

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terdahulu

Berdasarkan penelitian yang dilakukan oleh Faradilla, dkk (2018) mengenai aplikasi absensi siswa untuk kemudahan informasi antara pihak sekolah dan orang tua pada sekolah dasar negeri 3 banda aceh berbasis android, Faradilla, dkk menyimpulkan bahwa aplikasi absensi siswa berbasis android berhasil memberi pemberitahuan kehadiran siswa kepada orang tua.

Berdasarkan penelitian yang dilakukan oleh Husain, dkk (2017) mengenai perancangan sistem absensi *online* menggunakan *android* guna mempercepat proses kehadiran karyawan pada PT. Sintech Berkah Abadi, Husein, dkk menyimpulkan bahwa dengan adanya sistem absensi online ini, karyawan dapat mengoptimalkan pekerjaan dengan sangat baik. Karena sistem absensi ini diakses sangat mudah dan waktu pun dapat digunakan secara efisien.

Berdasarkan penelitian yang dilakukan oleh Putra, dkk (2018) mengenai aplikasi absensi perkuliahan berbasis *android*, Putra, dkk menyimpulkan bahwa Aplikasi absensi perkuliahan berbasis *android* memberikan kemudahan kepada mahasiswa dalam melakukan proses absen, dan untuk dosen dalam mengontrol absensi para mahasiswa.

Berdasarkan penelitian yang dilakukan oleh Sumolang, dkk (2018) mengenai aplikasi absensi jemaat berbasis android, Sumolang, dkk menyimpulkan bahwa pelaporan persembahan dapat di implementasikan dengan menggunakan

android yang dijalankan pada ponsel pintar pada setiap peribadatan di tingkat kolom.

Dari keempat penelitian terdahulu diatas yang menggunakan sistem aplikasi absensi berbasis *android* untuk berbagai kasus, maka peneliti menggunakan aplikasi absensi dan impal guru berbasis *android*. Dari keempat penelitian terdahulu diatas juga tidak ditemukan adanya kesamaan penelitian dengan penelitian ini sehingga keempat penelitian terdahulu diatas dapat dijadikan referensi untuk penelitian ini.

II.2. Landasan Teori

Berikut ini adalah beberapa landasan teori yang dikutip dari beberapa jurnal referensi yang berkaitan dengan penelitian ini :

II.2.1. Sistem

Sistem yaitu sekumpulan prosedur organisasi yang pada saat dilaksanakan akan memberikan informasi bagi pengambil keputusan atau untuk mengendalikan organisasi. Menurut Azhar Susanto (2007), kumpulan dari sub-sub sistem baik fisik maupun non fisik yang saling berhubungan satu sama lain dan bekerja sama secara harmonis untuk mencapai tujuan yaitu mengolah data menjadi informasi yang berguna. (Ikhlas, 2018 : 48).

II.2.2. Aplikasi

Aplikasi adalah suatu sub kelas perangkat lunak komputer yang memanfaatkan kemampuan komputer langsung untuk melakukan suatu tugas yang

diinginkan pengguna. Biasanya dibandingkan dengan perangkat lunak sistem yang mengintegrasikan berbagai kemampuan komputer, tapi tidak secara langsung menerapkan kemampuan tersebut untuk mengerjakan suatu tugas yang menguntungkan pengguna. (Sumolang, dkk, 2018 : 17).

II.2.3. Absensi

Absensi adalah daftar kehadiran pegawai/siswa, yang berisi jam datang, jam pulang, serta alasan/keterangan kehadiran pegawai. Absensi juga merupakan suatu daftar pendataan kehadiran seseorang dari suatu aktifitas di sebuah institusi yang diatur dan disusun sedemikian rupa sehingga dapat digunakan sewaktu - waktu oleh pihak yang berkepentingan. Dimana data dari pegawai yang tidak hadir akan tercatat di daftar kepegawaian dan dapat di periksa kapan saja oleh pihak instansi tersebut. Dalam bahasa Inggris, pemakaian kata absen sering digunakan sebagai istilah *List of Absent*, yang berarti seseorang yang tidak hadir dalam suatu pertemuan sedangkan pemakaian kata kehadiran sering digunakan sebagai istilah *List of Presence* atau *List of Participants*. (Sumolang, dkk, 2018 : 1).

II.2.4. Web Server

Web server merupakan sebuah perangkat lunak dalam *server* yang berfungsi menerima permintaan (*request*) berupa halaman *web* melalui HTTP atau HTTPS dari klien yang dikenal dengan *browser web* dan mengirimkan kembali (*response*) hasilnya dalam bentuk halaman-halaman *web* yang umumnya berbentuk dokumen HTML. (Kosasih, dkk, 2015 : 115).

II.2.5. *Android*

Android adalah sebuah sistem operasi untuk perangkat *mobile* berbasis *linux* yang mencakup sistem operasi, *middleware* dan aplikasi. *Android* menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka. Awalnya, Google Inc. membeli Android Inc. Yang merupakan pendatang baru yang membuat piranti lunak untuk ponsel atau *smartphone*. Kemudian untuk mengembangkan *Android* dibentuklah *Open Handset Alliance* (OHA), konsorium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi termasuk Google, HTC, Intel, Motorola, Qualcomm, TMobile dan Nvidia. (Sumolong, dkk, 2018 : 2).

Android merupakan OS (*Operating System*) *Mobile* yang tumbuh ditengah OS lainnya yang berkembang dewasa ini. OS lainnya seperti Windows Mobile, iPhone OS, Symbian, dan masih banyak lagi. Akan tetapi, OS yang ada ini berjalan dengan memprioritaskan aplikasi inti yang dibangun sendiri tanpa melihat potensi yang cukup besar dari aplikasi pihak ketiga. Oleh karena itu, adanya keterbatasan dari aplikasi pihak ketiga untuk mendapatkan data asli ponsel, berkomunikasi antar proses serta keterbatasan distribusi aplikasi pihak ketiga untuk platform mereka. Dapat disimpulkan bahwa *Android* adalah sistem operasi atau OS yang dikembangkan oleh linux guna menjalankan perangkat *smartphone* sehingga *smartphone* dapat digunakan dengan tampilan dan performa yang baik. (Husain, dkk, 2017 : 118).

II.2.6. *Hypertext Markup Language (HTML)*

Hypertext Markup Language (HTML) adalah bahasa standard yang digunakan untuk menampilkan halaman *web*. Yang bisa dilakukan dengan HTML yaitu :

1. Mengatur tampilan dari halaman *web* dan isinya.
2. Membuat tabel dalam halaman *web*.
3. Mempublikasikan halaman *web* secara *online*.
4. Membuat *form* yang bisa digunakan untuk menangani registrasi dan transaksi via *web*. Contoh: Setiap dokumen HTML diawali dan diakhiri dengan tag HTML. (Novendri, dkk, 2019 : 48).

II.2.7. *Hypertext Preprocessor (PHP)*

PHP adalah singkatan dari *Hypertext Preprocessor*. PHP adalah bahasa *server-side scripting* yang menyatu dengan *Hypertext Markup Language (HTML)* untuk membuat halaman web yang dinamis. Maksud dari *server-side scriping* adalah sintaks dan perintah-perintah yang diberikan akan sepenuhnya dijalankan di *server* tetapi disertakan pada dokumen HTML sebagai pembangun halaman *web*. (Kosasih, dkk, 2015 : 115).

II.2.8. MySQL

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (bahasa Inggris : *database management system*) atau DBMS yang *multithread*, *multi-user*, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL AB membuat

MySQL tersedia sebagai perangkat lunak gratis di bawah lisensi GNU General Public License (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL. (Kosasih, dkk, 2015 : 115).

MySQL merupakan sistem manajemen *database* yang bersifat relational. Artinya, data yang dikelola dalam database yang akan diletakkan pada beberapa tabel yang terpisah sehingga manipulasi data akan jauh lebih cepat. MySQL dapat digunakan untuk mengelola database mulai dari yang kecil sampai dengan yang sangat besar. (Novendri, dkk, 2019 : 49).

II.2.9. UML

UML adalah bahasa spesifik standar yang dipergunakan untuk mendokumentasikan, menspesifikan dan membangun perangkat lunak. UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML yaitu *Use Case Diagram*, *Activity Diagram*, *Class Diagram* dan *Sequence Diagram*. (Urva dan Siregar, 2015 : 95).

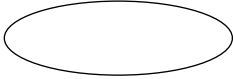
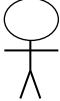

Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut :


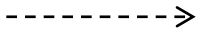
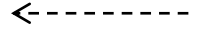
1. *Use Case Diagram*

Use case diagram 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. Dapat dikatakan

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 yang digunakan dalam *use case* diagram dapat dilihat pada tabel II.1 dibawah ini :

Tabel II.1. Simbol Use Case

Gambar	Keterangan
	<p><i>Use case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, dan dinyatakan dengan menggunakan kata kerja di awal nama <i>use case</i>.</p>
	<p>Aktor adalah <i>abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan <i>use case</i>, tetapi tidak memiliki <i>control</i> terhadap <i>use case</i>.</p>
	<p>Asosiasi antara aktor dan <i>use case</i>, digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengidikasikan aliran data.</p>




	Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem.
	<i>Include</i> , merupakan di dalam <i>use case</i> lain (<i>required</i>) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya adalah pemanggilan sebuah fungsi program.
	<i>Extend</i> , merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.

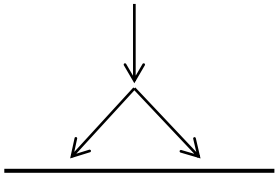
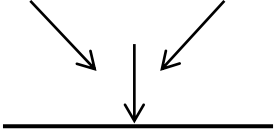
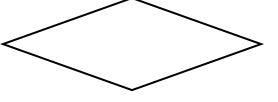
(Sumber : Urva dan Siregar, 2015 : 94)

2. Diagram Aktivitas (*Activity Diagram*)

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity diagram* dapat dilihat pada tabel II.2 dibawah ini :

Tabel II.2. Simbol *Activity Diagram*

Gambar	Keterangan
	<i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	<i>End point</i> , akhir aktifitas.
	<i>Activites</i> , menggambarkan suatu proses/kegiatan bisnis.

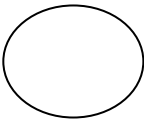
	<i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.
	<i>Join</i> (penggabungan) atau <i>rake</i> , digunakan untuk menunjukkan adanya dekomposisi.
	<i>Decision Points</i> , menggambarkan pilihan untuk pengambilan keputusan, <i>true</i> , <i>false</i> .
<div style="border: 1px solid black; padding: 2px; display: inline-block;">New Swimlane</div>	<i>Swimlane</i> , untuk menunjukkan siapa melakukan apa.

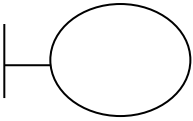
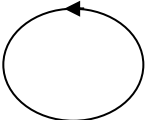
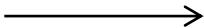
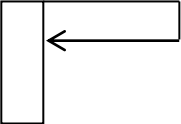

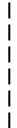
(Sumber : Urva dan Siregar, 2015 : 94)

3. Diagram Urutan (*Sequence Diagram*)

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam *sequence diagram* dapat dilihat pada tabel II.3 dibawah ini :

Tabel II.3. Simbol *Sequence Diagram*

Gambar	Keterangan
	<i>Entity Class</i> , merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.

	<p><i>Boundary Class</i>, berisi kumpulan kelas yang menjadi <i>interface</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan formentry dan <i>form</i> cetak.</p>
	<p><i>Control class</i>, suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.</p>
	<p><i>Message</i>, simbol mengirim pesan antar <i>class</i>.</p>
	<p><i>Recursive</i>, menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.</p>
	<p><i>Activation</i>, <i>activation</i> mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.</p>
	<p><i>Lifeline</i>, garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i>.</p>

(Sumber : Urva dan Siregar, 2015 : 95)

4. *Class Diagram* (Diagram Kelas)

Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem. *Class diagram* juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan. *Class diagram* secara khas

meliputi: Kelas (*Class*), Relasi, *Associations*, *Generalization* dan *Aggregation*, Atribut (*Attributes*), Operasi (*Operations/Method*), *Visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *multiplicity* atau kardinaliti yang dapat dilihat pada tabel II.4 dibawah ini :

Tabel II.4. *Multiplicity Class Diagram*

Multiplicity	Penjelasan
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih
1..*	1 atau lebih
0..1	Boleh tidak ada, maksimal 1
n..n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimum 4

(Sumber : Urva dan Siregar, 2015 : 95)