

BAB III

LANDASAN TEORI

3.1 Pengertian Sistem

Sistem adalah data yang diolah menjadi lebih berguna dan berarti bagi penerimanya, serta untuk mengurangi ketidakpastian dalam proses pengambilan keputusan mengenai suatu keadaan (Tyoso, 2016).

3.2 Pengertian Informasi

Informasi merupakan data yang sudah diolah menjadi sebuah bentuk yang berarti bagi pengguna, yang bermanfaat dalam pengambilan keputusan saat ini atau mendukung sumber informasi (Frisdayanti, 2019).

3.3 Pengertian Sistem Informasi

Pengertian sistem informasi adalah data yang diolah menjadi lebih berguna dan berarti bagi penerimanya, serta untuk mengurangi ketidakpastian dalam proses pengambilan keputusan mengenai suatu keadaan. Suatu kombinasi manusia, fasilitas atau alat teknologi, dan pengendalian bermaksud menata jaringan komunikasi yang penting bagi pengguna atau penerima (Ahmad & Hasti, 2018).

3.4 Pengertian Konsultasi

Konsultasi merupakan suatu bentuk hubungan tolong menolong yang dilakukan oleh seorang profesional yang disebut konsultan. Selain itu konsultasi diartikan sebagai pertimbangan orang terhadap suatu masalah (Abdulghani et al., 2019). Konsultasi biasanya melibatkan pertukaran informasi antara konsultan dan klien, di mana konsultan mendengarkan masalah atau kebutuhan klien, menganalisis situasi dan memberikan solusi yang sesuai. Konsultasi bisa bersifat formal, melibatkan kontrak resmi dan pembayaran, atau informal, di mana orang mencari nasihat dari teman,

kolega, atau ahli dalam lingkup sosial mereka.

3.5 Pengertian Website

Website merupakan kumpulan dari halaman – halaman web yang berhubungan dengan file file lain yang saling terkait. Dalam sebuah *website* terdapat suatu halaman yang dikenal dengan sebutan *homepage*. Dari *home page*, pengunjung dapat mengklik *hyperlink* untuk pindah ke halaman lain yang terdapat dalam *website* tersebut. Sebuah *home page* biasanya merupakan sebuah file dengan nama *index.html* atau *index* (Novendri et al., 2019).

3.6 Pengertian HTML

HTML (*Hyper Text Markup Language*) merupakan sebuah bahasa pemrograman terstruktur yang dikembangkan untuk membuat halaman *website* yang dapat diakses atau ditampilkan menggunakan *web browser*. HTML sendiri secara resmi lahir pada tahun 1989 oleh Tim Berners Lee dan dikembangkan oleh *World Wide Web Consortium*, yang kemudian pada tahun 2004 dibentuklah *Web Hypertext Application Technology Working Group* yang hingga kini bertanggung jawab akan perkembangan bahasa HTML ini (Setiawan, 2017).

3.7 Pengertian CSS

CSS (*Cascading Style Sheet*) secara sederhana adalah sebuah metode yang digunakan untuk mempersingkat penulisan tag HTML, seperti *font*, *color*, *text* dan tabel menjadi lebih ringkas. *Cascading Style Sheet* (CSS) merupakan salah satu bahasa untuk mengatur tampilan dalam web yang berfungsi memisahkan antara desain dengan content (Oktarini et al., 2019). CSS dapat ditulis untuk mendefinisikan ulang tag-tag HTML ataupun dengan menambah atribut *id* dan *class* pada tag HTML (Basuki & Bambang, 2018).

3.8 Pengertian PHP

PHP merupakan singkatan dari *Hypertext Preprocessor* dengan Bahasa yang berbentuk skrip yang bersifat *server side* yang dimana proses pengerjaan

kode program dilalukan di *server*, dan hasilnya akan ditampilkan di *browser*. PHP bekerja di dalam sebuah dokumen *Hypertext Markup Language* (HTML) untuk dapat menghasilkan isi dari sebuah halaman *web* sesuai permintaan. PHP bersifat software *open source* dan juga software *cross platform*, jadi bisa berjalan dengan baik pada sistem operasi *Windows*, *Mac OS*, maupun *Linux* (N. M. Fauzan & Nurhidayah, 2020).

3.9 Pengertian *JavaScript*

JavaScript merupakan bahasa pemrograman tingkat tinggi (*High Level Language*) dan dinamis. *JavaScript* merupakan suatu bahasa *script* yang banyak digunakan dalam dunia teknologi terutama internet, bahasa ini dapat bekerja di sebagian besar *web browser* seperti *Internet Explorer* (IE), *Mozilla Firefox*, *Netscape*, *opera* dan *web browser* lainnya *JavaScript* populer di internet dan dapat bekerja di sebagian besar penjelajah (*browser*) web populer, seperti *Google Chrome*, *Internet Explorer*, *Mozilla Firefox*, *Netscape*, dan *Opera* (Supardi, 2020).

3.10 Pengertian *Mysql*

MySQL merupakan *database engine* atau *server database* yang mendukung bahasa *database SQL* sebagai bahasa interaktif dalam mengelola data. MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL atau DBMS yang *multithread* dan *multi-user* (Fitri, 2020). Kegunaan bahasa SQL yaitu membangun basis data, menjalankan *query* terhadap basis data, melakukan penambahan, pengurangan, perubahan terhadap data yang ada (Pratama et al., 2018).

3.11 Pengertian *XAMPP*

XAMPP merupakan paket PHP yang berbasis *Open Source* yang dikembangkan oleh sebuah komunitas *Open Source*. Dengan menggunakan XAMPP tidak perlu lagi bingung untuk melakukan penginstalan program-program yang lain, karena semua kebutuhan telah disediakan oleh XAMPP.

XAMPP ialah sebuah perangkat lunak program aplikasi (*aplication software*) yang menyediakan *Apache*, *MySQL*, dll. Sebenarnya tanpa aplikasi ini kita juga dapat membuat *database MySQL* dan juga menjalankan *PHP* pada komputer kita, yaitu dengan cara instal *apache* dan *MySQL* secara langsung. Tetapi XAMPP menyediakan *interface* atau tampilan yang memudahkan para pengguna, apalagi bagi para pemula (Muhidin et al., 2017).

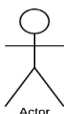
3.12 Pengertian UML




UML (*Unified Modeling Language*) merupakan bahasa untuk visualisasi, spesifikasi, membangun sistem perangkat lunak, serta dokumentasi. Secara khusus, UML menspesifikasi langkah-langkah penting dalam pengembangan keputusan analisis, perancangan, serta implementasi dalam sistem perangkat lunak (Nugroho & Rohimi, 2020). Alat bantu yang di gunakan untuk perancangan berorientasi objek menggunakan UML sebagai berikut.

a. Use Case Diagram

Use Case Diagram merupakan pemodelan untuk sistem informasi yang akandibuat. *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 pada *use case* ditunjukkan pada Tabel 3.1 *Use Case Diagram* (Satzinger et al., 2015).

Tabel 3. 1 *Use Case Diagram*



Simbol	Nama	Keterangan
	<i>Actor</i>	Seseorang atau <i>admin</i> yang berhubungan dengan sistem yang sedang dibangun.







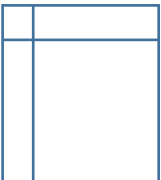
	<i>Use Case</i>	Digunakan untuk menggambarkan aktivitas apa saja yang dapat dilakukan oleh <i>actor</i> .
	<i>System</i>	Sistem yang sedang dikembangkan.
	<i>Assosiation</i>	Menggambarkan interaksi antara aktor dengan use case. Mengindikasikan apa dan siapa yang meminta interaksi secara langsung, bukan mengindikasikan data.

b. Activity Diagram

Activity Diagram digunakan untuk mendokumentasikan aliran aktivitas untuk satu kasus penggunaan. *Activity Diagram* yang menggambarkan alur kerja, aktivitas dari suatu sistem atau pelaku atau proses bisnis. Menurut beberapa ahli, *Activity Diagram* merupakan gambaran rinci dari proses *use case diagram*. Simbol-simbol pada *activity diagram* ditunjukkan pada Tabel 3.2 *Activity Diagram* (Satzinger et al., 2015).

Tabel 3. 2 *Activity Diagram*

Simbol	Deskripsi
	<i>Start Point</i> Menggambarkan status awal sistem.
	<i>Stop Point</i> Status akhir yang dilakukan oleh sistem.




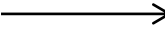

	Aktivitas Menggambarkan suatu proses atau kegiatan bisnis.
	Control Flow Menunjukkan rangkain pelaksaan.
	Penggabungan/ <i>Join</i> Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
	<i>Decision</i> Aktivitas yang tidak dapat dilakukan secara bersamaan.
	<i>Fork</i> Menunjukkan kegiatan yang dilakukan secara paralel.
	<i>Join</i> Menunjukkan kegiatan yang digabungkan.
	<i>Swimlane</i> Digunakan untuk membedakan apa yang dikerjakan oleh actor dan apa yang dikerjakan oleh sistem

c. *Sequence Diagram*

Sequence Diagram digunakan untuk menggambarkan aliran informasi masuk dan keluar dari bagian otomatis sistem. *Sequence Diagram* merupakan alat yang efektif untuk membantu dalam desain awal antarmuka pengguna dengan mengidentifikasi informasi spesifik yang mengalir dari pengguna ke dalam sistem dan informasi yang mengalir keluar dari sistem kembali ke pengguna. *Sequence Diagram*

memberikan gambaran visual yang jelas tentang urutan peristiwa dalam suatu skenario atau proses, membantu pengembang dan analis sistem untuk memahami dinamika sistem dengan lebih baik. Simbol-simbol pada *Sequence Diagram* ditunjukkan pada Tabel 3.3 *Sequence Diagram* (Satzinger et al., 2015).

Tabel 3. 3 *Sequence Diagram*

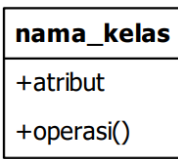

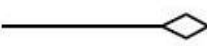
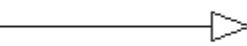
Gambar	Nama	Keterangan
	<i>Lifeline</i>	Objek entity, antarmuka yang saling berinteraksi
	<i>Note</i>	Menunjukkan catatan untuk komentar dari suatu pesan bersifat opsional.
	<i>Actor</i>	Menspesifikasi himpunan peran yang pengguna mainkan ketika berinteraksi dengan sesama actor atau lifeline.
	<i>Message</i>	Symbol untuk pengiriman pesan.
	<i>Lifeline</i>	Garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i> .

d. *Class Diagram*

Class Diagram UML digunakan untuk menunjukkan kelas objek untuk suatu sistem. Salah satu jenis diagram kelas UML yang menunjukkan hal-hal dalam masalah pengguna domain disebut diagram

kelas model domain. *Class Diagram* digambar dengan menunjukkan kelas dan asosiasi antar kelas. Selain itu, masalah yang lebih kompleks tentang kelas dan asosiasi dapat diilustrasikan dengan menggunakan diagram kelas model domain. Berikut merupakan simbol pada *class diagram* yang ditunjukan pada Tabel 3.4 *Class Diagram* (Satzinger et al., 2015).

Tabel 3. 4 *Class Diagram*

Gambar	Nama	Keterangan
	Class and Attribute	<p>Kelas adalah penggambaran dari struktur sistem yang akan di tampilkan dalam sistem informasi.</p> <p>Attribute adalah penggambaran mengenai keadaan dari suatu objek didalam kelas.</p>
	Associaton	Relasi antar kelas dengan makna umum,asosiasi biasanya juga disertai dengan <i>multiplecities</i> .
	Agregation	Relasi antar kelas dengan makna semua-bagian (whole-part)
	Generalisasi	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)

3.13 Pengujian Sistem

Metode adalah kegiatan dimana suatu sistem atau komponen dieksekusi dibawah kondisi tertentu, hasilnya diamati atau dicatat untuk kemudian dievaluasi berdasarkan aspek sistem atau komponen. Teknik pengujian pada penelitian ini menggunakan *black box testing*, dimana akan terlihat hasil pengujiannya dengan bentuk tabel dengan field hasil pengujian

dan kesimpulan yang dapat dilihat hasilnya (Masripah & Ramayanti, 2020).

3.13.1 White Box Testing

White box testing merupakan pengujian perangkat lunak pada tingkat alur kode program, apakah masukan dan keluaran yang sesuai dengan spesifikasi yang dibutuhkan dan pengujian yang didasarkan pada pengujian design program secara prosedural, secara struktural, pengujian berbasis logika atau pengujian berbasis kode. Metode jalur dasar adalah salah satu metode white box testing, di mana dalam proses pengujian diperlukan untuk membuat flow graph dari program skrip dan juga menentukan nilai kompleksitas siklomatik. Tes ini bertujuan untuk menganalisis kebenaran struktur program yang dibuat dan kinerja program (Pratala et al., 2020).

3.13.2 Black Box Testing

Metode *Black box Testing* adalah sebuah metode yang dipakai untuk menguji sebuah software tanpa harus memperhatikan detail software. Proses *Black Box Testing* dengan cara mencoba program yang telah dibuat dengan mencoba memasukkan data pada setiap formnya. Pengujian ini diperlukan untuk mengetahui program tersebut berjalan sesuai dengan yang dibutuhkan oleh perusahaan (Made et al., 2021).