

## **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 Lubulngau”. Penelitian ini menghasilkan sebuah sistem informasi mengenai peringkat hotel yang dapat dijadikan tempat refrensi untuk tetap memperhatikan beberapa kriteria, termasuk tarif kamar, jarak ke pusat kotam, fasilitas dan layanan.

## **II.2. Landasan Teoritis**

### **II.2.1 Sistem**

Menurut Gaol (2008:9) mendefinisikan bahwa “sistem adalah hubungan satu unit dengan unit-unit lainnya yang saling berhubungan satu sama lainnya dan yang tidak dapat dipisahkan serta menuju suatu kesatuan dalam rangka mencapai tujuan yang telah ditetapkan”.

## II.2.2 Informasi

Menurut Laudon dan Laudonetal (2012:46),Informasia adalah data yangtelah dibuat ke dalam bentuk yang memiliki arti dan berguna bagi manusia. Sedangkan menurut O'Brien dan Marakas (2008:32), Informasi adalah data yang telah diubah menjadi kontek syang berarti dan berguna bagi para pemakaia khir tertentu. Sementara dari bukunya Stair dan Reynolds (2010:5) mendefinisikan Informasi sebagai kumpulan fakta yang terorganisir sehingga memiliki nilai tambah selain nilai fakta individu.

## II.2.3. Sistem Pendukung Keputusan

Sistem Pendukung Keputusan (SPK) atau *Decision Support System* (DSS) mulai dikembangkan pada tahun 1970-an oleh *Michael S.CottMorton* 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 bebagai 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.

#### **II.2.4. Aplikasi**

Secara istilah pengertian aplikasi adalah suatu program yang siap untuk digunakan yang dibuat untuk melaksanakan suatu fungsi bagi pengguna jasa aplikasi serta penggunaan aplikasi lain yang dapat digunakan oleh suatu sasaran yang akan dituju. Menurut kamus computer eksekutif, aplikasi mempunyai arti yaitu pemecahan masalah yang menggunakan salah satu tehnik pemrosesan data aplikasi yang biasanya berpacu pada sebuah komputansi yang diinginkan atau diharapkan maupun pemrosesan data yang di harapkan. Pengertian aplikasi menurut Kamus Besar Bahasa Indonesia, “Aplikasi adalah penerapan dari rancang sistem untuk mengolah data yang menggunakan aturan atau ketentuan bahasa pemrograman tertentu.

#### **II.2.5. Website**

Dimuat dalam jurnal Guntur Wibisono, Wahyu Eko Susanto (2015) bahwa menurut pendapat Arief (2011:7), “*Web* adalah salah satu aplikasi yang berisikan dokumen–dokumen multimedia (teks, gambar, suara, animasi, video) di dalamnya yang menggunakan protokol HTTP (*hypertexttransferprotokol*) dan untuk mengakses menggunakan perangkat lunak yang disebut *browser*”. Fungsi *website* diantaranya media promosi, media pemasaran, media informasi, media pendidikan dan media komunikasi.

#### **II.2.6. Database**

Data adalah fakta yang dapat direkam dan memiliki arti secara implisit. Sedangkan kumpulan data yang memiliki hubungan secara implisit itu disebut *Database*. Basis data merupakan kumpulan data, umumnya mendeskripsikan aktivitas satu organisasi atau lebih yang berhubungan. Dalam *database* juga dikenal istilah *DBMS (Database Management Systems)* yaitu sekumpulan program yang memungkinkan pengguna untuk membuat dan memelihara suatu *database*. Bisa juga dikatakan bahwa *DBMS* merupakan perangkat *General Purpose Software*

*System* yang berfungsi untuk mewadahi proses-proses dalam *database* seperti pendefinisian, pembuatan, *sharing*, maupun manipulasi *database*(Fitri Marisa, 2015).

### II.2.7. *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 :

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

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.2.8. Unified Modeling Language (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).

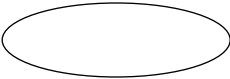
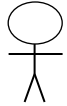

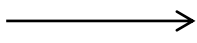
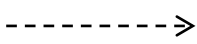
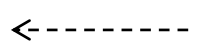
Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis *UML* adalah sebagai berikut:

#### 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**




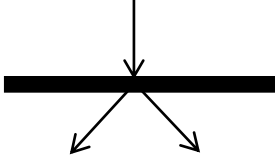
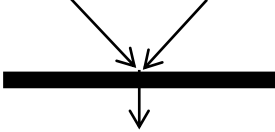
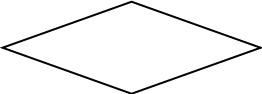

Gambar	Keterangan
	<i>Use case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama <i>use case</i> .
	Aktor adalah <i>abstraction</i> dari orang atau sistem yang lain yang 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> .
	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 mengidikasikan aliran data.
	Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengidinkasikan bila aktor berinteraksi secara pasif dengan sistem.
	<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.
	<i>Extend</i> , merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.

(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**

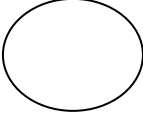
Gambar	Keterangan
	<i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	<i>End point</i> , akhir aktifitas.
	<i>Activites</i> , menggambarkan suatu proses/kegiatan bisnis.
	<i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau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

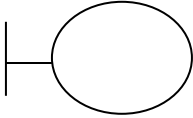
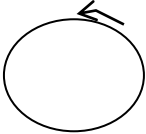
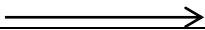
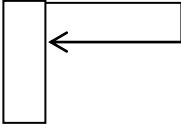


(Sumber : Gellysa Urva dan Helmi Fauzi Siregar : 2015: 94)

### 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
	<i>EntityClass</i> , merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.

	<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.
	<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.
	<i>Message</i> , simbol mengirim pesan antar <i>class</i> .
	<i>Recursive</i> , menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.
	<i>Activation</i> , <i>activation</i> mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.
	<i>Lifeline</i> , garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i> .

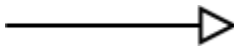
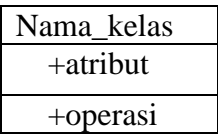


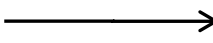
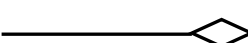

(Sumber : Gellysa Urva dan Helmi Fauzi Siregar:2015: 95)

#### 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
	<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.
	<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.
	<i>Interface</i> , untuk simbol <i>interface</i> atau dalam bahasa indonesianya antar muka. Konsep yang digunakan pun sama dengan pemrograman berorientasi object (OOP).
	<i>Association</i> , digunakan untuk menghubungkan atau merelasikan kelas satu dengan kelas yang lainnya dengan makna umum.
	<i>Directed Association</i> , adalah relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain.
	<i>Aggregation</i> , adalah relasi antar kelas dengan makna semua bagian.
	<i>Dependency</i> , adalah relasi antar kelas dengan makna kebergantungan antar kelas.

(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.2.9. 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)*.

### **II.2.10. MySQL**

*MySQL* adalah *Relation Database Management System (RDBMS)* yang di distribusikan 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 datanya dikerjakan dengan mudah secara otomatis. (Inayah, dkk, 2012 : 39).