berbeda.

Kata kunci : AHP, perangkat lunak, SPK.

DESIGN AND IMPLEMENTATION SOFTWARE OF MULTI CRITERIA DECISION SUPPORT SYSTEM USING AHP METHOD

Name of Student : Siti Maghfiroh
NRP : 1212 100 033
Department : Mathematics
Supervisor : Prof. Dr. Mohammad Isa Irawan, MT

ABSTRACT

By the rapid development of industry, accurate and quick decision making being important in an organization so that all activity going well. For supporting it, computer based system is build named Decision Support System (DSS). Analytic Hierarchy Process (AHP) is one of Multi Criteria Decision Making method that use in decision making. In order to be effectively used in decision making, AHP needs to be customized so that it is more user friendly with ease of used features. In this paper, DSS software is build by implementation of AHP model to solve decision making problems. In AHP method, the problem is modeled in a hierarchy and the rank alternative obtain by given priority. Priority is calculated by input value of decision matrix in each level of hierarchy. The result show that software obtain the same value with previous research. So that, decision maker do not need to built new software to solve another decision making problem.

Keyword : AHP, DSS, Software.

KATA PENGANTAR

Segala puji syukur penulis panjatkan ke hadirat Allah SWT, karena dengan ridlo-Nya penulis dapat menyelesaikan Tugas Akhir yang berjudul

“PERANCANGAN DAN IMPLEMENTASI PERANGKAT LUNAK SISTEM PENDUKUNG KEPUTUSAN MULTI KRITERIA MENGGUNAKAN METODE AHP”

yang merupakan salah satu persyaratan akademis dalam menyelesaikan Program Sarjana Jurusan Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Institut Teknologi Sepuluh Nopember Surabaya.

Tugas Akhir ini dapat diselesaikan dengan baik berkat kerja sama, bantuan, dan dukungan dari banyak pihak. Sehubungan dengan hal tersebut, penulis ingin mengucapkan terima kasih kepada :

1. Dr. Imam Mukhlash, S.Si, MT selaku Ketua Jurusan Matematika ITS.
2. Dr. Mahmud Yunus, M.Si selaku Dosen Wali yang telah memberikan arahan akademik selama penulis menempuh pendidikan di Jurusan Matematika ITS.
3. Prof. Dr. Mohammad Isa Irawan, MT selaku Dosen Pembimbing yang telah memberikan bimbingan dan motivasi kepada penulis dalam mengerjakan Tugas Akhir ini sehingga dapat terselesaikan dengan baik.
4. Drs. Soetrisno, Ml.Komp., Drs. Daryono Budi Utomo, M.Si, Valeriana Lukitosari, S.Si, MT, Alvida Mustika Rukmi, S.Si, M.Si selaku Dosen Penguji yang telah memberikan saran demi perbaikan Tugas Akhir ini.
5. Dr. Didik Khusnul Arif, S.Si, M.Si selaku Ketua Program Studi S1 Jurusan Matematika ITS.
6. Seluruh jajaran dosen dan staf Jurusan Matematika ITS.
7. Keluarga tercinta yang senantiasa memberikan dukungan dan do'a yang tak terhingga.

8. Tri Wahyuni yang telah meluangkan banyak waktu menjadi sahabat terbaik.
9. Teman seperjuangan Aul yang sudah banyak membantu sejak awal penyusunan Tugas Akhir ini.
10. Lailatul Izza yang telah memberikan bantuan dalam pengerjaan Tugas Akhir ini.
11. Teman-teman angkatan 2012 yang saling mendukung dan memotivasi.
12. Semua pihak yang tak bisa penulis sebutkan satu-persatu, terima kasih telah membantu sampai terselesaikannya Tugas Akhir ini.

Penulis menyadari bahwa Tugas Akhir ini masih jauh dari kesempurnaan. Oleh karena itu, penulis mengharapkan kritik dan saran dari pembaca. Akhir kata, semoga Tugas Akhir ini dapat bermanfaat bagi semua pihak yang berkepentingan.

Surabaya, Juli 2016
Penulis

DAFTAR ISI

HALAMAN JUDUL	i
LEMBAR PENGESAHAN	v
ABSTRAK	vii
ABSTRACT	ix
KATA PENGANTAR	xi
DAFTAR ISI	xiii
DAFTAR GAMBAR	xv
DAFTAR TABEL	xvii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah	3
1.4 Tujuan	4
1.5 Manfaat	4
1.6 Sistematika Penulisan Tugas Akhir	4
BAB II TINJAUAN PUSTAKA	7
2.1 Sistem Pendukung Keputusan (SPK)	7
2.2 <i>Analytic Hierarchy Process (AHP)</i>	8
2.3 Java	19
2.4 MySQL	20
2.5 Rekayasa Perangkat Lunak	21
BAB III METODOLOGI	25
3.1 Tahapan Penelitian	25
3.2 Diagram Alir Metode Penelitian	26
BAB IV ANALISIS DAN PERANCANGAN PERANGKAT LUNAK	29
4.1 Gambaran Umum Perangkat Lunak	29
4.2 Analisis Kebutuhan Perangkat Lunak	29
4.3 Perancangan Perangkat Lunak	30
4.2.1 <i>Usecase Diagram</i>	30

4.2. 2	<i>Activity Diagram</i>	31
4.2. 3	<i>Pseudocode</i>	34
4.2. 4	Desain <i>Interface</i>	34
4.2. 5	Database Arsitektur	38
BAB V IMPLEMENTASI DAN PENGUJIAN PERANGKAT LUNAK.....		41
4. 1	Implementasi Perangkat Lunak.....	41
4.1. 1	Form Input Kasus Baru	42
4.1. 2	Form Input Perbandingan Berpasangan	43
4.1. 3	Form Hasil Ranking	43
4.1. 4	Form Daftar Kasus	44
4.1. 5	Form Detail Kasus.....	44
4.1. 6	Form Help	46
4. 2	Pengujian Perangkat Lunak.....	46
BAB VI KESIMPULAN DAN SARAN.....		61
5.1	Kesimpulan.....	61
5.2	Saran	61
DAFTAR PUSTAKA		63
LAMPIRAN A <i>Source Code</i>		65
LAMPIRAN B Data Uji.....		95
LAMPIRAN C Biodata Penulis		101

DAFTAR GAMBAR

Gambar 2.1	Skema SPK.....	8
Gambar 2.2	Struktur Hirarki.....	9
Gambar 2.3	Struktur Hirarki.....	14
Gambar 2.4	Tahapan model sekuensial linear.....	22
Gambar 3.1	Metode Penelitian.....	27
Gambar 4.1	<i>Usecase Diagram</i>	31
Gambar 4.2	<i>Activity Diagram</i> Perankingan Alternatif ...	32
Gambar 4.3	<i>Activity Diagram</i> Kelola Data.....	33
Gambar 4.4	<i>Activity Diagram</i> Bantuan.....	33
Gambar 4.5	Desain <i>Homepage</i>	35
Gambar 4.6	Desain Input Kasus Baru.....	35
Gambar 4.7	Desain Input Perbandingan Berpasangan ...	36
Gambar 4.8	Desain Halaman Hasil Ranking.....	36
Gambar 4.9	Desain Halaman Daftar Kasus.....	37
Gambar 4.10	Desain Halaman Detail Kasus.....	37
Gambar 4.11	Desain Halaman Help.....	38
Gambar 4.12	Database Arsitektur.....	39
Gambar 5.1	Menu Utama SPK – AHP.....	41
Gambar 5.2	Form Masukkan Kasus Baru.....	42
Gambar 5.3	Form Input Perbandingan Berpasangan.....	43
Gambar 5.4	Form Hasil Ranking.....	43
Gambar 5.5	Form Daftar Kasus.....	44
Gambar 5.6	Form Detail Kasus.....	45
Gambar 5.7	Kotak Dialog Direktori Penyimpanan File.....	45
Gambar 5.8	Form Help.....	46
Gambar 5.9	Input Data Kasus.....	47
Gambar 5.10	Perbandingan Berpasangan Alternatif Subkriteria TGI.....	48
Gambar 5.11	Perbandingan Berpasangan Alternatif Subkriteria NPV.....	48

Gambar 5.12	Perbandingan Berpasangan Alternatif Subkriteria SDT.....	48
Gambar 5.13	Matriks Normalisasi Alternatif Subkriteria TGI	49
Gambar 5.14	Matriks Normalisasi Alternatif Subkriteria NPV	49
Gambar 5.15	Matriks Normalisasi Alternatif Subkriteria SDT	49
Gambar 5.16	Vektor Preferensi Alternatif	49
Gambar 5.17	Matriks Normalisasi Subkriteria untuk Kriteria Finansial	50
Gambar 5.18	Matriks Normalisasi Subkriteria untuk Kriteria Teknis.....	50
Gambar 5.19	Vektor Preferensi Subkriteria.....	51
Gambar 5.20	Matriks Normalisasi dan Vektor Preferensi Kriteria.....	51
Gambar 5.21	Bobot Global Subkriteria.....	52
Gambar 5.22	Bobot Global Alternatif.....	52
Gambar 5.23	Ranking Alternatif	53
Gambar 5.24	Kotak Dialog <i>Inconsisten</i>	53

DAFTAR TABEL

Tabel 2.1	Skala Penilaian Perbandingan Pasangan	11
Tabel 2.2	Daftar <i>Index Random Consistency</i>	13
Tabel 4.1	Kebutuhan Perangkat Lunak	30
Tabel 5.1	Hasil Perankingan.....	47
Tabel 5.2	Perbandingan Berpasangan Kriteria.....	54
Tabel 5.3	Matriks Normalisasi Kriteria.....	54
Tabel 5.4	Perbandingan Berpasangan Subkriteria untuk Kriteria Finansial	54
Tabel 5.5	Matriks Normalisasi Subkriteria untuk Kriteria Finansial	54
Tabel 5.6	Perbandingan Berpasangan Subkriteria untuk Kriteria Teknis.....	55
Tabel 5.7	Matriks Normalisasi Subkriteria untuk Kriteria Teknis	55
Tabel 5.8	Perbandingan Berpasangan Alternatif untuk Subkriteria TGI.....	56
Tabel 5.9	Matriks Normalisasi Alternatif untuk Subkriteria TGI	56
Tabel 5.10	Perbandingan Berpasangan Alternatif untuk Subkriteria NPV	57
Tabel 5.11	Matriks Normalisasi Alternatif untuk Subkriteria NPV	57
Tabel 5.12	Perbandingan Berpasangan Alternatif untuk Subkriteria SDT.....	58
Tabel 5.13	Matriks Normalisasi Alternatif untuk Subkriteria SDT	58
Tabel 5.14	Matriks Preferensi Alternatif.....	59
Tabel 5.15	Ranking Alternatif.....	59

BAB I PENDAHULUAN

Pada bab ini dibahas mengenai latar belakang yang mendasari penulisan Tugas Akhir ini. Di dalamnya mencakup identifikasi permasalahan pada topik Tugas Akhir kemudian dirumuskan menjadi permasalahan yang diberikan batasan-batasan dalam pembahasan pada Tugas Akhir ini.

1.1 Latar Belakang

Pengambilan keputusan merupakan suatu bentuk pemilihan alternatif terbaik diantara berbagai alternatif untuk mencapai tujuan tertentu. Keputusan yang diambil memiliki tingkat kesulitan yang berbeda-beda tergantung dari banyaknya alternatif atau pilihan yang tersedia. Salah satu metode yang dapat digunakan dalam pengambilan keputusan yaitu *Multi Criteria Decision Making* (MCDM). MCDM merupakan suatu metode untuk menetapkan alternatif terbaik dari sejumlah alternatif berdasarkan beberapa kriteria tertentu. Kriteria yang dimaksud dapat berupa ukuran-ukuran, aturan-aturan atau standar yang digunakan dalam pengambilan keputusan. Ada beberapa metode yang dapat digunakan untuk menyelesaikan masalah MCDM, antara lain metode *Simple Additive Weighting* (SAW), *Weighted Product* (WP), ELECTRE, *Technique for Order Preference by Similarity to Ideal Solution* (TOPSIS), dan *Analytic Hierarchy Process* (AHP). [1]

Sejalan dengan pesatnya perkembangan industri, pengambilan keputusan yang cepat dan akurat menjadi bagian yang penting dalam suatu instansi agar roda organisasi berjalan dengan lancar. Untuk mendukung hal tersebut, dibuat suatu sistem yang dapat memudahkan proses pengambilan keputusan yaitu Sistem Pendukung Keputusan (SPK). SPK merupakan sistem interaktif berbasis komputer yang mendukung *decision maker* dalam proses pengambilan keputusan. Hingga saat ini, banyak aplikasi SPK yang telah dibuat untuk membantu

menyelesaikan pengambilan keputusan salah satunya dengan metode AHP. Pada metode AHP, permasalahan dimodelkan dalam sebuah hirarki. Dengan hirarki, suatu masalah yang kompleks dapat diuraikan ke dalam kelompok-kelompok sehingga permasalahan akan tampak lebih terstruktur dan sistematis. Selain itu, pada metode AHP juga ditentukan penilaian dan dipertimbangkan konsistensinya untuk menentukan prioritas. Prioritas inilah yang nantinya digunakan untuk menentukan alternatif terbaik.

Adapun beberapa peneliti yang menggunakan metode AHP untuk menyelesaikan permasalahan pengambilan keputusan antara lain J.L. Garcia dkk (2013), Saefudin dkk (2014), A. Y. Ranius (2015) dll. Garcia menggunakan metode AHP untuk memilih lokasi distribusi hasil tani dengan memperhatikan enam kriteria meliputi kemudahan akses, jarak, keamanan, jumlah kebutuhan, penerimaan, dan biaya. [2] Kemudian Saefudin menggunakan SPK untuk penilaian kinerja pegawai pada RSUD Serang menggunakan metode AHP. Saefudin menentukan empat alternatif pegawai dengan kriteria prestasi kerja, tanggung jawab, kejujuran, kerjasama, prakarsa, ketaatan, kepemimpinan, dan kesetiaan. [3] Sedangkan Ranius menggunakan SPK untuk penentuan destinasi wisata unggulan di kota Palembang. Ranius menetapkan 15 destinasi untuk pengujian sistem dengan kriteria wisata alam, wisata sejarah budaya, dan wisata minat khusus. Kemudian diambil tiga destinasi wisata dengan ranking terbaik pada masing-masing kriteria wisata. [4] Selain itu, dalam bidang industri yang lain juga banyak permasalahan pengambilan keputusan dengan kriteria yang kompleks sehingga digunakan metode AHP untuk memodelkan permasalahan agar lebih terstruktur. Misalnya untuk promosi karyawan, pemilihan supplier [5], pemilihan lokasi *Floating Storage and Regasification Unit* (FSRU) [6], pemilihan *crusher* untuk tambang besi [7], dan masih banyak lagi yang lainnya.

Berdasarkan beberapa contoh tersebut, terlihat bahwa metode AHP banyak digunakan untuk menyelesaikan

pengambilan keputusan terutama untuk permasalahan dengan kriteria yang kompleks. Sehingga pada Tugas Akhir ini dibuat perangkat lunak SPK yang dinamis menggunakan metode AHP yang dapat digunakan untuk menyelesaikan berbagai permasalahan pengambilan keputusan dengan jumlah kriteria yang dapat diubah-ubah sesuai kebutuhan masalah.

1.2 Rumusan Masalah

Berdasarkan latar belakang tersebut, maka permasalahan yang dirumuskan dalam Tugas Akhir ini adalah sebagai berikut :

1. Bagaimana merancang perangkat lunak SPK multi kriteria menggunakan metode AHP?
2. Bagaimana mengimplementasikan rancangan perangkat lunak sehingga dapat digunakan untuk berbagai permasalahan pengambilan keputusan?
3. Bagaimana menguji perangkat lunak menggunakan data penelitian yang sudah ada?

1.3 Batasan Masalah

Batasan masalah yang digunakan dalam Tugas Akhir ini antara lain :

1. Perangkat lunak dibangun menggunakan bahasa pemrograman Java dan database MySQL
2. Perangkat lunak yang dibangun hanya berlaku untuk pengambilan keputusan dengan metode AHP
3. Jumlah maksimal level hirarki adalah empat yang meliputi tujuan, kriteria, subkriteria, dan alternatif. Jika terdapat subkriteria maka setiap kriteria harus memiliki subkriteria dan alternatif berhubungan langsung dengan subkriteria
4. Jumlah maksimal kriteria, subkriteria, dan alternatif adalah 15
5. Tidak dilakukan proses perbandingan dengan metode lainnya

1.4 Tujuan

Tujuan yang ingin dicapai dari penulisan Tugas Akhir ini adalah :

1. Merancang perangkat lunak SPK multi kriteria menggunakan metode AHP
2. Mengimplementasikan rancangan perangkat lunak sehingga dapat digunakan untuk menyelesaikan berbagai permasalahan pengambilan keputusan
3. Menguji perangkat lunak menggunakan data penelitian yang sudah ada

1.5 Manfaat

Manfaat yang diharapkan dengan adanya penelitian ini adalah dapat membantu menyelesaikan berbagai permasalahan pengambilan keputusan multi kriteria menggunakan metode AHP.

1.6 Sistematika Penulisan Tugas Akhir

Penulisan laporan Tugas Akhir ini disusun dalam enam bab sebagai berikut :

1. BAB I PENDAHULUAN

Bab ini menjelaskan latar belakang penyusunan Tugas Akhir, rumusan masalah, batasan masalah, tujuan, manfaat dan sistematika penulisan Tugas Akhir.

2. BAB II TINJAUAN PUSTAKA

Bab ini menjelaskan tentang dasar teori yang mendukung penelitian antara lain tentang sistem pendukung keputusan, metode AHP (*Analytic Hierarchy Process*), bahasa pemrograman Java, database MySQL, dan rekayasa perangkat lunak.

3. BAB III METODOLOGI

Bab ini menjelaskan tentang tahap-tahap yang dilakukan dalam penyusunan Tugas Akhir.

4. BAB IV ANALISIS DAN PERANCANGAN PERANGKAT LUNAK

Bab ini menjelaskan secara detail tentang spesifikasi kebutuhan perangkat lunak yang meliputi domain informasi serta fungsi yang dibutuhkan untuk performansi dan *interface*.

5. BAB V IMPLEMENTASI DAN PENGUJIAN PERANGKAT LUNAK

Bab ini menjelaskan tentang implementasi rancangan perangkat lunak dalam bahasa pemrograman Java dan hasil uji coba perangkat lunak menggunakan data penelitian yang sudah ada.

6. BAB VI PENUTUP

Bab ini menjelaskan tentang penarikan kesimpulan yang diperoleh dari pembahasan masalah pada bab sebelumnya serta saran yang diberikan untuk pengembangan penelitian selanjutnya.

BAB II

TINJAUAN PUSTAKA

Pada bab ini dibahas mengenai dasar teori yang digunakan dalam penyusunan Tugas Akhir ini. Dasar teori yang dijelaskan dibagi menjadi beberapa subbab yaitu sistem pendukung keputusan, metode AHP (*Analytic Hierarchy Process*), bahasa pemrograman Java, database MySQL, dan rekayasa perangkat lunak.

2.1 Sistem Pendukung Keputusan (SPK)

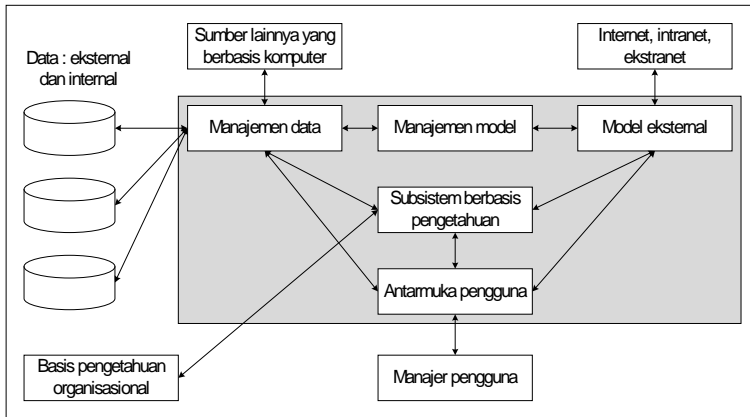
Sistem pendukung keputusan (SPK) adalah suatu sistem informasi spesifik yang ditujukan untuk membantu manajemen dalam mengambil keputusan yang berkaitan dengan persoalan yang bersifat semi terstruktur. Sistem ini memiliki fasilitas untuk menghasilkan berbagai alternatif yang secara interaktif dapat digunakan oleh pemakai.

SPK terdiri dari beberapa komponen atau subsistem sebagai berikut.

1. Subsistem manajemen data
Subsistem manajemen data merupakan komponen SPK sebagai penyedia data bagi sistem. Data tersebut disimpan dalam suatu basis data yang diorganisasikan oleh suatu sistem yang disebut *database management system* (DBMS).
2. Subsistem manajemen model
SPK mampu mengintegrasikan data dengan model-model keputusan. Model tersebut diorganisasikan oleh pengelola model yang disebut *model base*.
3. Subsistem antarmuka pengguna (*user interface*)
Sistem diimplementasikan menggunakan *user interface* sehingga pengguna dapat berkomunikasi dengan sistem yang dirancang.
4. Subsistem manajemen berbasis pengetahuan
Subsistem ini dapat mendukung subsistem yang lain atau bertindak sebagai suatu komponen independen. Subsistem

ini juga memberikan inteligensi untuk memperbesar pengetahuan *decision maker*.

SPK harus mencakup tiga komponen utama yang terdiri dari DBMS, *model base*, dan *user interface*. Subsistem manajemen berbasis pengetahuan bersifat opsional, namun dapat memberikan banyak manfaat karena memberikan inteligensi bagi tiga komponen utama tersebut. Skema SPK dan komponen-komponennya ditunjukkan pada Gambar 2.1. [8]



Gambar 2.1 Skema SPK

Sumber : Efraim Turban, et al. "*Decision Support Systems and Intelligent Systems (Sistem Pendukung Keputusan dan Sistem Cerdas)*".

2.2 Analytic Hierarchy Process (AHP)

Analytic Hierarchy Process (AHP) dikenalkan oleh Thomas L. Saaty pada tahun 1980. AHP merupakan salah satu teknik MCDM untuk memilih alternatif terbaik diantara beberapa alternatif melalui proses perbandingan berpasangan. AHP menyediakan sarana dekomposisi masalah ke dalam hirarki sehingga permasalahan lebih mudah dipahami dan dievaluasi secara subjektif. Evaluasi subjektif diubah menjadi nilai numerik

A adalah matriks ($n \times n$) dengan $a_{[i,j]}$ adalah perbandingan berpasangan antara alternatif i dan alternatif j , n jumlah alternatif. Diagonal bernilai 1 yakni ketika $i = j$ dan matriks segitiga bawah diisi dengan persamaan berikut, [9]

$$a_{[j,i]} = 1/a_{[i,j]} \quad (2)$$

AHP memungkinkan pengambilan keputusan berkelompok sehingga penilaian dilakukan oleh lebih dari satu *decision maker*. Masing-masing *decision maker* secara terpisah melakukan penilaian berdasarkan pengalaman, nilai-nilai, dan pengetahuannya. Untuk mencapai mufakat, digunakan teknik rata-rata geometri (*Geometric Mean*) yang dinyatakan oleh persamaan berikut, [10]

$$GM = \sqrt[n]{a_1 \times a_2 \times a_3 \times \dots \times a_n} \quad (3)$$

dengan $GM = \text{Geometric Mean}$

$n =$ jumlah *decision maker*

$a_n =$ penilaian *decision maker* ke- n

Decision maker dapat menilai dengan perbandingan sama penting, sedikit lebih penting, lebih penting, mutlak lebih penting, dan mutlak penting. Penilaian dibuat untuk masing-masing kriteria dan dikonversi menjadi nilai kuantitatif sesuai Tabel 2.1.

3. Sintesis

- a. Jumlahkan nilai pada setiap kolom matriks perbandingan berpasangan
- b. Bagilah setiap nilai pada setiap kolom matriks perbandingan berpasangan dengan jumlah kolom yang sesuai, ini disebut matriks normalisasi
- c. Rata-rata nilai pada setiap baris matriks normalisasi, ini disebut vektor preferensi
- d. Gabungkan vektor preferensi setiap alternatif (hasil langkah 3c) ke dalam satu matriks preferensi

Tabel 2.1 Skala Penilaian Perbandingan Pasangan

Intensitas Kepentingan	Definisi
1	Kedua elemen sama penting
3	Elemen yang satu sedikit lebih penting dibandingkan elemen yang lain
5	Elemen yang satu lebih penting dibandingkan elemen yang lain
7	Elemen yang satu mutlak lebih penting dibandingkan elemen yang lain
9	Elemen yang satu mutlak penting dibandingkan elemen yang lain
2, 4, 6, 8	Nilai-nilai antara dua pertimbangan yang berdekatan
Nilai kebalikan dari nilai di atas	Jika aktivitas i dibandingkan dengan aktivitas j , maka i memiliki nilai kebalikannya dibandingkan dengan j .

Sumber : Kusriani, “*Konsep dan Aplikasi Sistem Pendukung Keputusan*”.

4. Membuat matriks perbandingan berpasangan setiap kriteria ($B_{n \times n}$)
5. Menghitung matriks normalisasi dengan membagi setiap nilai pada setiap kolom matriks dengan jumlah kolom yang sesuai
6. Membuat vektor preferensi kriteria dengan menghitung rata-rata baris matriks normalisasi ($V_{n \times 1}$)
7. Menghitung nilai keseluruhan setiap alternatif dengan mengalikan vektor preferensi kriteria (hasil langkah 6) dengan matriks preferensi (hasil langkah 3d)
8. Alternatif diranking berdasarkan nilai yang diperoleh dari langkah 7

9. Menghitung λ_{maks}
- Kalikan matriks perbandingan berpasangan setiap kriteria (hasil langkah 4) dengan vektor preferensi kriteria (hasil langkah 6)

$$M_{n \times 1} = B_{(n \times n)} \cdot V_{(n \times 1)} \quad (4)$$

- Bagilah hasil 9a dengan vektor preferensi kriteria (hasil langkah 6) yang bersesuaian

$$d_{i1} = \frac{m_{i1}}{v_{i1}} \quad (5)$$

- Jumlahkan hasil 9b, kemudian dibagi dengan jumlah kriteria

$$\lambda_{maks} = \frac{\sum_{i=1}^n d_{i1}}{n} \quad (6)$$

10. Menghitung *Consistency Index* (CI)

$$CI = \frac{\lambda_{maks} - n}{n - 1} \quad (7)$$

11. Menghitung *Consistency Ratio* (CR)

$$CR = \frac{CI}{IR} \quad (8)$$

dengan $CR = \text{Consistency Ratio}$

$CI = \text{Consistency Index}$

$IR = \text{Index Random consistency}$

Saaty mengatakan bahwa nilai CR harus kurang dari 0,1. Jika nilainya lebih dari 0,1, maka penilaian harus diperbaiki. Nilai *Index Random consistency* dapat dilihat pada Table 2.2. [9]

Tabel 2.2 Daftar *Index Random Consistency*

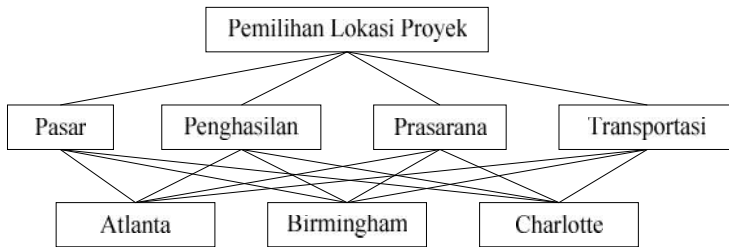
Ukuran Matriks	Nilai IR
1,2	0
3	0,58
4	0,90
5	1,12
6	1,24
7	1,32
8	1,41
9	1,45
10	1,49
11	1,51
12	1,48
13	1,56
14	1,57
15	1,59

Sumber : Kusriani, "*Konsep dan Aplikasi Sistem Pendukung Keputusan*".

Contoh operasional metode AHP pada kasus Pemilihan Lokasi Proyek [9]

Perusahaan pengembang perumahan hendak memilih salah satu lokasi untuk pembangunan proyek baru. Dengan mempertimbangkan aspek pasar pelanggan, penghasilan penduduk, sarana prasarana, dan sistem transportasi terpilih tiga alternatif lokasi yaitu Atlanta, Birmingham, dan Charlotte.

1. Struktur hirarki



Gambar 2.3 Struktur Hirarki

2. Membuat matriks perbandingan berpasangan setiap alternatif untuk setiap kriteria menggunakan persamaan (1)

Matriks perbandingan berpasangan kriteria Pasar

Pasar	A	B	C
A	1	3	2
B	1/3	1	1/5
C	1/2	5	1

Matriks perbandingan berpasangan kriteria Penghasilan

Penghasilan	A	B	C
A	1	6	1/3
B	1/6	1	1/9
C	3	9	1

Matriks perbandingan berpasangan kriteria Prasarana

Prasarana	A	B	C
A	1	1/3	1
B	3	1	7
C	1	1/7	1

Matriks perbandingan berpasangan kriteria Transportasi

Transportasi	A	B	C
A	1	1/3	1/2
B	3	1	4
C	2	1/4	1

3. Sintesis

- a. Jumlahkan nilai pada setiap kolom matriks perbandingan berpasangan

Pasar	A	B	C
A	1	3	2
B	1/3	1	1/5
C	1/2	5	1
Jumlah	1.8333	9	3.2

$$1.8333 = 1 + 1/3 + 1/2$$

$$9 = 3 + 1 + 5$$

$$3.2 = 2 + 1/5 + 1$$

- b. Bagilah setiap nilai pada setiap kolom matriks perbandingan berpasangan dengan jumlah kolom yang sesuai, ini disebut matriks normalisasi

Pasar	A	B	C
A	0.5455	0.3333	0.6250
B	0.1818	0.1111	0.0625
C	0.2727	0.5556	0.3125
Jumlah	1	1	1

$$0.5455 = 1/1.8333, \quad 0.3333 = 3/9, \quad 0.6250 = 2/3.2$$

$$0.1818 = 1/5.5, \quad 0.1111 = 1/9, \quad 0.0625 = 1/16$$

$$0.2727 = 1/3.6, \quad 0.5556 = 5/9, \quad 0.3125 = 1/3.2$$

$$0.5455 = 1/1.8333, \quad 0.3333 = 3/9, \quad 0.6250 = 2/3.2$$

$$0.1818 = 1/5.5, \quad 0.1111 = 1/9, \quad 0.0625 = 1/16$$

$$0.2727 = 1/3.6, \quad 0.5556 = 5/9, \quad 0.3125 = 1/3.2$$

$$0.5455 = 1/1.8333, \quad 0.3333 = 3/9, \quad 0.6250 = 2/3.2$$

$$0.1818 = 1/5.5, \quad 0.1111 = 1/9, \quad 0.0625 = 1/16$$

- c. Rata-rata nilai pada setiap baris matriks normalisasi, ini disebut vektor preferensi

Pasar	A	B	C	Prioritas
A	0.5455	0.3333	0.6250	0.5013
B	0.1818	0.1111	0.0625	0.1185
C	0.2727	0.5556	0.3125	0.3803
Jumlah	1	1	1	1

$$0.5013 = (0.5455 + 0.3333 + 0.6250)/3$$

$$0.1185 = (0.1818 + 0.1111 + 0.0625)/3$$

$$0.3803 = (0.2727 + 0.5556 + 0.3125)/3$$

Berikut vektor preferensi untuk kriteria Penghasilan, Prasarana, dan Transportasi

Lokasi	Penghasilan	Prasarana	Transportasi
A	0.2819	0.1790	0.1560
B	0.0598	0.6851	0.6196
C	0.6583	0.1360	0.2243

- d. Gabungkan vektor preferensi setiap alternatif (hasil langkah 3c) ke dalam satu matriks preferensi

Lokasi	Pasar	Penghasilan	Prasarana	Transportasi
A	0.5013	0.2819	0.1790	0.1560
B	0.1185	0.0598	0.6851	0.6196
C	0.3803	0.6583	0.1360	0.2243

4. Membuat matriks perbandingan berpasangan setiap kriteria ($B_{n \times n}$)

Kriteria	Pasar	Penghasilan	Prasarana	Transportasi
Pasar	1	1/5	3	4
Penghasilan	5	1	9	7
Prasarana	1/3	1/9	1	2
Transportasi	1/4	1/7	1/2	1

5. Menghitung matriks normalisasi dengan membagi setiap nilai pada setiap kolom matriks dengan jumlah kolom yang sesuai

Kriteria	Pasar	Penghasilan	Prasarana	Transportasi
Pasar	0.1519	0.1376	0.2222	0.2857
Penghasilan	0.7595	0.6878	0.6667	0.5000
Prasarana	0.0506	0.0764	0.0741	0.1429
Transportasi	0.0380	0.0983	0.0370	0.0714
Jumlah	1	1	1	1

6. Membuat vektor preferensi kriteria dengan menghitung rata-rata baris matriks normalisasi ($V_{n \times 1}$)

$$0.1993 = (0.1519 + 0.1376 + 0.2222 + 0.2857)/4$$

$$\mathbf{0.6535} = (0.7595 + 0.6878 + 0.6667 + 0.5)/4$$

$$\mathbf{0.0860} = (0.0506 + 0.0764 + 0.0741 + 0.1429)/4$$

$$\mathbf{0.0612} = (0.0380 + 0.0983 + 0.0370 + 0.0714)/4$$

7. Menghitung nilai keseluruhan setiap alternatif dengan mengalikan vektor preferensi kriteria (hasil langkah 6) dengan matriks preferensi (hasil langkah 3d)

Lokasi	Pasar	Penghasilan	Prasarana	Transportasi
A	0.5013	0.2819	0.1790	0.1560
B	0.1185	0.0598	0.6851	0.6196
C	0.3803	0.6583	0.1360	0.2243

Pasar	0.1993
Penghasilan	0.6535
Prasarana	0.0860
Transportasi	0.0612

$$\begin{bmatrix} 0.5013 & 0.2819 & 0.1790 & 0.1560 \\ 0.1185 & 0.0598 & 0.6851 & 0.6196 \\ 0.3803 & 0.6583 & 0.1360 & 0.2243 \end{bmatrix} \times \begin{bmatrix} 0.1993 \\ 0.6535 \\ 0.0860 \\ 0.0612 \end{bmatrix} = \begin{bmatrix} 0.3091 \\ 0.1595 \\ 0.5314 \end{bmatrix}$$

A	0.3091
B	0.1595
C	0.5314

8. Alternatif diranking berdasarkan nilai yang diperoleh dari langkah 7

Ranking	Alternatif	Bobot
1	Charlotte	0.5314
2	Atlanta	0.3091
3	Birmingham	0.1595

9. Menghitung λ_{maks}
- Kalikan matriks perbandingan berpasangan setiap kriteria (hasil langkah 4) dengan vektor preferensi kriteria (hasil langkah 6) menggunakan persamaan (4)

$$\begin{aligned}
 M_{n \times 1} &= B_{(n \times n)} \cdot V_{(n \times 1)} \\
 &= \begin{bmatrix} 1 & 1/5 & 3 & 4 \\ 5 & 1 & 9 & 7 \\ 1/3 & 1/9 & 1 & 2 \\ 1/4 & 1/7 & 1/2 & 1 \end{bmatrix} \times \begin{bmatrix} 0.1993 \\ 0.6535 \\ 0.0860 \\ 0.0612 \end{bmatrix} \\
 &= \begin{bmatrix} 0.8327 \\ 2.8524 \\ 0.3474 \\ 0.2474 \end{bmatrix}
 \end{aligned}$$

- b. Bagilah hasil 9a dengan vektor preferensi kriteria (hasil langkah 6) yang bersesuaian menggunakan persamaan (5)

$$\begin{bmatrix} 0.8327 \\ 2.8524 \\ 0.3474 \\ 0.2474 \end{bmatrix} \div \begin{bmatrix} 0.1993 \\ 0.6535 \\ 0.0860 \\ 0.0612 \end{bmatrix} = \begin{bmatrix} 4.1773 \\ 4.3649 \\ 4.0397 \\ 4.0436 \end{bmatrix}$$

- c. Jumlahkan hasil 9b, kemudian dibagi dengan jumlah kriteria menggunakan persamaan (6)

$$\begin{aligned}
 \lambda_{maks} &= \frac{4.1773 + 4.3649 + 4.0397 + 4.0436}{4} \\
 &= 4.1564
 \end{aligned}$$

10. Menghitung *Consistency Index* (CI) menggunakan persamaan (7)

$$CI = \frac{4.1564 - 4}{4 - 1} = \frac{0.1564}{3} = 0.0521$$

11. Menghitung Rasio *Consistency Ratio* (CR) menggunakan persamaan (8)

$$CR = \frac{0.0521}{0.90} = 0.0579$$

Karena nilai $CR < 0.1$ maka penilaian dapat dikatakan konsisten.

2.3 Java

Java adalah bahasa pemrograman yang dikembangkan oleh Sun Microsystem pada tahun 1991. Java merupakan bahasa pemrograman yang berorientasi objek (OOP) dan dapat dijalankan pada berbagai platform sistem operasi serta bersifat open *source*. Platform Java terdiri dari kumpulan library, *Java Virtual Machine* (JVM), sebuah *compiler*, *debugger* dan kaskas lain yang dipaket dalam *Java Development Kit* (JDK). Aplikasi Java yang diciptakan dengan file teks berekstensi *.java akan dikompilasi oleh JVM dan menghasilkan file *bytecode* berekstensi *.class. Hal inilah yang membuat program Java dapat dijalankan pada berbagai platform sistem operasi, asalkan pada sistem operasi tersebut terdapat JVM.

Program Java dapat dibedakan menjadi dua jenis, yaitu applet dan aplikasi. Applet adalah program Java yang diletakkan pada *Web server* dan diakses melalui *Web browser*. Dalam hal ini *browser* yang digunakan adalah yang memiliki kemampuan Java misalnya Internet Explorer. Sedangkan aplikasi adalah program Java yang bersifat umum. Aplikasi dapat dijalankan secara langsung, tidak perlu perangkat lunak *browser* untuk menjalankannya. Setelah dikompilasi, program dapat dieksekusi secara langsung. [11]

Platform Java memiliki tiga edisi yang berbeda, yaitu :

1. J2SE (Java 2 Standard Edition)

J2SE digunakan untuk mengembangkan aplikasi desktop dan applet. Java applet adalah program Java yang diakses melalui *Web browser* sehingga untuk menjalankannya sebuah komputer harus memiliki program penjelajah web yang dapat menjalankan Java, seperti Microsoft Internet Explore, Netscape Navigator, Mozilla Firefox, dan Opera.

2. J2EE (Java 2 Enterprise Edition)

J2EE adalah pengembangan Java yang ditujukan untuk sebuah perusahaan dengan skala cukup besar (*enterprise*). J2EE mengandung lebih banyak API (*Application Programming Interface*) serta arsitektur yang lebih besar

daripada J2SE. J2EE terdiri dari tiga bagian utama yaitu Client, Java EE Server, dan database.

3. J2ME (Java 2 Micro Edition)

J2ME adalah pengembangan Java untuk perangkat mobile seperti handphone, pocket PC, dan PDA [12].

2.4 MySQL

MySQL merupakan salah satu *relational database management system* (RDBMS) yang bersifat *open source*. MySQL dapat dijalankan pada berbagai platform sistem operasi seperti Windows, Linux, Unix, Sun OS dll. MySQL juga mendukung SQL standar. SQL atau *Structure Query Language* adalah bahasa komputer yang digunakan untuk mengakses dan melakukan manipulasi data di dalam sebuah *database relational*. Dalam SQL terdapat beberapa perintah yang berguna untuk mengakses dan manajemen data yang terdapat dalam database. Jenis perintah SQL secara umum dibagi dalam dua sub perintah yaitu DDL (*Data Definition Language*) dan DML (*Data Manipulation Language*).

1. DDL (*Data Definition Language*)

DDL adalah perintah SQL yang digunakan untuk membangun kerangka sebuah database, dalam hal ini database dan table. Terdapat tiga perintah penting dalam DDL yaitu CREATE, ALTER, DROP.

a. CREATE

Perintah ini digunakan untuk membuat, termasuk di dalamnya membuat database baru, tabel baru, dan kolom baru.

b. ALTER

Perintah ALTER berfungsi untuk mengubah struktur tabel yang telah dibuat seperti mengubah nama tabel, menambah kolom, mengubah kolom, menghapus kolom, dan memberikan atribut pada kolom.

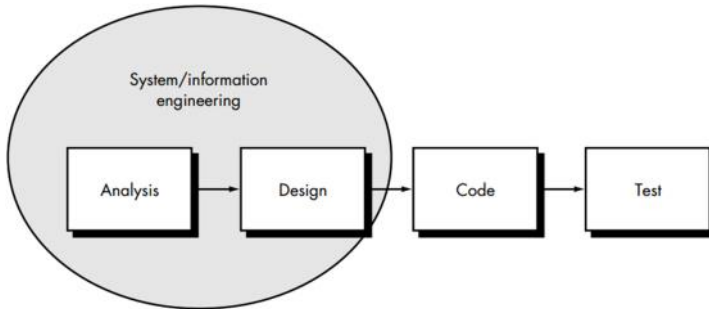
- c. DROP
Perintah DROP berfungsi untuk menghapus database atau tabel.
- 2. DML (*Data Manipulation Language*)
DML adalah perintah SQL yang digunakan untuk memanipulasi data dalam database. Terdapat empat perintah penting dalam DML yaitu INSERT, SELECT, UPDATE, dan DELETE.
 - a. INSERT
Perintah ini digunakan untuk memasukkan data baru ke dalam sebuah tabel.
 - b. SELECT
Perintah ini digunakan untuk mengambil dan menampilkan data dari tabel atau dari beberapa tabel dengan menggunakan relasi.
 - c. UPDATE
Perintah update digunakan untuk memperbarui data pada sebuah tabel.
 - d. DELETE
Perintah delete digunakan untuk menghapus data dari sebuah tabel. [13]

2.5 Rekayasa Perangkat Lunak

Perangkat lunak adalah suatu aplikasi program komputer yang didalamnya terdapat program itu sendiri, konfigurasi yang digunakan, dokumentasi penjelasan struktur sistem, dokumentasi penggunaan sistem, serta informasi pengembangan sistem. Terdapat beberapa model pengembangan perangkat lunak yaitu model sekuensial linier atau *Waterfall Model*, *Prototype Model*, RAD (*Rapid Application Development*) *Model*, dan *Evolutionary Software Process Model*. Model yang akan digunakan pada Tugas Akhir ini adalah model sekuensial linier.

Model sekuensial linier merupakan model pengembangan perangkat lunak dengan pendekatan sistematis dan sekuensial yang dimulai dari tingkat dan kemajuan sistem pada seluruh

tahapan analisis, desain, pengkodean, pengujian, dan pemeliharaan.



Gambar 2.4 Tahapan model sekuensial linear

Sumber : Roger S. Pressman, "Software Engineering".

Tahap-tahap pengembangan model sekuensial linear adalah sebagai berikut.

1. Rekayasa dan pemodelan sistem (*System engineering and modeling*)

Pada tahap ini dibangun keseluruhan elemen sistem dan dipilih bagian-bagian yang akan dijadikan bahan pengembangan perangkat lunak, dengan memperhatikan hubungannya dengan hardware, user, dan database.

2. Analisis kebutuhan perangkat lunak (*Software requirements analysis*)

Pada tahap ini dilakukan analisis kebutuhan perangkat lunak yang meliputi domain informasi serta fungsi yang dibutuhkan untuk performansi dan *interface*. Hasil analisis didokumentasikan dan ditunjukkan kepada pelanggan.

3. Desain (*Design*)

Pada tahap desain, kebutuhan perangkat lunak diwujudkan dalam sebuah gambaran sehingga dapat diperkirakan kualitasnya sebelum proses pengkodean (*coding*). Proses ini fokus pada struktur data, arsitektur perangkat lunak, representasi *interface*, dan detail prosedur.

4. Pengkodean (*Code*)
Pengkodean merupakan proses menerjemahkan desain ke dalam bentuk yang dapat dimengerti oleh mesin dengan menggunakan bahasa pemrograman.
5. Pengujian (*Testing*)
Setelah proses pengkodean selesai, dilanjutkan dengan proses pengujian perangkat lunak, baik pengujian logika internal maupun pengujian eksternal fungsional untuk memeriksa segala kemungkinan terjadinya kesalahan dan memeriksa apakah hasil dari pengembangan tersebut sesuai dengan hasil yang diinginkan.
6. Pemeliharaan (*Maintenance*)
Perangkat lunak akan mengalami perubahan setelah dikirim ke pelanggan. Perubahan terjadi apabila terdapat kesalahan pada perangkat lunak setelah dilakukan penyesuaian dengan lingkungan yang baru, misalnya hardware, periperal, sistem operasi baru, atau sebagai tuntutan atas perkembangan sistem komputer atau ketika pelanggan memerlukan peningkatan kinerja atau fungsional. [14]

BAB III METODOLOGI

Pada bab ini dijelaskan langkah-langkah yang digunakan dalam penyusunan Tugas Akhir. Disamping itu, dijelaskan pula prosedur dan proses pelaksanaan tiap-tiap langkah yang dilakukan dalam menyelesaikan Tugas Akhir.

3.1 Tahapan Penelitian

Langkah-langkah yang digunakan dalam menyelesaikan Tugas Akhir ini adalah sebagai berikut :

a. Studi Literatur

Pada tahap ini akan dipelajari teori tentang metode AHP dan perancangan perangkat lunak serta literatur-literatur ilmiah yang memiliki hubungan dengan topik penelitian yang sedang penulis lakukan.

b. Analisis Kebutuhan Perangkat Lunak

Perangkat lunak yang akan dibuat adalah sebuah perangkat lunak yang dapat menyelesaikan berbagai permasalahan pengambilan keputusan menggunakan metode AHP sehingga input kriteria dan alternatif bersifat dinamis baik jumlah maupun jenisnya sesuai kebutuhan permasalahan. Selanjutnya perangkat lunak direpresentasikan dalam suatu sistem dengan *interface* yang interaktif disertai sebuah database sebagai penyimpanan data.

c. Perancangan Perangkat Lunak

Kebutuhan perangkat lunak yang telah dianalisis direpresentasikan dalam sebuah desain atau gambaran yang mudah dipahami berupa *usecase diagram*, *activity diagram*, desain *interface*, dan arsitektur database.

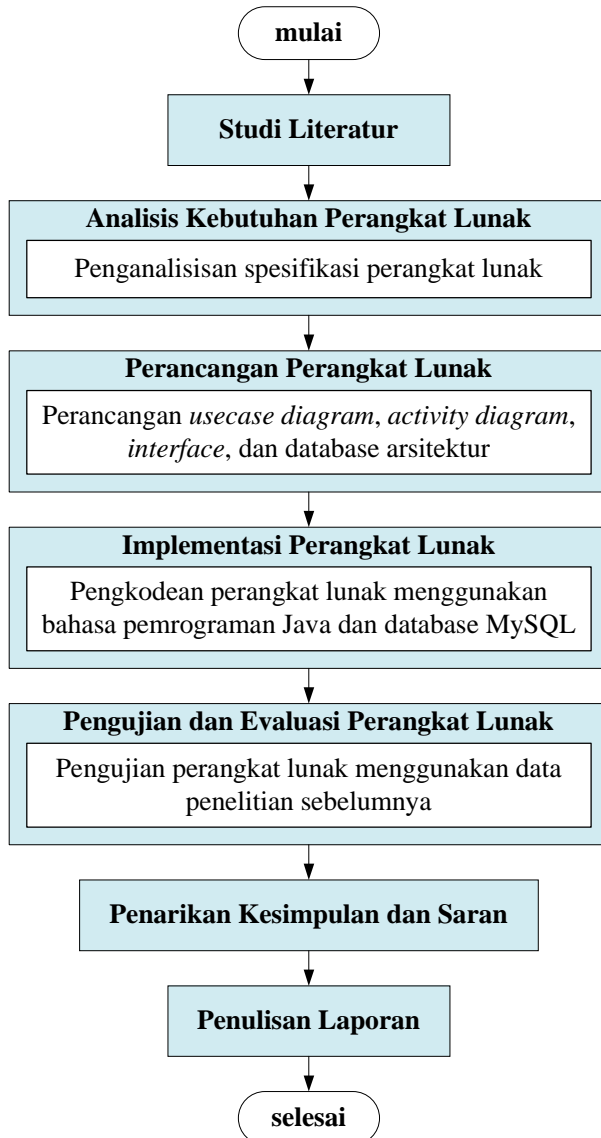
d. Implementasi Perangkat Lunak

Desain perangkat lunak yang telah dirancang diimplementasikan menggunakan bahasa pemrograman Java dan database MySQL.

- e. **Pengujian dan Evaluasi Perangkat Lunak**
Pengujian perangkat lunak dilakukan dengan menggunakan data yang telah digunakan pada penelitian sebelumnya. Data yang dapat diinputkan pada perangkat lunak adalah data sekunder yang berupa nilai perbandingan berpasangan. Tahap ini dilakukan untuk memvalidasi program sehingga akan diperbaiki jika terdapat kesalahan.
- f. **Penarikan Kesimpulan dan Saran**
Setelah dilakukan pengujian kemudian dilakukan penarikan kesimpulan dari hasil yang diperoleh. Selain itu juga akan diberikan saran sebagai masukan untuk penelitian selanjutnya.
- g. **Penulisan Laporan**
Pada tahap ini akan dilakukan penulisan laporan dan pembuatan user manual dari perangkat lunak yang dihasilkan.

3. 2 Diagram Alir Metode Penelitian

Berdasarkan uraian tersebut diatas, metode penelitian Tugas Akhir ini dapat dinyatakan dalam diagram alir sebagai berikut.



Gambar 3.1 Metode Penelitian

BAB IV

ANALISIS DAN PERANCANGAN PERANGKAT LUNAK

Pada bab ini akan dijelaskan secara detail tentang spesifikasi kebutuhan perangkat lunak yang meliputi domain informasi serta fungsi yang dibutuhkan untuk performansi dan *interface*. Hasil analisis kemudian didokumentasikan dalam bentuk desain yang berupa *usecase diagram*, *activity diagram*, desain *interface*, dan arsitektur database sehingga mudah dipahami.

4.1 Gambaran Umum Perangkat Lunak

Perangkat lunak SPK – AHP akan berfungsi sebagai aplikasi yang memudahkan *decision maker* dalam pengambilan keputusan dengan memasukkan atribut kasus yang telah ditentukan. Perangkat lunak ini juga menyediakan fitur penyimpanan data hasil perhitungan. Metode pengambilan keputusan yang digunakan adalah metode AHP.

4.2 Analisis Kebutuhan Perangkat Lunak

Berdasarkan gambaran umum tersebut, dianalisis kebutuhan dari perangkat lunak yang akan dibuat. Perangkat lunak tersebut harus dapat melakukan proses perankingan alternatif, menyimpan data kasus yang pernah diinputkan, mengubah data kasus yang sudah ada, dan menghapus data kasus dari daftar kasus. Perangkat lunak yang dibuat juga harus memiliki tampilan yang menarik, mudah dipelajari dan dioperasikan serta dapat memberikan respon dengan cepat. Untuk menjaga keamanan data perangkat lunak yang dibuat harus memberikan informasi mengenai user yang membuat perhitungan dan user yang mengubah data perhitungan. Analisis tersebut dapat dirangkum menjadi kebutuhan fungsional dan kebutuhan non fungsioanl yang ditunjukkan dalam Tabel 4.1.

Tabel 4.1 Kebutuhan Perangkat Lunak

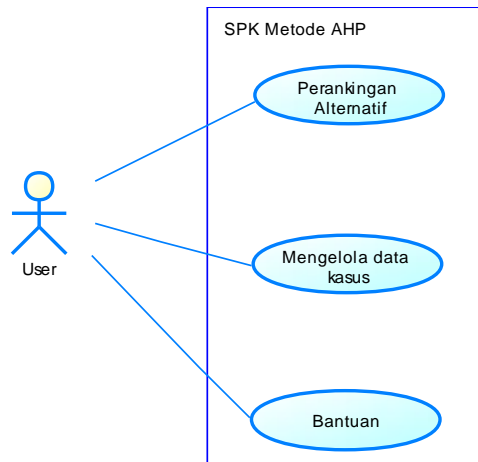
Kebutuhan	Keterangan
Kebutuhan Fungsional	<ol style="list-style-type: none"> 1. Sistem mampu melakukan proses perankingan alternatif 2. Sistem mampu mengelola data kasus yang meliputi : lihat dan cetak detail kasus, mengubah data kasus, dan menghapus data kasus 3. Sistem mampu melakukan proses pencarian bantuan
Kebutuhan Non Fungsional	<ol style="list-style-type: none"> 1. Ketersediaan perangkat lunak 2. Desain perangkat lunak harus <i>user friendly</i> 3. Media penyimpanan 4. Waktu aplikasi untuk merespon <i>request</i> dari user 5. Keamanan data 6. Keamanan perangkat lunak

4.3 Perancangan Perangkat Lunak

Berdasarkan kebutuhan perangkat lunak tersebut, dibuat rancangan perangkat lunak yang meliputi *usecase diagram*, *activity diagram*, desain *interface*, dan arsitektur database.

4.2.1 Usecase Diagram

Usecase diagram menggambarkan interaksi apa saja yang dapat dilakukan di dalam sistem dan siapa saja aktor yang akan menjalankan interaksi tersebut. Aktor dari perangkat lunak ini adalah user. User dapat melakukan perankingan alternatif, mengelola data, dan mencari bantuan. *Usecase diagram* perangkat lunak SPK – AHP dapat dilihat pada Gambar 4.1.



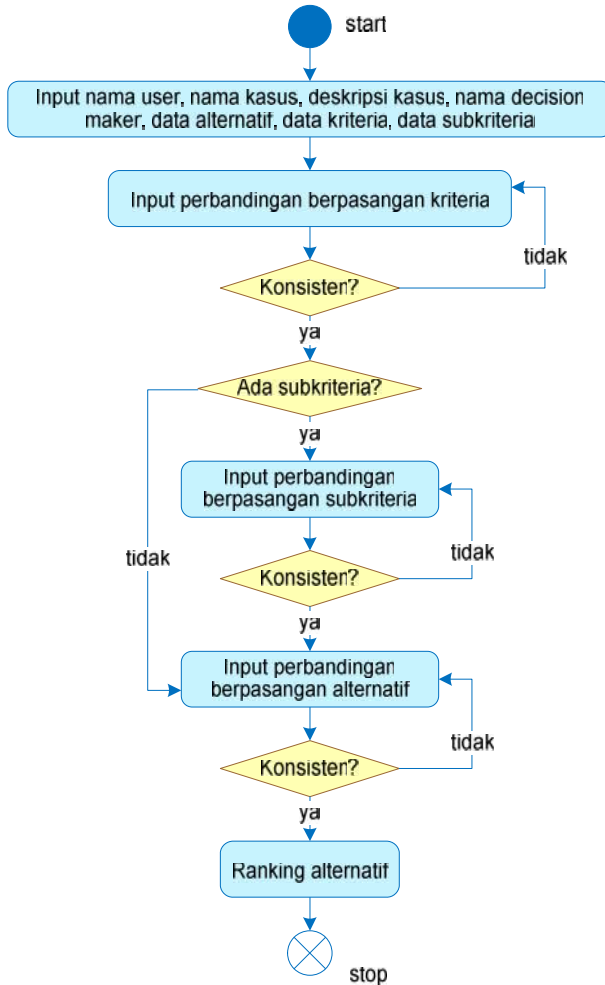
Gambar 4.1 *Usecase Diagram*

4.2. 2 *Activity Diagram*

Berdasarkan *usecase diagram* yang ditunjukkan pada Gambar 4.1, dibuat *activity diagram* untuk menggambarkan alur proses dari setiap interaksi.

4.2.2. 1 *Activity Diagram* Perankingan Alternatif

Gambar 4.2 menunjukkan alur proses perankingan alternatif yang dimulai dari user input data kasus meliputi nama user, judul kasus, deskripsi kasus, nama DM, data alternatif, data kriteria, dan data subkriteria jika ada. Kemudian dilanjutkan dengan input nilai perbandingan berpasangan tiap level untuk menentukan prioritas yang selanjutnya digunakan untuk perankingan alternatif. Sebelum nilai perbandingan berpasangan ditambahkan, data tersebut akan dicek oleh sistem apakah nilai yang dimasukkan konsisten atau tidak. Jika tidak konsisten, maka data penilaian tidak dapat ditambahkan dan user diminta untuk input ulang data penilaian yang sudah diperbaiki.

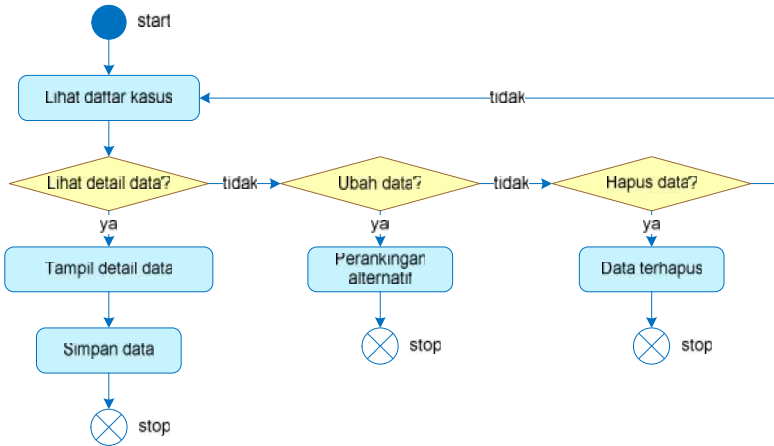


Gambar 4.2 Activity Diagram Perankingan Alternatif

4.2.2.2 Activity Diagram Kelola Data

Gambar 4.3 menunjukkan alur proses kelola data kasus yang meliputi aktivitas melihat detail kasus, mengubah data kasus, dan menghapus data kasus. Mengubah data kasus berarti mengganti data yang sudah ada mulai dari data atribut kasus sampai dengan

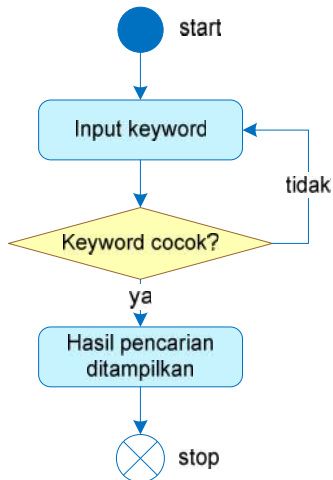
penilaian perbandingan berpasangan. Sehingga proses yang terjadi sama dengan proses perankingan alternatif.



Gambar 4.3 Activity Diagram Kelola Data

4.2.2.3 Activity Diagram Bantuan

Gambar 4.4 menunjukkan alur proses pencarian bantuan terkait tatacara penggunaan perangkat lunak.



Gambar 4.4 Activity Diagram Bantuan

4.2.3 Pseudocode

Proses perankingan alternatif metode AHP dapat dijelaskan dengan algoritma berikut :

```

procedure Ranking;
begin
  initialize jumlah_level L;
  input pb;
  foreach level  $i \in L$ , do  $i := \text{prioritas\_global}_i$ 
  repeat
    for  $i=0$  to  $\text{level}_i.\text{size}$  do
      for  $j=0$  to  $\text{level}_i.\text{size}$  do
         $\text{jum}_i += \text{pb}_{j,i}$  (jumlah dalam baris);
      endfor;
    endfor;
    for  $i=0$  to  $\text{level}_i.\text{size}$  do
      for  $j=0$  to  $\text{level}_i.\text{size}$  do
         $\text{norm}_{i,j} = \text{pb}_{i,j} / \text{jum}_i$ ;
      endfor;
    endfor;
    for  $i=0$  to  $\text{level}_i.\text{size}$  do
      for  $j=0$  to  $\text{level}_i.\text{size}$  do
         $\text{prioritas}_i += \text{norm}_{i,j}$  (jumlah dalam
kolom) /  $\text{level}_i.\text{size}$ ;
      endfor;
       $\text{prioritas\_global}_i = \text{prioritas}_i$ .
     $\text{prioritas\_global}_i(\text{level } L-1)$ ;
    add  $\text{prioritas\_global}_i$  to L;
  endfor;
  endfor;
  rank = select  $\text{prioritas\_global}_i(L)$ ;
until L.size;
end;

```

4.2.4 Desain Interface

4.2.4.1 Desain *Homepage* Perangkat Lunak

Gambar 4.5 merupakan desain tampilan awal perangkat lunak SPK – AHP. Terdapat tiga menu utama yaitu New, Data, dan Help.

SPK – AHP

Software SPK untuk menyelesaikan permasalahan pengambilan keputusan
menggunakan metode AHP

NEW

DATA

HELP

Gambar 4.5 Desain *Homepage*

4.2.4. 2 Desain Halaman Input Kasus Baru

Gambar 4.6 merupakan desain halaman input kasus baru yang terdiri dari input data kasus, data DM, data alternatif, data kriteria, dan data subkriteria.

Masukkan Kasus Baru

Nama user

Nama Kasus

Deskripsi

Nama DM* (+)

*Decision Maker

No	Nama DM

Alternatif (+)

No	Nama Alternatif

Kriteria (+)

No	Nama Kriteria

Sub-kriteria Kriteria (+)

No	Nama Kriteria	Nama Sub-kriteria

Gambar 4.6 Desain Input Kasus Baru

4.2.4. 3 Desain Halaman Input Perbandingan Berpasangan

Gambar 4.7 merupakan desain halaman input perbandingan berpasangan yang terdiri dari perbandingan berpasangan kriteria, perbandingan berpasangan subkriteria, dan perbandingan berpasangan alternatif.

Perbandingan Berpasangan

Skala Kepentingan

Daftar Kriteria

Nc	Nama Kriteria

Pilih DM

	K1	K2	K3	K4	K5	K6	...
K1							
K2							
K3							
...							

Gambar 4.7 Desain Input Perbandingan Berpasangan

4.2.4.4 Desain Halaman Hasil Ranking

Gambar 4.8 merupakan desain halaman hasil ranking alternatif keputusan yang dilengkapi dengan perhitungan berdasarkan metode AHP.

Hasil Ranking

User

Judul Kasus

Tanggal

Deskripsi

Data DM

Data Alternatif

Data Kriteria

Data Subkriteria

Nc	Nama Alternatif	Bobot

	A1	A2	A3	A4	A5	A6	...
A1							
A2							
A3							
...							

	A1	A2	A3	A4	A5	A6	...
A1							
A2							
A3							
...							

	A1	A2	A3	A4	A5	A6	...
A1							
A2							
A3							
...							

Gambar 4.8 Desain Halaman Hasil Ranking

4.2.4. 5 Desain Halaman Daftar Kasus

Gambar 4.9 merupakan desain halaman daftar kasus. User dapat melihat detail kasus, mengubah data kasus, dan menghapus data kasus.

Daftar Kasus				
No	Tujuan	Action		
1		Lihat	Ubah	Hapus
2		Lihat	Ubah	Hapus
3		Lihat	Ubah	Hapus

Gambar 4.9 Desain Halaman Daftar Kasus

4.2.4. 6 Desain Halaman Detail Kasus

Gambar 4.10 merupakan desain halaman detail kasus yang terdiri dari data kasus dan hasil ranking alternatif.

Detail Kasus																								
Judul Kasus		Data DM																						
Nama User		<input type="text"/>																						
Tanggal Create		Data Alternatif																						
Diupdate Oleh		<input type="text"/>																						
Tanggal Update		Data Kriteria																						
Deskripsi		<input type="text"/>																						
Ranking Alternatif		Data Subkriteria																						
<input type="text"/>		<input type="text"/>																						
<table border="1"> <thead> <tr> <th>No</th> <th>Nama Alternatif</th> <th>Bobot</th> </tr> </thead> <tbody> <tr><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td></tr> </tbody> </table>	No	Nama Alternatif	Bobot																				<input type="text"/>	
No	Nama Alternatif	Bobot																						

Gambar 4.10 Desain Halaman Detail Kasus

4.2.4. 6 Desain Halaman Help

Gambar 4.11 merupakan desain halaman pencarian bantuan terkait tatacara penggunaan perangkat lunak.

Content	Find
Tentang Aplikasi	SPK – AHP Ver: 1.0 ©2016
Tambah Kasus	
Ubah Kasus	
Hapus Kasus	SPK – AHP dapat digunakan untuk menyelesaikan permasalahan pengambilan keputusan multi kriteria menggunakan metode AHP (Analytic Hierarchy Process). SPK – AHP juga menyediakan fasilitas simpan data dan cetak hasil perhitungan.
Cetak Data Kasus	
	Klik pilihan pada tab Contents untuk melihat tatacara penggunaan aplikasi

Gambar 4.11 Desain Halaman Help

4.2. 5 Database Arsitektur

Sebelum rancangan perangkat lunak diimplementasikan, terlebih dahulu dibuat database arsitektur yang merupakan skema sistematis mengenai keseluruhan entitas yang terdapat dalam sistem database. Database arsitektur menggambarkan relasi antar entitas yang ada.

Perangkat lunak SPK – AHP terdiri dari 10 entitas yang terdiri dari entitas kasus, decision_maker, alternatif, kriteria, subkriteria, pbk, pbs, pbsa, pbka, dan help. Gambaran rinci database arsitektur perangkat lunak SPK – AHP dapat dilihat pada Gambar 4.12.

BAB V

IMPLEMENTASI DAN PENGUJIAN PERANGKAT LUNAK

Pada bab ini dijelaskan tentang implementasi rancangan perangkat lunak dalam bahasa pemrograman Java dan hasil uji coba perangkat lunak menggunakan data yang telah digunakan pada penelitian sebelumnya.

4.1 Implementasi Perangkat Lunak

Database arsitektur dan desain *interface* yang telah dibuat selanjutnya diimplementasikan dalam bahasa pemrograman Java dengan menggunakan *software* Netbeans IDE 7.3.1. Web server yang digunakan adalah XAMPP 3.2.1 dengan database MySQL. Gambar 5.1 merupakan *homepage* perangkat lunak SPK – AHP.



Gambar 5.1 Menu Utama SPK – AHP

Agar dapat mengelola data kasus, perangkat lunak SPK – AHP harus dikoneksikan dengan database MySQL. Berikut kode untuk koneksi perangkat lunak dengan database :


```

import java.sql.*;
import javax.swing.*;

public class koneksi {
    Connection conn = null;
    public static Connection ConnectDb(){
        try{
Class.forName("com.mysql.jdbc.Driver");
            Connection conn =
DriverManager.getConnection("jdbc:mysql://localhost
:3306/db_keputusan", "root", "");
            return conn;
        }catch(Exception e){
            JOptionPane.showMessageDialog(null, e);
            return null;
        }
    }
}

```

Hasil implementasi rancangan perangkat lunak dalam bahasa pemrograman Java dapat dilihat pada gambar berikut.

4.1.1 Form Input Kasus Baru

Gambar 5.2 merupakan form untuk menambahkan kasus baru. User diminta untuk memasukkan data kasus. Jika data masih kosong maka user tidak dapat lanjut ke tahap selanjutnya.

Gambar 5.2 Form Masukkan Kasus Baru

4.1.2 Form Input Perbandingan Berpasangan

Gambar 5.3 merupakan form untuk memasukkan penilaian subjektif dari *decision maker*. Jika penilaian tidak konsisten maka user tidak dapat lanjut ke tahap selanjutnya.

Perbandingan Berpasangan Kriteria

Skala Kepentingan

Intensitas	Keterangan
1	Sama penting
3	Sedikit lebih penting
5	Lebih penting
7	Mutlak lebih penting
9	Mutlak penting
2, 4, 6, 8	Nilai antara
Kebalikan	$a_{ij} = 1/a_{ji}$

Daftar Kriteria

Kode	Nama Kriteria
K1	Fungsional
K2	Kesempangan
K3	Manajemen Proyek
K4	Flexibilitas
K5	Kecandaban
K6	Biaya Perolehan
K7	Reputasi

Nama DM: DM 1

	K1	K2	K3	K4	K5	K6	K7
K1							
K2							
K3							
K4							
K5							
K6							
K7							

Buttons: **Batal** **Tambah** **Next**

Gambar 5.3 Form Input Perbandingan Berpasangan

4.1.3 Form Hasil Ranking

Gambar 5.4 merupakan form hasil ranking alternatif keputusan. Data kasus, data perbandingan berpasangan dan perhitungan AHP ditampilkan pada form ini termasuk ranking alternatif dan bobotnya.

Hasil Ranking

JUDUL Kasus: Pemilihan Sistem Informasi Akuntansi

Nama User: Jafrianni

Tanggal: 2019-09-22

Deskripsi: Penetapan AHP Sistem Pengambilan Keputusan Pemilihan Sistem Informasi Akuntansi pada PT. ...

Data DM: DM 2

Data Alternatif: A1 Accurate, A2 Accoy, A3 Seventh Soft

Data Kriteria: K1 Fungsional, K2 Kesempangan, K3 Manajemen Proyek, K4 Flexibilitas, K5 Kecandaban

Ranking Alternatif

Ranking	Nama Alternatif	Bobot
1	Accurate	0.5281
2	Accoy	0.2737
3	Seventh Soft	0.2012

	K1	K2	K3	K4	K5	K6	K7	K1	K2	K3	K4	K5	K6	K7	Phi	Psi
K1	1.0	6.4	3.4	5.0	1.8	8.4	7.0	0.3	0.4	0.3	0.2	0.2	0.2	0.2	0.2	0.3
K2	0.3	1.0	0.5	0.5	5.9	1.0	0.0	0.0	0.0	0.0	0.0	0.1	0.1	0.0	0.0	0.0
K3	0.3	1.7	1.0	0.7	0.3	8.4	3.4	0.0	0.1	0.0	0.0	0.0	0.2	0.1	0.1	0.1
K4	0.2	1.7	1.4	1.0	0.2	4.0	3.0	0.0	0.1	0.1	0.0	0.0	0.1	0.0	0.0	0.0
K5	1.8	1.8	5.0	4.8	1.8	5.1	6.4	0.3	0.1	0.3	0.3	0.1	0.3	0.1	0.3	0.2
K6	0.3	0.1	0.1	0.2	0.1	1.0	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
K7	0.1	1.8	0.4	0.5	0.1	3.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Jumlah	2.7	14	13	12	3.2	36	20	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0

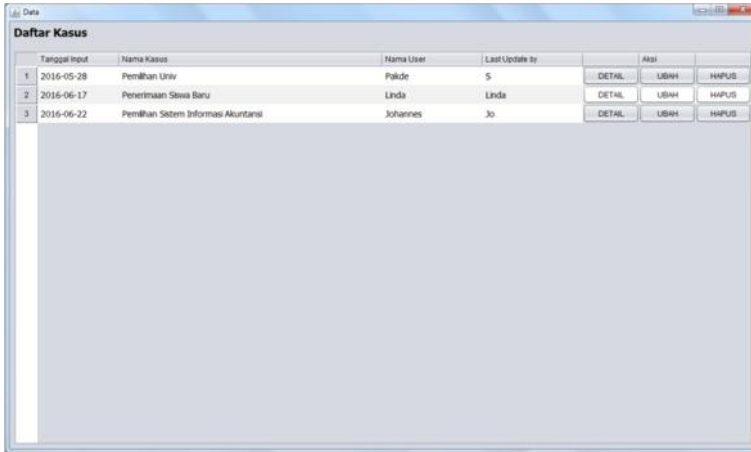
Pilih Kriteria: Fungsional

	A1	A2	A3	A1	A2	A3	Prioritas
A1	1.0000	1.2000	1.0000	0.2529	0.3960	0.3226	0.3538
A2	0.8333	1.0000	1.1000	0.2041	0.3216	0.3548	0.3235
A3	1.0000	0.9091	1.0000	0.2529	0.2624	0.3226	0.3226
Jumlah	2.8333	3.1091	3.1000	1.0000	1.0000	1.0000	

Gambar 5.4 Form Hasil Ranking

4.1. 4 Form Daftar Kasus

Gambar 5.5 merupakan form untuk mengelola data kasus. User dapat melihat detail kasus, ubah data kasus, dan hapus data kasus.



	Tanggal Input	Nama Kasus	Nama User	Last Update by	Aksi
1	2016-05-28	Pemilihan Utuh	Pakde	S	DETAIL UBAH HAPUS
2	2016-06-17	Penerimaan Sewa Baru	Linda	Linda	DETAIL UBAH HAPUS
3	2016-06-22	Pemilihan Sistem Informasi Akuntansi	Johannes	Jo	DETAIL UBAH HAPUS

Gambar 5.5 Form Daftar Kasus

4.1. 5 Form Detail Kasus

Dengan klik tombol **DETAIL** pada form daftar kasus (Gambar 5.5), maka akan muncul form detail kasus. Data kasus dan ranking alternatif ditampilkan dalam form ini. Dengan klik tombol **Save PDF**, maka akan muncul kotak dialog penyimpanan file (Gambar 5.7) dan data perhitungan akan disimpan dalam bentuk file *.pdf. Form detail kasus perangkat lunak dapat dilihat pada Gambar 5.6.

Detail Kasus Save PDF

Data Kasus

Judul Kasus: Pemilihan Sistem Informasi Akuntansi
 Nama User: Johannes
 Tanggal Create: 2016-06-22
 Di Update Oleh: Jo
 Tanggal Update: 2016-06-23
 Deskripsi: Penerapan AHP dalam Pengambilan Keputusan Pemilihan Sistem Informasi Akuntansi pada PT KDM di Surabaya

Melalui proses AHP dengan bobot yang diberikan oleh decision maker, diperoleh ranking alternatif dengan bobot sebagai berikut :

Ranking	Nama Alternatif	Bobot
1	Accurate	0.52310
2	Acoys	0.27567
3	Seventh Soft	0.20123

Data DM

DM 1
DM 2

Data Alternatif

Accurate
Acoys
Seventh Soft

Data Kriteria

Fungsional
Kesenjangan
Manajemen Proyek
Fleksibilitas
Keandalan
Biaya Perolehan
Reputasi

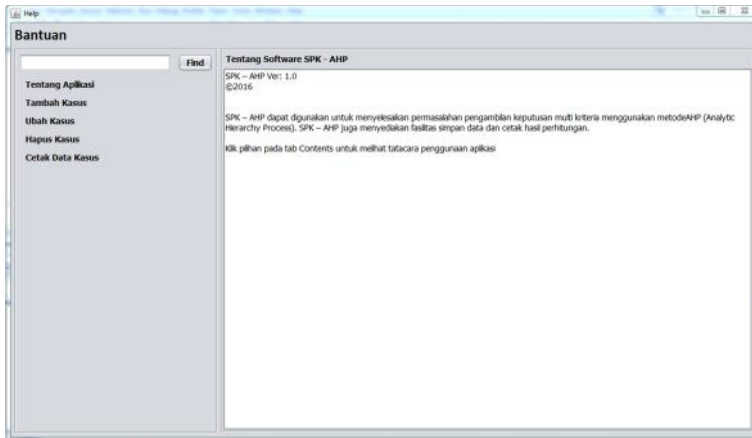
Gambar 5.6 Form Detail Kasus



Gambar 5.7 Kotak Dialog Direktori Penyimpanan File

4.1.6 Form Help

Gambar 5.8 merupakan form help yang berisi informasi terkait tatacara penggunaan perangkat lunak. User dapat melihat informasi lebih lengkap dengan klik pilihan pada tab sebelah kiri. Terdapat lima pilihan yaitu pilihan tentang aplikasi, tambah kasus, ubah kasus, hapus kasus, dan cetak data kasus.



Gambar 5.8 Form Help

4.2 Pengujian Perangkat Lunak

Pengujian perangkat lunak dilakukan dengan menggunakan data penelitian sebelumnya dengan judul “Analisa Keputusan Proyek Investasi Pemasangan Booster Kompresor sebagai Upaya Mempertahankan Produksi Gas Bumi Lapangan Offshore L-Parigi di PT. PEP dengan Metode AHP dan TOPSIS” oleh Risang Raheditya dan Suparno [15].

Hasil perancangan alternatif oleh sistem berdasarkan prosedur metode AHP dapat dijelaskan sebagai berikut :

1. Mendefinisikan masalah dan menentukan solusi yang diinginkan. Goal dari penelitian ini adalah Seleksi Proyek Investasi dengan dua kriteria yaitu kriteria finansial dan kriteria teknis. Kriteria finansial memiliki tiga subkriteria sedangkan kriteria teknis memiliki delapan subkriteria.

Terdapat 10 alternatif keputusan dengan hasil ranking seperti yang ditunjukkan pada Tabel 5.1.

Tabel 5.1 Hasil Perankingan

Ranking	Alternatif	Bobot
1	Alt. Proyek Investasi 8	0.150
2	Alt. Proyek Investasi 6	0.147
3	Alt. Proyek Investasi 2	0.132
4	Alt. Proyek Investasi 4	0.128
5	Alt. Proyek Investasi 10	0.098
6	Alt. Proyek Investasi 1	0.095
7	Alt. Proyek Investasi 7	0.077
8	Alt. Proyek Investasi 3	0.063
9	Alt. Proyek Investasi 9	0.057
10	Alt. Proyek Investasi 5	0.053

Gambar 5.9 menunjukkan hasil input data kasus pada perangkat lunak dapat.

The screenshot shows a software interface for inputting case data. It is organized into several sections:

- Nama User:** Input field containing "Raj".
- Nama Kasus:** Input field containing "Seleksi Proyek Investasi".
- Deskripsi:** Text area containing "Analisis Keputusan Proyek Investasi Pemasangan Booster Kompresor sebagai Upaya Mempertahankan Produksi Gas Bumi Lapangan Citanore L. Pang di PT. PEP dengan Metode WPM dan TOPSIS".
- Nama DM* (Decision Maker):** A list box containing "1. Rianing Rahaditya".
- Alternatif:** A list box containing "1. Alternatif Proyek Investasi 1", "2. Alternatif Proyek Investasi 2", "3. Alternatif Proyek Investasi 3", "4. Alternatif Proyek Investasi 4", "5. Alternatif Proyek Investasi 5", and "6. Alternatif Proyek Investasi 6".
- Kriteria:** A list box containing "1. Kriteria Finansial" and "2. Kriteria Teknis".
- Subkriteria:** A list box containing "1. Kriteria Finansial TGI", "2. Kriteria Finansial NPV", "3. Kriteria Finansial PGT", "4. Kriteria Teknis SPT", "5. Kriteria Teknis CNV", and "6. Kriteria Teknis FLX".

Gambar 5.9 Input Data Kasus

2. Buat matriks perbandingan berpasangan setiap alternatif untuk setiap subkriteria menggunakan persamaan (1).

Gambar berikut menunjukkan hasil input perbandingan berpasangan alternatif untuk setiap subkriteria.

	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10
A1	1.00000	0.50000	0.33333	0.25000	0.50000	0.25000	0.33333	0.25000	0.50000	0.25000
A2	2.00000	1.00000	0.33333	0.25000	0.33333	0.25000	0.50000	0.25000	0.50000	0.25000
A3	3.00000	3.00000	1.00000	0.50000	2.00000	0.33333	0.50000	0.25000	2.00000	2.00000
A4	4.00000	4.00000	2.00000	1.00000	1.00000	1.00000	2.00000	1.00000	2.00000	1.00000
A5	2.00000	3.00000	0.50000	1.00000	1.00000	0.33333	2.00000	0.33333	2.00000	0.33333
A6	4.00000	4.00000	3.00000	1.00000	3.00000	1.00000	3.00000	1.00000	3.00000	2.00000
A7	3.00000	2.00000	2.00000	0.50000	0.50000	0.33333	1.00000	0.33333	2.00000	0.33333
A8	4.00000	4.00000	4.00000	1.00000	3.00000	1.00000	3.00000	1.00000	3.00000	1.00000
A9	2.00000	2.00000	0.50000	0.50000	0.50000	0.33333	0.50000	0.33333	1.00000	0.33333
A10	4.00000	4.00000	0.50000	1.00000	3.00000	0.50000	3.00000	1.00000	3.00000	1.00000

Gambar 5.10 Perbandingan Berpasangan Alternatif Subkriteria TGI

	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10
A1	1	0.20000	0.25000	0.20000	0.25000	0.20000	0.25000	0.20000	0.25000	0.20000
A2	5.00000	1	3.00000	0.20000	3.00000	0.33333	3.00000	0.33333	2.00000	0.50000
A3	4.00000	0.33333	1	0.50000	1.00000	0.50000	2.00000	0.33333	2.00000	0.50000
A4	5.00000	5.00000	2.00000	1	4.00000	0.50000	2.00000	1.00000	2.00000	2.00000
A5	4.00000	0.33333	1.00000	0.25000	1	0.33333	1.00000	0.25000	1.00000	0.33333
A6	5.00000	3.00000	2.00000	2.00000	3.00000	1	3.00000	2.00000	5.00000	2.00000
A7	4.00000	0.33333	0.50000	0.50000	1.00000	0.33333	1	0.20000	2.00000	0.33333
A8	5.00000	3.00000	3.00000	1.00000	4.00000	0.50000	5.00000	1	5.00000	1.00000
A9	4.00000	0.50000	0.50000	0.50000	1.00000	0.20000	0.50000	0.20000	1	0.20000
A10	5.00000	2.00000	2.00000	0.50000	3.00000	0.50000	3.00000	1.00000	5.00000	1

Gambar 5.11 Perbandingan Berpasangan Alternatif Subkriteria NPV

	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10
A1	1	2.00000	5.00000	5.00000	5.00000	5.00000	4.00000	4.00000	3.00000	3.00000
A2	0.50000	1	5.00000	5.00000	5.00000	5.00000	4.00000	4.00000	3.00000	3.00000
A3	0.20000	0.20000	1	2.00000	0.50000	1.00000	0.25000	0.25000	1.00000	1.00000
A4	0.20000	0.20000	0.50000	1	1.00000	1.00000	0.25000	0.25000	1.00000	1.00000
A5	0.20000	0.20000	2.00000	1.00000	1	2.00000	0.33333	0.33333	0.50000	0.50000
A6	0.20000	0.20000	1.00000	1.00000	0.50000	1	0.25000	0.33333	0.50000	0.50000
A7	0.25000	0.25000	4.00000	4.00000	3.00000	4.00000	1	2.00000	0.33333	0.50000
A8	0.25000	0.25000	4.00000	4.00000	3.00000	3.00000	0.50000	1	0.50000	0.50000
A9	0.33333	0.33333	1.00000	1.00000	2.00000	2.00000	3.00000	2.00000	1	2.00000
A10	0.33333	0.33333	1.00000	1.00000	2.00000	2.00000	2.00000	2.00000	0.50000	1

Gambar 5.12 Perbandingan Berpasangan Alternatif Subkriteria SDT

3. Proses sintesis menghasilkan matriks normalisasi dan vektor preferensi alternatif.

Matriks normalisasi alternatif dan matriks preferensi alternatif dapat dilihat pada gambar berikut.

	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10
A1	0.034...	0.018...	0.023...	0.035...	0.033...	0.046...	0.021...	0.043...	0.026...	0.02...
A2	0.068...	0.036...	0.023...	0.035...	0.022...	0.046...	0.031...	0.043...	0.026...	0.02...
A3	0.103...	0.109...	0.070...	0.071...	0.134...	0.062...	0.031...	0.043...	0.105...	0.23...
A4	0.137...	0.145...	0.141...	0.142...	0.067...	0.187...	0.126...	0.173...	0.105...	0.11...
A5	0.068...	0.109...	0.035...	0.142...	0.067...	0.062...	0.126...	0.057...	0.105...	0.03...
A6	0.137...	0.145...	0.211...	0.142...	0.202...	0.187...	0.189...	0.173...	0.157...	0.23...
A7	0.103...	0.072...	0.141...	0.071...	0.033...	0.062...	0.063...	0.057...	0.105...	0.03...
A8	0.137...	0.145...	0.282...	0.142...	0.202...	0.187...	0.189...	0.173...	0.157...	0.11...

Gambar 5.13 Matriks Normalisasi Alternatif Subkriteria TGI

	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10
A1	0.023...	0.012...	0.016...	0.030...	0.011...	0.045...	0.012...	0.030...	0.009...	0.024...
A2	0.119...	0.063...	0.196...	0.030...	0.141...	0.075...	0.144...	0.051...	0.079...	0.061...
A3	0.095...	0.021...	0.065...	0.075...	0.047...	0.113...	0.096...	0.051...	0.079...	0.061...
A4	0.119...	0.318...	0.131...	0.150...	0.188...	0.113...	0.096...	0.153...	0.079...	0.247...
A5	0.095...	0.021...	0.065...	0.037...	0.047...	0.075...	0.048...	0.038...	0.039...	0.041...
A6	0.119...	0.191...	0.131...	0.300...	0.141...	0.227...	0.144...	0.306...	0.198...	0.247...
A7	0.095...	0.021...	0.032...	0.075...	0.047...	0.075...	0.048...	0.030...	0.079...	0.041...
A8	0.119...	0.191...	0.196...	0.150...	0.188...	0.113...	0.240...	0.153...	0.198...	0.123...

Gambar 5.14 Matriks Normalisasi Alternatif Subkriteria NPV

	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10
A1	0.288...	0.402...	0.204...	0.200...	0.217...	0.192...	0.256...	0.247...	0.264...	0.230...
A2	0.144...	0.201...	0.204...	0.200...	0.217...	0.192...	0.256...	0.247...	0.264...	0.230...
A3	0.057...	0.040...	0.040...	0.080...	0.021...	0.038...	0.016...	0.015...	0.088...	0.076...
A4	0.057...	0.040...	0.020...	0.040...	0.043...	0.038...	0.016...	0.015...	0.088...	0.076...
A5	0.057...	0.040...	0.081...	0.040...	0.043...	0.076...	0.021...	0.020...	0.044...	0.038...
A6	0.057...	0.040...	0.040...	0.040...	0.021...	0.038...	0.016...	0.020...	0.044...	0.038...
A7	0.072...	0.050...	0.163...	0.160...	0.130...	0.153...	0.064...	0.123...	0.029...	0.038...
A8	0.072...	0.050...	0.163...	0.160...	0.130...	0.115...	0.032...	0.061...	0.044...	0.038...

Gambar 5.15 Matriks Normalisasi Alternatif Subkriteria SDT

	Prioritas	Prioritas	Prioritas	Prioritas	Prioritas	Prioritas	Prioritas	Prioritas
A1	0.03127	0.02177	0.16100	0.25045	0.17325	0.21596	0.21934	0.19970
A2	0.03647	0.09634	0.19394	0.21589	0.17325	0.17997	0.11144	0.19970
A3	0.09675	0.07067	0.04804	0.04756	0.07737	0.03390	0.05470	0.02568
A4	0.13455	0.15979	0.14165	0.04370	0.06807	0.02694	0.02758	0.01980
A5	0.08149	0.05099	0.04398	0.04646	0.04687	0.03412	0.09040	0.05013
A6	0.17843	0.20079	0.14297	0.03582	0.04098	0.02514	0.04614	0.04296
A7	0.07506	0.05467	0.03882	0.09858	0.18118	0.08770	0.17174	0.11358
A8	0.17373	0.16255	0.11555	0.08681	0.18118	0.06761	0.09499	0.11358

Gambar 5.16 Matriks Preferensi Alternatif

4. Ulangi langkah 2-3 untuk menghitung subkriteria.
Gambar berikut menunjukkan hasil input perbandingan berpasangan subkriteria, matriks normalisasi, dan vektor preferensi subkriteria.

	S1	S2	S3
S1	1	0.25000	0.50000
S2	4.00000	1	2.00000
S3	2.00000	0.50000	1

Gambar 5.17 Perbandingan Berpasangan Subkriteria untuk Kriteria Finansial

	S1	S2	S3
S1	0.14286	0.14286	0.14286
S2	0.57143	0.57143	0.57143
S3	0.28571	0.28571	0.28571
Jumlah	1.00000	1.00000	1.00000

Gambar 5.18 Matriks Normalisasi Subkriteria untuk Kriteria Finansial

	S4	S5	S6	S7	S8
S4	1	0.20000	2.00000	0.33333	0.50000
S5	5.00000	1	4.00000	1.00000	3.00000
S6	0.50000	0.25000	1	0.50000	0.50000
S7	3.00000	1.00000	2.00000	1	0.50000
S8	2.00000	0.33333	2.00000	2.00000	1

Gambar 5.19 Perbandingan Berpasangan Subkriteria untuk Kriteria Teknis

	S4	S5	S6	S7	S8
S4	0.08696	0.07186	0.18182	0.06897	0.09091
S5	0.43478	0.35928	0.36364	0.20690	0.54545
S6	0.04348	0.08982	0.09091	0.10345	0.09091
S7	0.26087	0.35928	0.18182	0.20690	0.09091
S8	0.17391	0.11976	0.18182	0.41379	0.18182
Jumlah	1.00000	1.00000	1.00000	1.00000	1.00000

Gambar 5.20 Matriks Normalisasi Subkriteria untuk Kriteria Teknis

	Prioritas
S1	0.14286
S2	0.57143
S3	0.28571
S4	0.10010
S5	0.38201
S6	0.08371
S7	0.21995
S8	0.21422

Gambar 5.21 Vektor Preferensi Subkriteria

5. Ulangi langkah 2-3 untuk menghitung kriteria. Gambar berikut menunjukkan hasil input perbandingan berpasangan kriteria, matriks normalisasi, dan vektor preferensi kriteria.

	K1	K2
K1	1	3.00000
K2	0.33333	1

Gambar 5.22 Perbandingan Berpasangan Kriteria

	K1	K2	Prioritas
K1	0.75000	0.75000	0.75000
K2	0.25000	0.25000	0.25000
Jumlah	1.00000	1.00000	

Gambar 5.23 Matriks Normalisasi dan Vektor Preferensi Kriteria

6. Kalikan vektor preferensi kriteria dengan vektor preferensi subkriteria untuk mendapatkan bobot global subkriteria. Hasil perhitungan bobot global subkriteria dapat dilihat pada Gambar 5.24.

	Pri.Global
S1	0.10714
S2	0.42857
S3	0.21429
S4	0.02503
S5	0.09550
S6	0.02093
S7	0.05499
S8	0.05356

Gambar 5.24 Bobot Global Subkriteria

7. Kalikan bobot global subkriteria dengan matriks preferensi alternatif untuk mendapatkan bobot global alternatif. Gambar 5.25 menunjukkan hasil perhitungan bobot global alternatif.

	Bobot
A1	0.09727
A2	0.12929
A3	0.06462
A4	0.12399
A5	0.05402
A6	0.14598
A7	0.07692
A8	0.14738
A9	0.05735
A10	0.10318

Gambar 5.25 Bobot Global Alternatif

8. Ranking alternatif keputusan diperoleh berdasarkan bobot global alternatif. Hasil ranking alternatif keputusan dapat dilihat pada Gambar 5.26.

Ranking	Nama Alternatif	Bobot
1	Alternatif Proyek Investasi 8	0.14738
2	Alternatif Proyek Investasi 6	0.14598
3	Alternatif Proyek Investasi 2	0.12929
4	Alternatif Proyek Investasi 4	0.12399
5	Alternatif Proyek Investasi 10	0.10318
6	Alternatif Proyek Investasi 1	0.09727
7	Alternatif Proyek Investasi 7	0.07692
8	Alternatif Proyek Investasi 3	0.06462
9	Alternatif Proyek Investasi 9	0.05735
10	Alternatif Proyek Investasi 5	0.05402

Gambar 5.26 Ranking Alternatif

Selanjutnya dihitung konsistensi untuk setiap perbandingan berpasangan kriteria, subkriteria, dan alternatif. Penilaian dikatakan konsisten jika nilai $CR < 0,1$. Jika penilaian konsisten maka data akan disimpan dan jika tidak konsisten akan muncul kotak dialog yang menyatakan bahwa penilaian tidak konsisten dilengkapi dengan nilai CR. Gambar 5.27 merupakan kotak dialog informasi ketika penilaian tidak konsisten.

Gambar 5.27 Kotak Dialog *Inconsisten*

Hasil pengujian menunjukkan bahwa perangkat lunak SPK – AHP memberikan ranking alternatif yang sama dengan penelitian sebelumnya tetapi bobot setiap alternatif berbeda.

Berikut perhitungan kasus Seleksi Proyek Investasi menggunakan Microsoft Excel.

Tabel 5.2 Perbandingan Berpasangan Kriteria

Kriteria	Finansial	Teknis
Finansial	1	3
Teknis	0.333	1
Jumlah	1.333	4

Tabel 5.3 Matriks Normalisasi Kriteria

Kriteria	Finansial	Teknis	Prioritas
Finansial	0.75	0.75	0.75
Teknis	0.25	0.25	0.25
Jumlah	1	1	1

Tabel 5.4 Perbandingan Berpasangan Subkriteria untuk Kriteria Finansial

Kriteria Finansial	TGI	NPV	POT
TGI	1	0.25	0.5
NPV	4	1	2
POT	2	0.5	1
Jumlah	7	1.75	3.5

Tabel 5.5 Matriks Normalisasi Subkriteria untuk Kriteria Finansial

Kriteria Finansial	TGI	NPV	POT	Prioritas	Prioritas Global
TGI	0.143	0.143	0.143	0.143	0.107
NPV	0.571	0.571	0.571	0.571	0.429
POT	0.286	0.286	0.286	0.286	0.214
Jumlah	1	1	1	1	

Tabel 5.6 Perbandingan Berpasangan Subkriteria untuk Kriteria Teknis

Kriteria Teknis	SDT	CNV	FLX	PCS	INT
SDT	1	0.2	2	0.333	0.5
CNV	5	1	4	1	3
FLX	0.5	0.25	1	0.5	0.5
PCS	3	1	2	1	0.5
INT	2	0.333	2	2	1
Jumlah	11.5	2.783	11	4.833	5.5

Tabel 5.7 Matriks Normalisasi Subkriteria untuk Kriteria Teknis

Kriteria Teknis	SDT	CNV	FLX	PCS	INT	Prioritas	Prioritas Global
SDT	0.087	0.072	0.182	0.069	0.091	0.100	0.025
CNV	0.435	0.359	0.364	0.207	0.545	0.382	0.096
FLX	0.043	0.090	0.091	0.103	0.091	0.084	0.021
PCS	0.261	0.359	0.182	0.207	0.091	0.220	0.055
INT	0.174	0.120	0.182	0.414	0.182	0.214	0.054
Jumlah	1	1	1	1	1	1	

Tabel 5.14 Matriks Preferensi Alternatif

	TGI	NPV	POT	SDT	CNV	FLX	PCS	INT	Bobot
A1	0.031	0.022	0.161	0.250	0.173	0.216	0.219	0.200	0.097
A2	0.036	0.096	0.194	0.216	0.173	0.180	0.111	0.200	0.129
A3	0.097	0.071	0.048	0.048	0.077	0.034	0.055	0.026	0.065
A4	0.135	0.160	0.142	0.044	0.068	0.027	0.028	0.020	0.124
A5	0.081	0.051	0.044	0.046	0.047	0.034	0.090	0.050	0.054
A6	0.178	0.201	0.143	0.036	0.041	0.025	0.046	0.043	0.146
A7	0.075	0.055	0.039	0.099	0.181	0.088	0.172	0.114	0.077
A8	0.174	0.168	0.116	0.087	0.181	0.068	0.095	0.114	0.147
A9	0.053	0.045	0.049	0.097	0.029	0.179	0.126	0.117	0.057
A10	0.140	0.133	0.065	0.078	0.029	0.150	0.058	0.117	0.103

Tabel 5.15 Ranking Alternatif

No	Alternatif	Bobot
1	A8	0.147
2	A6	0.146
3	A2	0.129
4	A4	0.124
5	A10	0.103
6	A1	0.097
7	A7	0.077
8	A3	0.065
9	A9	0.057
10	A5	0.054

Hasil perhitungan menggunakan Microsoft Excel sama dengan hasil perhitungan menggunakan perangkat lunak SPK – AHP baik ranking alternatif maupun bobot setiap alternatif.

BAB VI

KESIMPULAN DAN SARAN

Bab ini berisi tentang kesimpulan yang dihasilkan berdasarkan penelitian yang telah dilaksanakan serta saran yang diberikan untuk pengembangan penelitian selanjutnya.

5.1 Kesimpulan

Berdasarkan analisis dan pengujian yang telah disajikan pada bab sebelumnya, dapat disimpulkan bahwa :

1. Rancangan perangkat lunak dibuat dengan menggunakan metode AHP dan didesain secara dinamis.
2. Hasil pengujian menggunakan data referensi [15] menunjukkan bahwa perangkat lunak yang dibangun memberikan ranking alternatif yang sama sehingga dengan adanya perangkat lunak ini *decision maker* tidak perlu membangun perangkat lunak lain untuk permasalahan yang berbeda.

5.2 Saran

Adapun saran dari Tugas Akhir ini adalah pengembangan perangkat lunak serupa dengan metode MCDM yang lain sehingga dapat digunakan untuk membandingkan hasil perankingan untuk kasus yang sama.

DAFTAR PUSTAKA

- [1] Kusumadewi, S., Hartati, S., Harjoko, A., & Wardoyo, R. (2006). *Fuzzy Multi-Attribute Decision Making (FUZZY MADM)*. Yogyakarta: Graha Ilmu.
- [2] Garcia, J. L., Alvarado, A., Blanco, J., Jimenez, E., Maldonado, A. A., & Cortes, G. (2013). Multi-attribute evaluation and selection of sites for agricultural product warehouses based on an Analytic Hierarchy Process. *Computers and Electronics in Agriculture*, 60-69.
- [3] Saefudin, & Wahyuningsih, S. (2014). Sistem Pendukung Keputusan untuk Penilaian Kinerja Pegawai Menggunakan Metode Analytical Hierarchy Process (AHP) pada RSUD Serang. *Sistem Informasi*, 33-40.
- [4] Ranius, A. Y. (2015). Sistem Pendukung Keputusan Penentuan Destinasi Wisata Unggulan di Kota Palembang. *Seminar Nasional Inovasi dan Tren (SNIT)*, 50-55.
- [5] Setiawan, A., Irawan, M. I., & Wijaya, R. (2007). Perancangan dan Pembuatan Aplikasi Decision Support System pada Departemen HRD dan Pembelian dengan Menggunakan Metode Analytic Hierarchy Process (AHP).
- [6] Artana, K. B. (2008). Pengambilan Keputusan Kriteria Jamak (MCDM) untuk Pemilihan Lokasi Floating Storage and Regasification Unit (FSRU): Studi Kasus Suplai LNG dari Ladang Tangguh ke Bali. *Pengambilan Keputusan Kriteria Jamak (MCDM)*, 97-111.
- [7] Javad, R. M., & Mohammad, A. (2014). Application of Analytical Hierarchy Process to Selection of Primary Crusher. *International Journal of Mining Science and Technology*, 519-523.
- [8] Turban, E., Aronson, J. E., & Liang, T.-P. (2005). *Decision Support Systems and Intelligent Systems (Sistem Pendukung Keputusan dan Sistem Cerdas)*. Yogyakarta: ANDI.
- [9] Taylor, B. W. (2013). *Introduction to Management Science*. New Jersey: Prentice Hall.

- [10] Rao, R. V. (2007). *Decision Making in the Manufacturing Environment : Using Graph Theory and Fuzzy Multiple Attribute Decision Making Methods*. London: Springer.
- [11] Kadir, A. (2005). *Dasar Pemrograman Java 2*. Yogyakarta: ANDI.
- [12] Hariyanto, B.(2011). *Esensi-Esensi Bahasa Pemrograman Java*. Bandung: Infomatika.
- [13] Nugroho, B. (2006). *Database Relational dengan MySQL*. Yogyakarta: ANDI.
- [14] Pressman, R. S. (2001). *Software Engineering*. New York: McGraw-Hill.
- [15] Raheditya, R., & Suparno. (2014). *Analisa Keputusan Proyek Investasi Pemasangan Booster Kompresor sebagai Upaya Mempertahankan Produksi Gas Bumi Lapangan Offshore L-Parigi di PT. PEP dengan Metode AHP dan TOPSIS*. Surabaya: ITS.

LAMPIRAN A

Source Code

1. Tambah Kasus Baru

```
public void insertKasus(){
    try {
        conn = koneksi.ConnectDb();
        PreparedStatement psts =
conn.prepareStatement("INSERT INTO kasus
(user_create, nama_kasus, deskripsi_kasus,
tgl_create)+"VALUES (?, ?, ?, ?)");
        psts.setString(1, tfUser.getText());
        psts.setString(2, tfKasus.getText());
        psts.setString(3,
txDeskripsi.getText());
        psts.setDate(4, new java.sql.Date(new
java.util.Date().getTime()));
        psts.executeUpdate();
        conn.close();
    }catch(Exception e)
        JOptionPane.showMessageDialog(null, e);
}

public void selectIDS(){
    try {
        conn = koneksi.ConnectDb();
        PreparedStatement psts =
conn.prepareStatement("SELECT id_kasus FROM kasus
ORDER BY id_kasus desc limit 1");
        ResultSet rss = psts.executeQuery();
        while (rss.next()) {
            id = rss.getInt("id_kasus");
        }conn.close();
    }catch(Exception e)
        JOptionPane.showMessageDialog(null, e);
}

public void insertDM(){
    try {
        conn = koneksi.ConnectDb();
        for (int i=0; i<tabelDM.getRowCount();i++){
            PreparedStatement pstd =
conn.prepareStatement("INSERT INTO decision_maker
```



```

(id_kasus, nama_dm) VALUES(?, ?)");
        pstd.setInt(1, id);
        pstd.setString(2, (String)
tabelDM.getValueAt(i,0));
        pstd.executeUpdate();
    }conn.close();
}catch(Exception e)
    JOptionPane.showMessageDialog(null, e);
}

public void insertAlternatif(){
    try {
        conn = koneksi.ConnectDb();
        for (int i=0;
i<tabelAlternatif.getRowCount(); i++){
            PreparedStatement psta =
conn.prepareStatement("INSERT INTO alternatif
(id_kasus, nama_alternatif) VALUES(?, ?)");
            psta.setInt(1, id);
            psta.setString(2, (String)
tabelAlternatif.getValueAt(i,0));
            psta.executeUpdate();
        }conn.close();
    }catch(Exception e)
        JOptionPane.showMessageDialog(null, e);
}

public void insertKriteria(){
    try {
        conn = koneksi.ConnectDb();
        for (int i=0;
i<tabelKriteria.getRowCount(); i++){
            PreparedStatement pstr =
conn.prepareStatement("INSERT INTO kriteria
(id_kasus, nama_kriteria) VALUES(?, ?)");
            pstr.setInt(1, id);
            pstr.setString(2, (String)
tabelKriteria.getValueAt(i,0));
            pstr.executeUpdate();
        }conn.close();
    }catch(Exception e)
        JOptionPane.showMessageDialog(null, e);
}

```

```

public void insertSubkriteria(){
    PreparedStatement pstr, pstb;
    ResultSet rsr;
    try {
        conn = koneksi.ConnectDb();
        for (int i=0;
i<tabelSubkriteria.getRowCount(); i++){
            int idr=0;
            String namak = (String)
tabelSubkriteria.getValueAt(i, 0);
            pstr = conn.prepareStatement("SELECT
id_kriteria FROM kriteria WHERE nama_kriteria
='"+namak+"'");
            rsr = pstr.executeQuery();
            while(rsr.next()){
                idr = rsr.getInt("id_kriteria");
                pstb =
conn.prepareStatement("INSERT INTO subkriteria
(id_kriteria, nama_subkriteria) VALUES(?, ?)");
                pstb.setInt(1, idr);
                pstb.setString(2, (String)
tabelSubkriteria.getValueAt(i,1));
                pstb.executeUpdate();
            }}conn.close();
        }catch(Exception e)
            JOptionPane.showMessageDialog(null, e);
    }
}

```

2. Input Perbandingan Berpasangan

```

public void insertPB(){
    PreparedStatement pstd, pst;
    ResultSet rsd;
    try {
        conn = koneksi.ConnectDb();
        int dm=0;
        String namap = (String)
cboDM.getSelectedItemAt();
        String id_dm = "SELECT id_dm FROM
decision_maker WHERE id_kasus='"+id+"' AND nama_dm
='"+namap+"'";
        pstd = conn.prepareStatement(id_dm);
        rsd = pstd.executeQuery();
        while(rsd.next()){

```

```

        dm = rsd.getInt(1);
    }double[][] pbk = new
double[tabelPBKriteria.getRowCount()][tabelPBKriter
ia.getColumnCount()];
    for(int i=0;
i<tabelPBKriteria.getRowCount(); i++){
        for(int j=0;
j<tabelPBKriteria.getColumnCount(); j++){
            pbk[i][j]=Double.parseDouble(tabelPBKriteria.
getValueAt(i, j).toString());
            String insert_pbk = "INSERT INTO
pbkriteria (id_dm, id_kriterial, id_kriteria2,
nilai_pb) VALUES(?, ?, ?, ?)";
            pst =
conn.prepareStatement(insert_pbk);
            pst.setInt(1, dm);
            pst.setInt(2, (idk.get(i)));
            pst.setInt(3, (idk.get(j)));
            pst.setDouble(4, pbk[i][j]);
            pst.executeUpdate();
        }}hapus(tabelPBKriteria);
    }catch(Exception e)
    System.out.print(e);
}

```

3. Konsistensi

```

public void setKonsistensi(JTable tabel){
    double[][] pb_input=new
double[tabel.getRowCount()][tabel.getColumnCount()]
;
    double[][] norm=new
double[tabel.getRowCount()][tabel.getColumnCount()]
;
    double[] jum = new
double[tabel.getColumnCount()];
    double[] Jumlah = new
double[tabel.getRowCount()];
    double[] prioritas = new
double[tabel.getRowCount()];
    double[] mv = new double[tabel.getRowCount()];
    double[] mvpv = new double[tabel.getRowCount()];
    double alfa=0, alfa_maks=0, ci=0;
    double[] ir =

```

```

{0,0,0.58,0.90,1.12,1.24,1.32,1.41,1.45,1.49,1.51,1
.48,1.56,1.57,1.59};
    for(int i=0; i<tabel.getRowCount(); i++){
        for(int j=0; j<tabel.getColumnCount(); j++){
            pb_input[i][j]=
Double.parseDouble(tabel.getValueAt(i,
j).toString());
        }for(int i=0; i<tabel.getColumnCount(); i++){
            for(int j=0; j<tabel.getRowCount(); j++){
                jum[i]+=pb_input[j][i];
            }for(int i=0; i<tabel.getRowCount(); i++){
                for(int j=0; j<tabel.getColumnCount(); j++){
                    norm[i][j]=pb_input[i][j]/jum[j];
                }for(int i=0; i<tabel.getRowCount(); i++){
                    for(int j=0; j<tabel.getColumnCount(); j++){
                        Jumlah[i]+=norm[i][j];
                    }for(int i=0; i<tabel.getRowCount(); i++){
                        prioritas[i]=Jumlah[i]/tabel.getColumnCount()
;
                    }for(int i=0; i<tabel.getRowCount(); i++){
                        for(int j=0; j<tabel.getColumnCount(); j++){
                            mv[i]+=pb_input[i][j]*prioritas[j];
                        }for(int i=0; i<tabel.getRowCount(); i++){
                            mvpv[i]=mv[i]/prioritas[i];
                        }for(int i=0; i<tabel.getRowCount(); i++){
                            alfa+=mvpv[i];
                            alfa_maks=alfa/tabel.getRowCount();
                            ci=(alfa_maks-
tabel.getRowCount())/(tabel.getRowCount()-1);
                            cr=ci/ir[tabel.getRowCount()-1];
                        }
}

```

4. Bobot Global Kriteria

```

public void hitungKriteria(){
    DefaultTableModel ahpR = new DefaultTableModel();
    try{
        conn = koneksi.ConnectDb();
        double[][] pbr_sum;
        for(int i=0; i<idk.size(); i++){
            ahpR.addColumn("K"+(i+1));
            ahpR.addRow(new Object[][] { });
        }ahpR.addRow(new Object[][] { });
        ahpR.addColumn("");
        ahpKriteria.setModel(ahpR);
        ahpKriteria1.setModel(ahpR);
    }
}

```



```

        ahpKriterial.setValueAt(String.format("%.5f"
, prioritas[i]), i, (2*idk.size()+1);
        conn.close();
    }catch(Exception e)
        JOptionPane.showMessageDialog(null, e);
}

```

5. Bobot Global Subkriteria

```

public void hitungSubkriteria(){
    PreparedStatement pstr, pstb;
    ResultSet rsr, rsb;
    DefaultTableModel ahpB = new
DefaultTableModel();
    ahpB.addColumn("");
    idb = new ArrayList<Integer>();
    idm = new ArrayList<Integer>();
    try{
        conn = koneksi.ConnectDb();
        double[][][] pbb_dm;
        double[][] pbb_sum, pbb;
        int idr=0;
        String namak=null;
        PreparedStatement pstd =
conn.prepareStatement("SELECT id_dm FROM
decision_maker WHERE id_kasus='"+id+"'");
        ResultSet rsd = pstd.executeQuery();
        while(rsd.next()){
            idm.add(new Integer(rsd.getInt(1)));
        }
        pstr = conn.prepareStatement("SELECT
id_kriteria, nama_kriteria FROM kriteria WHERE
id_kasus='"+id+"'");
        rsr = pstr.executeQuery();
        while(rsr.next()){
            idr = rsr.getInt(1);
            namak = rsr.getString(2);
            if(cboR.getSelectedItem().toString().equals(namak))
            {
                pstb = conn.prepareStatement("SELECT
id_subkriteria, nama_subkriteria FROM subkriteria
WHERE id_kriteria='"+idr+"'");
                rsb = pstb.executeQuery();
                while(rsb.next()){

```

```

        idb.add(new Integer(rsb.getInt(1)));
        ahpB.addColumn(rsb.getString(2));
        ahpB.addRow(new
Object[] {rsb.getString(2)});
        }ahpB.addColumn("");
        ahpSubkriteria.setModel(ahpB);
        pbb_dm = new
double[idm.size()][idb.size()][idb.size()];
        pbb_sum = new
double[idb.size()][idb.size()];
        pbb = new
double[idb.size()][idb.size()];
        for(int i=0; i<idm.size(); i++){
            for(int j=0; j<idb.size(); j++){
                for(int k=0; k<idb.size(); k++){
                    PreparedStatement prep_pbb =
conn.prepareStatement("SELECT nilai_pb FROM
pbbsubkriteria WHERE id_dm='"+idm.get(i)+"' AND
id_subkriteria1='"+idb.get(j)+"' AND
id_subkriteria2='"+idb.get(k)+"'");
                    ResultSet res_pbb =
prep_pbb.executeQuery();
                    while(res_pbb.next()){
                        pbb_dm[i][j][k]=res_pbb.getDouble(1);
pbb_sum = new double[idb.size()][idb.size()];
                        for(int f=0; f<pbb_sum.length;
f++){
                            for(int g=0;
g<pbb_sum.length; g++){
pbb_sum[f][g]=pbb_dm[0][f][g];
                            }}for(int f=0; f<pbb_sum.length; f++){
                                for(int g=0; g<pbb_sum.length; g++){
                                    for(int h=1; h<idm.size(); h++){
pbb_sum[f][g]=pbb_sum[f][g]*pbb_dm[h][f][g];
                                    }}
                                }
pbb[j][k]=Math.pow(pbb_sum[j][k], 1/(double)
idm.size());
ahpSubkriteria.setValueAt((pbb[j][k]), j, k+1);
                }}}}String nb;
                for(int x=0; x<ahpSubkriteria.getRowCount();
x++){
                    nb = (String) ahpSubkriteria.getValueAt(x, 0);
                    ahpB.addColumn(nb);
                }ahpB.addColumn("Prioritas");

```



```

        ahpB.addColumn("Pri.Global");
        double[][] pbb2 = new
double[ahpSubkriteria.getRowCount()][ahpSubkriteria
.getRowCount()];
        for(int a=0; a<pbb2.length; a++)
            for(int b=0; b<pbb2.length; b++){
                pbb2[a][b]=(double)
ahpSubkriteria.getValueAt(a, b+1);
            }
        double[] jumb = new double[pbb2.length];
        for(int d=0; d<jumb.length; d++)
            for(int e=0; e<jumb.length; e++){
                jumb[d]+=pbb2[e][d];
            }
        normb = new double[pbb2.length][pbb2.length];
        for(int g=0; g<normb.length; g++)
            for(int h=0; h<normb.length; h++){
                normb[g][h]=pbb2[g][h]/jumb[h];
            }
        for(int i=0; i<ahpSubkriteria.getRowCount();
i++){
            for(int j=ahpSubkriteria.getRowCount()+2;
j<2*ahpSubkriteria.getRowCount()+2; j++){
                ahpSubkriteria.setValueAt(normb[i][j-
(ahpSubkriteria.getRowCount()+2)], i, j);
            }
        }Jumlahb=new double[pbb2.length];
        for(int i=0; i<Jumlahb.length; i++){
            for(int j=0; j<Jumlahb.length; j++){
                Jumlahb[i]+=normb[i][j];
            }prioritasb = new double[pbb2.length];
            for(int i=0; i<prioritasb.length; i++)
                prioritasb[i]=Jumlahb[i]/pbb2.length;
        }conn.close();
    }catch(Exception e)
        JOptionPane.showMessageDialog(null, e);
}

```

6. Robot Lokal Alternatif

```

public void hitungAlternatif(){
    DefaultTableModel ahpA = new DefaultTableModel();
    try{
        conn = koneksi.ConnectDb();
        double[][][] pba_dm = new
double[idd.size()][ida.size()][ida.size()];
        double[][] pba_sum = new
double[ida.size()][ida.size()];
        double[][] pba = new
double[ida.size()][ida.size()];
        for(int p=0; p<idb.size(); p++){
            if(cboB.getSelectedItem().toString().equals(nm
b.get(p).getText())){
                for(int q=0; q<ida.size(); q++){
                    ahpA.addColumn("A"+(q+1));
                    ahpA.addRow(new Object[][]{});
                }ahpA.addRow(new Object[][]{});
                ahpAlternatif.setModel(ahpA);
                for(int i=0; i<idd.size(); i++){
                    for(int j=0; j<ida.size(); j++){
                        for(int k=0; k<ida.size(); k++){
                            PreparedStatement prep_pba =
conn.prepareStatement("SELECT nilai_pb FROM
pbsalternatif WHERE id_dm='"+idd.get(i)+"' AND
id_subkriteria='"+idb.get(p)+"' AND
id_alternatif1='"+ida.get(j)+"' AND
id_alternatif2='"+ida.get(k)+"'");
                            ResultSet res_pba =
prep_pba.executeQuery();
                            while(res_pba.next()){
                                pba_dm[i][j][k]=res_pba.getDouble(1);
                                gm kali start
                                pba_sum = new
double[ida.size()][ida.size()];
                                for(int f=0; f<pba_sum.length; f++)
                                    for(int g=0; g<pba_sum.length; g++){
                                        pba_sum[f][g]=pba_dm[0][f][g];
                                    }
                                for(int f=0; f<pba_sum.length; f++)
                                    for(int g=0; g<pba_sum.length; g++)
                                        for(int h=1; h<idd.size(); h++)
                                            pba_sum[f][g]=pba_sum[f][g]*pba_dm[h][f][g];
                                pba[j][k]=Math.pow(pba_sum[j][k], 1/(double)

```

```

idd.size());
ahpAlternatif.setValueAt(String.format("%.5f"),pba[j
][k]), j, k);
    ahpA.addColumn("");
    for(int q=0; q<ida.size(); q++)
        ahpA.addColumn("A"+(q+1));
    ahpA.addColumn("Prioritas");
    ahpAlternatif1.setModel(ahpA);
    double[] juma = new double[ida.size()];
    for(int d=0; d<juma.length; d++)
        for(int e=0; e<juma.length; e++){
            juma[d]+=pba[e][d];
        }ahpAlternatif.setValueAt(String.format("%.5f
"), juma[d]), ida.size(), d);
    norma = new double[ida.size()][ida.size()];
    for(int g=0; g<norma.length; g++)
        for(int h=0; h<norma.length; h++)
            norma[g][h]=pba[g][h]/juma[h];
    for(int i=0; i<ida.size(); i++)
        for(int j=ida.size()+1; j<(2*ida.size()+1;
j++)
    ahpAlternatif.setValueAt(String.format("%.5f"),
norma[i][j-(ida.size()+1)]), i, j);
    double[] juma2 = new double[ida.size()];
    for(int d=0; d<juma2.length; d++)
        for(int e=0; e<juma2.length; e++)
            juma2[d]+=norma[e][d];
    for(int j=ida.size()+1; j<(2*ida.size()+1;
j++)
    ahpAlternatif.setValueAt(String.format("%.5f"),
juma2[j-(ida.size()+1)]), ida.size(), j);
    Jumlaha=new double[ida.size()];
    for(int i=0; i<Jumlaha.length; i++)
        for(int j=0; j<Jumlaha.length; j++)
            Jumlaha[i]+=norma[i][j];
    prioritas = new double[ida.size()];
    for(int i=0; i<prioritasa.length; i++)
        prioritas[i]=Jumlaha[i]/ida.size();
    conn.close();
    }catch(Exception e)
        JOptionPane.showMessageDialog(null, e);
}

```



```

        Jumlaha=new double[pba.length];
        for(int g=0; g<Jumlaha.length; g++)
            for(int h=0; h<Jumlaha.length; h++)
                Jumlaha[g]+=norma[g][h];
        prioritas = new double[pba.length];
        for(int g=0; g<prioritasa.length; g++)
            prioritas[g]=Jumlaha[g]/pba.length;
    }}}pa[n][m]=prioritasa[n];
    }}

    PreparedStatement prep_bg =
conn.prepareStatement("SELECT bg_kriteria FROM
kriteria WHERE id_kasus='"+id+"'");
    ResultSet res_bg = prep_bg.executeQuery();
    while(res_bg.next()){
        bg_kri.add(new
Double(res_bg.getDouble(1)));
    }pg2 = new double[ida.size()];
    for(int i=0; i<ida.size(); i++){
        for(int j=0; j<idk.size(); j++){
            pg2[i]+=pa[i][j]*bg_kri.get(j);
        }
    }PreparedStatement prep =
conn.prepareStatement("UPDATE alternatif SET
bobot_alternatif=? WHERE
id_alternatif='"+ida.get(i)+"'");
    prep.setDouble(1, pg2[i]);
    prep.executeUpdate();
    }conn.close();
}catch(Exception e)
    JOptionPane.showMessageDialog(null, e);
}

```

8. Ubah Data Kasus

```

public void updateKasus(){
    try {
        conn = koneksi.ConnectDb();
        PreparedStatement pst =
conn.prepareStatement("UPDATE kasus SET
user_update=?, nama_kasus=?, tgl_update=? WHERE
id_kasus=?");
        pst.setString(1, tfUser.getText());
        pst.setString(2, tfKasus.getText());
        pst.setDate(3, new java.sql.Date(new
java.util.Date().getTime()));
    }
}

```

```

        pst.setInt(4, id);
        pst.executeUpdate();
        conn.close();
    }catch(Exception e)
        JOptionPane.showMessageDialog(null, e);
}

public void updateDM(){
    PreparedStatement pstd, pstd1;
    try {
        conn = koneksi.ConnectDb();
        for (int i=0; i<idd.size(); i++){
            pstd = conn.prepareStatement("UPDATE
decision_maker SET nama_dm=? WHERE id_dm=?");
            pstd.setString(1, (String)
tabelDM.getValueAt(i,1));
            pstd.setInt(2, idd.get(i));
            pstd.executeUpdate();
        }if (tabelDM.getRowCount(>idd.size()){
            for (int j=idd.size();
j<tabelDM.getRowCount(); j++){
                pstd1 =
conn.prepareStatement("INSERT INTO decision_maker
(id_kasus, nama_dm) VALUES(?, ?)");
                pstd1.setInt(1, id);
                pstd1.setString(2, (String)
tabelDM.getValueAt(j,1));
                pstd1.executeUpdate();
            }}conn.close();
        }catch(Exception e)
            JOptionPane.showMessageDialog(null, e);
    }

public void updateAlternatif(){
    PreparedStatement psta, pstal;
    try {
        conn = koneksi.ConnectDb();
        for (int i=0; i<ida.size(); i++){
            psta = conn.prepareStatement("UPDATE
alternatif SET nama_alternatif=? WHERE
id_alternatif=?");
            psta.setString(1, (String)
tabelAlternatif.getValueAt(i,1));
            psta.setInt(2, ida.get(i));

```

```

        psta.executeUpdate();
    }if(tabelAlternatif.getRowCount()>ida.size()){
        for (int j=ida.size();
j<tabelAlternatif.getRowCount(); j++){
            pstal =
conn.prepareStatement("INSERT INTO alternatif
(id_kasus, nama_alternatif) VALUES(?, ?)");
            pstal.setInt(1, id);
            pstal.setString(2, (String)
tabelAlternatif.getValueAt(j,1));
            pstal.executeUpdate();
        }}conn.close();
    }catch(Exception e)
        JOptionPane.showMessageDialog(null, e);
}

public void updateKriteria(){
    PreparedStatement pstr, pstr1;
    try {
        conn = koneksi.ConnectDb();
        for (int i=0; i<idk.size(); i++){
            pstr = conn.prepareStatement("UPDATE
kriteria SET nama_kriteria=? WHERE id_kriteria=?");
            pstr.setString(1, (String)
tabelKriteria.getValueAt(i,1));
            pstr.setInt(2, idk.get(i));
            pstr.executeUpdate();
        }if(tabelKriteria.getRowCount()>idk.size()){
            for (int j=idk.size();
j<tabelKriteria.getRowCount(); j++){
                pstr1 =
conn.prepareStatement("INSERT INTO kriteria
(id_kasus, nama_kriteria) VALUES(?, ?)");
                pstr1.setInt(1, id);
                pstr1.setString(2, (String)
tabelKriteria.getValueAt(j,1));
                pstr1.executeUpdate();
            }}conn.close();
        }catch(Exception e)
            JOptionPane.showMessageDialog(null, e);
    }

public void updateSubkriteria(){
    PreparedStatement pstrb, pstrbl, pstr;

```

```

    try {
        conn = koneksi.ConnectDb();
        for (int i=0; i<idb.size(); i++){
            pstb = conn.prepareStatement("UPDATE
subkriteria SET nama_subkriteria=? WHERE
id_subkriteria=?");
            pstb.setString(1, (String)
tabelSubkriteria.getValueAt(i,2));
            pstb.setInt(2, idb.get(i));
            pstb.executeUpdate();
        }if(tabelSubkriteria.getRowCount()>idb.size()){
            for (int j=idb.size();
j<tabelSubkriteria.getRowCount(); j++){
                int idr=0;
                String namak = (String)
tabelSubkriteria.getValueAt(j, 1);
                pstr =
conn.prepareStatement("SELECT id_kriteria FROM
kriteria WHERE nama_kriteria ='"+namak+"' AND
id_kasus='"+id+"'");
                ResultSet rsr =
pstr.executeQuery();
                while(rsr.next()){
                    idr = rsr.getInt(1);
                    pstbl =
conn.prepareStatement("INSERT INTO subkriteria
(id_kriteria, nama_subkriteria) VALUES(?, ?)");
                    pstbl.setInt(1, idr);
                    pstbl.setString(2, (String)
tabelSubkriteria.getValueAt(j,2));
                    pstbl.executeUpdate();
                }
            }
            conn.close();
        }catch(Exception e)
            JOptionPane.showMessageDialog(null, e);
    }

    public void edit(){
        PreparedStatement psts, prep;
        ResultSet rs, res;
        try {
            conn = koneksi.ConnectDb();
            String select = "SELECT nama_kasus,
user_create FROM kasus where id_kasus='"+id+"'";

```



```

        psts = conn.prepareStatement(select);
        rs = psts.executeQuery(select);
        while (rs.next()) {
            tfKasus.setText(rs.getString(1));
            tfUser.setText(rs.getString(2));
        }String cek = "SELECT
subkriteria.id_subkriteria FROM subkriteria,
kriteria WHERE
subkriteria.id_kriteria=kriteria.id_kriteria AND "
            + "kriteria.id_kasus='"+id+"'";
        prep = conn.prepareStatement(cek);
        res = prep.executeQuery();
        while(res.next()){
            idb.add(new Integer(res.getInt(1)));
        }if(idb.size()==0){
            selectDM();
            selectAlternatif();
            selectKriteria();
        }else{
            selectDM();
            selectAlternatif();
            selectKriteria();
            selectSubkriteria();
        }conn.close();
    }catch(Exception e)
        JOptionPane.showMessageDialog(null, e);
}

```

9. Update Perbandingan Berpasangan

```

public void updatePB(){
    PreparedStatement pstd, pst;
    ResultSet rsd;
    pb = new ArrayList <Double>();
    try {
        conn = koneksi.ConnectDb();
        int idd=0, idk1;
        double[][] pb2, pb3, pb4;
        String namad = (String)
cboDM.getSelectedItemAt();
        String id_dm = "SELECT id_dm FROM
decision_maker WHERE id_kasus='"+id+"' AND nama_dm
='"+namad+"'";
        pstd = conn.prepareStatement(id_dm);

```

```

        rsd = pstd.executeQuery();
        while(rsd.next()){
            idd = rsd.getInt(1);
        }PreparedStatement prep_pbr =
conn.prepareStatement("SELECT pbkriteria.nilai_pb
FROM pbkriteria, kriteria WHERE id_dm='"+idd+"' AND
pbkriteria.id_kriterial=kriteria.id_kriteria AND
kriteria.id_kasus='"+id+"'");
        ResultSet res_pbr= prep_pbr.executeQuery();
        while(res_pbr.next()){
            pb.add(new
Double(res_pbr.getDouble(1)));
            }idk1=(int)Math.sqrt(pb.size());
            if(idk1<idk.size()){
                pb2 = new
double[idk.size()][(idk.size()-idk1)];
                for(int i=0; i<idk.size(); i++){
                    for(int j=idk1; j<idk.size(); j++){
pb2[i][j]=Double.parseDouble(tabelPBKriteria.getVal
ueAt(i, j+1).toString());
                        String insert1 = "INSERT INTO
pbkriteria (id_dm, id_kriterial, id_kriteria2,
nilai_pb) VALUES(?, ?, ?, ?)";
                        pst =
conn.prepareStatement(insert1);
                        pst.setInt(1, idd);
                        pst.setInt(2, (idk.get(i)));
                        pst.setInt(3, (idk.get(j)));
                        pst.setDouble(4, pb2[i][j]);
                        pst.executeUpdate();
                        pst.close();
                    }
                }
            pb3 = new double[(idk.size()-
idk1)][idk1];
            for(int i=idk1; i<idk.size(); i++){
                for(int j=0; j<idk1; j++){
pb3[i][j]=Double.parseDouble(tabelPBKriteria.getVal
ueAt(i, j+1).toString());
                    String insert2 = "INSERT INTO
pbkriteria (id_dm, id_kriterial, id_kriteria2,
nilai_pb) VALUES(?, ?, ?, ?)";
                    pst =
conn.prepareStatement(insert2);
                    pst.setInt(1, idd);

```

```

        pst.setInt(2, (idk.get(i)));
        pst.setInt(3, (idk.get(j)));
        pst.setDouble(4, pb3[i][j]);
        pst.executeUpdate();
        pst.close();
    }
}
else{
    pb4 = new
double[idk.size()][idk.size()];
    for(int i=0; i<idk.size(); i++){
        for(int j=0; j<idk.size(); j++){
pb4[i][j]=Double.parseDouble(tabelPBKriteria.getValueAt
ueAt(i, j+1).toString());
            String update_pb = "UPDATE
pbkriteria SET nilai_pb=? WHERE id_dm=?,
id_kriterial=?, AND id_kriteria2=?";
            pst =
conn.prepareStatement(update_pb);
            pst.setDouble(1, pb4[i][j]);
            pst.setInt(2, idd);
            pst.setInt(3, (idk.get(i)));
            pst.setInt(4, (idk.get(j)));
            pst.executeUpdate();
            pst.close();
        }
    }
    JOptionPane.showMessageDialog(null, "Data
Berhasil Diupdate");
    hapus();
    cboDM.removeItemAt(cboDM.getSelectedIndex());
}catch(Exception e)
    JOptionPane.showMessageDialog(null, e);
}

```

10. Hapus Kasus

```

public void hapus(int id){
    try {
        conn = koneksi.ConnectDb();
        String hapus = "DELETE FROM kasus WHERE
id_kasus=?";
        PreparedStatement prep =
conn.prepareStatement(hapus);
        prep.setInt(1, id);
        prep.executeUpdate();
        conn.close();
    }catch(Exception e)
        JOptionPane.showMessageDialog(null, e);
}

```

11. Detail Kasus

```

public void detailKasus(){
    PreparedStatement psts, pstd, psta, pstr, pstb;
    ResultSet rss, rsd, rsa, rsr, rsb;
    try {
        conn = koneksi.ConnectDb();
        String nms=null, users=null, tglc=null,
updaters=null, tglu=null;
        psts = conn.prepareStatement("SELECT * FROM
kasus WHERE id_kasus='"+id+"'");
        rss = psts.executeQuery();
        while(rss.next()){
            nms = rss.getString("nama_kasus");
            users = rss.getString("user_create");
            tglc = rss.getString("tgl_create");
            updaters =
rss.getString("user_update");
            tglu = rss.getString("tgl_update");
        }dtKasus.setText(": "+nms);
        dtUser.setText(": "+users);
        dtTglCreate.setText(": "+tglc);
        if(updaters==null){
            jLabel15.setText("");
            dtTglUpdate.setText("");
        }else{
            dtTglUpdate.setText(": "+tglu);
        }if(tglu==null){
            jLabel14.setText("");
        }
    }
}

```

```

        dtUpdater.setText("");
    }else{
        dtUpdater.setText(": "+updaters);
    }pstd = conn.prepareStatement("SELECT
nama_dm FROM decision_maker WHERE
id_kasus='"+id+"'");
    rsd = pstd.executeQuery();
    while(rsd.next()){
        nmd.add(new JLabel(rsd.getString(1)));
    }for(int i=0; i<nmd.size(); i++){
        taDM.append(nmd.get(i).getText()+"\n");
    }psta = conn.prepareStatement("SELECT
nama_alternatif FROM alternatif WHERE
id_kasus='"+id+"'");
    rsa = psta.executeQuery();
    while(rsa.next()){
        nma.add(new JLabel(rsa.getString(1)));
    }for(int i=0; i<nma.size(); i++){
taAlternatif.append(nma.get(i).getText()+"\n");
    }pstr = conn.prepareStatement("SELECT
nama_kriteria FROM kriteria WHERE
id_kasus='"+id+"'");
    rsr = pstr.executeQuery();
    while(rsr.next()){
        nmk.add(new JLabel(rsr.getString(1)));
    }for(int i=0; i<nmk.size(); i++){
taKriteria.append(nmk.get(i).getText()+"\n");
    }pstb = conn.prepareStatement("SELECT
subkriteria.nama_subkriteria FROM subkriteria,
kriteria WHERE
subkriteria.id_kriteria=kriteria.id_kriteria AND
kriteria.id_kasus='"+id+"'");
    rsb = pstb.executeQuery();
    while(rsb.next()){
        nmb.add(new JLabel(rsb.getString(1)));
    }for(int i=0; i<nmb.size(); i++){
taSubkriteria.append(nmb.get(i).getText()+"\n");
    }conn.close();
    }catch(Exception e)
        JOptionPane.showMessageDialog(null, e);
}

public void tabelBobot(){
    DefaultTableModel tbl = new

```

```

DefaultTableModel();
tbl.addColumn("Ranking");
tbl.addColumn("Nama Alternatif");
tbl.addColumn("Bobot");
tblBobot.setModel(tbl);
nma = new ArrayList<JLabel>();
bg_alt = new ArrayList<Double>();
try{
    conn = koneksi.ConnectDb();
    PreparedStatement psta =
conn.prepareStatement("SELECT nama_alternatif,
bobot_alternatif FROM alternatif WHERE
id_kasus='"+id+"'ORDER BY bobot_alternatif desc");
    ResultSet rsa = psta.executeQuery();
    while(rsa.next()){
        nma.add(new JLabel(rsa.getString(1)));
        bg_alt.add(new
Double(rsa.getDouble(2)));
    }for(int i=0; i<nma.size(); i++){
        tbl.addRow(new Object[]{i+1,
nma.get(i).getText(), bg_alt.get(i)});
        tblBobot.setModel(tbl);
    }conn.close();
}catch(Exception e)
    JOptionPane.showMessageDialog(null, e);
}

```

12. Save PDF

```

try{
    conn = koneksi.ConnectDb();
    Document doc=new Document();
    JFileChooser saveFile = new JFileChooser();
    int save = saveFile.showSaveDialog(this);
    if(save==JFileChooser.APPROVE_OPTION){
        File f = saveFile.getSelectedFile();
        PdfWriter.getInstance(doc, new
FileOutputStream(saveFile.getCurrentDirectory()+f
.getName()+".pdf"));
    }
    doc.open();
    PdfPTable pb = new PdfPTable(idk.size()+2);
    pb.setHorizontalAlignment(0);
    pb.addCell(new PdfPCell(new Phrase("Nama

```

```

DM" )).setBackgroundColor(BaseColor.LIGHT_GRAY);
    pb.addCell(new PdfPCell(new
Phrase("Kriteria" )).setBackgroundColor(BaseColor
.LIGHT_GRAY);
        for(int i=0; i<idk.size(); i++)
            pb.addCell(new PdfPCell(new
Phrase("K" +(i+1) )).setBackgroundColor(BaseColor.
LIGHT_GRAY);
                PdfPTable pb3 = new
PdfPTable(ida.size()+3);
                pb3.setHorizontalAlignment(0);
                pb3.addCell(new PdfPCell(new
Phrase("Nama
DM" )).setBackgroundColor(BaseColor.LIGHT_GRAY);
                    pb3.addCell(new PdfPCell(new
Phrase("Kriteria" )).setBackgroundColor(BaseColor
.LIGHT_GRAY);
                        pb3.addCell(new PdfPCell(new
Phrase("Alternatif" )).setBackgroundColor(BaseCol
or.LIGHT_GRAY);
                            for(int i=0; i<ida.size(); i++)
                                pb3.addCell(new PdfPCell(new
Phrase("A" +(i+1) )).setBackgroundColor(BaseColor.
LIGHT_GRAY);
                                    PdfPTable hasil = new PdfPTable(3);
                                    hasil.setHorizontalAlignment(0);
                                    hasil.addCell(new PdfPCell(new
Phrase("Ranking" )).setBackgroundColor(BaseColor.
LIGHT_GRAY);
                                        hasil.addCell(new PdfPCell(new
Phrase("Nama
Alternatif" )).setBackgroundColor(BaseColor.LIGHT
_GRAY);
                                            hasil.addCell(new PdfPCell(new
Phrase("Bobot" )).setBackgroundColor(BaseColor.LI
GHT_GRAY);
                                                PdfPTable dataDM = new PdfPTable(2);
                                                dataDM.setHorizontalAlignment(0);
                                                dataDM.addCell(new PdfPCell(new
Phrase("No" )).setBackgroundColor(BaseColor.LIGHT
_GRAY);
                                                    dataDM.addCell(new PdfPCell(new
Phrase("Nama
DM" )).setBackgroundColor(BaseColor.LIGHT_GRAY);

```

```

        PdfPTable dataAlternatif = new
PdfPTable(2);
dataAlternatif.setHorizontalAlignment(0);
        dataAlternatif.addCell(new PdfPCell(new
Phrase("Kode"))).setBackgroundColor(BaseColor.LIG
HT_GRAY);
        dataAlternatif.addCell(new PdfPCell(new
Phrase("Nama
Alternatif"))).setBackgroundColor(BaseColor.LIGHT
_GRAY);
        PdfPTable dataKriteria = new
PdfPTable(2);
dataKriteria.setHorizontalAlignment(0);
        dataKriteria.addCell(new PdfPCell(new
Phrase("Kode"))).setBackgroundColor(BaseColor.LIG
HT_GRAY);
        dataKriteria.addCell(new PdfPCell(new
Phrase("Nama
Kriteria"))).setBackgroundColor(BaseColor.LIGHT_G
RAY);
        doc.add(new Paragraph("LAPORAN
PERHITUNGAN", FontFactory.getFont("Calibri", 16,
1, BaseColor.BLACK)));
        doc.add(new Paragraph(" ",
FontFactory.getFont("Calibri", 8, 0,
BaseColor.BLACK)));
        doc.add(new Paragraph("Judul Kasus :
"+dtKasus1.getText(),
FontFactory.getFont("Calibri", 12,
0,BaseColor.BLACK)));
        doc.add(new Paragraph("Nama User :
"+dtUser1.getText(),
FontFactory.getFont("Calibri", 12, 0,
BaseColor.BLACK)));
        doc.add(new Paragraph("Tanggal Create :
"+dtTglCreatel.getText(),
FontFactory.getFont("Calibri", 12, 0,
BaseColor.BLACK)));
        doc.add(new Paragraph("Terakhir Diupdate
Oleh : "+dtUpdater1.getText(),
FontFactory.getFont("Calibri", 12, 0,
BaseColor.BLACK)));
        doc.add(new Paragraph("Tanggal Update :
"+dtTglUpdate1.getText(),

```



```

FontFactory.getFont("Calibri", 12, 0,
BaseColor.BLACK));
    doc.add(new Paragraph("Deskripsi Kasus :
"+dtDeskripsi1.getText(),
FontFactory.getFont("Calibri", 12,
0,BaseColor.BLACK));
doc.add(new Paragraph("Data DM",
FontFactory.getFont("Calibri", 12, 0,
BaseColor.BLACK));
    doc.add(new Paragraph(" ",
FontFactory.getFont("Calibri", 4, 0,
BaseColor.BLACK));
        for (int i=0; i<idd.size(); i++){
dataDM.addCell(String.valueOf(i+1));
dataDM.addCell(nmd.get(i).getText());
    }doc.add(dataDM);
    doc.add(new Paragraph("Data Alternatif",
FontFactory.getFont("Calibri", 12,
0,BaseColor.BLACK));
    doc.add(new Paragraph(" ",
FontFactory.getFont("Calibri", 4, 0,
BaseColor.BLACK));
        for (int i=0; i<ida.size(); i++){
dataAlternatif.addCell(String.valueOf("A"+(i+1)))
; dataAlternatif.addCell(nma.get(i).getText());
    }doc.add(dataAlternatif);
doc.add(new Paragraph("Data Kriteria",
FontFactory.getFont("Calibri", 12, 0,
BaseColor.BLACK));
doc.add(new Paragraph(" ",
FontFactory.getFont("Calibri", 4, 0,
BaseColor.BLACK));
for(int i=0; i<idk.size(); i++){
dataKriteria.addCell(String.valueOf("K"+(i+1)));
dataKriteria.addCell(nmk.get(i).getText());
    }doc.add(dataKriteria);
doc.add(new Paragraph(" ",
FontFactory.getFont("Calibri", 5, 0,
BaseColor.BLACK));
doc.add(new Paragraph("Perbandingan Berpasangan
Kriteria", FontFactory.getFont("Calibri", 12,
0,BaseColor.BLACK));
doc.add(new Paragraph(" ",
FontFactory.getFont("Calibri", 4, 0,

```



```

        PdfPCell cell2 = new PdfPCell(new
Phrase(nmk.get(j).getText()));
cell2.setBackgroundColor(BaseColor.LIGHT_GRAY);
cell2.setRowspan(nma.size());
        pb3.addCell(cell2);
        for(int k=0; k<nma.size(); k++){
            pb3.addCell(new PdfPCell(new
Phrase("A"+(k+1))).setBackgroundColor(BaseColor.
LIGHT_GRAY);
            for(int l=0; l<nma.size(); l++){
                PreparedStatement prep_pba =
conn.prepareStatement("SELECT nilai_pb FROM
pbkalternatif WHERE id_dm='"+idd.get(i)+"' "+
"AND id_kriteria='"+idk.get(j)+"' AND
id_alternatif1='"+ida.get(k)+"' AND
id_alternatif2='"+ida.get(l)+"'");
                ResultSet res_pba = prep_pba.executeQuery();
                while(res_pba.next()){
                    select_pbka[i][k][l]=res_pba.getDouble(1);
                    pb3.addCell(String.valueOf(select_pbka[i][k][l]))
                    ;
                }
            }
        }
        doc.add(pb3);
        doc.add(new Paragraph("Ranking Alternatif",
FontFactory.getFont("Calibri", 12,
0,BaseColor.BLACK));
        doc.add(new Paragraph(" ",
FontFactory.getFont("Calibri", 4, 0,
BaseColor.BLACK));
        for(int i=0;
i<tblBobot1.getRowCount(); i++){
            for(int j=0;
j<tblBobot1.getColumnCount();
j++){
                hasil.addCell(String.format("%.5f",
tblBobot1.getValueAt(i, j).toString()));
            }
        }
        doc.add(hasil);
        doc.close();
    }
}
catch(Exception ex)
    System.out.println(ex);
}

```

13. Bantuan

```
try{
    Class.forName("com.mysql.jdbc.Driver");
    conn =
DriverManager.getConnection("jdbc:mysql://localhost
:3306/db_keputusan", "root", "");
    psth = conn.prepareStatement("SELECT
judul_help, content_help FROM help WHERE
id_help=1");
    rsh = psth.executeQuery();
    while (rsh.next()) {
        judul = rsh.getString(1);
        isi = rsh.getString(2);
    }labelContent.setText(judul);
    areaContent.setText(isi);
    conn.close();
}catch(Exception e){
    JOptionPane.showMessageDialog(null, e);
}
```


LAMPIRAN B

Data Uji

Data yang digunakan pada Tugas Akhir sebagai berikut.

Goal : Seleksi Proyek Investasi

Data Kriteria

No	Kriteria
1	Kriteria Finansial
2	Kriteria Teknis

Data Subkriteria

No	Kriteria	Subkriteria
1	Kriteria Finansial	TGI (Total Government Income)
2	Kriteria Finansial	NPV (Net Present Value)
3	Kriteria Finansial	POT (Pay Out Time)
4	Kriteria Teknis	S/DT (Schedule/Delivery Time Project)
5	Kriteria Teknis	CNV (Conventionality)
6	Kriteria Teknis	FLX (Flexibility)
7	Kriteria Teknis	PCS (Process Design)
8	Kriteria Teknis	INT (Integrity)

Data Alternatif

No	Alternatif
1	Alternatif Proyek Investasi 01
2	Alternatif Proyek Investasi 02
3	Alternatif Proyek Investasi 03
4	Alternatif Proyek Investasi 04
5	Alternatif Proyek Investasi 05
6	Alternatif Proyek Investasi 06
7	Alternatif Proyek Investasi 07
8	Alternatif Proyek Investasi 08

9	Alternatif Proyek Investasi 09
10	Alternatif Proyek Investasi 10

Hasil Ranking Alternatif

Ranking	Alternatif	Bobot
1	Alt. Proyek Investasi 08	0.150
2	Alt. Proyek Investasi 06	0.147
3	Alt. Proyek Investasi 02	0.132
4	Alt. Proyek Investasi 04	0.128
5	Alt. Proyek Investasi 10	0.098
6	Alt. Proyek Investasi 01	0.095
7	Alt. Proyek Investasi 07	0.077
8	Alt. Proyek Investasi 03	0.063
9	Alt. Proyek Investasi 09	0.057
10	Alt. Proyek Investasi 05	0.053

Data Perbandingan Berpasangan Kriteria

	Finansial	Teknis
Finansial	1	3
Teknis	0.33	1

Data Perbandingan Berpasangan Subkriteria

Finansial	TGI	NPV	POT
TGI	1	0.25	0.5
NPV	4	1	2
POT	2	0.5	1

Teknis	S/DT	CNV	FLX	PCS	INT
S/DT	1	0.2	2	0.33	0.5
CNV	5	1	4	1	3
FLX	0.5	0.25	1	0.5	0.5
PCS	3	1	2	1	0.5
INT	2	0.33	2	2	1

Data Perbandingan Berpasangan Alternatif

TGI	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10
A1	1	0.5	0.33	0.25	0.5	0.25	0.33	0.25	0.5	0.25
A2	2	1	0.33	0.25	0.33	0.25	0.5	0.25	0.5	0.25
A3	3	3	1	0.5	2	0.33	0.5	0.25	2	2
A4	4	4	2	1	1	1	2	1	2	1
A5	2	3	0.5	1	1	0.33	2	0.33	2	0.33
A6	4	4	3	1	3	1	3	1	3	2
A7	3	2	2	0.5	0.5	0.33	1	0.33	2	0.33
A8	4	4	4	1	3	1	3	1	3	1
A9	2	2	0.5	0.5	0.5	0.33	0.5	0.33	1	0.33
A10	4	4	0.5	1	3	0.5	3	1	3	1

NPV	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10
A1	1	0.2	0.25	0.2	0.25	0.2	0.25	0.2	0.25	0.2
A2	5	1	3	0.2	3	0.33	3	0.33	3	1
A3	4	0.33	1	0.5	1	0.5	2	0.33	2	0.5
A4	5	5	2	1	4	0.5	2	1	2	2
A5	4	0.33	1	0.25	1	0.33	1	0.25	1	0.33
A6	5	3	2	2	3	1	3	2	5	2
A7	4	0.33	0.5	0.5	1	0.33	1	0.2	2	0.33
A8	5	3	3	1	4	0.5	5	1	5	1
A9	4	0.33	0.5	0.5	1	0.2	0.5	0.2	1	0.2
A10	5	1	2	0.5	3	0.5	3	1	5	1

POT	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10
A1	1	0.33	3	2	3	2	4	2	4	2
A2	3	1	4	2	4	1	5	1	5	2
A3	0.33	0.25	1	0.33	1	0.5	1	0.5	2	0.5
A4	0.5	0.5	3	1	3	1	4	2	6	3
A5	0.33	0.25	1	0.33	1	0.33	1	0.33	2	0.5
A6	0.5	1	2	1	3	1	4	2	5	3
A7	0.25	0.2	1	0.25	1	0.25	1	0.33	2	0.33
A8	0.5	1	2	0.5	3	0.5	3	1	5	3
A9	0.25	0.2	0.5	0.17	0.5	0.2	0.5	0.2	1	5
A10	0.5	0.5	2	0.33	2	0.33	3	0.33	0.2	1

S/DT	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10
A1	1	2	5	5	5	5	4	4	3	3
A2	0.5	1	5	5	5	5	4	4	3	3
A3	0.2	0.2	1	2	0.5	1	0.25	0.25	1	1
A4	0.2	0.2	0.5	1	1	1	0.25	0.25	1	1
A5	0.2	0.2	2	1	1	2	0.33	0.33	0.5	0.5
A6	0.2	0.2	1	1	0.5	1	0.25	0.33	0.5	0.5
A7	0.25	0.25	4	4	3	4	1	2	0.33	0.5
A8	0.25	0.25	4	4	3	3	0.5	1	0.5	0.5
A9	0.33	0.33	1	1	2	2	3	2	1	2
A10	0.33	0.33	1	1	2	2	2	2	0.5	1

CNV	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10
A1	1	1	3	3	4	4	1	1	5	5
A2	1	1	3	3	4	4	1	1	5	5
A3	0.33	0.33	1	2	2	2	0.33	0.33	3	3
A4	0.33	0.33	0.5	1	2	2	0.33	0.33	3	3
A5	0.25	0.25	0.5	0.5	1	2	0.2	0.2	2	2
A6	0.25	0.25	0.5	0.5	0.5	1	0.2	0.2	2	2
A7	1	1	3	3	5	5	1	1	5	5
A8	1	1	3	3	5	5	1	1	5	5
A9	0.2	0.2	0.33	0.33	0.5	0.5	0.2	0.2	1	1
A10	0.2	0.2	0.33	0.33	0.5	0.5	0.2	0.2	1	1

FLX	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10
A1	1	1	7	7	5	5	3	3	3	2
A2	1	1	6	5	5	5	3	3	2	1
A3	0.14	0.17	1	2	1	2	0.25	0.33	0.2	0.25
A4	0.14	0.2	0.5	1	0.5	1	0.33	0.5	0.2	0.25
A5	0.2	0.2	1	2	1	2	0.2	0.25	0.17	0.2
A6	0.2	0.2	0.5	1	0.5	1	0.2	0.25	0.17	0.2
A7	0.33	0.33	4	3	5	5	1	2	0.2	0.25
A8	0.33	0.33	3	2	4	4	0.5	1	0.2	0.25
A9	0.33	0.5	5	5	6	6	5	5	1	2
A10	0.5	1	4	4	5	5	4	4	0.5	1

PCS	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10
A1	1	3	4	6	2	3	1	3	3	5
A2	0.33	1	2	4	1	2	1	2	1	2
A3	0.25	0.5	1	3	0.5	2	0.33	0.5	0.33	0.5
A4	0.17	0.25	0.33	1	0.33	0.5	0.25	0.33	0.25	0.33
A5	0.5	1	2	3	1	2	0.5	0.33	1	2
A6	0.33	0.5	0.5	2	0.5	1	0.33	0.5	0.33	0.5
A7	1	1	3	4	2	3	1	3	2	4
A8	0.33	0.5	2	3	3	2	0.33	1	0.5	2
A9	0.33	1	3	4	1	3	0.5	2	1	5
A10	0.2	0.5	2	3	0.5	2	0.25	0.5	0.2	1

INT	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10
A1	1	1	7	7	5	5	2	2	2	2
A2	1	1	7	7	5	5	2	2	2	2
A3	0.14	0.14	1	3	0.25	0.25	0.2	0.2	0.2	0.2
A4	0.14	0.14	0.33	1	0.25	0.25	0.2	0.2	0.2	0.2
A5	0.2	0.2	4	4	1	2	0.33	0.33	0.25	0.25
A6	0.2	0.2	4	4	0.5	1	0.25	0.25	0.25	0.25
A7	0.5	0.5	5	5	3	4	1	1	1	1
A8	0.5	0.5	5	5	3	4	1	1	1	1
A9	0.5	0.5	5	5	4	4	1	1	1	1
A10	0.5	0.5	5	5	4	4	1	1	1	1

LAMPIRAN C

Biodata Penulis



Penulis bernama Siti Maghfiroh, lahir di Jember, 20 Desember 1993. Penulis merupakan anak bungsu dari pasangan Ngasiman dan Samijatun. Penulis menempuh pendidikan formal mulai dari SD Negeri Tanjungrejo 07 (2000-2006), SMP Negeri 1 Wuluhan (2006-2009), dan SMA Negeri Ambulu (2009-2012). Setelus lulus dari SMA, pada tahun 2012 penulis melanjutkan studi ke jenjang S1 di Jurusan Matematika ITS Surabaya. Di Jurusan

Matematika, penulis mengambil Bidang Minat Ilmu Komputer. Hingga saat ini penulis memiliki kemampuan dalam bidang database, pemrograman Java dan PHP. Selain aktif kuliah, penulis juga aktif berorganisasi di KM ITS melalui HIMATKA ITS sebagai staf Dep. Sosial Masyarakat (2013-2014) dan BEM FMIPA ITS sebagai staf Dep. Ekososial (2013-2014). Penulis juga aktif dalam kepanitian di ITS sebagai panitia GERIGI ITS (2013), panitia INTERN FMIPA ITS (2013 dan 2014), panitia Olimpiade Matematika ITS (2014 dan 2015). Selain itu, penulis juga melaksanakan Kerja Praktek di PT. Duta Cipta Konsultama bidang pengembangan web sistem informasi pada tahun 2015. Informasi lebih lanjut mengenai Tugas Akhir ini dapat ditujukan ke penulis melalui email: smaghfiroh44@gmail.com.

