

## **BAB II TINJAUAN PUSTAKA**

### **2.1 Sistem Informasi**

Sistem informasi adalah suatu yang berkaitan dengan sistem manajerial atau pengelolaan. Sistem informasi ini merupakan kombinasi dari prosedur kerja, informasi, individu, dan teknologi informasi yang terorganisir (Siahaan, 2020).

Pengertian sistem informasi merupakan kombinasi dari teknologi informasi serta aktivitas dari orang-orang yang menggunakan teknologi tersebut guna mendukung operasi serta manajemen. Apabila diartikan secara luas, istilah dari sistem informasi sering digunakan untuk merujuk pada interaksi yang terjadi di antara orang, proses algoritmik, teknologi serta data (Jogiyanto, 2017).

### **2.2 Monitoring**

E-Monitoring merupakan pemantauan dan pelaporan dengan penyampaian data secara elektronik (online) serta dapat dipantau secara terus menerus untuk penilaian terhadap kualitas dan efektivitas sistem pengendalian untuk meyakinkan bahwa prosedur berjalan sebagaimana mestinya (Saputra *et al.*, 2019).

### **2.3 Mobile**

Aplikasi *mobile* yaitu program siap pakai yang direkap untuk melaksanakan suatu fungsi bagi pengguna atau aplikasi yang lain dan dapat digunakan oleh sasaran yang dituju sedangkan *mobile* dapat di artikan sebagai perpindahan dari suatu tempat ketempat yang lain (Maldhan, 2019).

Maka aplikasi *mobile* dapat di artikan sebuah program aplikasi yang dapat dijalankan atau digunakan walaupun pengguna berpindah-pindah dari satu tempat ketempat yang lain serta mempunyai ukuran yang kecil. HTML 5 adalah HTML5 merupakan bahasa markah untuk menstrukturkan dan menampilkan isi dari jejagat jembar, sebuah teknologi yang dipakai sebagai standar dari internet, HTML5 adalah revisi ke lima dari HTML dengan tujuan untuk mempermudah penerapan pada media teknologi terbaru (Nazli, 2019).

## 2.4 Web

Website adalah media yang digunakan untuk menampung data teks, gambar, suara, dan animasi yang dapat ditampilkan di internet/dan dapat diakses oleh computer yang terhubung dengan internet secara global. Website merupakan media informasi berbasis jaringan komputer yang dapat diakses dimana saja dengan biaya relatif murah (Oetomo and Mahargiono, 2020). Website merupakan bentuk implementasi dari bahasa pemrograman web (web programming). PHP (Hypertext Preprocessor) merupakan Bahasa pemrograman berbasis web yang memiliki kemampuan untuk memproses/dan mengolah data secara dinamis (Wulandari and Riyanto, 2021).

## 2.5 MySql

MySQL adalah sebuah *database management system* (manajemen basis data) menggunakan perintah dasar SQL (*Structured Query Language*) yang cukup terkenal. Database *management system* (DBMS) MySQL multi pengguna dan bersifat gratis. Mysql digunakan sebagai wadah dalam mengelola data yang dapat disimpan digunakan kembali dengan cara yang lebih efisien (Setyawan and Pratiwi, 2019).

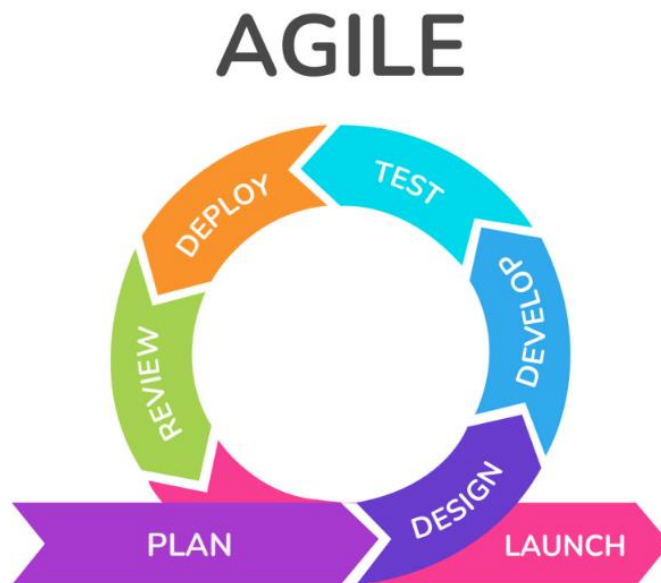
## 2.6 PHP (*Hypertext Preprocessor*)

*Hypertext Preprocessor* atau hanya PHP saja, adalah bahasa skrip dengan fungsi umum yang terutama digunakan untuk pengembangan *web*. Bahasa ini awalnya dibuat oleh seorang pemrogram Denmark-Kanada Rasmus Lerdorf pada tahun 1994. Implementasi referensi PHP sekarang diproduksi oleh The PHP Group. Salah satu Bahasa scripting open source yang banyak digunakan oleh Web Developer untuk pengembangan Web (Oetomo and Mahargiono, 2020).

PHP adalah bahasa penulisan skrip sisi server yang memiliki berbagai kegunaan, tapi yang paling umum adalah untuk membuat konten website yang dinamis. Selain itu, PHP sangat populer karena sifatnya yang open-source dan serbaguna (Fajri, Bahar and Setiawan, 2020).

## 2.7 Metode Agile

*Agile* adalah metode pengembangan proyek yang menggunakan siklus pengembangan yang singkat atau disebut dengan "sprint" yang berfokus pada peningkatan berkelanjutan dalam pengembangan suatu produk atau layanan. Metode Pengembangan Agile memiliki 12 prinsip utama yang dijadikan dasar acuan dalam pengembangan proyek (Nova, Widodo and Warsito, 2022).



**Gambar 2.1** Metode *Agile*  
Sumber : (Rosa dan Shalahuddin, 2019)

Alasan utama penerapan metodologi Agile yaitu mampu memberikan fleksibilitas yang tinggi terhadap perubahan kebutuhan proyek. Dengan fokus pada pengembangan bertahap, tim dapat dengan mudah menyesuaikan perencanaan mereka sesuai dengan perubahan dalam kebutuhan pelanggan atau kondisi pasar. Maka proses yang cepat dalam membangun sistem yang dibutuhkan telah sesuai dengan metode yang diterapkan menggunakan agile.

Prinsip-prinsip yang dimiliki oleh metode pengembangan perangkat lunak Agile yaitu sebagai berikut.

1. Kepuasan pelanggan/pengguna menjadi prioritas utama
2. Menerima perubahan kebutuhan/persyaratan, meskipun di tahapan akhir pengembangan layanan. Kebutuhan/persyaratan yang dirubah akan

dimanfaatkan menjadi suatu keunggulan kompetitif dengan proses yang tangkas.

3. Pengiriman proses tahapan pengembangan/perilisan proyek secara rutin agar mendapatkan feedback yang lebih baik.
4. Kolaborasi antara pengembang proyek dan stakeholder.
5. Bekerja dengan motivasi yang tinggi.
6. Komunikasi tatap muka dengan metode yang efisien dan efektif.
7. Software yang berfungsi dengan baik adalah kunci utama dari kemajuan.
8. Proses yang cepat dalam pengerjaan proyek
9. Perhatian terhadap keunggulan desain dan teknis yang baik.
10. Kesederhanaan - seni memaksimalkan jumlah pekerjaan yang belum selesai.
11. Arsitektur, persyaratan, dan desain terbaik muncul dari tim yang mengatur dirinya sendiri.
12. Secara berkala, tim merefleksikan bagaimana menjadi lebih efektif, kemudian menyesuaikan dan menyesuaikan perilakunya.


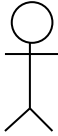

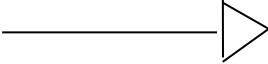
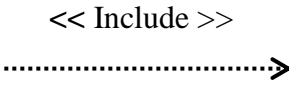

## **2.8 *Unified Modelling Language (UML)***

UML (*Unified Modelling Language*) adalah suatu metode dalam pemodelan secara visual yang digunakan sebagai sarana perancangan sistem berorientasi objek dan merupakan sekumpulan diagram yang digunakan untuk melakukan abstraksi terhadap sebuah sistem atau perangkat lunak berbasis objek. UML dapat digunakan untuk mempermudah pengembangan aplikasi yang berkelanjutan.

### **2.8.1 *Use Case Diagram***

*Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut (Rosa dan Salahuddin, 2019). Berikut simbol-simbol yang akan digunakan dalam menggambarkan *Use Case Diagram* dapat dilihat pada Tabel 2.1.



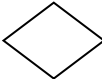

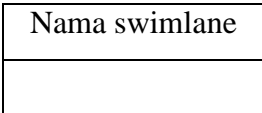

**Tabel 2.1** Simbol *Use Case Diagram*

No	Simbol	Deskripsi
1.		<i>Use case</i> : Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal <i>frase</i> nama <i>use case</i> .
2.		Aktor: seseorang/sesuatu yang berinteraksi dengan yang akan dibuat. diluar sistem informasi. Biasanya dinyatakan menggunakan kata benda
3.		Asosiasi ( <i>association</i> ): merupakan 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.		Generalisasi ( <i>generalization</i> ): merupakan hubungan (umum – khusus) antara dua buah <i>use case</i> dimana fungsi yang satu yang lebih umum
5.		Include berarti <i>use case</i> yang ditambahkan akan dipanggil saat <i>use case</i> tambahan dijalankan.
6.		Ekstensi ( <i>extend</i> ) merupakan <i>use case</i> tambahan ke sebuah <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu.

### 2.8.2 Activity Diagram

*Activity* diagram menggambarkan aliran kerja atau aktivitas dari sebuah sistem atau proses bisnis atau menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem (Rosa dan Salahuddin, 2019). Berikut simbol-simbol yang akan digunakan dalam menggambarkan *activity diagram* dapat dilihat pada Tabel 2.2.

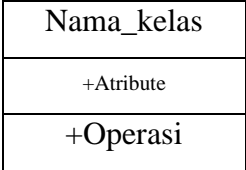
**Tabel 2.2** Simbol *Activity Diagram*


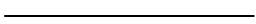
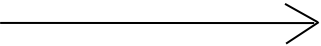
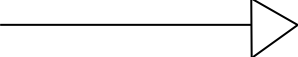
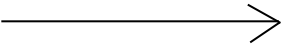
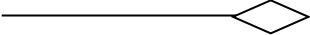
No.	Simbol	Keterangan
1.		Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
2.		Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3.		Percabangan ( <i>Decision</i> ) merupakan asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
4.		Penggabungan ( <i>Join</i> ) merupakan asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
5.		Swimlane Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas.
6.		Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.

### 2.8.3 Class Diagram

*Class diagram* mengembangkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem (Rosa dan Salahuddin, 2019). Berikut simbol-simbol yang akan digunakan dalam menggambarkan *Class Diagram* dapat dilihat pada Tabel 2.3.

**Tabel 2.3** Simbol *Class Diagram*

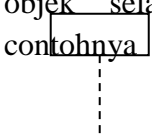
No.	Simbol	Deskripsi
1.		Kelas pada struktur sistem.

No.	Simbol	Deskripsi
2.	Antar Muka/Interface  Nama_Interface	Sama dengan konsep interface dalam pemrograman berorientasi objek.
3.	Asosiasi / Association 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>symbol</i>
4.	Asosiasi Berarah / <i>Digunakan Association</i> 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>symbol</i> .
5.	Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)
6.	Ketergantungan / dependency 	Relasi antar kelas dengan makna ketergantungan antar kelas.
7.	Agregasi / <i>aggregation</i> 	Relasi antar kelas dengan makna semua bagian ( <i>whole-part</i> )

#### 2.8.4 Sequence Diagram

Diagram *Sequence* menjelaskan bagaimana suatu operasi itu dilakukan; message (pesan) apa yang dikirim dan kapan pelaksanaannya (Rosa A.S. dan Shalahuddin, 2019). Berikut simbol *sequence diagram* pada Tabel 2.4.

**Tabel 2.4** Simbol *Sequence Diagram*

No.	Simbol	Deskripsi
1.	<i>Object lifeline</i> 	Menggambarkan panjang kehidupan suatu objek selama scenario sedang di buat contohnya

2.	<i>Activation</i>	Dimana proses sedang dilakukan oleh <i>object</i> atau <i>class</i> untuk memenuhi pesan atau perintah
3.	<i>Message</i>	Sebuah anak panah yang mengindikasikan pesan diantara objek. Dan objek dapat mengirimkan pesan ke dirinya sendiri

## 2.9 Pengujian *Black Box Testing*

Pengujian black-box adalah metode pengujian perangkat lunak yang memeriksa fungsionalitas aplikasi tanpa mengintip ke dalam struktur atau cara kerja internalnya. Metode pengujian ini dapat diterapkan secara virtual ke setiap tingkat pengujian perangkat lunak: unit, integrasi, sistem, dan penerimaan. Blackbox Testing adalah uji coba fungsionalitas sebuah aplikasi atau program yang sedang dikembangkan. Sementara Whitebox Testing adalah metode uji coba struktur internal, seperti pengujian pada code aplikasi (Rosa and Shalahuddin, 2019).

Pengujian black-box memungkinkan penguji dan developer bekerja sama secara independen tanpa mengganggu proses kerja satu sama lain. Guna menemukan inkonsistensi program, pengujian dilakukan berdasarkan sudut pandang pengguna.