



BAB II

TINJAUAN PUSTAKA

2.1 Teori Dasar

2.1.1 Aplikasi

Menurut Solichin (2016:1), Aplikasi atau perangkat lunak (*software*) merupakan bagian yang tidak terpisahkan dari suatu sistem komputer, disamping keberadaan pengguna (*brainware*), perangkat keras (*hardware*) dan jaringan (*networking*).

Menurut Kadir (2017:3) “Istilah program dan aplikasi lebih sering disebut untuk menyatakan perangkat lunak. Di kalangan profesional teknologi informasi, istilah program biasa digunakan untuk menyatakan hasil karya mereka yang berupa instruksi-instruksi untuk mengendalikan komputer. Di sisi pemakai, hal seperti itu biasa disebut sebagai aplikasi”.

Berdasarkan pengertian diatas dapat disimpulkan bahwa aplikasi merupakan suatu perangkat lunak yang terdapat dalam sistem tertentu berguna untuk membantu pengguna mengolah, mengedit data maupun informasi.

2.1.2 Sistem Pendukung Keputusan

Menurut Basuki dan Andharini (2016:13), Sistem Pendukung Keputusan (DSS) adalah sistem berbasis komputer yang interaktif, yang mampu membantu pengambilan keputusan dalam menggunakan data dan model untuk menyelesaikan masalah yang tidak terstruktur.

Menurut Sari (2018:1), Sistem Pendukung Keputusan atau dikenal juga dengan istilah Decision Support System (DSS) ini merupakan pengembangan lebih lanjut dari sistem informasi manajemen terkomputerisasi yang dirancang sedemikian rupa sehingga bersifat interaktif dengan pemakainya.

Berdasarkan pengertian diatas dapat disimpulkan bahwa sistem pendukung keputusan merupakan sistem informasi berbasis komputer dan dapat membantu pemakainya untuk mengambil suatu keputusan dalam menyelesaikan masalah semi-terstruktur.



2.1.3 Pembelajaran

Menurut Parwati, dkk. (2018:107), Istilah pembelajaran berhubungan erat dengan pengertian belajar dan mengajar. Belajar, mengajar dan pembelajaran terjadi Bersama-sama. Belajar dapat terjadi tanpa guru atau tanpa kegiatan mengajar dan pembelajaran formal lainnya. Sedangkan mengajar meliputi segala hal yang guru lakukan didalam kelas.

Menurut Hanafi, dkk (2018:57), Kata pembelajaran secara bahasa berarti; “proses, cara, menjadikan orang atau makhluk hidup belajar”.

Berdasarkan pengertian diatas dapat disimpulkan bahwa pembelajaran itu ialah proses belajar dan mengajar yang dilakukan oleh pendidik dalam membantu peserta didik untuk mencapai tujuan pendidikannya.

2.1.4 Dunia Kerja

Menurut Ginting (2017:9), Dunia Kerja adalah sebuah tempat atau wadah orang-orang yang berkumpul untuk bekerja bersama-sama. Mereka memanfaatkan sumber daya (uang, bahan baku, mesin, peralatan, alat bantu, fasilitas, perlengkapan, ruang, energi, metode, data, lingkungan, sarana-prasarana dan lain-lain) untuk mencapai tujuan pembentukan tempat atau wadah tersebut. Bekerja yang dimaksud adalah melakukan aktivitas yang melibatkan usaha mental atau fisik yang dilakukan untuk mencapai hasil.

2.1.5 *Sublime Text*

Menurut Supono dan Putratama (2018:14), *Sublime Text* merupakan perangkat lunak *text editor* yang digunakan untuk membuat atau meng-edit suatu aplikasi.

Menurut Rachmanto (2017:21), *Sublime Text* merupakan *text editor* kedua yang juga sering saya gunakan. Selain digunakan untuk belajar, sublime text terkadang saya gunakan untuk penulisan *coding time*.

Menurut pengertian diatas disimpulkan bahwa *sublime text* adalah *text editor* yang dapat digunakan untuk menulis dan mengedit suatu aplikasi.



2.1.6 *PHP MyAdmin*

Menurut Hikmah, dkk (2015:2), “*phpMyadmin* merupakan aplikasi yang dapat digunakan untuk membuat database, pengguna (*user*), memodifikasi tabel, maupun mengirim database secara cepat dan mudah tanpa harus menggunakan perintah (*command*) SQL”.

Menurut Haqi dan Setiawan (2019:10), *PhpMyAdmin* adalah bagian untuk mengelola *database MySQL* yang dikomputer.

Menurut pengertian diatas dapat disimpulkan bahwa *phpMyAdmin* adalah bagian dari *MySQL* yang digunakan untuk membuat, mengedit dan mengelola database.

2.1.7 *Multi Objective Optimization on the basis of Rasio Analysis (MOORA)*

Menurut Nofriansyah dan Sarjon (2017:85) Metode *Multi Objective Optimization by Ratio Analysis (MOORA)* adalah multiobjektif sistem mengoptimalkan dua atau lebih atribut yang saling bertentangan secara bersamaan. Metode ini diterapkan untuk memecahkan masalah dengan perhitungan matematika yang kompleks. MOORA diperkenalkan oleh Brauers dan Zavadkas pada tahun 2006. Pada awalnya metode ini diperkenalkan oleh Brauers dan Zavadkas tahun 2004 sebagai “*Multi-Objective Optimization*” yang dapat digunakan untuk memecahkan berbagai masalah pengambilan keputusan yang rumit pada lingkungan pabrik. Metode moora diterapkan untuk memecahkan banyak permasalahan ekonomi, manajerial dan kontruksi pada sebuah perusahaan maupun proyek.

Berikut ini adalah algoritma penyelesaian metode Moora yaitu sebagai berikut:

1. Langkah pertama : Menginput Nilai Kriteria. Menginputkan nilai kriteria pada suatu alternatif dimana nilai tersebut nantinya akan diproses dan hasilnya akan menjadi sebuah keputusan.
2. Langkah kedua : Merubah nilai kriteria menjadi matriks keputusan. Matriks keputusan berfungsi sebagai pengukuran kinerja dari alternative I^{th} pada atribut J^{th} , M adalah alternatif dan n adalah jumlah atribut dan kemudian sistem rasio



dikembangkan dimana setiap kinerja dari sebuah alternatif pada sebuah atribut dibandingkan dengan penyebut merupakan wakil untuk semua alternatif dari atribut tersebut. Berikut adalah perubahan nilai kriteria menjadi sebuah matriks keputusan :

$$X = \begin{matrix} & x_{11} & x_{12} & x_{1n} \\ x_{21} & x_{21} & x_{22} & x_{2n} \\ & x_{m1} & x_{m2} & x_{mn} \end{matrix}$$

Gambar 2.1 Pembentukan Matriks

3. Langkah ketiga : Normalisasi pada metode MOORA. Normalisasi bertujuan untuk menyatukan setiap element matriks sehingga element pada matriks memiliki nilai yang seragam. Normalisasi pada MOORA dapat dihitung menggunakan persamaan sebagai berikut.

$$X_{ij}^* = X_{ij} / \sqrt{[\sum_{i=1}^m X_{ij}^2]}$$

Gambar 2.2 Menentukan Matriks Normalisasi

4. Langkah keempat : Mengurangi nilai maximax minmax untuk menandakan bahwa sebuah atribut lebih penting itu bisa dikalikan dengan bobot yang sesuai (koefisien signifikasi). (Brauers etal.2009 dalam Ozcelik.2014). Saat atribut bobot dipertimbangkan perhitungan menggunakan persamaan sebagai berikut.

$$Y_i = \sum_j^g = W_j X_{ij}^* - \sum_{j=g+1}^n W_j W_{ij}^*$$

Gambar 2.3 Menentukan Nilai Preferensi

5. Langkah kelima : Menentukan rangking dari hasil perhitungan MOORA.



2.2 Teori Khusus

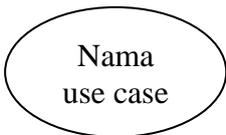
2.2.1 Unified Modeling Language (UML)

Menurut Sukamto (2018:13), *Unified Modeling Language* (UML) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung. UML hanya berfungsi untuk melakukan pemodelan. Jadi penggunaan UML tidak terbatas pada metodologi tertentu, meskipun pada kenyataannya UML paling banyak digunakan pada metodologi berorientasi objek.

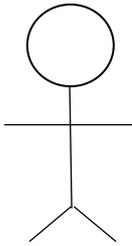
2.2.2 Use Case Diagram

Menurut Sukamto (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. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu. Berikut adalah simbol-simbol yang ada pada diagram *use case diagram* kelas dibuat agar pembuat program atau *programmer* membuat kelas-kelas sesuai rancangan di dalam diagram kelas agar antara dokumentasi perancangan dan perangkat lunak sinkron.

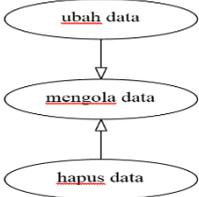
Tabel 2.1 Simbol-simbol *Use Case diagram*

No	Simbol	Deskripsi
1.	<p><i>Use case</i></p> 	<p>Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja diawal frase nama <i>use case</i>.</p>

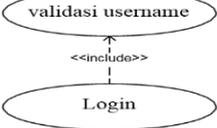
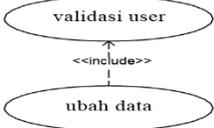
Lanjutan Tabel 2.1 Simbol-simbol *Use Case diagram*

No	Simbol	Deskripsi
2	Aktor/ <i>actor</i> 	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 aktor.
3.	Asosiasi/ <i>association</i> 	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.
4.	Exstensi/ <i>extend</i> <<extend>>	Relasi <i>use case</i> tambahan sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu, mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek, biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan, misal : 

Lanjutan **Tabel 2.1** Simbol-simbol *Use Case diagram*

No	Simbol	Deskripsi
		Arah panah mengarah pada <i>use case</i> yang ditambahkan, <i>use case</i> yang menjadi <i>extend</i> -nya merupakan jenis yang sama dengan <i>use case</i> yang menjadi induknya.
5.	Generalisasi/ <i>generalization</i> 	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua <i>buah use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya, misalnya :  arah panah mengarah pada <i>use case</i> yang menjadi generalisasinya (umum).
6.	Menggunakan / <i>include / uses</i> <<include>>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini Ada dua sudut pandang yang cukup besar mengenai <i>include</i> di <i>use case</i> : - <i>Include</i> berarti <i>use case</i> yang ditambahkan akan selalu di panggil saat <i>use case</i> tambahan dijalankan, missal pada kasus berikut :

Lanjutan Tabel 2.1 Simbol-simbol *Use Case diagram*

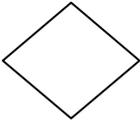
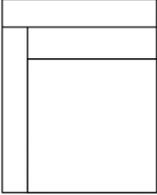
No	Simbol	Deskripsi
		<div style="text-align: center;">  <pre> graph TD Login((Login)) -- <<include>> --> validasi_username((validasi username)) </pre> </div> <p>- <i>Include</i> berarti <i>use case</i> yang tambahan akan selalu melakukan pengecekan apakah <i>use case</i> yang di tambahkan telah dijalankan sebelum <i>use case</i> tambahan di</p> <p>jalankan, misal pada kasus berikut:</p> <div style="text-align: center;">  <pre> graph TD ubah_data((ubah data)) -- <<include>> --> validasi_user((validasi user)) </pre> </div> <p>Kedua interpretasi di atas dapat dianut salah satu atau keduanya tergantung pada pertimbangan dan interpretasi yang dibutuhkan.</p>

Sumber : Sukamto (2018:156-158)

2.2.3 Activity Diagram

Menurut Sukamto (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. diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. Berikut adalah simbol-simbol yang ada pada diagram aktivitas :

Tabel 2.2 Simbol-simbol *activity diagram*

No	Simbol	Deskripsi
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 oleh sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
6.	Swimlane 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

Sumber : Sukamto (2018:162-163)

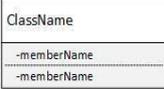
2.2.4 Class Diagram

Menurut Sukamto (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 *method* atau operasi. Berikut penjelasan atribut dan *method* :



1. Atribut merupakan variable-variabel yang dimiliki oleh suatu kelas.
2. Operasi atau *method* adalah fungsi-fungsi yang dimiliki oleh suatu kelas.

Tabel 2.3 Simbol-simbol *class diagram*

No	Simbol	Deskripsi
1.	Kelas 	Kelas pada struktur sistem.
2.	Antarmuka/ <i>interface</i> 	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.
3.	Asosiasi/ <i>association</i> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
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 <i>multiplicity</i> .
5.	Generalisasi 	Relasi antar kelas dengan makna generalisasi – spesialisasi (umum - khusus).
6.	Kebergantungan/ <i>dependensi</i> 	Relasi antar kelas dengan makna kebergantungan antar kelas.
7.	Agregasi/ <i>aggregation</i> 	Relasi antar kelas dengan makna semua-bagian (<i>whole-part</i>).

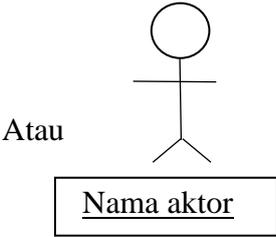
Sumber : Sukanto (2018:146-147)



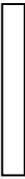
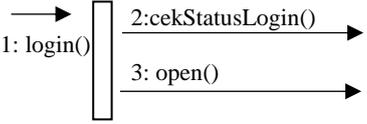
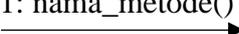
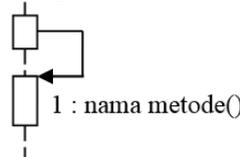
2.2.5 Sequence Diagram

Menurut Sukamto (2018:165), Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dengan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada *use case*. 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 interaksi jalannya pesan sudah dicakup dalam 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 :

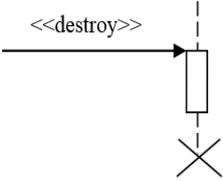
Tabel 2.4 Simbol-simbol *sequence diagram*

No	Simbol	Deskripsi
1.	<p>Aktor</p>  <p>Atau</p> <p>Tanpa Waktu Aktif</p>	<p>Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan dalam menggunakan kata benda diawal frase nama aktor.</p>
2.	<p>Garis hidup/<i>lifeline</i></p> 	<p>Menyatakan kehidupan suatu objek.</p>

Lanjutan Tabel 2.4 Simbol-simbol *sequence diagram*

No	Simbol	Deskripsi
3.	Objek <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 10px auto;"> Nama objek : nama kelas </div>	Menyatakan objek yang berinteraksi pesan.
4.	Waktu aktif 	<p>Menyatakan objek dalam keadaan aktif dan berinteraksi, semuanya yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya, misalnya</p>  <p>Maka cekStatusLogin() dan open() dilakukan didalam metode login(). Aktor tidak memiliki waktu aktif.</p>
5.	Pesan tipe create <<create>> 	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.
6.	Pesan tipe call 1: nama_metode() 	<p>Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri,</p>  <p>Arah panah mengarah pada objek yang memiliki operasi/metode, karena ini memanggil operasi/metode yang</p>

Lanjutan **Tabel 2.4** Simbol-simbol *sequence diagram*

No	Simbol	Deskripsi
		dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi.
7.	Pesan tipe send 1 : masukkan 	Menyatakan bahwa suatu objek mengirimkan data/masukkan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.
8.	Pesan tipe return 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 destroy 	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaliknya jika ada create maka ada destroy.

Sumber : : Sukamto (2018:165-167)

2.3 Penelitian Terdahulu

Menurut Kusuma, dkk. (2018:114) bertujuan untuk membantu proses pemilihan siswa/I teladan karena aplikasi ini memberikan output rekomendasi perangsingan siswa dari nilai alternatif yang terbesar hingga terkecil melalui metode MOORA. Aplikasi sistem pendukung keputusan pemilihan siswa/I teladan telah sesuai dengan spesifikasi kebutuhan fungsional yang diharapkan.



Menurut Alvita, dkk. (2018:66) bertujuan untuk membantu meningkatkan daya kerja mekanik agar menjadi lebih baik lagi dari yang sebelumnya.

Menurut Manurung. (2018:701) Metode MOORA dapat membantu proses penilaian dan sistem pendukung keputusan yang dilakukan ini menggunakan metode moora dan metode moora digunakan untuk menguji coba di dalam correctness yang bertujuan untuk mengetahui akurasi nilai yang diperoleh oleh system, uji coba sensitivitas diberikan pada nilai bobot kriteria dan uji modifikasi yang bertujuan untuk mengetahui seberapa banyak kriteria yang dapat ditambahkan.

Menurut Septi, dkk. (2018:61) bertujuan untuk mengkaji sistem pendukung keputusan penentuan handphone bekas terbaik dan untuk menentukan nilai bobot setiap atribut, proses perankingan yang akan menyeleksi alternatif handphone terbaik dari sejumlah alternatif yang ada. Sistem Pendukung Keputusan membuat proses penentuan Kualitas handphone bekas lebih efisien dibandingkan dengan sistem manual.

Menurut Nisa, dkk. (2018:305) bertujuan untuk memberikan kemudahan bagi masyarakat dalam pemilihan operator seluler yang akan digunakannya dan dapat memberikan hasil yang akurat dalam memilih operator seluler yang diinginkan.