

BAB II 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.

Tabel 1 Tinjauan Pustaka

Peneliti	Tahun	Kesimpulan Penelitian
Muhammad, Safriadi & Prihartini	2017	Sistem pendukung keputusan dengan metode SAW dapat memberi rekomendasi dalam perbaikan jalan.

Saputra	2015	Informasi yang dihasilkan oleh sistem pendukung keputusan dengan metode SAW dapat digunakan sebagai acuan dalam pengambilan keputusan nomor urut caleg.
Akhirina	2016	Pemilihan mitra jasa pengiriman barang menggunakan metode SAW dan <i>Profile Matching</i> dengan penilaian dan data yang sama menghasilkan alternatif terbaik yang sama juga.
Kaur & Bhatia	2015	Pemilihan proyek perangkat lunak menggunakan 5 parameter yaitu <i>project risks</i> , <i>project benefits</i> , <i>cost</i> , <i>resource utilization</i> and <i>completion time</i> dengan <i>priority vector</i> masing-masing 23.57%, 15.22%, 48.25%, 4.25% dan 8.67%.
Hoda & Murugesan	2016	<i>Task dependencies</i> merupakan salah satu tantangan dimana tugas memiliki ketergantungan yang harus diselesaikan agar tugas dapat dimulai jika dihiraukan dapat menyebabkan penurunan kinerja dan kecepatan pengerjaan proyek serta 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 (Aakhirina, 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 (Aakhirina, 2016), untuk memproses 2 kriteria pada SAW perlu menggunakan formulasi berikut:

$$r_{ij} = \begin{cases} \frac{x_{ij}}{\text{Max}_i x_{ij}} & \text{Jika } j \text{ adalah atribut keuntungan (benefit)} \\ \frac{\text{Min}_i x_{ij}}{x_{ij}} & \text{Jika } j \text{ adalah atribut biaya (cost)} \end{cases} \dots\dots\dots (1)$$

dimana:

- r_{ij} rating kinerja ternormalisasi
- X_{ij} rating alternatif A_i di atribut C_j
- Max_i nilai maksimum dari tiap baris
- 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 *Model-View-Controller* 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 detil 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*.