

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Pengertian Sistem**

Sistem adalah sebuah istilah yang secara kolektif mendeskripsikan fase-fase awal pengembangan sistem. Analisis sistem adalah teknik pemecahan masalah yang menguraikan bagian-bagian komponen tersebut bekerja dan berinteraksi untuk mencapai tujuan mereka [4]

#### **2.2 Pengertian Informasi**

Informasi adalah hasil pengolahan data sehingga menjadi bentuk yang penting bagi penerimannya dan mempunyai kegunaan sebagai dasar dalam pengambilan keputusan yang dapat dirasakan akibatnya secara langsung saat itu juga atau secara tidak langsung pada saat mendatang [5]

#### **2.3 Pengertian Sistem Informasi**

Sistem informasi adalah suatu rangkaian sistem yang dikelompokkan kedalam suatu organisasi yang terdiri dari kumpulan komponen baik yang berbasis computer maupun berbasis manual yang dibuat untuk mengumpulkan serta menyiapkan data-data yang berisikan informasi keluaran untuk pemakai atau kumpulan dari perangkat keras dan perangkat lunak yang dihubungkan guna menciptakan dan memproses data menjadi informasi yang berguna [6]

#### **2.4 Pengertian Rancang Bangun**

Rancang bangun merupakan penggambaran, perencanaan, dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam suatu kesatuan yang utuh dan berfungsi. Dengan demikian pengertian rancang bangun merupakan kegiatan menerjemahkan hasil analisa ke dalam bentuk paket perangkat lunak kemudian menciptakan sistem tersebut atau memperbaiki sistem yang sudah ada [7]

## 2.5 Pengertian Absensi

Absensi adalah pencatatan dan pengolahan kehadiran data yang dilakukan terus menerus, pencatatan dilakukan setiap jam kerja dan melapor ke HRD/perusahaan pengelola [8]

## 2.6 Website

Website memiliki pengertian yaitu sekumpulan halaman khusus yang dibuat dengan cara khusus yang isinya hanya ada di situs internet dan isinya memuat informasi tertentu. Website juga dapat dijelaskan menjadi gabungan dari halaman-halaman yang ada di internet dan isinya memuat informasi berupa data digital baik secara tulisan, suara, foto, maupun animasi yang telah tersedia dan dapat diakses menggunakan internet [9]

## 2.7 Javascript

JavaScript adalah bahasa yang digunakan untuk membuat program yang digunakan agar dokument HTML yang ditampilkan pada sebuah Browser menjadi lebih interaktif, tidak sekedar indah saja. JavaScript memberikan beberapa fungsionalitas ke dalam halaman web, sehingga dapat menjadi sebuah program yang disajikan dengan menggunakan antar muka web.

JavaScript merupakan bahasa script, yaitu bahasa yang tidak memerlukan kompiler untuk dapat menjalankannya, tetapi cukup dengan Interpreter. Tidak perlu ada proses kompilasi terlebih dahulu agar program dapat dijalankan. Browser web Netscape Navigator dan Internet Explorer adalah salah satu contoh dari salah banyak interpreter, karena kedua browser ini telah dilengkapi dengan Interpreter JavaScript. Tetapi tidak semua browser web dapat menjadi interpreter javascript karena belum tentu browser tersebut dilengkapi dengan interpreter JavaScript [10]

## 2.8 MySQL

Mysql bukan termasuk Bahasa pemrograman. Mysql merupakan salah satu database populer dan mendunia. Mysql bekerja menggunakan SQL language (Structure Query Language). Itu dapat diartikan bahwa mysql merupakan standar penggunaan database di dunia untuk pengolahan data [11]

## 2.9 PHP

PHP merupakan secara umum dikenal sebagai bahasa pemrograman script script yang membuat dokumen HTML secara on the fly yang dieksekusi di server web, dokumen HTML yang dihasilkan dari suatu aplikasi bukan dokumen HTML yang dibuat dengan menggunakan editor teks atau editor HTML. Dikenal juga sebagai bahasa pemrograman server side [12]

## 2.10 HTML

Pada mulanya web merupakan sebuah ruang informasi internet, melalui penggunaan teknologi hypertexts, pemakai akan dituntun menemukan informasi melalui link yang sudah disediakan oleh dokumen web pada tampilan browser web. Hyper Text Markup Language (HTML) merupakan suatu rangkaian kode pada program yang menjadi basic representasi visual halaman web. HTML terdiri dari kumpulan beberapa informasi yang tersimpan dalam bentuk tag tertentu, tag-tag tersebut dapat digunakan untuk melakukan format informasi yang dimaksud [13]

## 2.11 CSS

CSS adalah singkatan dari Cascading Style Sheet yang merupakan kumpulan perintah yang dibentuk dari berbagai sumber yang disusun menurut urutan tertentu sehingga mampu mengatasi konflik style atau bahasa yang digunakan untuk mengatur tampilan suatu dokumen yang ditulis dalam bahasa markup / markup language[14]

## 2.12 UML

UML atau *Unified Modelling Language* adalah sebuah visualisasi, merancang dan mendokumentasikan sistem piranti lunak. Salah satu standar yang digunakan diperindustrian untuk mengetahui dan mendefinisikan requirement, analisis dan desain, dari suatu arsitektur dalam pemrograman berorientasi objek [15]

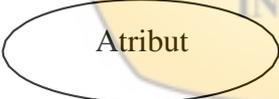
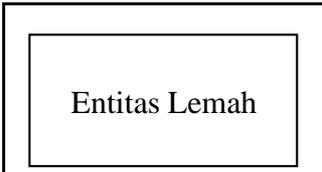
### 2.12.1 ERD

Entity Relationship Diagram (ERD) merupakan diagram yang digunakan untuk merancang tabel-tabel yang nantinya akan diimplementasikan pada basis data. ERD ini dibentuk berdasarkan tiga elemen, entitas, atribut, dan relasi. Entitas adalah objek dalam bentuk fisik atau konsep [16]

Elemen-elemen diagram hubungan entitas adalah sebagai berikut :

Adapun simbol-simbol yang digunakan dalam *Entity Relationship diagram* adalah sebagai berikut:

**Tabel 2.1 Simbol-Simbol ERD**

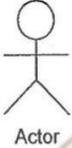
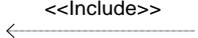
No	Simbol	Deskripsi
1		Jenis Entitas dapat berupa suatu elemen lingkungan, sumber daya atau transaksi yang <i>field-fieldnya</i> dipergunakan dalam aplikasi program.
2		Menunjukkan nama relasi antar satu entitas dengan entitas lainnya
3		Atribut adalah karakteristik dari sebuah entitas
4		Menunjukkan hubungan (keterkaitan) antar entitas.
5		Entitas yang kemunculannya tergantung dari entitas lain yang lebih kuat.

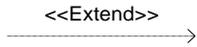
### 2.12.2 Use Case Diagram

*Use Case Diagram* bersifat statis, diagram ini memperlihatkan himpunan use case dan aktor-aktor (suatu jenis khusus dari kelas). Diagram ini sangat penting untuk mengorganisasi dan memodelkan perilaku dari suatu sistem yang dibutuhkan serta diharapkan pengguna.

Adapun simbol-simbol yang digunakan dalam *use case diagram* adalah sebagai berikut :

**Tabel 2.2 Simbol-Simbol Use Case Diagram**

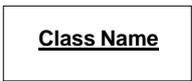
No	Simbol	Deskripsi
1	 Actor	<i>Actor/Role</i> , yaitu orang atau sistem lain yang berinteraksi dengan sistem saat ini.
2	 Use Case	<i>Use Case</i> adalah bagian utama dari fungsionalitas sistem. Bisa <i>extend</i> (memperluas) <i>use case</i> lainnya. Ditempatkan di dalam <i>system boundary</i> (batasan sistem). Dilabeli dengan kata kerja-frase kata benda.
3	 Subject Boundary	<i>Subject Boundary</i> , Berisi nama dari sistem yang diletakkan di dalam atau dibagian atas <i>boundary</i> . Mewakili ruang lingkup sistem. <i>Actor</i> berada di luar ruang lingkup sistem.
4	 * *	<i>Association Relationship</i> , menghubungkan <i>actor</i> dengan <i>use case</i> . Menunjukkan komunikasi dua arah (menunjukkan komunikasi satu arah jika menggunakan tanda panah). Tanda * untuk keragaman dari asosiasi ( <i>multiplicity of the association</i> ). Namun umumnya hanya digambarkan garis saja.
5	 <<Include>>	<i>Include Relationship</i> , merupakan satu <i>use case</i> dalam <i>use case</i> lainnya, perilaku ( <i>behavior</i> ) yang harus terpenuhi agar sebuah <i>event</i> dapat terjadi,

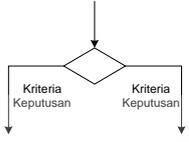
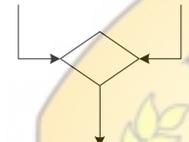
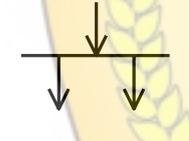
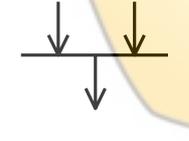
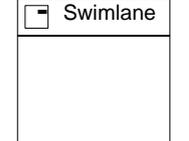
		dimana kondisi ini sebuah <i>use case</i> adalah bagian dari <i>use case</i> lainnya. Tanda panah mengarah dari <i>base use case</i> (pusat) menuju ke <i>use case</i> yang di-include.
6		<i>Extend Relationship</i> , memperluas <i>use case</i> untuk memasukkan perilaku opsional. Tanda panah mengarah dari <i>use case</i> tambahan ke <i>base use case</i> (pusat).
7		<i>Generalitisation Relationship</i> , mewakili <i>use case</i> khusus untuk <i>use case</i> yang lebih umum. Tanda panah mengarah dari <i>use case</i> khusus ( <i>specialized</i> ) ke <i>use case</i> yang umum.

### 2.12.3 Activity Diagram

*Activity* diagram secara esensial mirip dengan flowchart atau diagram alur yang menunjukkan aliran kendali dari sebuah aktivitas ke aktivitas lainnya. Dalam *activity diagram* terdapat aksi atau aktivitas, *activity nodes*, flows atau aliran, dan objek. Simbol-simbol yang dipakai dalam *activity diagram* yaitu:

**Tabel 2.3 Simbol-Simbol Activity Diagram**

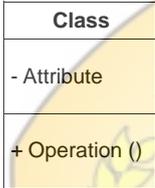
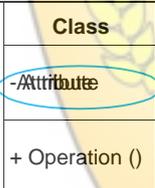
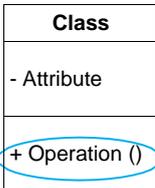
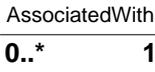
No	Simbol	Deskripsi
1		<i>Action</i> , perilaku yang sederhana dan tidak dapat diuraikan. Dilabeli dengan namanya.
2		<i>Activity</i> , digunakan untuk mewakili sekumpulan tindakan ( <i>action</i> ). Dilabeli dengan namanya.
3		<i>Object Node</i> , digunakan untuk mewakili sebuah objek yang terhubung ke sekumpulan arus objek. Dilabeli dengan nama <i>class</i> -nya.
4		<i>Control Flow</i> , menunjukkan urutan eksekusi.
5		<i>Object Flow</i> , menunjukkan aliran suatu objek dari satu aktivitas (atau tindakan) ke aktivitas (atau tindakan) lain.

6		<i>Initial Node</i> , menggambarkan awal dari serangkaian tindakan atau kegiatan.
7		<i>Final-Activity Node</i> , digunakan untuk menghentikan semua arus kontrol dan arus objek dalam suatu aktivitas (atau tindakan).
8		<i>Final-Flow Node</i> , digunakan untuk menghentikan aliran kontrol atau aliran objek tertentu.
9		<i>Decision Node</i> , digunakan untuk mewakili kondisi pengujian untuk memastikan bahwa aliran kontrol atau aliran objek hanya turun satu jalur. Dilabeli dengan kriteria keputusan untuk melanjutkan ke jalur tertentu.
10		<i>Merge Node</i> , digunakan untuk menyatukan kembali berbagai jalur keputusan yang dibuat menggunakan simpul keputusan.
11		<i>Fork Node</i> , digunakan untuk membantu perilaku menjadi seperangkat aktivitas yang paralel atau bersamaan dari aktivitas (atau tindakan).
12		<i>Join Node</i> , digunakan untuk menyatukan kembali serangkaian arus aktivitas (atau tindakan) yang paralel atau bersamaan.
13		<i>Swimlane</i> , digunakan untuk memecah diagram aktivitas menjadi baris dan kolom untuk menetapkan kegiatan individu (atau tindakan) kepada individu atau objek yang bertanggung jawab untuk melaksanakan aktivitas (atau tindakan). Dilabeli dengan nama individu atau objek yang bertanggung jawab.

### 2.12.4 Class Diagram

*Class* adalah sebuah spesifikasi yang akan menghasilkan objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu system. Class diagram menggambarkan struktur dan deskripsi kelas, package beserta hubungan satu sama lain. Simbol-simbol yang digunakan dalam class diagram yaitu:

**Tabel 2.4 Simbol-Simbol Class Diagram**

No	Simbol	Deskripsi
1		<i>Class</i> , mewakili orang, tempat, atau hal-hal yang dibutuhkan sistem untuk menangkap dan menyimpan informasi. Memiliki nama yang diketik dengan huruf yang tebal dan berpusat dibagian atas kotak. Memiliki daftar atribut di kotak tengahnya. Memiliki daftar operasi di kotak bawahnya. Tidak secara eksplisit menunjukkan operasi yang tersedia untuk semua kelas.
2		<i>Attribute</i> , merupakan properti yang menggambarkan keadaan suatu objek. Dapat diturunkan dari atribut lain, ditampilkan dengan menempatkan garis miring sebelum nama atribut.
3		<i>Operation</i> , mewakili tindakan atau fungsi yang dapat dilakukan oleh kelas. Dapat diklasifikasikan sebagai konstruktor, permintaan, atau operasi pembaruan. Termasuk tanda kurung yang mungkin berisi parameter atau informasi yang diperlukan untuk melakukan operasi.
4		<i>Generalization</i> , mewakili hubungan sejenis diantara beberapa kelas.
5		<i>Association</i> , mewakili hubungan antar beberapa kelas atau kelas dan dirinya sendiri. Dilabeli

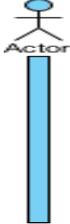
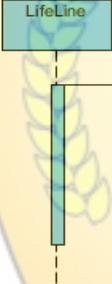
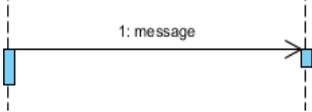
		menggunakan frasa kata kerja atau nama peran, yang lebih baik mewakili hubungan. Dapat ada di antara satu atau beberapa kelas. Berisi simbol multisplitas, yang mewakili waktu minimum dan maksimum <i>instance</i> kelas dapat dikaitkan dengan <i>instance</i> kelas terkait.
6		<i>Aggregation</i> , merupakan hubungan sebagian (a part of) yang logis diantara beberapa kelas atau suatu kelas dengan kelas itu sendiri. <i>Aggregation</i> terbuat dari objek-objek yang bisa dibagi atau ditukar. Angka disebelah kanan boleh satu, boleh juga banyak (*).
7		<i>Composition</i> , mewakili hubungan fisik antara beberapa kelas atau kelas dan kelas itu sendiri. <i>Composition</i> terdiri dari objek-objek yang tidak bisa dibagi atau ditukar dan hidup selama <i>composite</i> objek hidup. Angka disebelah kanan hanya boleh 1.

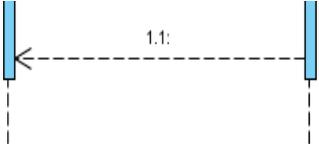
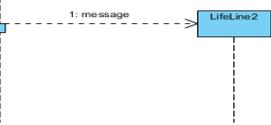
### 2.12.5 Sequence Diagram

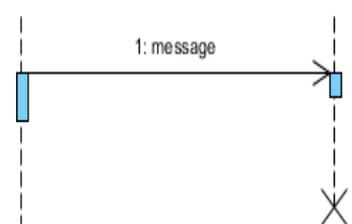
*Sequence* Diagram menggambarkan interaksi antara objek di dalam dan disekitar sistem (termasuk pengguna, display, dan sebagainya) berupa message yang digambarkan terhadap waktu, menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respon dari sebuah event untuk menghasilkan output tertentu. Collaboration diagram juga menggambarkan interaksi antara objek seperti sequence diagram, akan tetapi lebih menekankan pada masing masing objek dan bukan pada waktu penyampaian message. Setiap message memiliki sequence number, dimana message dari level tertinggi memiliki nomor 1.

Simbol-simbol yang digunakan dalam sequence diagram dijelaskan pada tabel :

Tabel 2.5 Simbol-Simbol Sequence Diagram

No	SIMBOL	DESKRIPSI
1		<p>Actor/Aktor, Jenis peran yang dimainkan oleh entitas yang berinteraksi dengan subjek (Misalnya dengan bertukar sinyal dan data) Di luar subjek (maksudnya adalah batasan/kendala dari aktor bukan bagian dari batasan/kendala dari subjek yang sesuai).</p>
2		<p>Lifeline, atau Garis hidup mewakili peserta individu dalam Interaksi.</p>
3		<p>Activation box, atau kotak aktivasi berbentuk sebuah persegi panjang tipis pada lifeline, mewakili periode di mana suatu elemen melakukan operasi.</p> <p>Bagian atas dan bawah dari kotak aktivasi disejajarkan dengan inisiasi dan waktu penyelesaian masing-masing</p>
4		<p>Call message atau pesan panggilan merupakan sebuah pesan yang mendefinisikan komunikasi tertentu antara Lifelines dari sebuah interaksi.</p> <p>Call Message adalah jenis pesan yang mewakili permintaan operasi dari target lifeline.</p>

5		<p>Return message atau pesan balik adalah sebuah pesan yang mendefinisikan komunikasi tertentu antara Lifelines dari sebuah interaksi.</p> <p>Return message adalah jenis pesan yang mewakili informasi yang dikirimkan kembali ke pengirim pesan atau pemanggil berdasarkan pesan sebelumnya.</p>
6		<p>Self message atau pesan mandiri adalah sebuah pesan yang mendefinisikan komunikasi tertentu antara Lifelines dari sebuah interaksi.</p> <p>Self message adalah jenis pesan yang mewakili permohonan pesan dari lifeline yang sama.</p>
7		<p>Recursive Message atau pesan rekursif merupakan sebuah pesan yang mendefinisikan komunikasi tertentu antara Lifelines dari sebuah interaksi.</p> <p>Recursive Message merupakan jenis pesan yang mewakili permohonan pesan dari lifeline yang sama. Recursive Message ini menargetkan aktivasi di atas aktivasi tempat pesan itu berasal.</p>
8		<p>Create Message merupakan sebuah pesan yang mendefinisikan komunikasi tertentu antara Lifelines dari sebuah interaksi.</p> <p>Create message atau membuat pesan adalah</p>

		jenis pesan yang mewakili Instansiasi (target) lifeline.
9		<p>Destroy Message merupakan sebuah pesan yang mendefinisikan komunikasi tertentu antara Lifelines dari sebuah interaksi.</p> <p>Destroy message atau hancurkan pesan merupakan sebuah jenis pesan yang mewakili permintaan untuk menghancurkan siklus hidup target.</p>
10		<p>Duration message atau pesan durasi yaitu sebuah pesan yang mendefinisikan komunikasi tertentu antara Lifelines dari sebuah interaksi. Pesan Durasi menunjukkan jarak antara dua batasan waktu untuk permohonan pesan.</p>
11		<p>Note atau Catatan (komentar) merupakan catatan atau lampiran berbagai komentar ke elemen. Sebuah komentar tidak memiliki kekuatan semantik, tetapi dapat berisi informasi yang berguna bagi pemodel.</p>