



BAB II

TINJAUAN PUSTAKA

2.1 Teori Umum

2.1.1 Pengertian Komputer

Menurut Abdul Kadir (2017:2), “Komputer merupakan peralatan elektronik yang bermanfaat untuk melaksanakan berbagai pekerjaan yang dilakukan oleh manusia. Meskipun computer berasal dari kata “komputasi”, komputasi yang memang dilaksanakannya mungkin tidak terlihat secara eksplisit. Ketika orang menggunakan computer untuk membuat dokumen, berbagai hitungan yang dilakukan tidak terlihat”.

Menurut Krisbianto (2018:1), “Komputer merupakan alat yang dipakai untuk mengolah dan memproses data menurut perintah yang telah dirumuskan. Komputer berasal dari kata *to compute* yang artinya menghitung. Jadi komputer adalah suatu peralatan elektronik yang digunakan untuk perhitungan”.

2.1.2 Pengertian Data

Menurut Fathansyah (2018:2), “Data adalah representasi fakta dunia nyata yang mewakili suatu objek seperti manusia (pegawai, siswa, pembeli, pelanggan), barang, hewan, peristiwa, konsep, keadaan, dan sebagainya, yang diwujudkan dalam bentuk angka, huruf, symbol, teks, gambar, bunyi, atau kombinasinya”.

Menurut Husda dan Wangdra (2017:13), Data merupakan sesuatu yang belum mempunyai arti bagi penerimanya dan masih memerlukan adanya suatu pengolahan. Data bisa berupa suatu keadaan, gambar, suara, huruf, angka, matematika, bahasa ataupun symbol-simbol lainnya yang bisa kita gunakan sebagai bahan untuk melihat lingkungan, obyek, kejadian ataupun suatu konsep.

2.1.3 Website

Menurut Abdulloh (2018:1), “Website atau disingkat web dapat diartikan sekumpulan halaman yang terdiri atas beberapa laman yang berisi informasi dalam bentuk data digital, baik berupa teks, gambar, video, audio, dan animasi lainnya yang disediakan melalui jalur koneksi internet”.



Menurut Sidik (2017:1), “Situs Web (web site) awalnya merupakan suatu layanan sajian informasi yang menggunakan konsep hyperlink, yang memudahkan surfer (sebutan bagi pemakai komputer yang melakukan penelusuran informasi di Internet) untuk mendapatkan informasi, dengan cukup mengklik suatu link berupa teks atau gambar, maka informasi dari teks atau gambar akan ditampilkan secara lebih rinci (detail)”.

Berdasarkan pengertian di atas, dapat disimpulkan bahwa pengertian website adalah sekumpulan halaman yang terdiri dari beberapa informasi berisi tentang video, gambar, teks, animasi yang diakses dengan menggunakan internet maupun tidak tergantung dengan laman web yang diakses.

2.2 Teori Judul

2.2.1 Pengertian Sistem

Sistem adalah kumpulan dari obyek-obyek seperti orang, *resources*, konsep dan prosedur yang ditujukan untuk melakukan fungsi tertentu atau memenuhi suatu tujuan. Kemudian sistem juga merupakan kumpulan dari komponen yang berinteraksi bersama-sama secara kolektif untuk melaksanakan tugas (Pratiwi, 2016:4).

Menurut Kristanto (2018:1-2) “Sistem merupakan kumpulan elemen-elemen yang saling terkait dan bekerja untuk memproses masukan (input) yang ditunjukkan kepada sistem tersebut dan mengolah masukan tersebut sampai menghasilkan keluaran (output) yang diinginkan.

2.2.2 Pengertian Sistem Pendukung Keputusan

Surbakti (2017:160), sistem pendukung keputusan merupakan Sistem berbasis komputer yang interaktif, yang membantu pengambil keputusan dengan memanfaatkan data dan model untuk menyelesaikan masalah-masalah yang tak terstruktur.

Turban (2017:160), mendefinisikan bahwa sistem pendukung keputusan merupakan suatu pendekatan untuk mendukung pengambilan keputusan. Sistem pendukung keputusan menggunakan data, memberikan antarmuka pengguna



yang mudah.

Berdasarkan pengertian tersebut dapat disimpulkan sistem pendukung keputusan adalah suatu sistem komputer yang dapat mengambil keputusan melalui suatu pendekatan dengan menggunakan data yang belum terstruktur.

2.2.2.1 Tahapan Pengambilan Keputusan

Alur/ proses pemilihan alternatif tindakan/ keputusan biasanya terdiri dari langkah-langkah berikut (Pratiwi, 2016:10-11):

1. Tahap *Intelligence*

Pencarian kondisi-kondisi yang dapat menghasilkan keputusan.

Suatu tahap proses seseorang dalam rangka pengambil keputusan untuk permasalahan yang dihadapi, terdiri dari aktivitas penelusuran, pendeteksian serta proses pengenalan masalah. Data masukan diperoleh, diuji dalam rangka mengidentifikasi masalah.

2. Tahap *Design*

Menemukan, mengembangkan dan menganalisis materi-materi yang mungkin untuk dikerjakan.

Tahap proses pengambil keputusan adalah tahap *intelligence* meliputi proses untuk mengerti masalah, mengenali solusi dan menguji kelayakan solusi. Aktivitas yang biasanya dilakukan seperti menemukan, mengembangkan dan menganalisa alternatif tindakan yang dapat dilakukan.

3. Tahap *Choice*

Pemilihan dari alternatif pilihan yang tersedia, mana yang akan dikerjakan.

Pada tahap ini dilakukan proses pemilihan diantara berbagai alternatif tindakan yang mungkin dijalankan. Hasil pemilihan tersebut kemudian diimplementasikan dalam proses pengambilan keputusan.

4. Tahap *Implementation*

Implementasi dari SPK yang telah dipilih. Tahap implementasi adalah tahap pelaksanaan dari keputusan yang telah diambil.



2.2.2.2 Karakteristik dari Sistem Pendukung Keputusan

Karakteristik dari sistem pendukung keputusan adalah :

- a. Mendukung pengambilan keputusan untuk membahas masalah-masalah terstruktur, semi struktur, dan tidak terstruktur.
- b. Mendukung di semua fase proses pengambilan keputusan: intelegensi, desain, pilihan.
- c. Adanya *interface* manusia atau mesin, dimana manusia (*user*) tetap mengontrol proses pengambilan keputusan
- d. Menggunakan model-model matematis dan statistik yang sesuai dengan pembahasan
- e. Memiliki kemampuan dialog untuk memperoleh informasi sesuai dengan kebutuhan.
- f. Memiliki subsistem-subsistem yang terintegrasi sedemikian rupa sehingga dapat berfungsi sebagai kesatuan sistem
- g. Membutuhkan struktur data komprehensif yang dapat melayani kebutuhan informasi seluruh tingkatan manajemen.
- h. Pendekatan *easy to use*, ciri suatu sistem pendukung keputusan yang efektif adalah kemudahannya untuk digunakan dan memungkinkan keleluasaan pemakai untuk memilih atau mengembangkan pendekatan-pendekatan baru dalam membahas masalah yang dihadapi
- i. Kemampuan sistem untuk beradaptasi secara cepat, dimana pengambil keputusan dapat menghadapi masalah-masalah baru dan pada saat yang sama dapat menanganinya dengan cara mengadaptasikan sistem terhadap kondisi- kondisi perubahan yang terjadi.

2.3 Metode Simple Additive Weighting (SAW)

2.3.1 Pengertian Metode Simple Additive Weighting (SAW)

Eniyati (2017:556), metode *Simple Additive Weighting* (SAW) sering juga dikenal istilah metode penjumlahan terbobot. Konsep dasar metode SAW adalah mencari penjumlahan terbobot dari rating kinerja pada setiap alternatif pada semua



$$r_{ij} = \begin{cases} \frac{x_{ij}}{\text{Max}_i x_{ij}} & \text{jika } j \text{ adalah atribut keuntungan (benefit)} \\ \frac{\text{Min}_i x_{ij}}{x_{ij}} & \text{jika } j \text{ adalah atribut biaya (cost)} \end{cases}$$

atribut. Metode SAW membutuhkan proses normalisasi matriks keputusan

(X) ke suatu skala yang dapat diperbandingkan dengan semua rating alternatif yang ada. Formula untuk melakukan normalisasi tersebut adalah sebagai berikut:

Formula untuk melakukan normalisasi dimana :

r_{ij} : Rating kinerja ternormalisasi

Max_i : Nilai maksimum dari setiap baris

dan kolom Min_i : Nilai minimum dari setiap

baris dan kolom X_{ij} : Baris dan kolom

dari matriks

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

Nilai preferensi untuk setiap alternative (V_i) diberikan sebagai:

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

Formula mencari nilai

preferensi V_i : Nilai

Akhir Alternative

W_i : Bobot yang telah diten-

tukan R_{ij} : Normalisasi

matriks

Nilai V_i yang lebih besar mengindikasikan bahwa alternative a_i lebih terpilih.



2.3.2 Keuntungan Metode Simple Additive Weighting (SAW)

Berikut merupakan keuntungan metode *Simple Additive Weighting* (SAW) :

1. *Simple Additive Weighting* (SAW) memberikan suatu model yang mudah dimengerti, luwes untuk bermacam-macam persoalan yang tidak terstruktur.
2. *Simple Additive Weighting* (SAW) mencerminkan cara berpikir alami untuk memilah-milih elemen-elemen dari suatu system ke dalam berbagai tingkat berlainan dan mengelompokan unsur yang serupa dalam setiap tingkat.
3. *Simple Additive Weighting* (SAW) memberikan suatu skala pengukuran dan memberikan metode untuk menetapkan prioritas.
4. *Simple Additive Weighting* (SAW) memberikan penilaian terhadap konsistensi logis dari pertimbangan-pertimbangan yang digunakan dalam menentukan prioritas.
5. *Simple Additive Weighting* (SAW) menuntun ke suatu pandangan menyeluruh terhadap alternatif-alternatif yang muncul untuk persoalan yang dihadapi.
6. *Simple Additive Weighting* (SAW) memberikan suatu sarana untuk penilaian yang tidak dipaksakan tetapi merupakan penilaian yang sesuai pandangan masing-masing.
7. *Simple Additive Weighting* (SAW) memungkinkan setiap orang atau kelompok untuk mempertajam kemampuan logic dan intuisinya terhadap persoalan yang dipetakan melalui *Simple Additive Weighting* (SAW).

2.3.3 Penerapan Metode Simple Additive Weighting (SAW) pada Sistem Pendukung Keputusan Pemilihan Kenaikan Jabatan Pegawai di PT. Sumsel Energi Gemilang(Perseroda)

Sistem pendukung keputusan pemilihan kenaikan jabatan pegawai menggunakan metode *Simple Additive Weighting* (SAW) pada PT. Sumsel Energi Gemilang adalah suatu sistem yang dapat membantu untuk menentukan pegawai



yang menerima kenaikan jabatan dengan menggunakan metode *Simple Additive Weighting (SAW)* yang mana akan memilih sesuai kriteria yang telah ditemukan.

2.4 Teori Khusus

2.4.1 *Unified Modeling Language (UML)*

2.4.1.1 Pengenalan *Unified Modeling Language (UML)*

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 (Rosa A.S dan M. Shalahudin, 2018:133).

Sugiarti (2018:100) menjelaskan bahwa “*Unified Modeling Language (UML)* adalah sebuah bahasa yang telah menjadi standar dalam industri untuk visualisasi, merancang, dan mendokumentasikan sistem peranti lunak.”

2.4.1.2 Sejarah *Unified Modeling Language (UML)*

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 yang kemudian diikuti dengan perkembangan bahasa pemrograman berorientasi objek yang lainnya seperti C objek, C++, Eiffel, dan CLOS.

Sekitar lima tahun setelah Smalltalk berkembang, maka berkembang pula metode pengembangan berorientasi objek. Karena banyaknya metodologi- metodologi yang berkembang pesat saat itu, maka muncullah ide untuk membuat sebuah bahasa yang dapat dimengerti semua orang. Maka dibuat bahasa yang merupakan gabungan dari beberapa konsep, seperti konsep



Object Modeling Technique (OMT) dari Rumbaugh dan Booch (1991), konsep *The Classes, Responsibilities, Collaborators (CRC)* dari Rebecca Wirfs-Brock (1990), konsep pemikiran Ivar Jacobson, dan beberapa konsep lainnya dimana James R. Rumbaugh, Grady Booch, dan Ivar Jacobson bergabung dalam sebuah perusahaan yang bernama Rational Software Corporation menghasilkan bahasa yang disebut dengan *Unified Modeling Language (UML)*.

Pada tahun 1996, *Object Management Group (OMG)* mengajukan proposal agar adanya standarisasi pemodelan berorientasi objek dan pada bulan September 1997 UML diakomodasi oleh OMG sehingga sampai saat ini UML telah memberikan kontribusinya yang cukup besar di dalam metodologi berorientasi objek dan hal-hal yang terkait di dalamnya (Rosa dan Shalahudin, 2018:138).

2.4.1.3 Diagram *Unified Modeling Language (UML)*

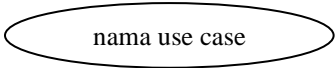
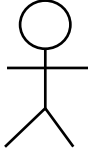

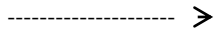
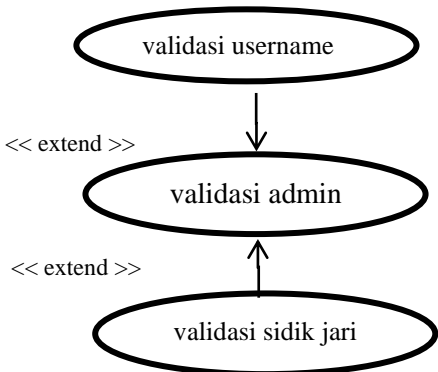
Rosa dan Shalahudin (2018:140), pada *UML* terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori. Berikut ini penjelasan singkat dari pembagian kategori tersebut.

1. *Structure diagram*, yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan. *Structure diagram* terdiri dari *class diagram*, *object diagram*, *component diagram*, *composite structure diagram*, *package diagram* dan *deployment diagram*.
2. *Behavior diagram* yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem. Behavior diagram terdiri dari *Use case diagram*, *Activity diagram*, *State Machine System*.
3. *Interaction diagram* yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar subsistem pada suatu sistem.

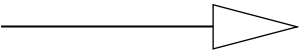
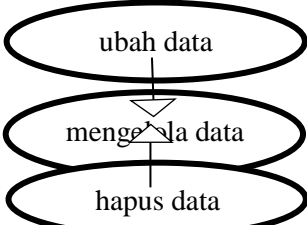

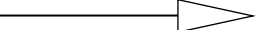
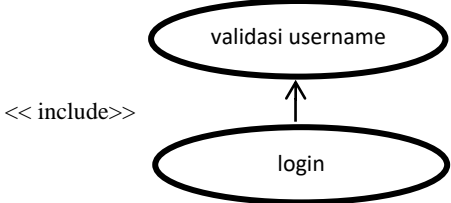
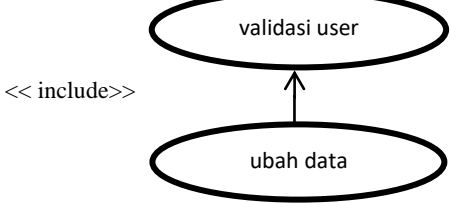
2.4.1.4 Use Case Diagram

Menurut Rosa & 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 actor 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.

Menurut Rosa & Shalahuddin (2018:156-158) simbol-simbol yang ada pada diagram *use case* yaitu:

No.	Simbol	Deskripsi
1.	<p><i>Uses case</i></p> 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau actor, biasanya dinyatakan dengan menggunakan kata kerja di awal di awal frase nama <i>use case</i> .
2.	<p>Aktor / <i>actor</i></p>  <p>nama aktor</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 actor adalah gambar orang, tapi actor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.
3.	<p>Asosiasi / <i>association</i></p> 	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.	<p>Ekstensi / <i>extend</i></p> <p><< extend >></p> 	<p>Relasi <i>use case</i> tambahan ke 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>.</p> <p><i>case</i> yang ditambahkan, misal</p>  <p>arah panah mengarah pada <i>use case</i> yang ditambahkan; biasanya <i>use case</i> yang menjadi <i>extend</i>-nya merupakan jenis yang sama <i>use case</i></p>

Tabel 2.1 Simbol-simbol *Use Case Diagram*


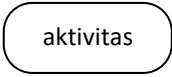



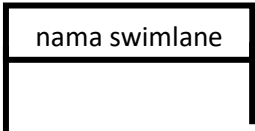
<p>5.</p>	<p>Generalisasi / generalization</p> 	<p>Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya, misalnya:</p>  <p>Arah panah mengaitkan pada <i>use case</i> yang menjadi generalisasinya (umum)</p>
<p>6.</p>	<p>Menggunakan / include / uses</p> <p><<include>></p>  <p><<uses>></p> 	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> di mana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini</p> <p>Ada dua sudut pandang yang cukup besar mengenai include di <i>use case</i>:</p> <ul style="list-style-type: none"> • Include berarti <i>use case</i> yang ditambahkan akan selalu dipanggil saat <i>use case</i> tambahan dijalankan, misal pada kasus berikut:  <ul style="list-style-type: none"> • Include 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 dijalankan, misal pada kasus berikut:  <p>Kedua interpretasi diatas dapat dianut salah satu atau keduanya tergantung ada pertimbangan dan interpretasi yang dibutuhkan.</p>

Sumber: Rosa &Salahudin, 2018.

Lanjutan Tabel 2.1 Simbol-simbol Use Case Diagram

2.4.1.5 Activity Diagram

Rosa & Shalahuddin (2018:161) mengemukakan, “Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan actor, jadi aktivitas yang dapat dilakukan oleh sistem. “Menurut Rosa & Shalahuddin (2018:162—163) simbol-simbol yang ada pada diagram aktivitas:

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.	Pecabangan / <i>decision</i> 	Asosiasi pecabangan 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 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.





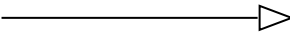
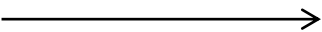
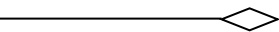
Sumber: Rosa & Salahudin, 2018.

Tabel 2.2 Simbol-simbol *Activity Diagram*

2.3.1.6 Class Diagram

Menurut Rosa & Shalahuddin (2018:141) diagram kelas atau *class diagram* digunakan untuk menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem.

Menurut Rosa & Shalahuddin (2018:146-147) simbol-simbol yang ada pada diagram kelas:

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>dependency</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: Rosa & Salahudin, 2018.




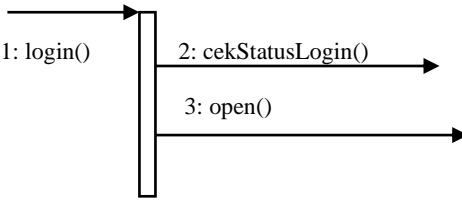

Tabel 2.3 Simbol-simbol *Class Diagram*

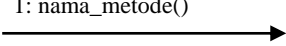
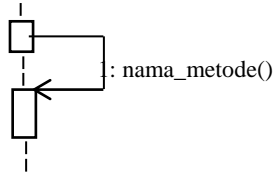
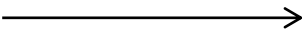
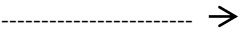
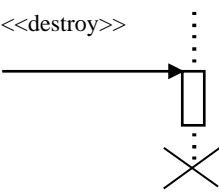
2.4.1.7 Sequence Diagram

Menurut Rosa & Shalahuddin (2018:165) Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambar 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*.

Menurut Rosa & Shalahuddin (2018:165-167) simbol-simbol yang ada pada *sequence diagram* yaitu:

No.	Simbol	Deskripsi
1.	Aktor  atau <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 5px auto;"> nama aktor </div> tanpa waktu aktif	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang kan 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.
2.	Garis hidup / lifeline 	Menyatakan kehidupan suatu objek.
3.	Objek <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 5px auto;"> nama objek : nama kelas </div>	Menyatakan objek yang berinteraksi pesan.
4.	Waktu aktif 	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya, misalnya:  Maka cekStatusLogin() dan open() dilakukan di dalam metode login(). Aktor tidak memiliki waktu aktif.
5.	Pesan tipe create <<create>> 	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.

6.	<p>Pesan tipe call</p> <p>l: nama_metode()</p> 	<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 maka operasi / metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi.</p>
7.	<p>Pesan tipe send</p> <p>l: masukan</p> 	<p>Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.</p>
8.	<p>Pesan tipe return</p> <p>l: keluaran</p> 	<p>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.</p>
9.	<p>Pesan tipe destroy</p> <p><<destroy>></p> 	<p>Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada create maka ada destroy.</p>

Sumber: Rosa &Salahudin, 2018.

Tabel 2.4 Simbol-simbol *Sequence Diagram*

2.5 Teori Program

2.5.1 Bootstrap

Bootstrap merupakan salah satu framework CSS paling populer dari sekian banyak framework CSS yang ada. Bootstrap memungkinkan desain sebuah web menjadi responsif sehingga dapat dilihat dari berbagai macam ukuran *device* dengan tampilan tetap menarik (Abdulloh,2018:261).



2.5.2 CSS (*Cascading Style Sheet*)

CSS adalah singkatan dari *Cascading Style Sheet* yaitu dokumen web yang berfungsi mengatur elemen HTML dengan berbagai property yang tersedia sehingga dapat tampil dengan berbagai gaya yang diinginkan (Abdulloh,2018:45).

2.5.3 Database

Basis data (*database*) dapat didefinisikan sebagai himpunan kelompok data saling berhubungan yang diorganisasikan sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah (Hidayatullah dan Kawistara, 2017:147).

2.5.4 HTML (*Hypertext Markup Language*)

Hypertext Markup Language (HTML) adalah bahasa standard yang digunakan untuk menampilkan halaman *web*. Yang bisa dilakukan HTML yaitu (Hidayatullah dan Kawistara, 2017:13).

1. Mengatur tampilan dari halaman *web* dan isinya.
2. Membuat tabel dalam halamn *web*.
3. Mempublikasikan halaman *web* secara *online*.
4. Membuat form yang bisa digunakan untuk menangani registrasi dan transaksi via *web*.
5. Menambahkan objek-objek seperti citra, audio, video, animasi, *java applet* dalam halaman *web*.
6. Menampilkan area gambar (*canvas*) di *browser*

2.5.5 MySQL

MySQL adalah salah satu aplikasi DBMS yang sudah sangat banyak digunakan oleh para pemrogram aplikasi web. Kelebihan dari *MySQL* adalah gratis, handal, selalu di-*update* dan banyak form yang memfasilitasi para pengguna jika memiliki kendala. *MySQL* juga menjadi DBMS yang sering dibundling dengan *web server* sehingga proses instalasinya jadi lebih mudah (Hidayatullah dan Kawistara, 2017: 180).



2.5.6 PHP (*Hypertext Preprocessor*)

PHP *Hypertext Preprocessor* atau disingkat dengan PHP ini adalah suatu bahasa *scripting* khususnya digunakan untuk *web development*. Karena sifatnya yang *server side scripting*, maka untuk menjalankan PHP harus menggunakan *web server* (Hidayatullah dan Kawistara, 2017: 231).

2.5.7 XAMPP

Sutanto (2017:72) XAMPP merupakan singkatan dari X (empat operasi apapun), Apache, MySQL, PHP, dan Perl. XAMPP merupakan tool yang menyediakan paket perangkat lunak ke dalam satu buah paket. Dalam paketnya sudah terdapat Apache (web server), MySQL (database), PHP (server side scripting), Perl, FTP server, PhpMyAdmin dan berbagai pustaka bantu lainnya.