



BAB II

TINJAUAN PUSTAKA

2.1. Teori Umum

2.1.1. Pengertian Komputer

Menurut Asropudin (2013:19) “Komputer adalah alat bantu pemrosesan data secara elektronik dan pemrosesan datanya berdasarkan urutan instruksi yang tersimpan dalam memori masing-masing komputer”. Sedangkan menurut Sujatmiko (2012:156), “Komputer adalah mesin yang dapat mengolah data digital dengan mengikuti serangkaian perintah atau program”.

Maka dapat ditarik kesimpulan bahwa pengertian komputer adalah alat bantu pengolah data secara elektronik berdasarkan serangkaian perintah atau program yang tersimpan dalam memori komputer.

2.1.2. Pengertian Perangkat Lunak

Menurut Kadir (2013:2) “Perangkat Lunak berupa instruksi-instruksi yang ditujukan kepada komputer agar komputer dapat melaksanakan tugas sesuai kehendak pemakai”. Sedangkan menurut Mulyono (2008:97), “*Software* adalah rangkaian instruksi elektronik yang memerintahkan komputer untuk melakukan tugas tertentu sesuai dengan perintah yang diberikan oleh seorang pengguna komputer.”

Jadi dapat disimpulkan bahwa pengertian perangkat lunak adalah serangkaian instruksi elektronik yang ditujukan kepada komputer untuk melakukan tugas tertentu sesuai dengan kehendak pengguna komputer.

2.1.3. Pengertian Data

Menurut Asropudin (2013:22) “Data adalah kumpulan dari angka-angka maupun karakter-karakter yang tidak memiliki arti. Sedangkan menurut Indrajani (2015:69), “Data merupakan fakta mentah tentang orang, tempat, kejadian, dan apapun yang penting bagi perusahaan, dimana data itu sendiri tidak memiliki arti”.

Dari pengertian diatas dapat disimpulkan bahwa pengertian perangkat lunak adalah kumpulan fakta mentah yang penting yang tidak memiliki arti.



2.1.4. Pengertian Internet

Menurut Sujatmiko (2012:138), “Internet adalah jaringan global yang menghubungkan berjuta-juta komputer di seluruh dunia melalui jalur telepon kabel maupun satelit”. Sedangkan menurut Sutarman (2012:283), “Internet adalah kumpulan dari berbagai macam jenis komputer yang saling terhubung dengan menggunakan media telekomunikasi (telepon, *wireless*, satelit, dan sebagainya) dengan jangkauan seluruh dunia/global”.

Berdasarkan pengertian diatas dapat disimpulkan bahwa pengertian internet adalah kumpulan jaringan global dari berbagai macam komputer di seluruh dunia yang saling terhubung melalui media telekomunikasi.

2.2 Teori Judul

2.2.1 Sistem Pendukung Keputusan

Menurut Nofriansyah dan Sarjon (2017:2), “Sistem Pendukung Keputusan adalah suatu sistem informasi spesifik yang ditunjukkan untuk membantu manajemen dalam mengambil keputusan yang berkaitan dengan persoalan yang bersifat semi terstruktur”.

Sedangkan menurut Little dalam buku *Multi Criteria Decision Making (MCDM) pada Sistem Pendukung Keputusan* yang dikutip Nofriansyah dan Sarjon (2017:1), mendefinisikan sistem pendukung keputusan sebagai suatu informasi berbasis komputer yang menghasilkan berbagai alternatif keputusan sehingga membantu manajemen dalam menangani berbagai permasalahan yang terstruktur maupun tidak terstruktur dengan menggunakan data dan model.

Dari berbagai definisi diatas dapat disimpulkan bahwa sistem pendukung keputusan adalah suatu sistem informasi yang dirancang untuk membantu seseorang maupun sekelompok orang dalam mengambil keputusan terhadap permasalahan yang bersifat terstruktur maupun tidak terstruktur melalui beberapa alternatif dan kriteria yang telah ditentukan sehingga dapat menghasilkan keputusan yang lebih cepat dan tepat.

Komponen-komponen Sistem Pendukung Keputusan (Nofriansyah dan Sarjon, 2017:3) terdiri dari:



- a. *Data Management*. Termasuk *database*, yang mengandung data yang relevan untuk berbagai situasi dan diatur oleh *software* yang disebut *Database Management System (DBMS)*.
- b. *Model Management*. Melibatkan model finansial, *statistical*, *management science*, atau berbagai model kuantitatif lainnya, sehingga dapat memberikan ke sistem suatu kemampuan analisis, dan manahemen *software* yang diperlukan.
- c. *Communication (dialog subsystem)*. *User* dapat berkomunikasi dan memberikan perintah pada DSS melalui subsistem ini. Ini bearti menyediakan antarmuka.
- d. *Knowledge Management*. Subsistem *optional* ini dapat mendukung subsistem lain atau bertindak sebagai komponen yang berdiri sendiri.

2.2.2 Komputer

Menurut Asropudin (2013:19) “Komputer adalah alat bantu pemrosesan data secara elektronik dan pemrosesan datanya berdasarkan urutan instruksi yang tersimpan dalam memori masing-masing komputer”. Sedangkan menurut Sujatmiko (2012:156), “Komputer adalah mesin yang dapat mengolah data digital dengan mengikuti serangkaian perintah atau program”.

Maka dapat ditarik kesimpulan bahwa pengertian komputer adalah alat bantu pengolah data secara elektronik berdasarkan serangkaian perintah atau program yang tersimpan dalam memori komputer.

2.2.3 Metode *Simple Additive Weighting (SAW)*

Menurut Nofriansyah dan Sarjon (2017:33), metode *Simple Additive Weighting* merupakan metode pembobotan sederhana atau penjumlahan terbobot pada penyelesaian masalah dalam sebuah sistem pendukung keputusan. Konsep dalam metode ini adalah dengan mencari peringkat kinerja (skala prioritas) pada setiap alternatif di semua atribut.



Adapun algoritma penyelesaian metode ini (Nofriansyah dan Sarjon, 2017:33) diantaranya sebagai berikut:

1. Langkah 1 : Mendefinisikan terlebih dahulu kriteria-kriteria yang akan di jadikan sebagai tolak ukur penyelesaian masalah.
2. Langkah 2 : Menormalisasi setiap nilai alternatif pada setiap atribut dengan cara menghitung nilai *rating* kinerja.
3. Langkah 3 : Menghitung nilai bobot preferensi pada setiap alternatif.
4. Langkah 4 : Melakukan perangkingan.

Rumus yang digunakan pada metode *Simple Additive Weighting* (Nofriansyah dan Sarjon, 2017:33) yaitu:

1. Menormalisasikan setiap alternatif (menghitung nilai *rating* kinerja).

$$r_{ij} = \left\{ \begin{array}{ll} \frac{x_{ij}}{\max_i x_{ij}} & \text{Jika } j \text{ adalah atribut keuntungan (benefit)} \\ \frac{\min_i x_{ij}}{x_{ij}} & \text{Jika } j \text{ adalah atribut biaya (cost)} \end{array} \right\}$$

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

Keterangan:

$\max x_{ij}$ = Nilai terbesar dari setiap kriteria i .

$\min x_{ij}$ = Nilai terkecil dari setiap kriteria i .

x_{ij} = Nilai atribut yang dimiliki dari setiap kriteria.

Benefit = Jika nilai terbesar adalah terbaik.

Cost = Jika nilai terkecil adalah terbaik.

2. Menghitung nilai bobot preferensi pada setiap alternatif.

$$V_i = \sum_{j=1}^n w_j r_{ij}$$

Keterangan:

V_i = Ranking untuk setiap alternatif.

w_j = Nilai bobot kriteria.

r_{ij} = Nilai *rating* kinerja ternormalisasi.

Nilai V_i yang lebih mengindikasikan bahwa alternatif A_i lebih terpilih.



Adapun kelebihan dari metode *Simple Additive Weighting* (Pratiwi, 2016:138), yaitu:

1. Menentukan nilai bobot untuk setiap atribut, kemudian dilanjutkan dengan proses perankingan yang akan menyeleksi alternatif terbaik dari sejumlah alternatif.
2. Penilaian akan lebih tepat karena didasarkan pada nilai kriteria dari bobot preferensi yang sudah ditentukan.
3. Adanya perhitungan normalisasi matriks sesuai dengan nilai atribut (antara nilai *benefit* dan *cost*).

2.3 Teori Khusus

2.3.1 Pengertian UML (*Unified Modeling Language*)

Sukamto dan Shalahuddin (2013:133), “UML (*Unified Modeling Language*) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan requirement, membuat analisis & desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek”.

Bahasa pemrograman berorientasi objek yang pertama dikembangkan dikenal dengan nama Simula-67 yang dikembangkan pada tahun 1967. Perkembangan aktif dari pemrograman berorientasi objek mulai menggeliat ketika berkembangnya bahasa pemrograman Smalltalk pada awal 1980-an. Pada 1996, Object Management Group (OMG) mengajukan proposal agar adanya standarisasi pemodelan berorientasi objek dan pada bulan September 1997 Unified Modeling Language (UML) diakomodasi oleh Object Management Group (OMG) sehingga sampai saat ini Unified Modeling Language (UML) telah memberikan kontribusinya yang cukup besar dalam metodologi berorientasi objek.

2.3.2 Usecase Diagram

Sukamto dan Shalahuddin (2018:155), “*Use case* atau diagram *use case* 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”.

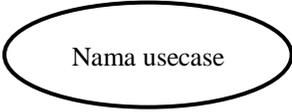
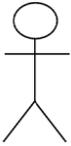
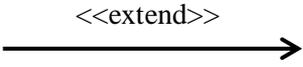
Syarat penamaan pada *use case* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada *use case* yaitu pendefinisian apa yang disebut aktor dan *use case*.

1. Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.

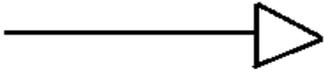
2. *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.

Berikut adalah simbol-simbol yang ada pada diagram *use case* menurut Rosa dan Shalahuddin (2013:156) :

Tabel 2.1 Simbol-simbol diagram *use case*

No.	Simbol	Keterangan
1.	<p>Use case</p>  <p>Nama usecase</p>	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama use case
2.	<p>Aktor / actor</p>  <p>nama actor</p>	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama actor.
3.	<p>Asosiasi / association</p> 	Komunikasi antar aktor dan use case yang berpartisipasi pada use case atau use case memiliki interaksi dengan actor.
4.	<p>Ekstensi / extend</p>  <p><<extend>></p>	Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan dapat berdiri sendiri walau tanpa use case tambahan itu; mirip dengan prinsip inheritance

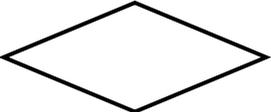


		pada pemrograman berorientasi objek; ditambahkan, misal arah panah mengarah pada use case yang ditambahkan; biasanya use case yang menjadi extend-nya merupakan jenis yang sama dengan use case yang menjadi induknya.
5.	Generalisasi / <i>generalization</i> 	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah use case dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya

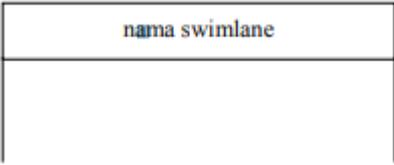
2.3.3 Activity Diagram

Sukanto dan Shalahuddin (2018:161), “Diagram aktivitas atau activity diagram menggambarkan workflow (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak”. Berikut adalah simbol-simbol yang ada pada diagram aktivitas menurut Rosa dan Shalahuddin (2013:162) :

Tabel 2.2 Simbol-simbol *activity diagram*

No.	Simbol	Keterangan
1.	Status Awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
2.	Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3.	Percabangan / <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
4.	Penggabungan / <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.



5.	Status Akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
6.	Swimlane 	Swimlane memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

2.3.4 ClassDiagram

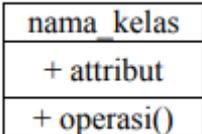
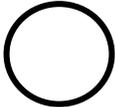
Sukamto dan Shalahuddin (2018:141), “Diagram kelas atau class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem”. Kelas memiliki apa yang disebut atribut dan metode atau operasi.

1. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas.
 2. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas.
- Susunan struktur kelas yang baik pada diagram kelas sebaiknya memiliki jenis-jenis kelas berikut:
1. Kelas main, kelas yang memiliki fungsi awal dieksekusi ketika sistem dijalankan.
 2. Kelas yang menangani tampilan sistem (view), kelas yang mendefinisikan dan mengatur tampilan ke pemakai.
 3. Kelas yang diambil dari pendefinisian use case (controller), kelas yang menangani fungsi-fungsi yang harus ada diambil dari pendefinisian use case, kelas ini biasanya disebut dengan kelas proses yang menangani proses bisnis pada perangkat lunak.
 4. Kelas yang diambil dari pendefinisian data (model), kelas yang digunakan untuk memegang atau membungkus data menjadi sebuah kesatuan yang diambil maupun akan disimpan ke basis data.



Berikut adalah simbol-simbol yang ada pada diagram kelas menurut Rosa dan Shalahuddin (2018:146) :

Tabel 2.3 Simbol-simbol *class diagram*

No.	Simbol	Keterangan
1.	Kelas 	Kelas pada struktur system.
2.	Antarmuka / <i>interface</i> 	Sama dengan konsep interface dalam pemrograman berorientasi objek.
3.	Asosiasi / <i>association</i> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan multiplicity.
4.	Asosiasi berarah / <i>directed association</i> 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan multiplicity.
5.	Agregasi / <i>aggregation</i> 	Relasi antar kelas dengan makna semua- bagian (whole-part).
6.	Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)
7.	Kebergantungan / <i>dependency</i> 	Relasi antar kelas dengan makna kebergantungan antarkelas

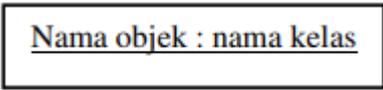


2.3.5 Sequence Diagram

Sukanto dan Shalahuddin (2018:165), “Sequence diagram atau diagram sekuen menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan message yang dikirim dan diterima antar objek”. Banyaknya diagram sekuen yang harus digambar adalah minimal sebanyak pendefinisian use case yang memiliki proses sendiri atau yang penting semua use case yang telah didefinisikan interaksinya jalannya pesan sudah dicakup pada diagram sekuen sehingga semakin banyak use case yang didefinisikan maka diagram sekuen yang harus dibuat juga semakin banyak.

Berikut adalah simbol-simbol yang ada pada diagram sekuen menurut Rosa dan Shalahuddin (2018:146) :

Tabel 2.4 Simbol-simbol *sequence diagram*

No.	Simbol	Keterangan
2.	Garis Hidup / lifeline 	Menyatakan kehidupan suatu objek.
3.	Objek 	Menyatakan objek yang berinteraksi pesan.
4.	Waktu Aktif 	Menyatakan objek dalam keadaan aktif dan berinteraksi, semuanya yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya, misalnya :



		<p>Maka cekStatusLogin() dan open() dilakukan didalam metode login(). Aktor tidak memiliki waktu aktif</p>
5.	Pesan tipe <i>create</i> <<create>> 	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.
6.	Pesan tipe <i>call</i> 1 : nama_metode() 	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri.
		Arah panah mengarah pada objek yang memiliki operasi/metode, karena ini memanggil operasi/metode maka operasi/metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi.
7.	Pesan tipe <i>send</i> 1 : masukkan 	Menyatakan bahwa suatu objek mengirimkan data/masukkan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.
8.	Pesan tipe <i>return</i> 1 : keluaran 	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.
9.	Pesan tipe <i>destroy</i> <<destroy>> 	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaliknya jika ada create maka ada destroy.



2.4 Teori Program

2.4.1 PHP

Menurut Gunawan (2010:12), “PHP merupakan bahasa pemrograman yang memungkinkan para *web developer* untuk membuat aplikasi *web* yang dinamis dengan cepat dan mudah”.

Menurut Winarno, dkk. (2014:49), “PHP atau PHP *Hypertext Preprocessor* adalah sebuah bahasa pemrograman *web* berbasis *server (server-side)* yang mampu memarsing kode PHP dari kode *web* dengan ekstensi *.php*, sehingga menghasilkan tampilan *website* yang dinamis di sisi *client (browser)*”.

Dari pengertian di atas dapat disimpulkan bahwa PHP merupakan sebuah bahasa pemrograman yang digunakan untuk membuat aplikasi *web*.

2.4.2 MySQL

Menurut Gunawan (2010:15), MySQL adalah aplikasi atau sistem untuk mengelola *database* atau manajemen data. MySQL bertugas mengatur dan mengelola data-data pada *database*. MySQL menggunakan struktur atau kerangka yang berbentuk tabel. Dalam tabel-tabel itulah data diatur dan dikelompokkan.

Menurut Winarno, dkk. (2014:49), “MySQL merupakan tipe data relasional yang artinya MySQL menyimpan datanya dalam bentuk table-tabel yang saling berhubungan”.

Dari pengertian di atas dapat disimpulkan bahwa MySQL merupakan suatu aplikasi atau sistem yang digunakan untuk menyimpan dan memanajemen data dalam tabel-tabel yang saling berhubungan.

2.4.3 XAMPP

Menurut Enterprise (2017:93), XAMPP merupakan *tool* yang digunakan untuk membantu kita bekerja menggunakan MySQL *Server*, namun dengan tampilan antarmuka grafis yang lebih ramah untuk siapa pun, terlebih bagi kalangan pemula.



Menurut Gunawan (2010:17), XAMPP adalah aplikasi *web server* instan yang dibutuhkan untuk membangun aplikasi berbasis *web*. Fungsi XAMPP adalah sebagai server yang berdiri sendiri (*localhost*), yang terdiri atas program *Apache HTTP Server*, *MySQL database*, dan penterjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl.

Dari pengertian di atas dapat disimpulkan bahwa XAMPP merupakan suatu aplikasi *web server* yang digunakan untuk mengelola *MySQL Server* dan membantu dalam pemrograman PHP.

2.5 Penelitian Terdahulu

Menurut penelitian yang dilakukan oleh Leni Natalia Zulita (2014) melakukan sebuah penelitian yang menyebutkan, metode *Simple Additive Weighting* (SAW) untuk penilaian dosen berprestasi di Universitas Dehasen Bengkulu diharapkan dapat membantu Staff dalam pengambilan keputusan untuk melakukan penilaian dosen berprestasi berdasarkan alternatif yang dipilih demi meningkatkan kualitas kinerja dosen sesuai dengan visi misi dan tujuan perguruan tinggi.

Elfrida Hutahaean, Dkk. (2019) telah ditemukan sebuah sistem pendukung keputusan dengan metode *Simple Additive Weighting* (SAW) perangkaan data konsumen penumpang kereta api. Metode ini dipilih karena merupakan metode yang sederhana secara perhitungan (komputasinya sederhana). Metode SAW memiliki keunggulan dalam mencari penjumlahan terbobot dari rating data penumpang.

Dalam penelitian yang dilakukan oleh Anita dan Sigit Syahputra Harahap (2019) mengatakan bahwa metode *Simple Additive Weighting* (SAW) digunakan untuk penyetujuan kartu kredit pada bank BRI. Hasil penelitian diharapkan dapat membantu bank memutuskan apakah sebuah permohonan pengajuan kartu kredit layak disetujui atau tidak secara lebih cepat dan akurat.

Muhammad Alfadin Salim (2018) melakukan penelitian yang menyatakan Sistem pendukung keputusan pemilihan penerima bantuan perbaikan rumah diproduksi dengan menggunakan beberapa metode, salah satunya adalah metode



Simple Additive Weighting (SAW) dengan menggunakan metode *Simple Additive Weighting* ini. Hasil yang didapatkan berupa urutan penerima bantuan Perbaikan rumah dari yang paling layak sampai yang paling tidak layak.

Dalam penelitian yang dibuat oleh Melisa Elistri, dkk (2014) mengatakan bahwa penulisan ini bertujuan untuk membuat suatu sistem pendukung keputusan pemilihan jurusan pada sekolah menggunakan microsoft visual basic 6.0. Hasil ini bisa dimanfaatkan sebagai salah satu pertimbangan dalam pengambilan suatu keputusan oleh pihak sekolah dalam mengambil keputusan jurusan siswa.