



BAB II
TINJAUAN PUSTAKA

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terkait

Untuk mendukung keberhasilan penelitian ini, penyusun melakukan pendekatan teoritis melalui beberapa literatur yang berhubungan dengan penelitian yang dilakukan. Beberapa uraian penelitian terdahulu yang menjadi acuan dalam penelitian ini yaitu :

1. Adam Hasbi Nurgroho, 2016, dengan judul “Sistem Pendukung Keputusan Penentuan Nilai Kenaikan Pangkat TNI Angkatan Darat Menggunakan Metode *Composite Performance Index* (CPI) (Jurnal Matematika, Volume 19, Nomor 3, Desember 2016)” Penelitian ini menghasilkan suatu sistem pendukung keputusan yang mampu menampilkan hasil keputusan dari nilai kenaikan pangkat TNI Angkatan Darat.
2. Renny Noer Fajarini, 2016, dengan judul “Sistem Pendukung Keputusan Menggunakan Metode *Composite Performance Index* Untuk Pemilihan Pemenang Tender Pengadaan Barang/Jasa (Jurnal Komputer, 2016)” Penelitian ini menghasilkan sebuah sistem pendukung keputusan yang mampu menampilkan hasil dari keputusan untuk pemilihan pemenang tender pengadaan barang/jasa.

3. Magdalena Karismariyanti, 2015, dengan judul “Simulasi Pendukung Keputusan Penerima Beasiswa Menggunakan Metode *Composite Performance Index* (Jurnal Teknologi Informasi, Volume 1, Nomor 2, November 2015)” Penelitian ini menghasilkan sebuah sistem pendukung keputusan yang mampu menampilkan hasil keputusan penerima beasiswa.
4. Charisma Putri Arlida, 2017, dengan judul “Sistem Pendukung Keputusan Untuk Menentukan Jenis Bibit Tanaman pada Penjualan Online menggunakan Metode *Composite Performance Index* (CPI) (Konferensi Nasional ICT-M Politeknik Telkom (KNIP) 2017)” Penelitian ini menghasilkan sebuah sistem pendukung keputusan yang mampu menampilkan hasil keputusan menentukan jenis bibit tanaman.
5. Andi Anto Tri Susilo, 2017, dengan judul “Penerapan Metode CPI Pada Pemilihan Hotel Di Kota Lubulinggau”. Penelitian ini menghasilkan sebuah sistem informasi mengenai peringkat hotel yang dapat dijadikan tempat referensi untuk tetap memperhatikan beberapa kriteria, termasuk tarif kamar, jarak ke pusat kotam, fasilitas dan layanan.
6. Yulia Purba, 2020, dengan judul “Sistem Pendukung Keputusan Penentuan Kelayakan Penggunaan Alat Berat Menggunakan Metode *Composite Performance Index* Pada PT. Utama Karya Infrastruktur”. Penelitian ini menghasilkan sebuah sistem informasi mengenai penentuan kelayakan penggunaan alat berat dengan sistem perankingan dari objek yang diteliti sehingga perusahaan dapat menentukan dengan cara yang efektif dan efisien.

II.2. Landasan Teoritis

II.2.1. Sistem Pendukung Keputusan

Sistem pendukung keputusan adalah sistem berbasis komputer interaktif yang membantu pengguna dalam penilaian dan pemilihan. Sistem tidak hanya menyediakan penyimpanan dan pengambilan data tapi juga meningkatkan akses informasi tradisional dengan dukungan untuk pembuatan model pengambilan keputusan dan penalaran berbasis model (Roger & Marek, 2007).

Konsep Sistem Pendukung Keputusan atau *Decision Support System* pertama kali diungkapkan pada awal tahun 1970-an oleh Michael S. Scott Morton dengan istilah *Management Decision System*. Definisi dari Sistem Pendukung Keputusan adalah suatu sistem berbasis komputer yang ditujukan untuk membantu pengambilan keputusan dalam memanfaatkan data dan model tertentu untuk memecahkan berbagai persoalan yang tidak terstruktur (Daihani, 2001).

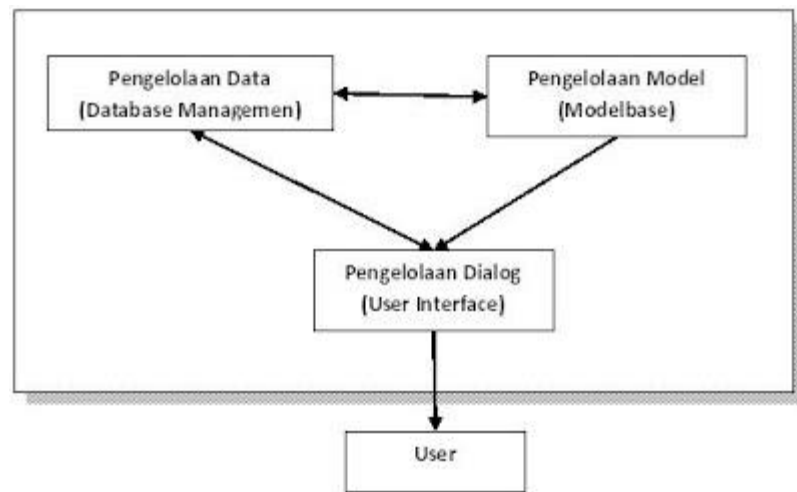
Sistem Pendukung Keputusan mengacu pada suatu sistem yang memanfaatkan dukungan komputer dalam proses pengambilan keputusan. Hal ini dikemukakan oleh beberapa ahli, diantaranya Little Man dan Watson memberi definisi bahwa Sistem Pendukung Keputusan merupakan suatu sistem yang interaktif, yang membantu pengambil keputusan melalui penggunaan data dan model-model keputusan untuk memecahkan masalah-masalah yang sifatnya semi terstruktur dan tidak terstruktur (Suryadi, 2001).

Dari berbagai definisi di atas dapat dikatakan bahwa Sistem Pendukung Keputusan adalah suatu sistem informasi spesifik yang ditujukan untuk

membantu manajemen dalam mengambil keputusan yang berkaitan dengan persoalan yang bersifat semi terstruktur (hukum sebab-akibat dari adanya suatu variabel belum pasti/bukan suatu rutinitas). Sistem ini mampu menghasilkan berbagai alternatif yang secara interaktif dapat digunakan oleh pemakai (Davis,1999).

Sistem Pendukung Keputusan (SPK) atau *Decision Support System* (DSS) mulai dikembangkan pada tahun 1970-an oleh Michael S.Cott Morton dengan istilah *Management Decision System* (Turban dkk, 2005).Sistem tersebut adalah sistem berbasis komputer yang ditujukan untuk membantu pengambilan keputusan dalam memanfaatkan data dan model tertentu untuk memecahkan berbagai persoalan yang tidak terstruktur. Menurut Kendal, 2002, *Decision Support System* (DSS) atau sistem pendukung keputusan hampir sama dengan sistem informasi manajemen tradisional karena keduanya sama-sama tergantung pada basis data sebagai sumber data dimana DSS menekankan pada fungsi pendukung pembuatan keputusan diseluruh tahap-tahapnya, walaupun keputusan aktual masih tetap wewenang eksekutif sebagai pembuat keputusan.

Secara umum Sistem Pendukung Keputusan dibangun oleh tiga komponen besar yaitu *database Management*, *Model Base* dan *Software System/User Interface*. Komponen SPK tersebut dapat digambarkan seperti gambar di bawah ini.



Gambar II. 1 Komponen Sistem Pendukung Keputusan (SPK)

- a. *Database Management* merupakan subsistem data yang terorganisasi dalam suatu basis data. Data yang merupakan suatu sistem pendukung keputusan dapat berasal dari luar maupun dalam lingkungan. Untuk keperluan SPK, diperlukan data yang relevan dengan permasalahan yang hendak dipecahkan melalui simulasi.
- b. *Model Base* merupakan suatu model yang merepresentasikan permasalahan kedalam format kuantitatif (model matematika sebagai contohnya) sebagai dasar simulasi atau pengambilan keputusan, termasuk didalamnya tujuan dari permasalahan (objektif), komponen-komponen terkait, batasan-batasan yang ada (*constraints*), dan hal-hal terkait lainnya. *Model Base* memungkinkan pengambil keputusan menganalisa secara utuh dengan mengembangkan dan membandingkan solusi alternatif.
- c. *User Interface / Pengelolaan Dialog* Terkadang disebut sebagai subsistem dialog, merupakan penggabungan antara dua komponen sebelumnya yaitu

Database Management dan *Model Base* yang disatukan dalam komponen ketiga (*user interface*), setelah sebelumnya dipresentasikan dalam bentuk model yang dimengerti computer. *User Interface* menampilkan keluaran sistem bagi pemakai dan menerima masukan dari pemakai kedalam Sistem Pendukung Keputusan

II.2.2. Composite Performance Index

Composite Performance Index merupakan indeks gabungan atau *composite index* yang dapat digunakan untuk menentukan penilaian atau peringkat dari berbagai alternatif (i) berdasarkan beberapa kriteria (j) (Nur S Tanjung, 2018).

Penyelesaian CPI dalam mencari alternatif yang layak, yaitu:

1. Identifikasi kriteria tren positif (semakin tinggi nilainya semakin baik) dan tren negatif (semakin rendah nilainya semakin buruk).
2. Untuk kriteria tren positif, nilai minimum pada setiap kriteria ditransformasi ke seratus, sedangkan nilai lainnya ditransformasi secara profesional lebih tinggi.
3. Untuk kriteria tren negatif, nilai minimum pada setiap kriteria ditransformasi ke seratus, sedangkan nilai lainnya ditransformasi secara profesional lebih rendah.

Index yang digunakan untuk menentukan penilaian atau peringkat dari berbagai alternatif keputusan berdasarkan beberapa kriteria dari setiap alternatif, dirumuskan sebagai berikut :

1. Normalisasi Matrix

Normalisasi matrix dilakukan dengan menyesuaikan Nilai Kriteria dengan nilai Bobot yang sudah ditentukan.

2. Menentukan Nilai Kriteria :

$$A_{ij} = X_{ij} / X_{ij(\min)} * 100 \quad ; \quad i = 1, 2, \dots, n \text{ dan } j = 1, 2, \dots, n$$

3. Menentukan Nilai CPI :

$$I_i = \sum_{j=1}^n A_{ij} B_j \quad ; \quad i = 1, 2, \dots, n \text{ dan } j = 1, 2, \dots, n$$

$$j=i$$

Dimana :

A_{ij} = Nilai alternatif ke i pada kriteria ke j

X_{ij} = Nilai awal alternatif ke i pada kriteria ke j

$X_{ij(\min)}$ = Nilai alternatif ke i pada kriteria minimum ke j

B_j = Bobot kepentingan kriteria ke j

I_i = indeks gabungan kriteria pada alternatif ke-i

II.2.3.UML

Unified Modeling Language (UML) adalah bahasa spesifik standar yang dipergunakan untuk mendokumentasikan, menspesifikan dan membangun perangkat lunak. UML merupakan metodologi dalam mengembangkan sistem

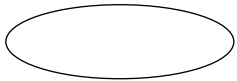
berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. UML saat ini sangat banyak digunakan dalam dunia industri yang merupakan standar bahasa pemodelan umum dalam industri perangkat lunak dan pengembangan sistem. Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML yaitu *Use Case Diagram*, *Activity Diagram*, *Class Diagram* dan *Sequence Diagram*. (Urva dan Siregar, 2012 : 95).

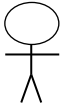


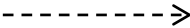
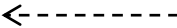
Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut:

1. *Use case Diagram*

Use case diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Dapat dikatakan *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Simbol-simbol yang digunakan dalam *use case diagram* dapat dilihat pada tabel II.1 dibawah ini :

Tabel II.1. Simbol *Use Case*

Gambar	Keterangan
	<p><i>Use case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama <i>use case</i>.</p>




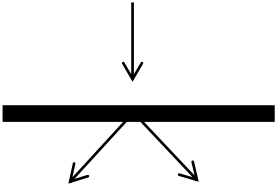
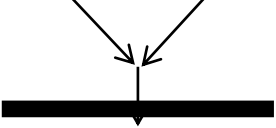
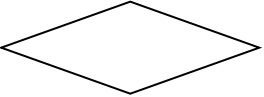

	<p>Aktor adalah <i>abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan <i>use case</i>, tetapi tidak memiliki control terhadap <i>use case</i>.</p>
	<p>Asosiasi antara aktor dan <i>use case</i>, digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan aliran data.</p>
	<p>Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem.</p>
	<p><i>Include</i>, merupakan di dalam <i>use case</i> lain (<i>required</i>) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya adalah pemanggilan sebuah fungsi program.</p>
	<p><i>Extend</i>, merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.</p>

(Sumber : Gellysa Urva dan Helmi Fauzi Siregar : 2015: 94)

2. Diagram Aktivitas (*Activity Diagram*)

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity diagram* dapat dilihat pada tabel II.2:

Tabel II.2. Simbol *Activity Diagram*

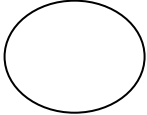
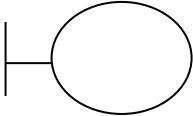
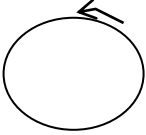
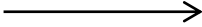
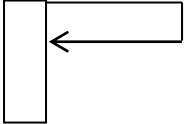
Gambar	Keterangan
	<p><i>Start point</i>, diletakkan pada pojok kiri atas dan merupakan awal aktifitas.</p>
	<p><i>End point</i>, akhir aktifitas.</p>
	<p><i>Activites</i>, menggambarkan suatu proses/kegiatan bisnis.</p>
	<p><i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.</p>
	<p><i>Join</i> (penggabungan) atau rake, digunakan untuk menunjukkan adanya dekomposisi.</p>
	<p><i>Decision Points</i>, menggambarkan pilihan untuk pengambilan keputusan, <i>true</i>, <i>false</i>.</p>
	<p><i>Swimlane</i>, pembagian <i>activity</i> diagram untuk menunjukkan siapa melakukan apa.</p>

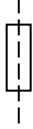

(Sumber : Gellysa Urva dan Helmi Fauzi Siregar : 2015: 94)

3. Diagram Urutan (*Sequence Diagram*)

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam *sequence diagram* dapat dilihat pada tabel II.3:

Tabel II.3. Simbol *Sequence Diagram*

Gambar	Keterangan
	<p><i>Entity Class</i>, merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.</p>
	<p><i>Boundary Class</i>, berisi kumpulan kelas yang menjadi <i>interface</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan formentry dan <i>form</i> cetak.</p>
	<p><i>Control class</i>, suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.</p>
	<p><i>Message</i>, simbol mengirim pesan antar <i>class</i>.</p>
	<p><i>Recursive</i>, menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.</p>






	<p><i>Activation, activation</i> mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.</p>
	<p><i>Lifeline</i>, garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i>.</p>


(Sumber : Gellysa Urva dan Helmi Fauzi Siregar:2015: 95)

4. Diagram Kelas (*Class Diagram*)

Class diagram juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan. *Class diagram* secara khas meliputi: Kelas (*Class*), Relasi, *Associations*, *Generalization* dan *Aggregation*, Atribut (*Attributes*), Operasi (*Operations/ Method*), *Visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *multiplicity* atau kardinaliti (Ade Hendini, 2016 : 111). Simbol *class diagram* dan *multiplicity class diagram* dapat dilihat pada Tabel II.4 dan Tabel II.5. dibawah ini:

Tabel II.4. Simbol *Class Diagram*

Gambar	Keterangan			
	<p><i>Generalization</i>, untuk menghubungkan antar kelas dengan arti umum-khusus. Jadi jika ada kelas dengan arti umum-khusus. Jadi jika ada kelas bermakna umum dan kelas bermakna khusus dapat menggunakan simbol ini.</p>			
<table border="1" data-bbox="336 842 553 1057"> <tr> <td data-bbox="336 842 553 913">Nama_kelas</td> </tr> <tr> <td data-bbox="336 913 553 985">+atribut</td> </tr> <tr> <td data-bbox="336 985 553 1057">+operasi</td> </tr> </table>	Nama_kelas	+atribut	+operasi	<p><i>Class</i>, untuk sebuah kelas pada struktur sistem. Penulisan tidak boleh menggunakan spasi. Simbol ini memiliki 3 susunan, yaitu kotak pertama adalah nama kelas, kedua atribut dan ketiga operasi.</p>
Nama_kelas				
+atribut				
+operasi				
	<p><i>Interface</i>, untuk simbol <i>interface</i> atau dalam bahasa indonesianya antar muka. Konsep yang digunakan pun sama dengan pemrograman berorientasi object (OOP).</p>			
	<p><i>Association</i>, digunakan untuk menghubungkan atau merelasikan kelas satu dengan kelas yang lainnya dengan makna umum.</p>			
	<p><i>Directed Association</i>, adalah relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain.</p>			
	<p><i>Aggregation</i>, adalah relasi antar kelas dengan makna semua bagian.</p>			

	<p><i>Dependency</i>, adalah relasi antar kelas dengan makna kebergantungan antar kelas.</p>
---	--

(Sumber : Gellysa Urva dan Helmi Fauzi Siregar : 2015 : 95)

Tabel II.5. Multiplicity Class Diagram

Multiplicity	Penjelasan
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih
1..*	1 atau lebih
0..1	Boleh tidak ada, maksimal 1
n..n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimum 4

(Sumber : Gellysa Urva dan Helmi Fauzi Siregar : 2015 : 95)

II.2.4 Hypertext Preprocessor(PHP)

PHP adalah bahasa pemrograman server-side yang dirancang untuk perkembangan web. PHP dikatakan server-side karena program yang diberikan akan dijalankan/diproses pada komputer yang bertindak sebagai server. PHP yang merupakan sebuah bahasa scripting yang terpasang pada HTML (*Hyper Text Model Language*). Menurut Nugroho (2006 : 61) “PHP atau singkatan dari Personal Home Page merupakan bahasa skrip yang tertanam dalam HTML untuk dieksekusi bersifat server side”.

II.2.5. MySQL

MySQL adalah *Relation Database Management System* (RDBMS) yang didistribusikan secara gratis di bawah lisensi GPL (*General Public License*). MySQL merupakan turunan dari salah satu konsep utama dalam *database* sejak lama, yaitu SQL (*Structure Query Language*). SQL merupakan salah satu konsep pengoperasian *database*, terutama sebagai seleksi dan pemasukan data, yang memungkinkan pengoperasian data yang dikerjakan dengan mudah secara otomatis. (Inayah, dkk, 2012 : 39)

