

# Analisis Dan Perancangan Sistem Pendukung Keputusan Untuk Menentukan Prioritas Pengerjaan Aplikasi Menggunakan Metode *Simple Additive Weighting*

Kevin Gautama<sup>1</sup>, Suwarno, S.T., M.M.<sup>2</sup>

Program Sistem Informasi, Fakultas Ilmu Komputer, Universitas Internasional Batam, Sei Ladi, Jl.  
Gajah Mada, Baloi Permai, Kec. Sekupang, Kota Batam, Kepulauan Riau, 29442

Email: [kevin@uib.ac.id](mailto:kevin@uib.ac.id), [suwarno@uib.ac.id](mailto:suwarno@uib.ac.id)

## Abstrak

Pada industri pengembangan perangkat lunak yang telah bergeser dari model pengembangan tradisional menjadi model pengembangan *agile-based* yang mana metode *agile* digolongkan memiliki desain dan tugas yang terus berubah mengakomodasi kebutuhan *user* yang semakin kompleks dan dinamis. Oleh karena itu, seorang manajer perlu membuat banyak keputusan penting. Laporan skripsi ini dibuat untuk merancang sistem pendukung keputusan yang dapat menunjang keputusan seorang manajer dalam menentukan pekerjaan yang perlu diprioritaskan dalam pengembangan aplikasi.

Metode yang digunakan dalam sistem pendukung keputusan adalah *Simple Additive Weighting* (SAW) yang mencari penjumlahan terbobot dari rating kinerja pada setiap alternatif pada semua atribut. Penilaian yang digunakan pada sistem pendukung keputusan adalah risiko, pengeluaran, penggunaan sumber daya, lama waktu penyelesaian dan keuntungan dari penyelesaian tugas.

**Kata kunci:** *simple additive weighting, sistem pendukung keputusan*

## Abstract

*In the software development industry that has shifted from traditional development models to agile-based development models in which agile methods are classified as having designs and tasks that are constantly changing to accommodate increasingly complex and dynamic users requirement. Therefore, a manager needs to make many important decisions. This thesis report is conducted to design a decision support system that can support a manager's decision in determining which work needs to be prioritized in application development.*

*The method used in the decision support system is Simple Additive Weighting (SAW), which seeks the weighted sum of performance ratings on each alternative on all attributes. Assessments used in the decision support system are the risk, expenditure, resources, length of completion time and benefits of a task's completion.*

**Key word:** *simple additive weighting, decision support system*

## 1 PENDAHULUAN

Terus berkembangnya teknologi informasi membuat pemanfaatan teknologi di bidang usaha lainnya terus meningkat. Berbagai sistem

dikembangkan dengan tujuan meningkatkan akurasi data, mempercepat proses data dan mengurangi biaya untuk meningkatkan kinerja suatu perusahaan.

Menggunakan sistem pendukung untuk menunjang keputusan merupakan salah satu pemanfaatan teknologi dalam menyelesaikan masalah yang bersifat manajerial pada operasional kantor sehari-hari. Salah satu masalah manajerial pada kantor adalah menentukan pekerjaan mana yang perlu diprioritaskan.

Industri pengembangan perangkat lunak telah bergeser dari model pengembangan tradisional menjadi model pengembangan *agile-based* yang mana metode *agile* digolongkan memiliki siklus pengembangan yang lebih pendek, interaksi pelanggan yang lebih banyak, rilis berkala serta desain ulang untuk mengakomodasi kebutuhan *user* yang semakin kompleks dan dinamis (Matharu et al., 2015).

Tujuan pokok dari pengembangan sistem perangkat lunak adalah memenuhi berbagai macam kebutuhan pemangku kepentingan. Oleh sebab itu, mengelola kebutuhan perangkat lunak merupakan peran penting menuju keberhasilan proyek (Dabbagh & Lee, 2014).

Sehingga seorang manajer perlu membuat banyak keputusan penting. Dikarenakan tenggat waktu dan anggaran bisa menjadi tantangan bagi manajer untuk memutuskan persyaratan mana yang perlu didahulukan dan mengarah pada kepuasan pemangku keputusan. Dengan risiko yang dihasilkan ketika melakukan keputusan, seorang manajer lebih memilih untuk mengurangi risiko dan mengambil keputusan ketika ada kepastian.

Dengan menggunakan sistem pendukung keputusan, manajer dapat menggunakan pendekatan yang sistemik untuk mengetahui kelebihan dan kekurangan berbagai pekerjaan yang dilalui di proyek serta memberikan pilihan atau menawarkan pendekatan terbaik dalam memenuhi tujuan sambil menghemat investasi.

Dengan menggunakan data dan model keputusan, sistem pendukung keputusan dapat membantu menyelesaikan masalah yang sifat semi terstruktur maupun yang tidak terstruktur (Kemal Farouq, 2014).

Berdasarkan latar belakang yang telah diuraikan, penulis akan menerapkan metode *simple additive weighting* pada sistem pendukung keputusan dalam menentukan prioritas pengerjaan. Judul yang digunakan adalah “**Analisis dan Perancangan Sistem Pendukung Keputusan untuk Menentukan Prioritas Pengerjaan Aplikasi Menggunakan Metode Simple Additive Weighting**”.

Adapun rumusan masalah dalam penelitian ini sebagai berikut:

1. Bagaimana sistem pendukung keputusan dapat memberi informasi sebagai penunjang dalam pengambilan keputusan prioritas pengerjaan aplikasi?
2. Bagaimana metode *simple additive weighting* diterapkan dalam sistem pendukung keputusan untuk menentukan prioritas pengerjaan?

Tujuan penelitian ini adalah Mengembangkan sistem pendukung keputusan yang dapat menghasilkan urutan daftar tugas pada pengembangan aplikasi dengan menggunakan metode *simple additive weighting*.

## 2 LANDASAN TEORI

### 2.1 TINJAUAN PUSTAKA

Telah banyak penelitian yang dilakukan pada sistem pendukung keputusan. Penelitian dengan judul Implementasi Metode *Simple Additive Weighting*(SAW) pada Sistem Pendukung Keputusan dalam Menentukan Prioritas Perbaikan Jalan yang ditulis oleh (Muhammad et al., 2017) menjelaskan sistem pendukung keputusan dengan metode *simple additive weighting* dapat memberikan

rekomendasi perbaikan sebagai perbantuan perencanaan.

Penelitian dengan judul Sistem Pendukung Keputusan Penentuan Nomor Urut Caleg dengan Metode SAW ditulis oleh (Saputra, 2015) yang membangun sistem pendukung keputusan dengan menggunakan metode *simple additive weighting*. Konsep dasar dari metode *simple additive weighting* adalah mencari penjumlahan terbobot dari rating kinerja pada setiap alternatif pada semua atribut sehingga dikenal dengan terminologi metode dengan penjumlahan terbobot. Pada penelitian juga dijelaskan metode *Simple Additive Weighting* dapat digunakan sebagai acuan pengambilan keputusan karena memenuhi kriteria objektivitas dalam penentuan nomor urut caleg.

Penelitian berjudul Komparasi Metode *Simple Additive Weighting* dan *Profile Matching* pada Pemilihan Mitra Jasa Pengiriman Barang yang ditulis oleh (Akhirina, 2016) menjelaskan bahwa kedua metode dilakukan dengan menggunakan data yang sama dengan menggunakan proses penilaian yang sama mengeluarkan alternatif terbaik terpilih yang sama. Maka penelitian ini akan menerapkan metode *simple additive weighting* pada sistem pendukung keputusan untuk menentukan prioritas pekerjaan.

Selain itu penulis menggunakan parameter dari penelitian berjudul *Prioritizing Parameters for Software Project Selection using Analytical Hierarchical Process* yang ditulis oleh (Kaur & Bhatia, 2015), pada penelitian dijelaskan bahwa dalam memilih proyek perangkat lunak, "*Cost*" memiliki prioritas paling tinggi, dimana "*Resource Utilization*" memiliki prioritas terendah. Untuk membuat keputusan yang bermanfaat, pertama harus fokus pada total biaya dan kemudian profitabilitas proyek atau "*Project Benefits*". "*Completion Time*" berada pada tingkat prioritas keempat, sedangkan "*Project*

*Risks*" berada pada tingkat ketiga. *Project risks, project benefits, cost, resource utilization and completion time* masing-masing memiliki *priority vector* 23,57%, 15,22%, 48,25%, 4,25% dan 8,67%.

Penelitian yang dilakukan oleh (Hoda & Murugesan, 2016) dengan judul *Multi-Level Agile Project Management Challenges: A Self-Organizing Team Perspective* menjelaskan bahwa salah satu tantangan pada tingkat tugas (*task*) adalah ketergantungan tugas (*task dependencies*) yang ada antara tugas lainnya. Ketergantungan tugas merujuk pada situasi di mana penyelesaian satu tugas diperlukan untuk memulai tugas selanjutnya. Konsekuensi utama dari ketergantungan tugas adalah dapat dengan mudah mengakibatkan penurunan kinerja dan kecepatan pengerjaan proyek dan menunda pengiriman produk.

## **2.2 LANDASAN TEORI**

### **2.2.1 Data**

Data merupakan informasi yang diubah menjadi bentuk yang mudah dipindahkan dan diproses. Menurut (Andalia & Setiawan, 2015), data merupakan sebuah fakta yang menggambarkan suatu kejadian. Penjelasan lain dari (Irviani & Oktaviana, 2017) data merupakan nilai seperti karakter, angka atau simbol. Gagasan lain juga diungkapkan oleh (Iswandy, 2015), data merupakan informasi yang belum diolah dimana belum memiliki nilai bagi pihak-pihak yang membutuhkannya.

### **2.2.2 Sistem**

Menurut (Masykur & Atmaja, 2015), sistem adalah himpunan atau kumpulan elemen yang berkerja sama untuk memenuhi target atau sasaran tertentu. Sistem adalah kumpulan komponen yang saling berkaitan dan berinteraksi membentuk kesatuan yang utuh. Sistem berfungsi untuk memproses masukan data dan menghasilkan luaran yang memiliki makna, akurat, tepat

waktu dan relevan bagi pengguna. Gagasan lain dari (Muhammad et al., 2017), menjelaskan himpunan komponen yang berkerja sama secara harmonis untuk mencapai satu tujuan.

**2.2.3 Informasi**

Informasi merupakan himpunan data yang diproses untuk memberikan nilai dan makna bagi suatu organisasi atau perusahaan. Kualitas informasi dinilai berdasarkan ketepatan waktu, keakuratan, relevansi, kelengkapan dan bisa dimengerti (Andalia & Setiawan, 2015).

Menurut (Alannita & Suaryana, 2014), pengguna menggunakan informasi sebagai dasar dalam pengambilan keputusan saat ini atau masa yang akan datang. Informasi juga dapat mendukung ketepatan pengambilan keputusan.

**2.2.4 Sistem Informasi**

Menurut (Iswandy, 2015), sistem informasi adalah sebuah sistem yang mengolah atau memproses data transaksi untuk menunjang operasi suatu organisasi. Sistem informasi juga diartikan sebagai sekumpulan komponen yang saling terhubung atau berkomunikasi satu sama lainnya untuk mendapatkan atau mengumpulkan, menyebarkan atau mendistribusikan informasi, serta pengarsipan informasi untuk pengawasan dan dijadikan bahan pertimbangan dalam keputusan di suatu organisasi.

Menurut (Alannita & Suaryana, 2014) menjelaskan pemanfaatan teknologi komputer dalam organisasi yang melayani pengguna atau pihak-pihak yang membutuhkan dengan menyediakan hasil olahan data atau informasi disebut sebagai sistem informasi.

**2.2.5 Sistem Pendukung Keputusan**

Sistem pendukung keputusan merupakan sistem informasi yang terhubung atau terintegrasi antar komponen yang mempermudah proses pengambilan keputusan untuk

menangani masalah yang bersifat manajerial pada perusahaan (Nurdiyanto & Meilia, 2016). Ide yang serupa dinyatakan oleh (Muhammad et al., 2017), menjelaskan sistem pendukung keputusan merupakan satu set model berbasis prosedur yang membantu manajer dalam mengambil keputusan dengan memproses data dan pertimbangan. Dan dijelaskan oleh (Akhirina, 2016), sistem pendukung keputusan memiliki kemampuan dalam membantu orang membuat keputusan untuk menyelesaikan masalah semi struktur atau tidak terstruktur yang mungkin berubah dengan cepat dan tidak mudah sebelumnya.

**2.2.6 Simple Additive Weighting**

Menurut (Saputra, 2015), konsep dasar metode SAW adalah mencari penjumlahan terbobot dari rating kinerja pada setiap alternatif pada semua kriteria sehingga dikenal sebagai metode penjumlahan terbobot. Metode SAW ini membutuhkan proses normalisasi matriks keputusan (X) ke suatu skala yang dapat di perbandingkan dengan semua rating alternatif yang ada.

Ada 2 macam kriteria pada metode SAW yaitu *cost* merupakan kriteria biaya dan *benefit* merupakan kriteria keuntungan. Menurut (Akhirina, 2016), untuk memproses 2 kriteria pada SAW perlu menggunakan formulasi berikut:

$$r_{ij} = \begin{cases} \frac{x_{ij}}{\max_i x_{ij}} & \text{jika } j \text{ adalah atribut keuntungan (benefit)} \\ \frac{\min_i x_{ij}}{x_{ij}} & \text{jika } j \text{ adalah atribut biaya (cost)} \end{cases} \dots\dots\dots (1)$$

dimana:

- $r_{ij}$  rating kinerja ternormalisasi
- $X_{ij}$  rating alternatif  $A_i$  di atribut  $C_j$
- $\text{Max}_i$  nilai maksimum dari tiap baris
- $\text{Min}_i$  nilai minimum dari tiap baris

Dimana  $r_{ij}$  adalah rating kinerja ternormalisasi dari alternatif  $A_i$ , pada atribut  $C_j$ ;  $i= 1,2,\dots,m$  dan  $j=1,2,\dots,n$ .

Nilai preferensi untuk setiap alternatif ( $V_i$ ) diberikan sebagai  $V_i = \sum_{j=1}^n W_j r_{ij} \dots\dots\dots (2)$

dimana:

$V_i$  nilai preferensi

$W_j$  nilai bobot setiap

kriteria

Dimana alternatif  $A_i$  dengan nilai  $V_i$  terbesar merupakan alternatif terbaik.

### 2.2.7 ASP.NET MVC

Menurut (Galloway et al., 2014), *ASP.NET MVC* merupakan sebuah *framework* yang menerapkan konsep dalam membangun aplikasi *web*. *Model* adalah *class* yang mewakili struktur data. *Controller* adalah kumpulan *class* yang menangani interaksi dan memperbarui *model* kemudian meneruskan informasi ke *View* yang sesuai sedangkan *View* berfungsi untuk menyediakan antarmuka untuk menampilkan informasi yang diterima dari *controller*.

Menurut (Munro, 2015), *ASP.NET MVC 5* digunakan untuk membangun aplikasi web yang canggih (*controller*), mengakses dengan basis data (*model*) dan secara dinamis membuat *HTML* (*view*).

### 2.2.8 Entity Framework

*Entity Framework* merupakan *Object Relational Mapper (ORM)* yang dikembangkan oleh Microsoft. *Entity Framework* dapat yang mempermudah pembuatan *data access layer* dengan memungkinkan akses data, dengan mewakili data sebagai model konseptual atau kumpulan entitas dan relasi (Singh, 2015).

Keuntungan menggunakan *Entity Framework* dapat mengurangi waktu pengembangan yang diperlukan karena dapat melakukan akses data tanpa harus menulis kode *ADO.NET*, dapat menulis logika akses data pada tingkat yang lebih tinggi seperti *C#* atau *VB* daripada menulis *query SQL* dan *stored procedure* (Singh, 2015).

### 2.2.9 Language Integrated Query

*Language Integrated Query* atau disebut *LINQ* dengan merupakan sebuah *query* yang digunakan untuk menyimpan dan mengambil data dari sumber yang

berbeda dengan menggunakan bahasa pemrograman seperti *C#* atau *VB*. *LINQ* terintegrasi dalam *.NET*, sehingga memberikan cara tunggal untuk *query* data, tanpa memerhatikan dari mana asal data.

Dengan *LINQ*, bahasa pemrograman *.NET* dapat melakukan *query* data dengan cara yang serupa seperti *SQL* melakukan *query* data dalam database (Spaanjaars, 2014).

### 2.2.10 Website

Menurut (R. F. Nugroho et al., 2016), *website* adalah kumpulan halaman yang dirangkum dalam domain atau dalam *World Wide Web*.

*Website* adalah halaman-halaman yang menyajikan informasi bersifat statis ataupun dinamis yang membentuk satu rangkaian bangunan yang saling terhubung dengan jaringan-jaringan halaman (Syukroon & Hasan, 2015).

### 2.2.11 HTML

Dengan nama panjang *hypertext markup language*, merupakan bahasa standar yang dirancang untuk merangkai isi dan struktur antarmuka dari halaman *web* (R. F. Nugroho et al., 2016).

### 2.2.12 Javascript

Menurut (Wahyudi et al., 2017), *javascript* adalah bahasa *scripting* berbasis objek untuk membuat halaman web yang interaktif dengan menanggapi *input* dari *form*, respon klik dan *event* lainnya dari pengguna ke halaman web.

### 2.2.13 JQuery

Menurut (Hidayat et al., 2017), *jquery* adalah kumpulan-kumpulan kode *javascript* siap pakai yang dapat mempermudah pengembangan aplikasi *web* tanpa perlu membuat kode secara panjang lebar.

*JQuery* dapat digunakan untuk mengatur atau memanipulasi animasi *css*, *HTML DOM*, pengaturan *event* dan *Ajax*. (Wahyudi et al., 2017).

### 2.2.14 CSS

*Cascading Style Sheet* yang disingkat *CSS* merupakan bahasa

pemrograman yang berfungsi untuk mengatur komponen antarmuka suatu halaman web sehingga terstruktur dan seragam (Iriadi & Indrasari, 2017).

Salah satu fitur dari *CSS* adalah dapat memisahkan presentasi situs dari isi web. Pemisahan ini berguna untuk mengurangi duplikasi dan kompleksitas yang diasosiasikan dengan memasukkan informasi antarmuka ke isi struktural. Pemisahan ini mempermudah menjaga konsistensi antarmuka pada seluruh situs karena dapat diubah dari satu dokumen *CSS* (Olsson, 2014).

#### **2.2.15 Bootstrap**

*Bootstrap* adalah *framework* yang menyediakan kode *CSS* dan *JavaScript* siap pakai untuk mempersingkat dan menyederhanakan proses pengembangan *website* (Ariansyah et al., 2017). *Bootstrap* memiliki templat desain berdasarkan *CSS* dan *JavaScript* untuk tipografi, formulir, navigasi, tombol dan komponen antarmuka lainnya pada halaman web. *Bootstrap* juga dapat mempermudah pembangunan *website* dengan antarmuka *responsive* yang mana antarmuka yang dapat merespons atau mengubah ukurannya sendiri tergantung pada jenis perangkat.

#### **2.2.16 Database**

*Database* atau basis data merupakan sekumpulan data yang berkaitan yang diorganisasikan berdasarkan struktur tertentu untuk mendukung operasi aplikasi dalam sistem tertentu (Nugrahanti, 2015). Dan juga gagasan dari (Efendi et al., 2014) menjelaskan basis data menyimpan sekumpulan data dalam perangkat keras komputer yang kemudian diakses oleh perangkat lunak. Basis data juga berfungsi memberi batasan atau aturan pada data agar menjaga keakuratan data dalam tabel dan diperoleh kemudahan dan kecepatan saat mengakses data dalam basis data.

#### **2.2.17 DBMS**

*DBMS* atau Sistem manajemen basis data merupakan sistem untuk menciptakan dan mengelola basis data (Septa et al., 2014). *DBMS* menyediakan cara yang sistematis untuk membuat, mengambil, memperbarui dan menghapus. *DBMS* juga menyediakan antarmuka antara *database* dengan pengguna.

#### **2.2.18 SQL Server**

*Microsoft SQL Server* adalah perangkat lunak yang didesain untuk melakukan proses manipulasi basis data berukuran besar dengan berbagai fasilitas (Suripto & Triyono, 2014). Penjelasan lain dari (B. Nugroho et al., 2015), *SQL Server* adalah sistem manajemen basis data relasional yang dibuat oleh *Microsoft*. Bahasa *query* yang digunakan oleh *SQL Server* adalah *Transact-SQL*.

#### **2.2.19 Entity Relationship Diagram**

Diagram yang menggunakan notasi dan simbol untuk menunjukkan struktur dan relasi data dalam basis data disebut *Entity Relationship Diagram* atau disingkat sebagai *ERD* (Susila & Triyono, 2015). Dan dijelaskan juga oleh (Iswandy, 2015) bahwa *ERD* memiliki 2 komponen utama yaitu entitas dan relasi, masing-masing memiliki atribut untuk merepresentasikan fakta yang terjadi di dunia nyata.

#### **2.2.20 Unified Modelling Language**

*Unified Modelling Language* atau disebut *UML* merupakan bahasa pemodelan standar yang menggunakan notasi-notasi grafis dengan tujuan menggambarkan aturan, proses, objek dan juga relasi antar objek yang berlangsung pada sistem (Lenti, 2014).

Menurut (Lenti, 2014), sketsa, cetak biru dan bahasa pemrograman merupakan tiga karakter penting yang dalam *UML*. *UML* sebagai sketsa, *UML* mampu berfungsi sebagai penghubung dalam komunikasi antar aspek dari sistem. *UML* sebagai cetak biru dapat menyampaikan informasi secara detail dan lengkap dalam menjelaskan logika

dan proses alur program yang berlangsung pada program (*Forward Engineering*) atau sebaliknya, membaca atau membongkar program dan menerjemahkan kembali ke dalam diagram (*Reverse Engineering*). UML sebagai bahasa pemrograman, diagram dapat dipelajari dan diubah menjadi kode program. Beberapa diagram yang umum ada dalam pemodelan UML adalah diagram *use case*, *sequence* dan *class*.

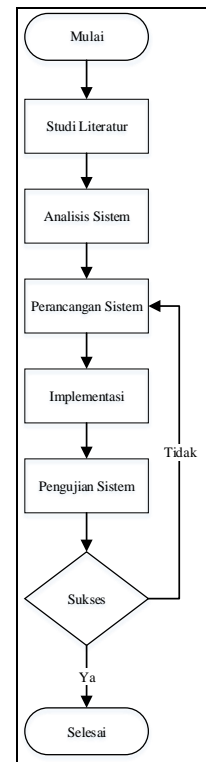
### 2.2.21 Flowchart

*Flowchart* merupakan diagram yang mewakili proses kerja atau aksi suatu proses yang di lambangkan dengan menggunakan simbol yang disusun secara sistematis (Iswandy, 2015). Simbol *flowchart* dihubungkan dengan tanda panah yang menunjukkan arah alur proses. Simbol *flowchart* paling umum adalah *terminator* dengan bentuk oval, *process* dengan bentuk persegi panjang, *decision* dengan bentuk belah ketupat, *connector* dengan bentuk lingkaran kecil dengan label, *data* dengan bentuk jajar genjang dan *document*.

## 3 METODOLOGI PENELITIAN

### 3.1 Alur Penelitian

Ketika melakukan perancangan sistem pendukung keputusan, ada beberapa tahapan yang dilalui secara sistematis sebagai kerangka acuan penelitian. Alur penelitian ini akan menjelaskan bagaimana sistem dirancang dan dianalisa. Berikut alur dapat dilihat pada gambar dibawah ini.



Gambar 1 Alur Penelitian

#### 1. Studi Literatur

Pada tahap studi literatur, penulis menentukan dan mengumpulkan data buku, jurnal dan sumber lainnya yang berkaitan dengan sistem pendukung keputusan yang akan dikembangkan.

#### 2. Analisis Sistem

Pada tahap analisa, penulis mempelajari data yang dikumpulkan dari tahap sebelumnya dan disesuaikan dengan metode *simple additive weighting*.

#### 3. Perancangan Sistem

Pada tahap implementasi, penulis mendefinisikan struktur basis data, alur proses sistem dan antarmuka sistem yang akan dikembangkan.

#### 4. Implementasi

Pada tahap implementasi, penulis akan membangun sistem sesuai dengan desain yang telah ditentukan.

#### 5. Pengujian Sistem

Pada tahap pengujian sistem, sistem akan diuji oleh penulis untuk membuktikan sistem berjalan dengan baik dan sesuai dengan desain dan persyaratan yang telah ditentukan.

Setelah tahapan sudah dilakukan maka selanjutnya adalah mengumpulkan hasil analisis data dan perancangan yang telah dilakukan, penulis memulai proses penyusunan laporan dengan format tugas akhir dan membuat kesimpulan.

### 3.2 Analisis Permasalahan

Tujuan penelitian adalah mendukung manajer dalam memutuskan prioritas pengerjaan pada proyek aplikasi. Dengan mengetahui prioritas pengerjaan, manajer dapat mengurangi pengeluaran operasional dan administrasi.

#### 3.2.1 Analisis Kriteria Penilaian

Kriteria yang digunakan berdasarkan parameter yang digunakan pada penelitian berjudul *Prioritizing Parameters for Software Project Selection using Analytical Hierarchical Process* yang ditulis (Kaur & Bhatia, 2015). Adapun perubahan nama dari *project risks*, *project benefits* dan *cost* menjadi *task risks*, *task benefits* dan *task cost*. *Task benefits* merupakan kemampuan memberikan manfaat. *Task cost* merupakan pengeluaran moneter langsung maupun tidak langsung. *Resource utilization* merupakan sumber daya yang terpakai pada tiap tahap pembangunan proyek. *Completion time* merupakan lamanya proses pembuatan proyek perangkat lunak yang harus diselesaikan sebelum batas waktu. Tabel berikut merupakan kriteria yang digunakan dalam sistem pendukung keputusan dalam menentukan prioritas pekerjaan.

**Tabel 1** Kriteria Penilaian

Nama	Tipe	Bobot
<i>Task Risks</i>	<i>Cost</i>	23.57%
<i>Task Cost</i>	<i>Cost</i>	48.25%

<i>Resource Utilization</i>	<i>Cost</i>	4.25%
<i>Completion Time</i>	<i>Cost</i>	8.67%
<i>Task Benefits</i>	<i>Benefit</i>	15.26%

Berikut merupakan klasifikasi pembobotan tiap kriteria yang akan digunakan:

**Tabel 2** Pembobotan Kriteria *Task Risks* & *Task Benefits*

Jenis Risiko	Bobot
Rendah	0.25
Menengah	0.5
Tinggi	0.75
Sangat Tinggi	1

**Tabel 3** Pembobotan Kriteria *Task Cost* & *Resource Utilization*

Jenis Pengeluaran	Bobot
Murah	0.25
Menengah	0.5
Mahal	0.75
Sangat Mahal	1

**Tabel 4** Pembobotan Kriteria *Completion Time*

Waktu	Bobot
Cepat (< 7 Hari)	0.25
Menengah (7 - 14 Hari)	0.5
Lama (14 - 30 Hari)	0.75
Sangat Lama (> 30 Hari)	1

#### 3.2.2 Analisis Metode

Sistem pendukung keputusan yang dikembangkan akan menggunakan metode *simple additive weighting*. Sistem pendukung keputusan akan mengeluarkan hasil berupa daftar tugas yang telah di beri peringkat. Adapun beberapa proses yang perlu dilalui untuk menghasilkan *output*, pertama menentukan kriteria dari alternatif kemudian tentukan nilai bobot pada setiap kriteria. Buat matriks keputusan



sesuai kriteria yang kemudian akan melalui proses normalisasi ke skala berdasarkan jenis atribut atau kriteria *cost* atau *benefit* sehingga diperoleh matriks ternormalisasi dengan rumus:

$$r_{ij} = \begin{cases} \frac{x_{ij}}{\text{Max}_i x_{ij}} & \text{Jika } j \text{ adalah atribut keuntungan (benefit)} \\ \frac{\text{Min}_i x_{ij}}{x_{ij}} & \text{Jika } j \text{ adalah atribut biaya (cost)} \end{cases} \quad (3)$$

dimana:

- $r_{ij}$  rating kinerja ternormalisasi
- $x_{ij}$  rating alternatif  $A_i$  di atribut  $C_j$
- $\text{Max}_i$  nilai maksimum dari tiap baris
- $\text{Min}_i$  nilai minimum dari tiap baris

Matriks ternormalisasi akan melalui proses preferensi dengan menggunakan formulasi berikut:

$$V_i = \sum_{j=1}^n W_j r_{ij} \dots \dots \dots (4)$$

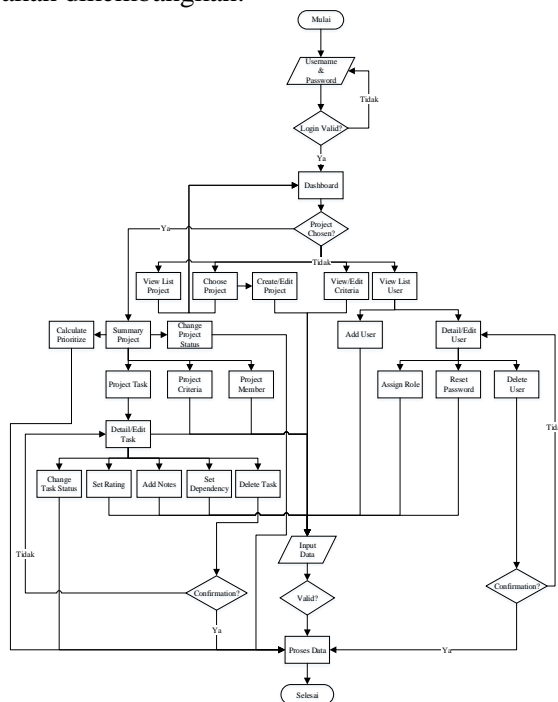
dimana:

- $V_i$  nilai preferensi
- $W_j$  nilai bobot setiap kriteria
- Dimana alternatif  $A_i$  dengan nilai  $V_i$  terbesar merupakan alternatif terbaik.

### 3.3 Perancangan Sistem

#### 3.3.1 Flowchart

Berikut adalah *flowchart* yang menggambarkan alur dari sistem yang akan dikembangkan.



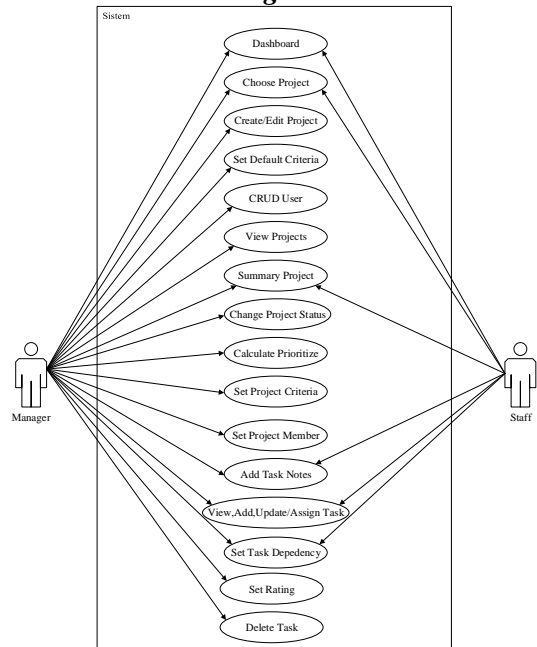
Gambar 2 Flowchart Sistem

Pada saat membuka sistem, *user* akan menampilkan halaman login yang memerlukan masukan berupa *username* dan *password* untuk lanjut menggunakan sistem. Jika *login* berhasil, *user* akan ditampilkan halaman *dashboard* yang menampilkan ulasan umum atau ringkasan dari sistem.

Jika proyek telah terpilih, sistem akan menampilkan menu *summary project* yang menampilkan informasi umum tentang proyek terpilih. *User* juga dapat mengakses menu *project member* untuk mengatur anggota proyek, menu *criteria* untuk mengatur kriteria penilaian proyek dan menu *task* dimana *user* dapat mengubah data, menambah catatan, mengubah status, menentukan nilai, menentukan *dependency* dan hapus *task*.

Jika proyek belum dipilih, *user* akan ditampilkan menu *project* yang menampilkan daftar proyek, menu *criteria* yang menampilkan kriteria penilaian umum dan menu *user* untuk mengatur pengguna sistem.

#### 3.3.2 Use Case Diagram



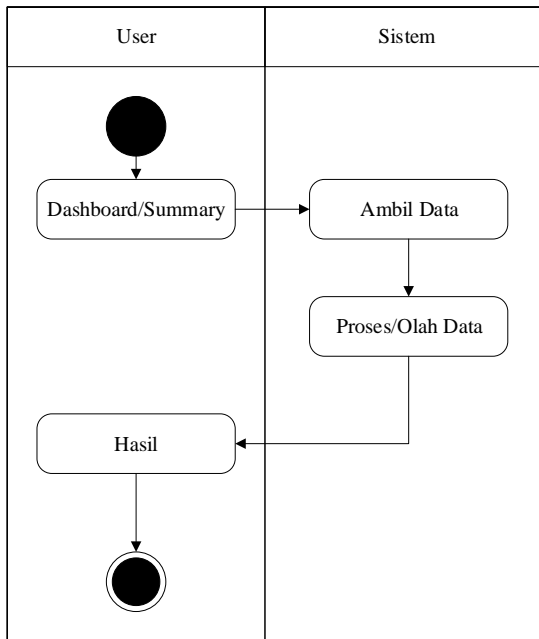
Gambar 3 Use Case Diagram Sistem

*Use case diagram* digunakan untuk menjelaskan aksi-aksi yang dapat

dilakukan oleh aktor pada sistem. Sistem akan digunakan oleh 2 macam aktor yaitu *manager* dan *staff*.

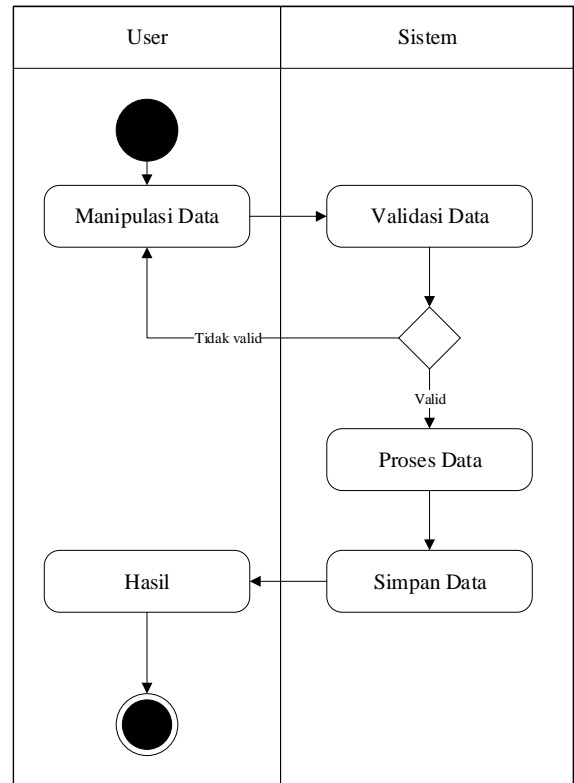
### 3.3.3 Activity Diagram

*Activity diagram* berfungsi untuk menggambarkan alur-alur aktivitas umum sebuah sistem. *Activity diagram* berikut menjelaskan proses halaman *dashboard* dan *summary*.



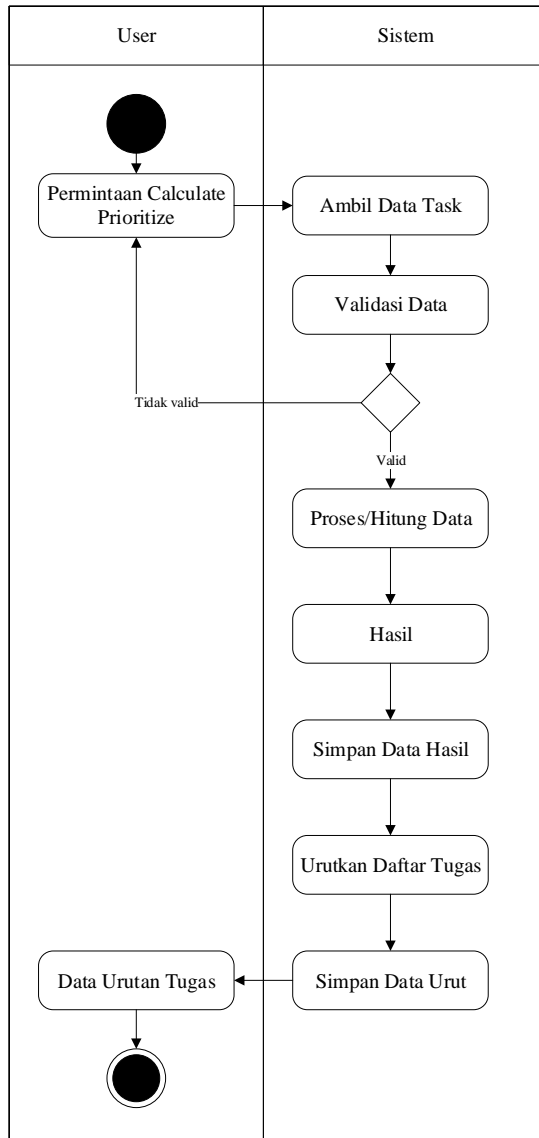
**Gambar 4** Activity Diagram Dashboard dan Summary

Berikut adalah *activity diagram* yang menjelaskan bagaimana data di proses sebelum disimpan ke basis data.



**Gambar 5** Activity Diagram Manipulasi Data

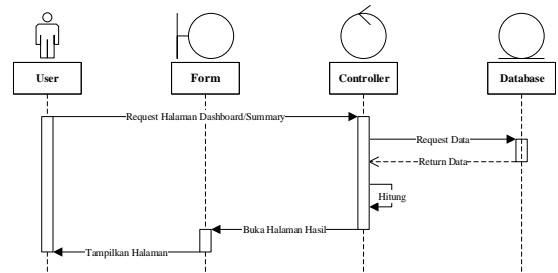
*Activity diagram* berikut akan menjelaskan alur yang terjadi saat *user* melakukan proses *calculate prioritize* untuk melakukan proses ini *user* perlu menentukan data kriteria seperti nama, tipe kriteria beserta bobotnya. *User* juga perlu menambah data *project*, data *task* dan memberi nilai kinerja atau *rating* ke masing-masing data *task*. Berikut adalah *activity diagram calculate prioritize*.



Gambar 6 Activity Diagram Calculate Prioritize

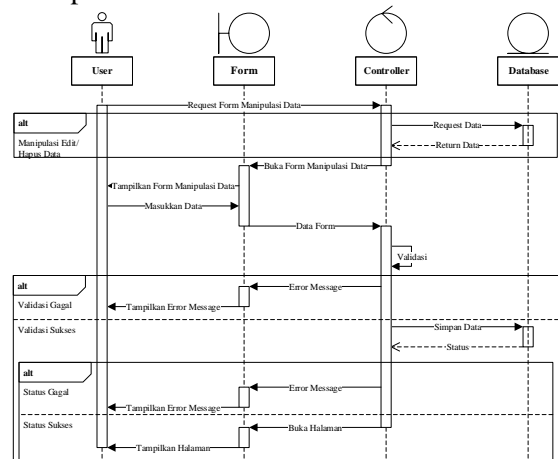
### 3.3.4 Sequence Diagram

Sequence diagram berfungsi untuk menunjukkan interaksi antar objek secara prosedural. Berikut adalah proses pengambilan data untuk halaman dashboard dan summary.



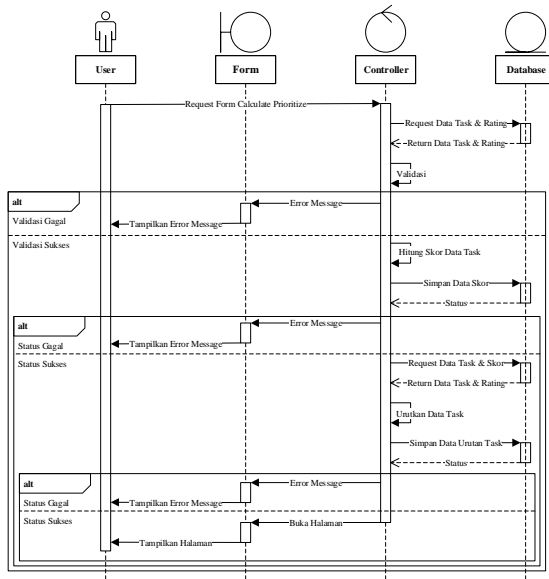
Gambar 7 Sequence Diagram Dashboard atau Summary

Sequence diagram berikut adalah proses manipulasi data yang dilakukan oleh user. Untuk proses manipulasi data seperti hapus atau ubah data memiliki proses tambahan yaitu mengambil data ke database sebelum membuka form manipulasi data.



Gambar 8 Sequence Diagram Manipulasi Data

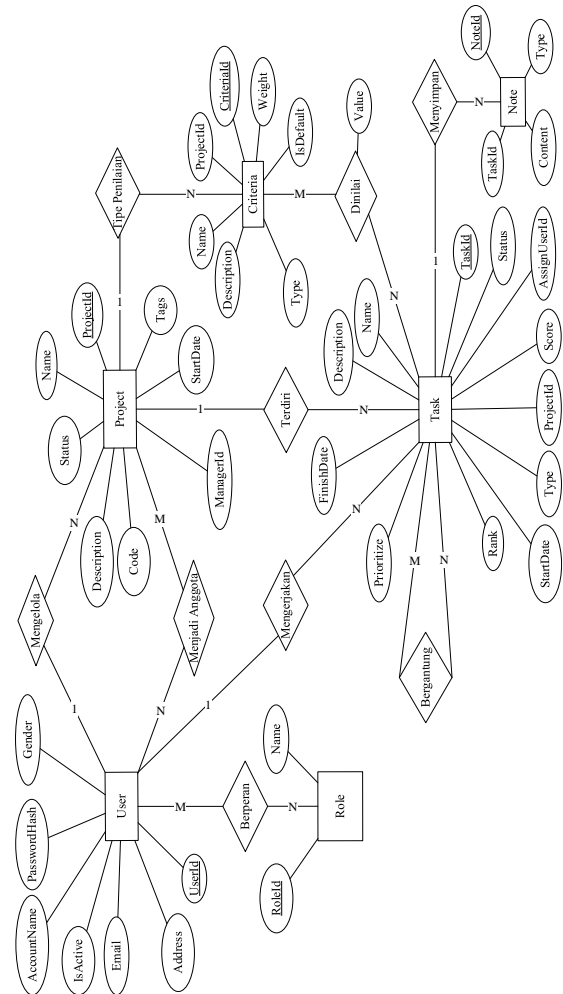
Berikut merupakan sequence diagram yang menjelaskan alur proses perhitungan yang dilakukan oleh sistem ketika user melakukan calculate prioritize di halaman dashboard.



Gambar 9 Sequence Diagram Calculate Prioritize

### 3.3.5 Entity Relationship Diagram

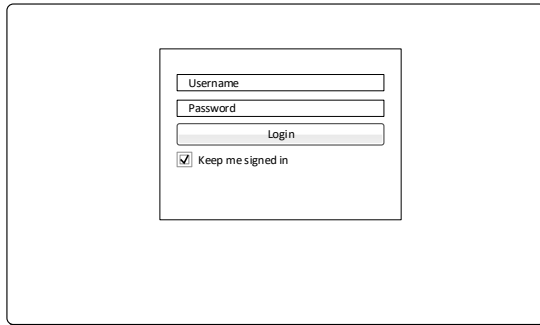
Entity relationship diagram berfungsi menjelaskan relasi antar entitas pada basis data. Total enam entitas pada basis data yaitu *role*, *user*, *project*, *criteria*, *task* dan *note*. Berikut adalah *entity relationship diagram* sistem pendukung keputusan yang akan dibangun.



Gambar 10 Entity Relationship Diagram Sistem

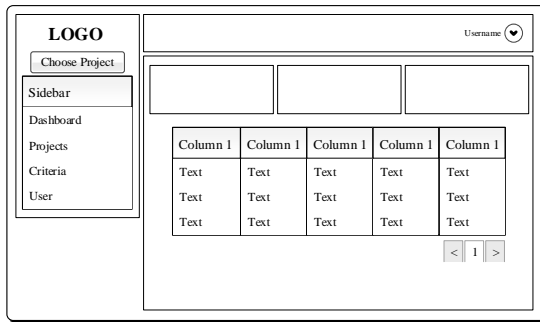
### 3.3.6 Rancangan Antarmuka

Berikut adalah gambar halaman *login* dimana *user* mengisi form untuk melakukan proses autentikasi sistem. Halaman *login* berisi sebuah form dengan masukan berupa *username* dan *password*. Halaman *login* memiliki opsi *remember me* sehingga *user* tidak perlu mengisi ulang form ketika kembali ke sistem.



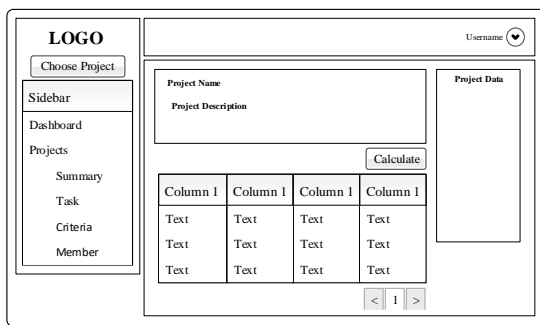
Gambar 11 Halaman Login

Sistem akan menampilkan halaman *dashboard* setelah *user* berhasil melakukan proses autentikasi. Pada halaman *dashboard* menyajikan informasi keseluruhan proyek secara ringkas.



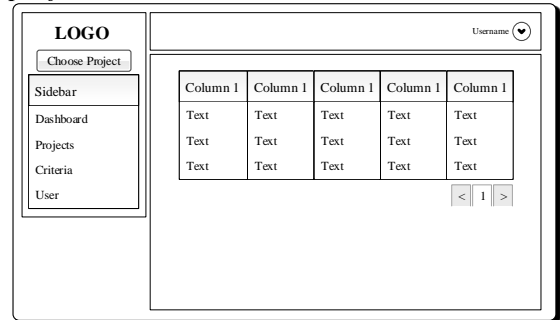
Gambar 12 Halaman Dashboard

Halaman *summary* akan dapat diakses ketika *user* telah memilih proyek dan perubahan menu terhadap opsi pada *sidebar*. Pada halaman *summary* menyajikan informasi tentang proyek secara ringkas dan memberikan akses tombol perhitungan peringkat pengerjaan aplikasi. Berikut adalah halaman *summary*.



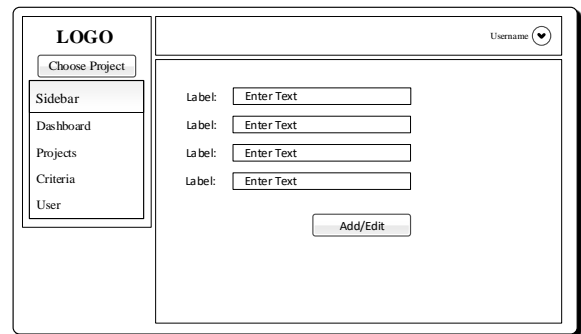
Gambar 13 Halaman Summary

Halaman berikut menampilkan daftar data dimana *user* dapat melakukan manipulasi seperti tambah atau ubah data. Halaman ini digunakan pada menu *project*, *user* dan *task*.



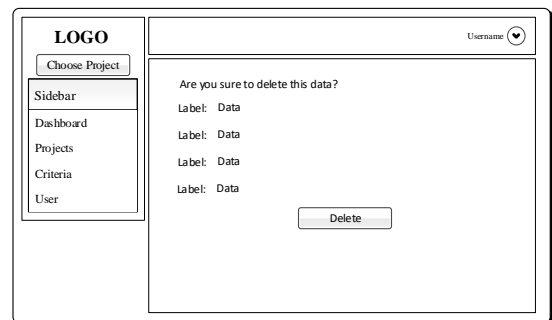
Gambar 14 Halaman Daftar Data

Halaman berikut merupakan bentuk umum yang digunakan ketika *user* melakukan manipulasi data berbentuk tambah atau ubah. Bentuk halaman ini digunakan pada menu *project*, *user* dan *task*.



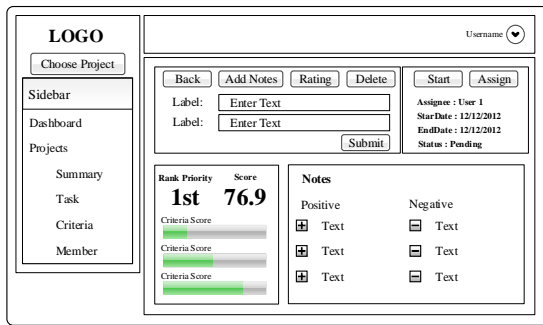
Gambar 15 Halaman Manipulasi Data Tambah atau Ubah

Antarmuka berikut akan menampilkan halaman untuk manipulasi data berbentuk hapus. Bentuk halaman ini digunakan pada menu *task* dan *user*.



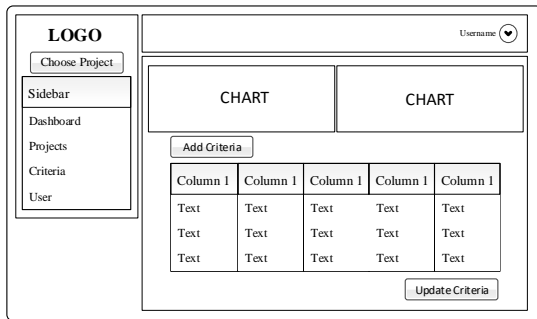
**Gambar 16** Halaman Manipulasi Data Hapus

Berikut merupakan antarmuka ubah *task* dimana menampilkan data *task*, nilai kinerja *task* dan catatan *task*. Pada antarmuka ini juga, *user* dapat melakukan aksi mengubah data, memberi nilai kinerja, menambah catatan *task*.



**Gambar 17** Halaman Ubah *Task*

Pada menu *criteria*, *user* dapat memanipulasi data seperti menambah, mengubah atau menghapus.



**Gambar 18** Halaman *Criteria*

## 4 IMPLEMENTASI DAN PEMBAHASAN

### 4.1 Implementasi

Tahap implementasi dilakukan setelah perancangan sistem telah selesai dilakukan. Implementasi yang dilakukan adalah pembangunan basis data sesuai dengan *entity relationship diagram*, membangun serta menerapkan antarmuka sistem dan diakhiri dengan penulis menguji sistem.

## 4.2 Implementasi Antarmuka

Berikut merupakan penerapan dari rancangan antarmuka:

### 4.2.1 Halaman *Login*

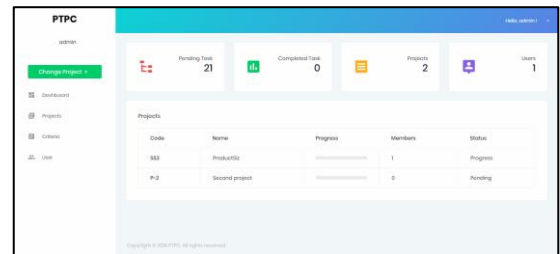
Sebelum menggunakan sistem, *user* perlu melakukan proses autentikasi dengan memasukkan *username* dan *password*.



**Gambar 19** Halaman *Login*

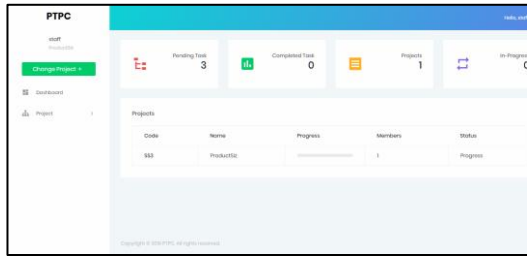
### 4.2.2 Halaman *Dashboard*

Berikut merupakan halaman yang ditampilkan setelah *user* berhasil melakukan proses *log in*. Pada halaman ini, sistem menampilkan data singkat seperti jumlah *task*, daftar *project* kepada *user* sesuai dengan *role* masing-masing.

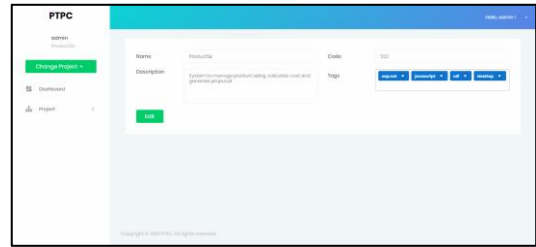


**Gambar 20** Halaman *Dashboard Manager*

Untuk *user* dengan *role staff*, halaman *dashboard* akan menampilkan data *task* dengan status *in-progress* melainkan data jumlah *user* yang ditemukan pada halaman *dashboard* untuk *role manager*.



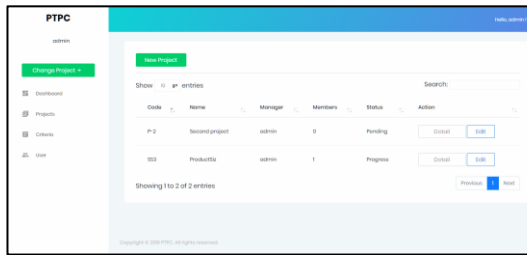
Gambar 21 Halaman Dashboard Staff



Gambar 24 Halaman Edit Project

#### 4.2.3 Halaman View Project

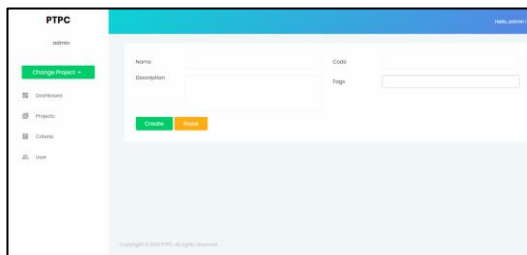
Halaman *view project* menampilkan daftar *project* yang telah diisi oleh *user*. *User* dapat mengakses halaman tambah, edit dan detail *project*. Halaman ini hanya dapat diakses oleh *user* dengan *role manager*.



Gambar 22 Halaman View Project

#### 4.2.4 Halaman Add Project

Halaman berikut digunakan untuk memasukkan data *project* baru.

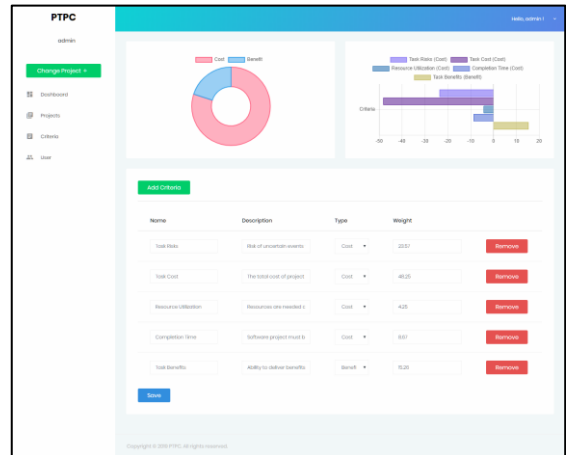


Gambar 23 Halaman Add Project

#### 4.2.5 Halaman Edit Project

Halaman *edit project* digunakan oleh *user* untuk mengubah data *project*.

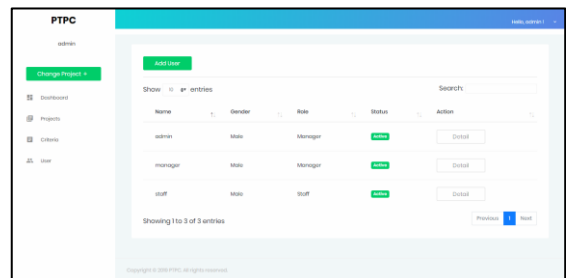
Halaman *criteria* terdiri dari 2 grafik yang menampilkan data dalam bentuk gambar yang disesuaikan dengan masukan data dari *user*. *User* dapat menambah, mengubah atau menghapus data *criteria*.



Gambar 25 Halaman Criteria

#### 4.2.7 Halaman View User

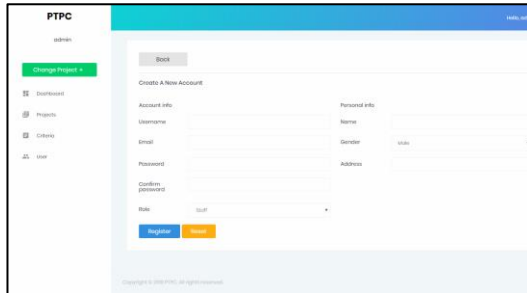
Halaman *view user* menampilkan semua *user* yang terdaftar dalam sistem. *User* dapat melihat data detail *user* dengan menekan tombol *detail* yang ditemukan di tiap baris data.



Gambar 26 Halaman View User

#### 4.2.8 Halaman Add User

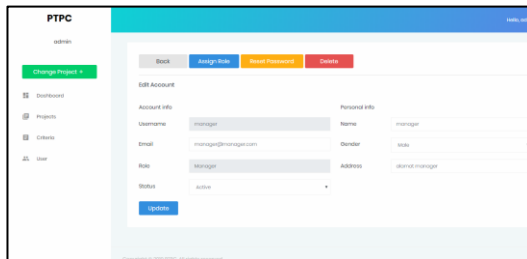
User dapat memasukkan user baru dengan mengisi form yang dapat diakses dengan menekan tombol *add user* pada halaman *view user*.



Gambar 27 Halaman Add User

#### 4.2.9 Halaman Edit User

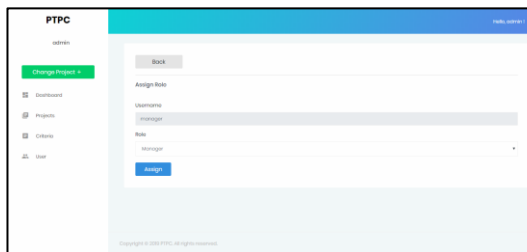
User dapat mengubah data user dengan menekan tombol *detail* pada halaman *view user*.



Gambar 28 Halaman Edit User

#### 4.2.10 Halaman Assign Role

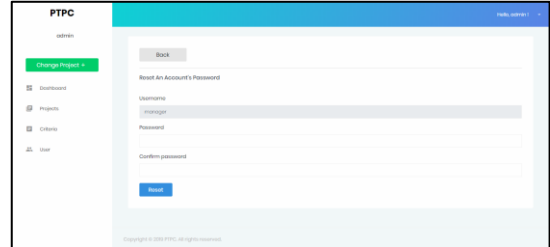
User dapat mengakses halaman *assign role* dengan menekan tombol *assign role* pada halaman *edit user*. Halaman ini berfungsi untuk mengubah hak akses user.



Gambar 29 Halaman Assign Role

#### 4.2.11 Halaman Reset Password

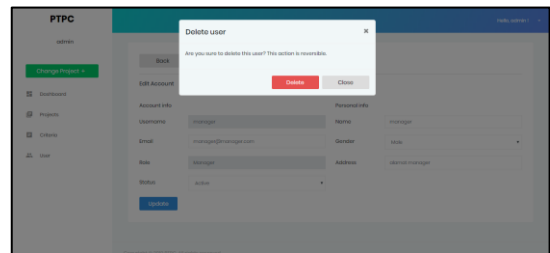
User dengan hak akses *manager* dapat melakukan *reset password* pada user lain yang lupa *password*.



Gambar 30 Halaman Reset Password

#### 4.2.12 Halaman Delete User

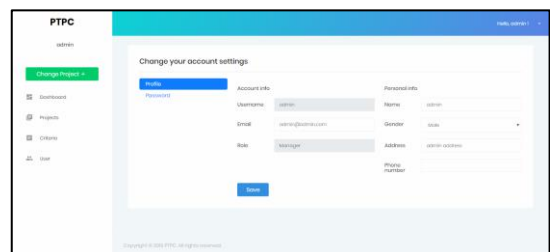
Halaman hapus user dapat diakses dengan menekan tombol *delete user* dimana sistem akan memberikan pesan yang perlu dikonfirmasi oleh user untuk melanjutkan proses hapus user.



Gambar 31 Halaman Delete User

#### 4.2.13 Halaman Manage Account

Untuk mengubah profil, user dapat mengakses halaman dengan memilih menu *manage account* yang dapat ditemukan pada menu *drop-down* diatas kanan setiap halaman.



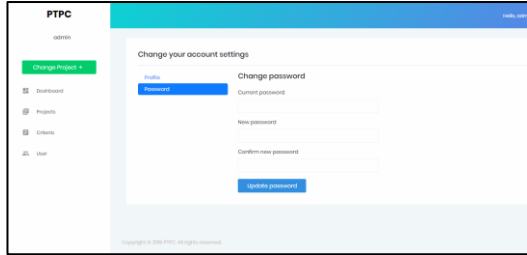
Gambar 32 Halaman Manage Account

#### 4.2.14 Halaman Change Password

Untuk mengubah profil, user dapat mengakses halaman dengan memilih menu *change password* yang



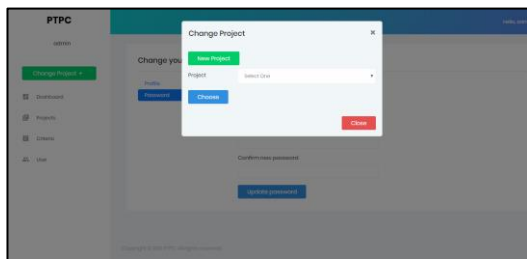
dapat ditemukan pada menu *drop-down* diatas kanan setiap halaman.



**Gambar 33** Halaman *Change Password*

#### 4.2.15 Halaman *Choose Project*

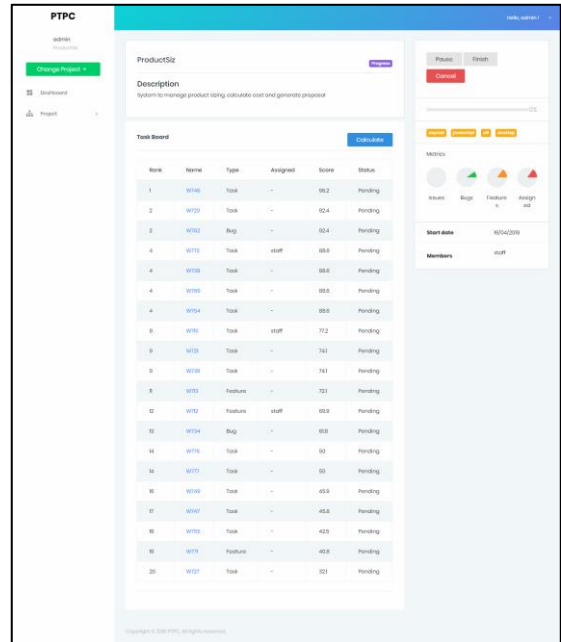
Halaman *choose project* berfungsi untuk menentukan *project* mana yang sedang diatur atau dikerjakan. Setelah memilih *project*, *user* akan diberi akses menu baru seperti menu *task*, *criteria* dan *member*.



**Gambar 34** Halaman *Choose Project*

#### 4.2.16 Halaman *Summary Project*

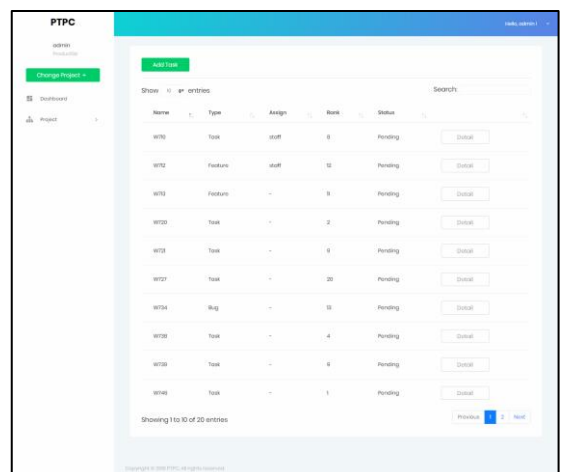
Halaman *summary project* menampilkan informasi tentang sebuah *project* secara pintas. Pada halaman *summary project*, *user* dapat melihat nama, deskripsi, daftar member dan status *project*. *User* juga dapat melihat daftar *task* yang belum dikerjakan serta grafik yang menunjukkan macam *task*. Pada halaman ini, *user* dapat mengubah status *project* dan menentukan peringkat *task* dengan menekan tombol *calculate*.



**Gambar 35** Halaman *Summary Project*

#### 4.2.17 Halaman *View Task*

Halaman *view task* hanya dapat diakses setelah memilih *project* di halaman *choose project*. Pada halaman ini, *user* dapat melihat daftar *task* dari *project* yang dipilih. Tombol *add task* untuk menambah *task* baru. Tombol *detail* untuk melihat detil atau mengubah data *task*.

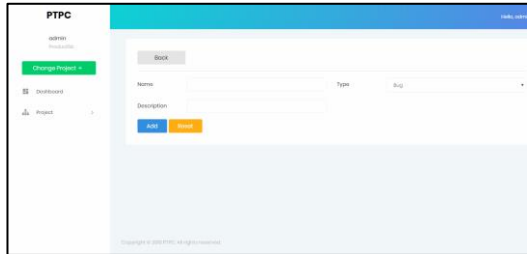


**Gambar 36** Halaman *View Task*

#### 4.2.18 Halaman *Add Task*

Halaman ini dapat diakses dengan menekan tombol *add task* pada

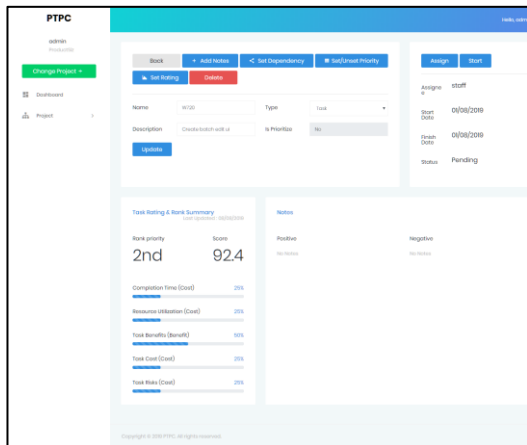
halaman *view task*. *User* perlu mengisi form dan menekan tombol *add* untuk menambah *task*.



Gambar 37 Halaman Add Task

#### 4.2.19 Halaman Edit Task

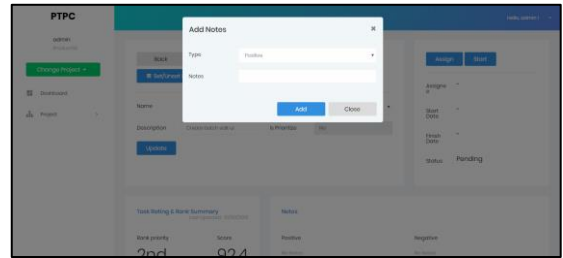
Berikut merupakan halaman yang menampilkan data *task* seperti peringat, catatan, *user* yang bertanggung jawab.



Gambar 38 Halaman Edit Task

#### 4.2.20 Halaman Add Notes

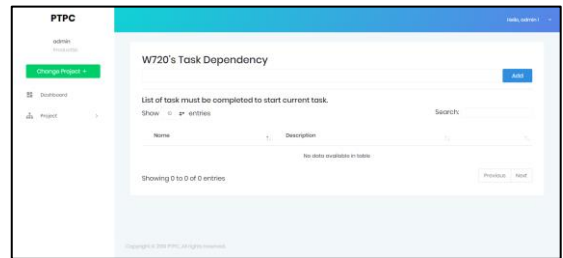
Halaman ini dapat diakses dengan menekan tombol *add notes* pada halaman *edit task*. Sistem akan menampilkan form yang perlu diisi oleh *user*.



Gambar 39 Halaman Add Notes

#### 4.2.21 Halaman Set Dependency

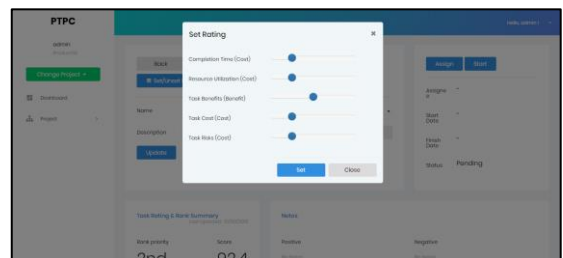
Halaman *set dependency* berfungsi untuk memasukkan data ketergantungan tugas. Halaman ini juga menampilkan daftar tugas yang perlu diselesaikan.



Gambar 40 Halaman Set Dependency

#### 4.2.22 Halaman Set Rating

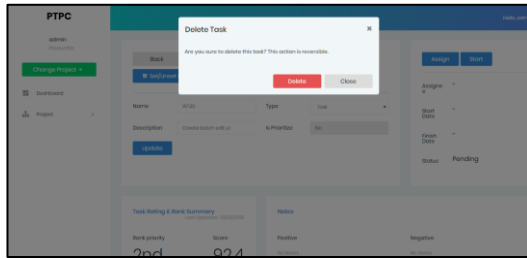
Halaman berikut berfungsi untuk memberikan nilai kinerja suatu *task* dalam skala satu sampai seratus pada tiap kriteria.



Gambar 41 Halaman Set Rating

#### 4.2.23 Halaman Delete Task

*User* dapat menghapus *task* dengan menekan tombol *delete* pada halaman *edit task* kemudian sistem akan menampilkan pesan konfirmasi yang perlu dijawab oleh *user*.



Gambar 42 Halaman Delete Task

### 4.3 Pembahasan

Sistem yang telah dibangun akan diuji. Pengujian dilakukan untuk memastikan seluruh fungsi pada sistem berjalan sesuai dengan desain yang telah dirancang sebelumnya. Pengujian yang dilakukan dengan memasukkan data dan membandingkan hasil dengan hasil yang diharapkan. Kriteria penilaian yang digunakan adalah *task risks (C1)*, *task cost (C2)*, *resource utilization (C3)*, *completion time (C4)* dan *task benefits (C5)*. Berikut adalah daftar tugas yang akan diurutkan:

Tabel 5 Matriks Keputusan

Kode	C1	C2	C3	C4	C5
W710	0.5	0.2 5	0.7 5	0.5	0.7 5
W712	0.2 5	0.5	0.5	0.2 5	0.7 5
W713	0.2 5	0.5	0.2 5	0.2 5	0.7 5
W720	0.2 5	0.2 5	0.2 5	0.2 5	0.5
W721	0.5	0.2 5	0.5	0.5	0.5
W727	0.7 5	0.7 5	0.7 5	0.7 5	0.2 5
W734	0.2 5	0.5	0.5	0.5	0.5
W738	0.2 5	0.2 5	0.2 5	0.2 5	0.2 5
W739	0.5	0.2 5	0.5	0.5	0.5

W746	0.2 5	0.2 5	0.2 5	0.2 5	0.7 5
W747	0.5	0.7 5	0.5	0.5	0.7 5
W749	0.7 5	0.5	0.7 5	0.2 5	0.2 5
W754	0.2 5	0.2 5	0.2 5	0.2 5	0.2 5
W755	0.5	0.7 5	0.5	0.2 5	0.2 5
W762	0.2 5	0.2 5	0.2 5	0.2 5	0.5
W769	0.2 5	0.2 5	0.2 5	0.2 5	0.2 5
W771	1	0.7 5	0.7 5	1	1
W772	0.2 5	0.2 5	0.2 5	0.2 5	0.2 5
W776	0.5	0.5	0.5	0.5	0.5
W777	0.5	0.5	0.5	0.5	0.5

Ada dua macam batasan yaitu prioritas yang ditentukan oleh *user* dimana urutan tidak dipengaruhi oleh nilai akhir dan ketergantungan merupakan hubungan dimana tugas bergantung dengan tugas lainnya yang harus diselesaikan sebelum dapat dilakukan, berikut merupakan batasan yang diisi oleh *user* sebelum melakukan kalkulasi:

Tabel 6 Batasan Tugas

Kode	Ketergantungan	Prioritas
W721	W720	-
W755	W754, W747	-
W747	W746	-
W771	-	2
W734	W727	1
W776	W771	-
W777	W771	-
W749	W747	-
W769	W747, W755	-

Berikut adalah tabel dengan hasil dari perhitungan manual dan hasil dari sistem:

**Tabel 7** Perbandingan Perhitungan Manual dan Sistem

Kode	Manual (Peringkat)	Sistem (Peringkat)
W710	77.2(8)	77.2(8)
W712	69.9(12)	69.9(12)
W713	72.1(11)	72.1(11)
W720	92.4(2)	92.4(2)
W721	74.1(9)	74.1(9)
W727	32.1(20)	32.1(20)
W734	61.8(13)	61.8(13)
W738	88.5(4)	88.5(4)
W739	74.1(9)	74.1(9)
W746	96.2(1)	96.2(1)
W747	45.7(17)	45.7(17)
W749	45.9(16)	45.9(16)
W754	88.5(4)	88.5(4)
W755	42.5(18)	42.5(18)
W762	92.4(2)	92.4(2)
W769	88.5(4)	88.5(4)
W771	40.8(19)	40.8(19)
W772	88.5(4)	88.5(4)
W776	50(14)	50(14)
W777	50(14)	50(14)

Pada tabel 7 dapat disimpulkan bahwa sistem berjalan sesuai dengan rancangan hasil yang dikeluarkan karena dari hasil perhitungan manual dan hasil dari sistem mengeluarkan peringkat alternatif yang sama.

Tabel berikut merupakan hasil urutan prioritas yang dihasilkan oleh sistem pendukung keputusan setelah menerapkan batasan tugas yang kemudian dibandingkan dengan realisasi pengerjaan tugas:

**Tabel 8** Perbandingan Hasil Sistem dan Realisasi

Kode	Nilai(Peringkat)	Peringkat Realisasi
W727	32.1(1)	1
W734	61.8(2)	2
W771	40.8(3)	3
W746	96.2(4)	4
W720	92.4(5)	8
W762	92.4(5)	8
W747	45.8(7)	6
W754	88.6(8)	4
W755	42.5(9)	6
W772	88.6(10)	10
W738	88.6(10)	10
W769	88.6(10)	10
W710	77.2(13)	13
W721	74.1(14)	14
W739	74.1(14)	14
W713	72.1(16)	16
W712	69.9(17)	16
W776	50(18)	18
W777	50(18)	18
W749	45.9(20)	20

Kesimpulan yang dapat diambil dari hasil perbandingan adalah sistem pendukung keputusan dapat

menghasilkan informasi berupa urutan tugas sesuai prioritas dengan tingkat akurasi sebesar 75% atau 15 dari 20 alternatif.

## 5 KESIMPULAN DAN SARAN

### 5.1 Kesimpulan

Berdasarkan hasil dari analisa yang telah diuraikan sebelumnya dan sistem yang sudah dikembangkan, maka penulis dapat mengambil beberapa kesimpulan sebagai berikut:

1. Sistem pendukung keputusan dapat memberikan informasi berupa daftar peringkat tugas yang perlu dikerjakan sebagai referensi untuk menentukan prioritas tugas yang dikerjakan lebih dahulu.
2. Metode *simple additive weighting* dengan menggunakan masukan berupa penilaian di tiap kriteria dapat berjalan dengan baik sesuai rancangan.
3. Sistem pendukung keputusan dengan metode *simple additive weighting* menggunakan *task benefit* dengan bobot 15.26% sebagai kriteria keuntungan(*benefit*) dan *task risk*, *task cost*, *resource utilization* dan *completion time* dengan bobot masing-masing sebesar 23.57%, 48.25%, 4.25% dan 8.67% sebagai kriteria biaya(*cost*) dapat menentukan urutan prioritas tugas pada pengembangan aplikasi dengan tingkat akurasi sebesar 75%.

### 5.2 Saran

Dalam pembuatan sistem, adapun saran yang dapat diterapkan pada sistem yaitu:

1. Membuat fitur untuk mengatur klasifikasi penilaian pada tiap kriteria.
2. Membuat fitur notifikasi ketika status tugas diserahkan ke *user* atau ketika tugas telah selesai.

3. Mengembangkan sistem yang dapat merekam riwayat perubahan yang terjadi pada data sehingga proses audit dapat dilakukan ketika terjadinya kerusakan atau kesalahan pada data.
4. Memberi informasi tambahan seperti deskripsi tugas, penilaian dan pengaturan prioritas pada halaman *summary* dalam bentuk yang lebih interaktif seperti *pop up* tanpa harus membuka halaman lainnya.
5. Penelitian lebih lanjut pada kriteria yang digunakan atau metode lainnya untuk menentukan urutan prioritas tugas pada pengembangan aplikasi.

## DAFTAR PUSTAKA

- Akhirina, T. Y. (2016). Komparasi Metode Simple Additive Weighting dan Profile Matching pada Pemilihan Mitra Jasa Pengiriman Barang. *Edukasi Dan Penelitian Informatika (JEPIN)*, 2(1), 27–33.
- Alannita, N. P., & Suaryana, I. G. N. A. (2014). Pengaruh Kecanggihan Teknologi Informasi, Partisipasi Manajemen, dan Kemampuan Teknik Pemakai Sistem Informasi Akuntansi Pada Kinerja Individu. *E-Jurnal Akuntansi Universitas Udayana*, 1(6.1), 33–45.
- Andalia, F., & Setiawan, E. B. (2015). Pengembangan Sistem Informasi Pengolahan Data Pencari Kerja Pada Dinas Sosial Dan Tenaga Kerja Kota Padang. *Ilmiah Komputer Dan Informatika (KOMPUTA)*, 4(2), 93–98.
- Ariansyah, Fajriyah, & Prasetyo, F. S. (2017). Rancang Bangun Sistem Informasi Pendataan Alumni Pada STIE Prabumulih Berbasis Website dengan Menggunakan Bootstrap. *Manajemen Dan Informatika Pelita Nusantara*, 1(2), 26–30.

- Dabbagh, M., & Lee, S. P. (2014). An Approach for Integrating the Prioritization of Functional and Nonfunctional Requirements. *The Scientific World*.
- Efendi, R., Ernawati, & Hidayati, R. (2014). Aplikasi Fuzzy Database Model Tahani dalam Memberikan Rekomendasi Pembelian Rumah Berbasis Web. *Pseudocode*, 1(1), 32–43.
- Galloway, J., Wilson, B., Allen, K. S., & Matson, D. (2014). *Professional ASP.NET MVC 5*.
- Hidayat, R., Marlina, S., & Utami, L. D. (2017). Perancangan Sistem Informasi Penjualan Barang Handmade Berbasis Website Dengan Metode Waterfall. *Simposium Nasional Ilmu Pengetahuan Dan Teknologi (SIMNASIPTEK)*, 175–183.
- Hoda, R., & Murugesan, L. K. (2016). Multi-Level Agile Project Management Challenges: A Self-Organizing Team Perspective The Journal of Systems and Software Multi-level agile project management challenges: A self-organizing team perspective. *The Journal of Systems & Software*, 117, 245–257.
- Iriadi, N., & Indrasari, A. U. (2017). Sistem Informasi Penjualan Berbasis Web Pada Cv . Bambu Jaya Jakarta. *Speed - Sentra Penelitian Engineering Dan Edukasi*, 9(3), 34–39.
- Irviani, R., & Oktaviana, R. (2017). Aplikasi Perpustakaan pada SMA N1 Kelumbayan Barat menggunakan Visual Basic. *Technology Acceptance Model*, 8(1), 63–69.
- Iswandy, E. (2015). Sistem Penunjang Keputusan untuk Menentukan Penerimaan Dana Santunan Sosial Anak Nagari dan Penyalurannya Bagi Mahasiswa dan Pelajar Kurang Mampu Di Kenagarian Barung–Barung Balantai Timur. *Teknoif*, 3(2), 70–79.
- Kaur, B., & Bhatia, R. (2015). Prioritizing Parameters for Software Project Selection using Analytical Hierarchical Process. *International Journal of Computer Applications*, 118(3), 36–40.
- Lenti, F. N. (2014). Rekayasa Database Terdistribusi pada Layanan Pemesanan Tiket Pesawat Terbang. *Jurnal Teknologi Technoscientia*, 6(2), 129–134.
- Masykur, F., & Atmaja, I. M. P. (2015). Sistem Administrasi Pengelolaan Arsip Surat Masuk dan Surat Keluar Berbasis Web. *Indonesian Journal on Networking and Security*, 4(3), 1–7.
- Matharu, G. S., Mishra, A., Singh, H., & Upadhyay, P. (2015). *Empirical Study of Agile Software Development Methodologies: A Comparative Analysis*. 40(1), 1–6.
- Muhammad, M., Safriadi, N., & Prihartini, N. (2017). Implementasi Metode Simple Additive Weighting (SAW) pada Sistem Pendukung Keputusan dalam Menentukan Prioritas Perbaikan Jalan. *Sistem Dan Teknologi Informasi*, 5(4), 223–228.
- Munro, J. (2015). *ASP.NET MVC 5 with Bootstrap Knockout.js: Building Dynamic Responsive Web Applications*. O'Reilly Media, Inc.
- Nugrahanti, F. (2015). *Perancangan Sistem Informasi Inventory Sparepart Mesin Fotocopy dengan Menggunakan Visual Delphi 7. 9*.
- Nugroho, B., Fitriasih, S. H., & Widada, B. (2015). Sistem Informasi Rekam Medis di Puskesmas Masaran I Sragen. *Teknologi Informasi Dan Komunikasi Sinar Nusantara*, 3(2), 49–56.
- Nugroho, R. F., Riza, T. A., & Hariyani, Y. S. (2016). Perancangan dan Implementasi Sistem Reservasi Servis Mobil Berbasis Website

- Studi Kasus Di Bengkel Mobil Pandawa 5 Motosport. *E-Proceeding of Applied Science*, 2(3), 1390–1396
- Nurdiyanto, H., & Meilia, H. (2016). Sistem Pendukung Keputusan Penentuan Prioritas Pengembangan Industri Kecil dan Menengah di Lampung Tengah Menggunakan Analitical Hierarchy Process (AHP). *Seminar Nasional Teknologi Informasi Dan Multimedia*, 6–7.
- Olsson, M. (2014). *CSS Quick Syntax Reference*.
- Saputra, A. Y. (2015). Sistem Pendukung Keputusan Penentuan Nomor Urut Caleg dengan Metode SAW. *Citec*, 2(2), 93–101.
- Septa, R., Hamzah, A., & Andayati, D. (2014). Sistem Informasi Pengolahan Data IKPM (Ikatan Keluarga Pelajar Mahasiswa) Muara Enim Berbasis Web Menggunakan Php dan My SQL. *SCRIPT*, 2(1), 59–68.
- Singh, R. R. (2015). Mastering Entity Framework. In *Packt Publishing*.
- Spaanjaars, I. (2014). *Beginning ASP.NET 4.5.1 in C# and VB*. John Wiley & Sons.
- Suripto, M. A. S., & Triyono, R. A. (2014). Pembangunan Sistem Informasi Akta Kelahiran Dinas Kependudukan dan Pencatatan Sipil Sragen. *Indonesian Journal on Networking and Security*, 3(3), 33–40.
- Susila, C. B., & Triyono, R. A. (2015). Sistem Informasi Nilai Mahasiswa Berbasis SMS Gateway pada Sekolah Tinggi Ilmu Tarbiyah (STIT) Muhammadiyah Pacitan. *Speed - Sentra Penelitian Engineering Dan Edukasi*, 7(3), 30–37.
- Syukroon, A., & Hasan, N. (2015). Perancangan Sistem Informasi Rawat Jalan Berbasis Web Pada Puskesmas Winong. *Biaglala Informatika*, 3(1), 1–9.s
- Wahyudi, K., Dewi, L. P., & Wibowo, A. (2017). Sistem Informasi Manajemen Pekerjaan Panel Listrik Berbasis Web di PT. Yoel Tricitra. *Infra*.