

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terkait

Untuk mendukung keberhasilan penelitian ini, penulis melakukan pendekatan teoritis melalui beberapa literatur yang berhubungan dengan penelitian yang dilakukan. Beberapa uraian penelitian terdahulu yang menjadi acuan dalam penelitian ini yaitu:

1. I, Wayan, D, S & Dewa, P, Y, A. 2020, dengan judul “Sistem Absensi Kegiatan Ormawa Berbasis *QR Code*”. Penelitian ini membantu menginputkan absensi suatu kegiatan dengan mudah dan cepat, serta mampu menjaga integritas data yang diinputkan.
2. Johanes, T, O, G., dkk. 2020, dengan judul “Rancang Bangun Sistem Informasi Akademi: Modul Sistem Absensi Berbasis *Mobile* dan *Web* pada Universitas Universal”. Penelitian ini menghasilkan fitur *QR Code* pada sistem absensi mahasiswa dapat digunakan sebagai media akses mahasiswa untuk melakukan absensi dikelas.
3. Khaerul Anam dan Asep, T, M. 2018, dengan judul “Analisa Dan Perancangan Sistem Informasi Akademik Berbasis *Web* Pada Mi Al-Mursyidiyyah Al-‘Asyrotussyafi’iyyah”. Penelitian ini dapat membantu dan mempercepat proses pencatatan data guru, siswa, kelas, pembuatan jadwal pelajaran dan penilaian menggunakan sistem akademik berbasis *web*.

4. Melan Susanti. 2016, dengan judul “Perancangan Sistem Informasi Akademik Berbasis *Web* Pada SMK Pasar Minggu Jakarta”. Penelitian ini dapat disimpulkan bahwa penggunaan sistem informasi akademik berbasis *web* dapat memberikan solusi yang lebih cepat dalam pengolahan nilai, pembuatan laporan lebih efektif dan efisien karena penyimpanan data sudah dalam *database*.
5. M. Rizki Alpiandi. 2016, dengan judul “Sistem Informasi Akademik Berbasis *Web* di SMP Negeri 2 Kecamatan Gaung Anak Serka”. Dengan adanya sistem informasi akademik berbasis *web* ini, sangat membantu pihak sekolah dalam mengelola administrasi data sekolah seperti data guru, data siswa, data nilai, data mata pelajaran, data pengajar, dan data jadwal pelajaran.
6. M. Teguh Prihandoyo. 2018, dengan judul “*Unified Modeling Language* (UML) Model Untuk Pengembangan Sistem Informasi Akademik Berbasis *Web*”. Penelitian ini memberikan sebuah solusi dalam pengolahan data nilai mahasiswa sehingga terjadinya efektifitas pelayanan kepada mahasiswa. Integrasi sistem dengan database yang dirancang menjadikan sistem informasi lebih efektif sehingga proses penyimpanan data, pencarian data yang tersimpan tidak redundan.
7. Nataniel Degen dan Dyna Marisa Kh. 2016, dengan judul “Sistem Informasi Akademik Berbasis *Web* SMP Negeri 4 Samarinda”. Penelitian ini memberikan informasi mengenai daftar data pegawai, data pengajar, data kelas, data sarana dan prasarana yang dimiliki serta data jadwal mata pelajaran pada tiap-tiap kelas.

II.2. Landasan Teoritis

II.2.1. Sistem

Menurut Reza Fahlevi Ahmad dan Novrini Hasti (2018), sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau menyelesaikan suatu sasaran tertentu.

II.2.2. Informasi

Menurut Reza Fahlevi Ahmad dan Novrini Hasti (2018), informasi adalah data yang diolah menjadi bentuk yang lebih berguna bagi penggunaannya.

II.2.3. Sistem Informasi

Menurut Bodnar dan Hopwood di dalam jurnal Nisma Apriani (2019) Sistem informasi adalah kumpulan perangkat keras dan perangkat lunak yang dirancang untuk mentransformasikan data ke dalam bentuk informasi yang berguna.

II.2.4. Sistem Informasi Akademik

Menurut Ahmar di dalam jurnal Johanes, dkk (2020) Sistem Informasi Akademi merupakan suatu aplikasi yang dirancang/dikembangkan untuk membantu lembaga pendidikan dalam mengelola data-data yang bersifat akademi. Sistem informasi akademi terdiri dari beberapa modul/menu di antaranya modul referensi, modul penerimaan mahasiswa baru, modul guru dan pelajaran, modul

jadwal dan kalender, modul kesiswaan, modul presensi/absensi, modul penilaian, modul kenaikan dan kelulusan, modul mutasi serta modul lainnya.

II.3. Basis Data

Menurut Anisah dan Mayasari di dalam jurnal Sri Mulyati, dkk, (2018) basis data merupakan kumpulan file-file yang saling berelasi, relasi dihubungkan dengan kunci tiap file yang ada. Satu *database* menunjukkan kumpulan data yang dipakai dalam ruang lingkup.

II.4. ERD (*Entity Relationship Diagram*)

Menurut Sukamto dan Salahudin di dalam jurnal Muhdar Abdurahman, dkk, (2018) ERD dikembangkan berdasarkan teori himpunan dalam bidang matematika. ERD digunakan untuk pemodelan basis data relational. Sehingga jika penyimpanan basis data menggunakan OODBMS maka perancangan basis data tidak perlu menggunakan ERD.

II.5. Normalisasi

Menurut Indrajani di dalam jurnal Sri Mulyati, dkk, (2018) normalisasi adalah suatu teknik dengan pendekatan *bootom-up* yang digunakan untuk membantu mengidentifikasikan hubungan. Dimulai dari menguji hubungan *functional dependencies* antara atribut. Pengertian lainnya adalah suatu teknik yang menghasilkan sekumpulan hubungan dengan sifat-sifat yang diinginkan dengan memenuhi kebutuhan pada perusahaan.

II.6. *Hypertext Preprocessor (PHP)*

PHP adalah bahasa pemrograman *server-side* yang dirancang untuk perkembangan web. PHP dikatakan *server-side* karena program yang diberikan akan dijalankan/diproses pada komputer yang bertindak sebagai server.

Menurut Supono di dalam Intan Lestari (2019) PHP adalah suatu bahasa pemrograman yang digunakan untuk menerjemahkan baris kode program menjadi kode mesin yang dapat dimengerti oleh komputer yang bersifat *server-side* yang dapat ditambahkan ke dalam HTML.

II.7. *MySQL*

MySQL adalah *database open source* terpopuler di dunia. Dengan kinerja, kehandalan dan kemudahan penggunaan yang terbukti, MySQL telah menjadi pilihan *database* terdepan untuk aplikasi berbasis *web*, yang digunakan oleh properti *web* profil tinggi termasuk Facebook, Twitter, YouTube, Yahoo! dan banyak lagi. Kata "SQL" dari "MySQL" adalah singkatan dari "*Structured Query Language*". SQL adalah bahasa standar yang paling umum digunakan untuk mengakses *database* (Cosmas, dkk, 2017).

II.8. *UML*

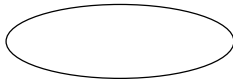
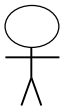
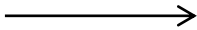
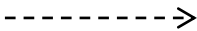
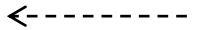
Menurut M. Teguh Prihandoyo (2018) *Unified Modeling Language* merupakan salah satu metode pemodelan visual yang digunakan dalam perancangan dan pembuatan sebuah *software* yang berorientasikan pada objek. UML merupakan sebuah standar penulisan atau semacam *blue print* dimana

didalamnya termasuk sebuah bisnis proses, penulisan kelas-kelas dalam sebuah bahasa yang spesifik. Terdapat beberapa diagram UML yang sering digunakan dalam pengembangan sebuah sistem, yaitu:

1. *Use case Diagram*

Merupakan gambaran dari fungsionalitas yang diharapkan dari sebuah sistem, dan merepresentasikan sebuah interaksi antara aktor dan sistem. Didalam *use case* terdapat actor yang merupakan sebuah gambaran entitas dari manusia atau sebuah sistem yang melakukan pekerjaan di sistem.. Simbol-simbol yang digunakan dalam *use case* diagram dapat dilihat pada tabel II.1 dibawah ini:

Tabel II.1. Simbol *Use Case*

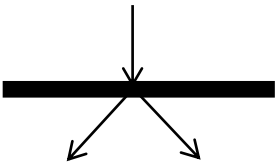
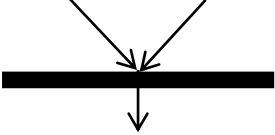
Gambar	Keterangan
	<i>Use case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama <i>use case</i> .
	Aktor adalah <i>abstraction</i> dari orang atau sistem yang lain yang 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 control terhadap <i>use case</i> .
	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 mengindikasikan aliran data.
	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: Gellysa Urva dan Helmi Fauzi Siregar, 2015)

2. Diagram Aktivitas (*Activity Diagram*)

Merupakan gambaran alir dari aktivitas-aktivitas didalam sistem yang berjalan. Simbol-simbol yang digunakan dalam *activity diagram* dapat dilihat pada tabel II.2:

Tabel II.2. Simbol *Activity Diagram*

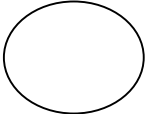
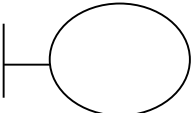
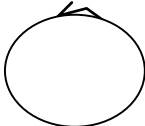
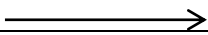
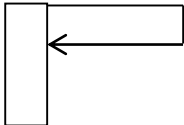
Gambar	Keterangan
	<i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	<i>End point</i> , akhir aktifitas.
	<i>Activites</i> , menggambarkan suatu proses/kegiatan bisnis.
	<i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan paralel menjadi satu.
	<i>Join</i> (penggabungan) atau rake, digunakan untuk menunjukkan adanya dekomposisi.
	<i>Decision Points</i> , menggambarkan pilihan untuk pengambilan keputusan, <i>true</i> , <i>false</i> .
	<i>Swimlane</i> , pembagian <i>activity diagram</i> untuk menunjukkan siapa melakukan apa.

(Sumber : Gellysa Urva dan Helmi Fauzi Siregar, 2015)

3. Diagram Urutan (*Sequence Diagram*)

Menggambarkan interaksi antar objek didalam dan di sekitar sistem yang berupa *message* yang digambarkan terhadap waktu. Simbol-simbol yang digunakan dalam *sequence diagram* dapat dilihat pada tabel II.3:

Tabel II.3. Simbol *Sequence Diagram*

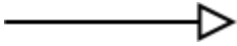
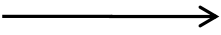
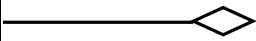
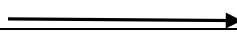
Gambar	Keterangan
	<i>Entity Class</i> , merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
	<i>Boundary Class</i> , berisi kumpulan kelas yang menjadi <i>interface</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan formentry dan <i>form</i> cetak.
	<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.
	<i>Message</i> , simbol mengirim pesan antar <i>class</i> .
	<i>Recursive</i> , menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.
	<i>Activation</i> , <i>activation</i> mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.
	<i>Lifeline</i> , garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i> .

(Sumber : Gellysa Urva dan Helmi Fauzi Siregar, 2015)

4. Diagram Kelas (*Class Diagram*)

Merupakan gambaran struktur dan deskripsi dari *class*, *package*, dan objek yang saling berhubungan seperti diantaranya pewarisan, asosiasi dan lainnya. Simbol *class diagram* dapat dilihat pada Tabel II.4. dibawah ini:

Tabel II.4. Simbol Class Diagram

Gambar	Keterangan
	<i>Generalization</i> , untuk menghubungkan antar kelas dengan arti umum-khusus. Jadi jika ada kelas dengan arti umum-khusus. Jadi jika ada kelas bermakna umum dan kelas bermakna khusus dapat menggunakan simbol ini.
Nama_kelas +atribut +operasi	<i>Class</i> , untuk sebuah kelas pada struktur sistem. Penulisan tidak boleh menggunakan spasi. Simbol ini memiliki 3 susunan, yaitu kotak pertama adalah nama kelas, kedua atribut dan ketiga operasi.
	<i>Interface</i> , untuk simbol <i>interface</i> atau dalam bahasa indonesianya antar muka. Konsep yang digunakan pun sama dengan pemrograman berorientasi object (OOP).
	<i>Association</i> , digunakan untuk menghubungkan atau merelasikan kelas satu dengan kelas yang lainnya dengan makna umum.
	<i>Directed Association</i> , adalah relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain.
	<i>Aggregation</i> , adalah relasi antar kelas dengan makna semua bagian.
	<i>Dependency</i> , adalah relasi antar kelas dengan makna kebergantungan antar kelas.

(Sumber : Gellysa Urva dan Helmi Fauzi Siregar, 2015)