

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 keaslian penelitian yang menjadi acuan dalam penelitian ini yaitu :

Tabel II.1. Keaslian Penelitian

No.	Peneliti Jurnal	Jurnal	Hasil Penelitian
1.	Adam Hasbi Nurgroho, dkk 2016	Sistem Pendukung Keputusan Penentuan Nilai Kenaikan Pangkat TNI Angkatan Darat Menggunakan Metode <i>Composite Performance Index</i> (CPI)	Penelitian ini menghasilkan suatu sistem pendukung keputusan yang mampu menampilkan hasil keputusan dari nilai kenaikan pangkat TNI Angkatan Darat.
2.	Renny Noer Fajarini, 2016	Sistem Pendukung Keputusan Menggunakan Metode <i>Composite Performance Index</i> Untuk Pemilihan Pemenang Tender Pengadaan Barang/Jasa	Penelitian ini menghasilkan sebuah sistem pendukung keputusan yang mampu menampilkan hasil keputusan menentukan jenis bibit tanaman.

3.	Nur S Tanjung dkk, 2018	Sistem Pendukung Keputusan Pemilihan Guru Teladan Dengan Menggunakan Metode Composite Performance Index (CPI)	Penelitian ini menghasilkan sebuah sistem pendukung keputusan yang mampu menampilkan hasil keputusan pemilihan guru teladan.
----	-------------------------	---	--

II.2. Sistem Informasi

Sistem Informasi didefinisikan sebagai sekumpulan objek, ide dan data yang telah ditempatkan pada konteks yang penuh arti oleh penerimanya, berikut keterkaitannya di dalam mencapai tujuan. Dengan kata lain, sistem informasi adalah sekumpulan komponen (sub-sistem fisik & non-fisik) dan informasi yang saling berhubungan satu sama lainnya dan bekerja sama untuk mencapai suatu tujuan. (Eddy Prahasta, 2014). Sistem Informasi adalah seperangkat komponen yang saling terkait yang mengumpulkan atau mengambil, memproses, menyimpan, dan mendistribusikan informasi untuk mendukung pengambilan keputusan dan kontrol dalam sebuah organisasi (Merpanto Kaneko dan Radiant Victor Imbar, 2017).

II.3. Sistem Pendukung Keputusan

Sistem Pendukung Keputusan (SPK) atau *Decision Support System* (DSS) adalah sebuah sistem yang mampu memberikan kemampuan dalam memecahkan suatu masalah dan memberikan solusi suatu masalah tersebut (Merpanto Kaneko dan Radiant Victor Imbar, 2017). 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. Sistem Pendukung Keputusan (SPK) merupakan sistem interaksi yang menyediakan informasi, pemodelan, dan manipulasi data. Sistem Pendukung Keputusan menggabungkan kemampuan sumber daya manusia dengan kemampuan komputer untuk meningkatkan hasil keputusan akhir (Andri Anto Tri Susilo, 2017)

II.4. *Composite Performance Index*

Composite Performance Index (CPI) digunakan untuk memilih beberapa alternatif. Teknik CPI (*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 dkk, 2018).

Penyelesaian CPI (*Composite Performance Index*) dalam mencari alternatif yang terbaik, yaitu :

- a. Identifikasi kriteria tren positif (semakin tinggi nilainya semakin baik) dan tren negatif (semakin rendah nilainya semakin buruk).

- b. Untuk kriteria tren positif, nilai minimum pada setiap kriteria ditransformasi ke seratus, sedangkan nilai lainnya ditransformasi secara profesional lebih tinggi.
- c. 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 :

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

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

Metode CPI (*Composite Performance Index*) merupakan salah satu metode perhitungan dari pengambilan keputusan berbasis indeks kinerja gabungan yang dapat digunakan untuk menentukan penilaian atau peringkat dari berbagai alternatif (i) berdasarkan beberapa kriteria (j). (Adam Hasbi Nugroho dkk, 2016).

II.5. Sistem Basis Data

II.5.1. Basis Data

Basis data adalah kumpulan *file-file* yang saling berelasi, relasi tersebut biasa ditunjukkan dengan kunci dari tiap *file* yang ada. Satu basis data menunjukkan kumpulan data yang dipakai dalam satu lingkup informasi. Dalam satu *file* terdapat *record-record* yang sejenis, sama besar, sama bentuk, merupakan satu kumpulan *entity* yang seragam. Satu *record* terdiri dari *field-field* yang saling berhubungan untuk menunjukkan bahwa *field* tersebut dalam satu pengertian yang lengkap dan direkam dalam satu *record*. Suatu sistem manajemen basis data berisi satu koleksi data yang saling berelasi dan satu set program untuk mengakses data tersebut. Jadi sistem manajemen basis data dan set program pengelola untuk menambah data, menghapus data, mengambil data dan membaca data. (Mhd. Bustanur dan Tedy Setiady, Vol 2 : 2014).

II.5.2. Konsep Basis Data

Pada basis data ini akan dibahas tentang definisi yang terdiri dari *Database, File, Entity, dan Record*.

1) *Entity*

Entity adalah orang, tempat, kejadian atau konsep yang informasinya direkam pada suatu basis data misalnya informasi lalulintas, *entity* antara lain kemacetan, kecelakaan dan lain sebagainya.

1. Atribut

Setiap *entity* mempunyai atribut atau sebutan untuk mewakili suatu entity lalulintas dengan atributnya, misalnya nama obyek, alamat, jenis obyek, dan lain sebagainya. Atribut juga disebut sebagai data elemen, data *field*, item.

2. Data *Value*

Data *value* adalah data aktual atau informasi yang disimpan pada tiap data elemen atau atribut.

2) *Database*

Database adalah kumpulan *field-field* yang mempunyai kaitanantara satu *field* dengan *field* yang lain sehingga membentuk bangunan data untuk menginformasikan kondisi lalulintas dalam bahasa tertentu.

3) *File*

File adalah kumpulan *record-record* sejenis yang mempunyai panjang elemen yang sama, atribut yang sama, namun berbeda-beda datanya.

4) *Record*

Record adalah kumpulan elemen-elemen yang saling berkaitan menginformasikan tentang suatu *entity* secara lengkap satu *record* mewakili satu data atau informasi. (Mhd. Bustanur dan Tedy Setiady, Vol 2 : 2014).

II.6. Normalisasi

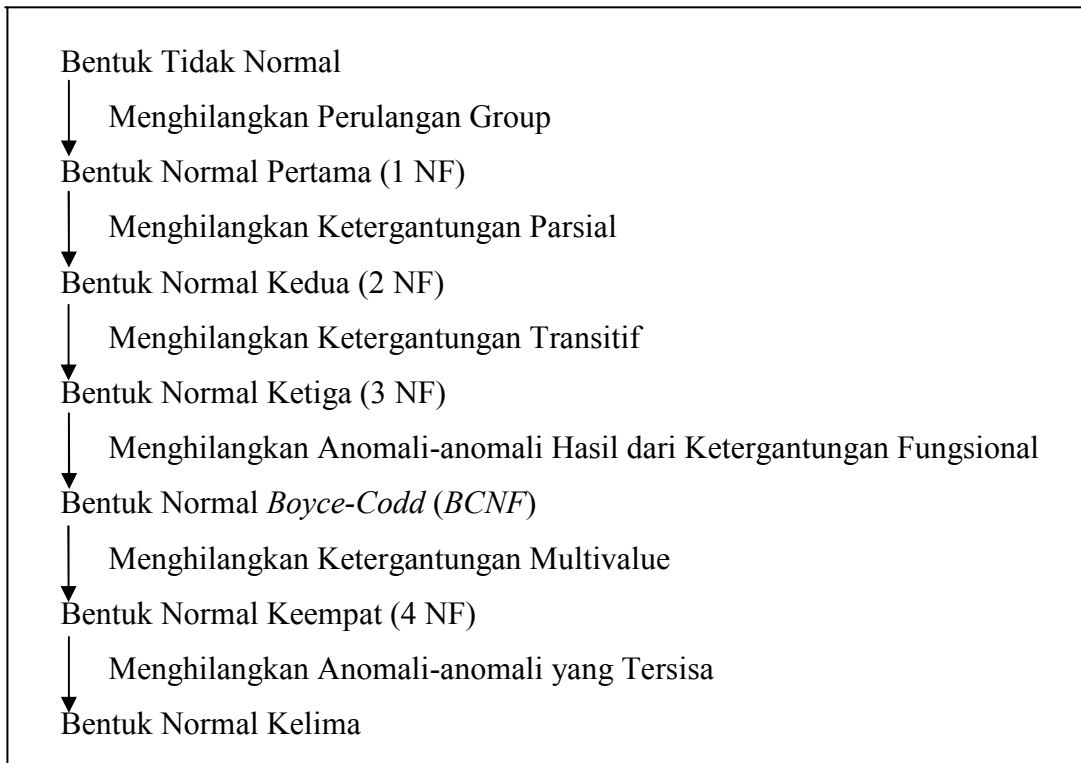
Normalisasi adalah proses pembentukan struktur basis data sehingga sebagian besar *ambiguity* bisa dihilangkan. Normalisasi merupakan sebuah teknik dalam *logical* desain sebuah basis data relasional yang mengelompokkan atribut dari suatu tabel sehingga membentuk struktur tabel yang normal. Adapun kriteria tabel dikatakan normal adalah ketika tidak ada kerangkapan data (redudansi data).

Tujuan dari normalisasi adalah untuk :

1. Untuk menghilangkan kerangkapan data sehingga meminimumkan pemakaian *storage* yang dipakai oleh base relations (*file*).
2. Untuk mengurangi kompleksitas.
3. Untuk mempermudah pemodifikasian data.

Adapun proses normalisasi sebagai berikut :

- a. Data diuraikan dalam bentuk tabel, selanjutnya dianalisis berdasarkan persyaratan tertentu ke beberapa tingkat kemudian.
- b. Apabila tabel yang diuji belum memenuhi persyaratan tertentu, maka tabel tersebut perlu dipecah menjadi beberapa tabel yang lebih sederhana sampai memenuhi bentuk yang optimal. Untuk melakukan proses tersebut dibutuhkan beberapa tahapan. Tahapan dalam normalisasi dimulai dari tahap paling ringan (1NF) hingga paling ketat (5NF). Biasanya hanya sampai pada tingkat 3NF atau BCNF karena sudah cukup memadai untuk menghasilkan tabel-tabel yang berkualitas baik. Urutan tahapan normalisasi dapat dilihat pada Gambar II.1.



Gambar II.I. Tahapan Normalisasi
(Sumber : Dwi Puspitasari, et al : 2016)

Adapun aturan dalam normalisasi adalah suatu tabel dikatakan baik (efisien) atau normal jika memenuhi 3 kriteria sbb:

1. Jika ada dekomposisi (penguraian) tabel, maka dekomposisinya harus dijamin aman (*Lossless-Join Decomposition*). Artinya, setelah tabel tersebut diuraikan / didekomposisi menjadi tabel-tabel baru, tabel-tabel baru tersebut bisa menghasilkan tabel semula dengan sama persis.
2. Terpeliharanya ketergantungan fungsional pada saat perubahan data (*Dependency Preservation*).
3. Tidak melanggar *Boyce-Codd* Normal Form (BCNF) Jika kriteria ketiga (BCNF) tidak dapat terpenuhi, maka paling tidak tabel tersebut tidak

melanggar Bentuk Normal tahap ketiga (3rd Normal Form / 3NF). Pada penelitian ini formula yang dibuat sampai memenuhi bentuk normal ke 3 yaitu 3NF. (Dwi Puspitasari, et al : 2016).

II.7. 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 datanya dikerjakan dengan mudah secara otomatis. (Inayah, dkk, 2012 : 39).

II.8. 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.9. Unified Modeling Language (UML)

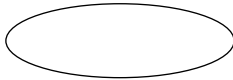
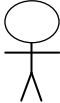

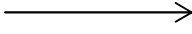
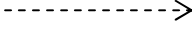
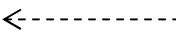
Unified Modeling Language (UML) adalah bahasa spesifik standar yang dipergunakan untuk mendokumentasikan, menspesifikan dan membangun perangkat lunak. UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. UML saat ini sangat 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 berbasiskan 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 berbasiskan UML adalah sebagai berikut:

II.9.1. 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.2. Simbol *Use Case*



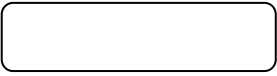
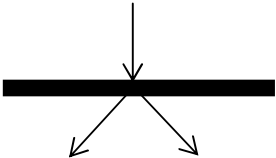
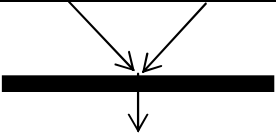
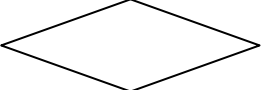

Gambar	Keterangan
	<i>Use case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama <i>use case</i> .
	Aktor adalah <i>abstraction</i> dari orang atau sistem yang lain yang 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 mengindikasikan aliran data.
	Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengindikasikan 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)

II.9.2. 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.3. Simbol *Activity Diagram*

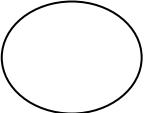
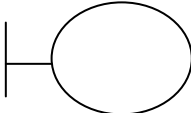
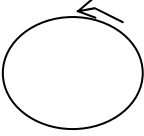

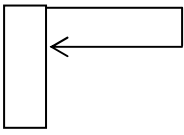


Gambar	Keterangan
	<i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	<i>End point</i> , akhir aktifitas.
	<i>Activites</i> , menggambarkan suatu proses/kegiatan bisnis.
	<i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel 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 diagram</i> untuk menunjukkan siapa melakukan apa.

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

II.9.3. 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.4. Simbol *Sequence Diagram*

Gambar	Keterangan
	<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.
	<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> .




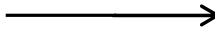


(Sumber : Gellysa Urva dan Helmi Fauzi Siregar:2015: 95)

II.9.4. 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.5. 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.			
<table border="1" data-bbox="335 1025 550 1153"> <tr> <td>Nama_kelas</td> </tr> <tr> <td>+atribut</td> </tr> <tr> <td>+operasi</td> </tr> </table>	Nama_kelas	+atribut	+operasi	<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.
Nama_kelas				
+atribut				
+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.6. 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)