

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terkait

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

Menurut penelitian yang dilakukan oleh Rudi Hermawan (2016) yaitu Sistem Informasi Kegiatan Belajar Mengajar Berbasis Web Aplikasi ini menggunakan metode *System Development Life Cycle* dimana palikasi dapat menghasilkan suatu sistem informasi kegiatan belajar mengajar berbasis web yang dapat mengatasi pengolahan jadwal belajar mengajar lebih terintegrasi dan terkelola dengan baik.

Menurut penelitian yang dilakukan oleh Irvan Azhary Chusna (2017) yaitu aplikasi Sistem Informasi Penjadwalan Kegiatan Belajar Mengajar Berbasis Web. Aplikasi dengan metode *System Development Life Cycle*, dimana aplikasi dapat dijalankan secara efisien dengan data yang menjadi inputannya adalah data guru, dan jam belajar yang akan menghasilkan suatu sistem informasi kegiatan belajar mengajar yang dapat diakses secara *online* dan sudah diuji dibidang pendidikan pada sekolah SMP Negeri 2 Dawarblondong, Mojokerto.

Menurut penelitian yang dilakukan oleh Udin Sidik Sidin (2016) melakukan penelitian yaitu Sistem Informasi Penjadwalan Mata Pelajaran Berbasis Web.

Aplikasi ini menggunakan metode *System Development Life Cycle* dengan inputan data yang berfokus pada data mata pelajaran. Penelitian ini menghasilkan suatu sistem untuk mempermudah dalam mengelola jadwal mata pelajaran.

Menurut penelitian yang dilakukan oleh Wahyu Triyanto (2017) melakukan penelitian yaitu Sistem Informasi Penjadwalan Mengajar Dan Pengolahan Nilai Raport Secara Multiuser Pada SMK Bhinneka Karya Simo Boyolali. Aplikasi ini menggunakan metode HIPO yang berfokus pada inputan data Guru, data kelas, data mata pelajaran, data *ekstrakurikuler*, data siswa, data jadwal mengajar, data *raport* dan data nilai kepribadian. Aplikasi berbasis *windows* dan dapat diakses secara langsung tanpa membutuhkan jaringan internet. Penelitian ini menghasilkan suatu sistem yang dapat membuat aplikasi dan Sistem Komputerisasi Penjadwalan Mengajar dan Pengolahan Nilai Raport pada SMK Bhinneka Karya Simo.

Menurut penelitian yang dilakukan oleh M Rizal Fahmi (2016) melakukan penelitian yaitu Sistem Informasi Penjadwalan Mata Pelajaran Pada Madrasah Tsanawiyah Ar Rohman Kencong. Aplikasi ini berfokus pada inputan data guru, data kelas dan data mata pelajaran. Aplikasi berbasis *web* dimana dapat diakses secara *online*. Dengan adanya sistem informasi penjadwalan mata pelajaran berbasis web ini dapat membantu pihak MTs Ar Rohman Kencong dalam membentuk dan mengelola proses penjadwalan sehingga proses tersebut berjalan secara akurat dan efisien.

II.2. Landasan Teoritis

II.2.1. Sistem Informasi

Sistem Informasi adalah serangkaian *hardware*, *software*, data, manusia, dan prosedur yang bekerja bersama untuk memproduksi informasi. Prosedur adalah suatu instruksi atau serangkaian instruksi, yang diikuti oleh *users* untuk menyelesaikan kegiatannya. Kegunaan umum dari sistem informasi termasuk kategori dari sistem informasi yang dapat digunakan oleh hampir semua bagian di dalam perusahaan. Sistem informasi terintegrasi digunakan oleh banyak bagian dan memfasilitasi berbagi informasi (*information Sharing*) dan komunikasi dalam perusahaan (Mas Ayoe Elhias, 2016 : 3).

II.2.2. Penjadwalan

Menurut Pinedo (2016), Penjadwalan adalah proses pengambilan keputusan yang berkenaan dengan pengalokasian sumber daya terbatas untuk tugas-tugas dari waktu ke waktu yang memiliki tujuan untuk mengoptimasi dari satu atau lebih tujuan.

Sumber daya dan tugas pada sebuah organisasi dapat berupa berbagai macam bentuk. Sumber daya dapat berupa mesin di bengkel, landasan pacu di bandara, kru di lokasi konstruksi, unit pengolahan dalam lingkungan komputasi, dan sebagainya. Tugas dapat berupa operasi-operasi dalam proses produksi, tinggal landas dan pendaratan pada bandara, tahapan-tahapan dalam proyek konstruksi, eksekusi program komputer, dan sebagainya.

Setiap tugas mungkin memiliki tingkat prioritas tertentu, kecepatan waktu mulai tertentu dan tanggal jatuh tempo tertentu. Tujuan juga dapat berupa berbagai macam bentuk. Tujuan penjadwalan organisasi yang satu mungkin untuk meminimalkan waktu yang dibutuhkan untuk menyelesaikan suatu tugas, sedang tujuan penjadwalan organisasi yang lain adalah untuk mengurangi jumlah tugas yang selesai setelah tanggal jatuh tempo. (Pinedo, 2016)

II.2.3. Database

Data adalah fakta yang dapat direkam dan memiliki arti secara implisit. Sedangkan kumpulan data yang memiliki hubungan secara implisit itu disebut *Database*. Basis data merupakan kumpulan data, umumnya mendeskripsikan aktivitas satu organisasi atau lebih yang berhubungan. Dalam *database* juga dikenal istilah DBMS (*Database Management Systems*) yaitu sekumpulan program yang memungkinkan pengguna untuk membuat dan memelihara suatu *database*. Bisa juga dikatakan bahwa DBMS merupakan perangkat *General Purpose Software System* yang berfungsi untuk mewadahi proses-proses dalam *database* seperti pendefinisian, pembuatan, *sharing*, maupun manipulasi *database* (Fitri Marisa, 2015).

II.2.4. PHP

PHP atau kependekan dari Hypertext Preprocessor adalah salah satu bahasa pemrograman open source yang sangat cocok atau dikhususkan untuk pengembangan web dan dapat ditanamkan pada sebuah skripsi HTML. Bahasa PHP dapat dikatakan menggambarkan beberapa bahasa pemrograman seperti C,

Java, dan Perl serta mudah untuk dipelajari. PHP merupakan bahasa scripting server – side, dimana pemrosesan datanya dilakukan pada sisi server. Sederhananya, serverlah yang akan menerjemahkan skrip program, baru kemudian hasilnya akan dikirim kepada client yang melakukan permintaan. Adapun pengertian lain PHP adalah akronim dari Hypertext Preprocessor, yaitu suatu bahasa pemrograman berbasis kode – kode (script) yang digunakan untuk mengolah suatu data dan mengirimkannya kembali ke web browser menjadi kode HTML”. Menurut Kustiyaningsih (2011:114), “PHP (atau resminya PHP: Hypertext Preprocessor) adalah skrip bersifat server – side yang ditambahkan ke dalam HTML”. (Astria Firman, 2016)

Pada prinsipnya server akan bekerja apabila ada permintaan dari client. Dalam hal ini client menggunakan kode-kode PHP untuk mengirimkan permintaan ke server. Sistem kerja dari PHP diawali dengan permintaan yang berasal dari halaman *website* oleh *browser*. Berdasarkan URL atau alamat *website* dalam jaringan *internet*, *browser* akan menemukan sebuah alamat dari *webserver*, mengidentifikasi halaman yang dikehendaki, dan menyampaikan segala informasi yang dibutuhkan oleh *webserver*. (Astria Firman, 2016)

Selanjutnya *webserver* akan mencarikan berkas yang diminta dan menampilkan isinya di *browser*. *Browser* yang mendapatkan isinya segera menerjemahkan kode HTML dan menampilkannya. Lalu bagaimana apabila yang dipanggil oleh user adalah halaman yang mengandung *script* PHP? Pada prinsipnya sama dengan memanggil kode HTML, namun pada saat permintaan dikirim ke *web-server*, *web-server* akan memeriksa tipe *file* yang diminta *user*.

Jika tipe file yang diminta adalah PHP, maka akan memeriksa isi script dari halaman PHP tersebut. (Astria Firman, 2016)

Apabila dalam file tersebut tidak mengandung script PHP, permintaan *user* akan langsung ditampilkan ke *browser*, namun jika dalam *file* tersebut mengandung script PHP, maka proses akan dilanjutkan ke modul PHP sebagai mesin yang menerjemahkan *script-script* PHP dan mengolah script tersebut, sehingga dapat dikonversikan ke kode-kode HTML lalu ditampilkan ke browser user. (Astria Firman, 2016)

II.2.5. MySQL

MySQL adalah salah satu jenis *database server* yang sangat terkenal. Kepopulerannya disebabkan MySQL menggunakan SQL sebagai bahasa dasar untuk mengakses *database*-nya. MySQL termasuk jenis RDBMS (*Relational Database Management System*). Pada MySQL, sebuah *database* mengandung satu atau sejumlah tabel. Tabel terdiri atas sejumlah baris dan setiap baris mengandung satu atau beberapa kolom (Emy Khuzainah, 2013 : 2-3).

MySQL adalah *Relational Database Management System (RDMS)* yang didistribusikan secara gratis dibawah lisensi *GPL (General Public License)*. Dimana setiap orang bebas untuk menggunakan MySQL, namun tidak boleh dijadikan produk turunan yang bersifat komersial. MySQL dapat dikatakan lebih unggul dibandingkan *database server* lainnya dalam *query* data. Hal ini terbukti untuk *query* yang dilakukan oleh *single user*, kecepatan *query* MySQL bisa sepuluh

kali lebih cepat dari *Postgre SQL* dan lima kali lebih cepat dibandingkan *Interbase* (Iman Saufik Suasana dan Jatu Suryantari, 2016: 37).

II.2.6. *Unified modeling language* (UML)

Unified Modeling Language (UML) adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem (Ade Hendini, 2016).


Adapun jenis-jenis yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut:

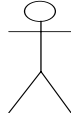
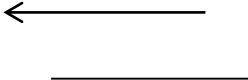
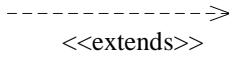
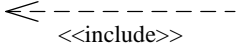
a. *Use Case Diagram*

Use Case digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut (Ade Hendini, 2016).

Simbol-simbol yang digunakan dalam *Use Case Diagram* dapat dilihat pada Tabel II.1

Tabel II.1. Simbol *Use Case Diagram*

Gambar	Nama	Keterangan
	<i>Use Case</i>	<i>Use Case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, yang dinyatakan dengan menggunakan kata kerja.

	<p><i>Actor</i></p>	<p><i>Actor</i> 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.</p>
	<p>Asosiasi</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 mengindikasikan data.</p>
	<p><i>Extends</i></p>	<p><i>Extends</i> merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.</p>
	<p><i>Include</i></p>	<p><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</p>



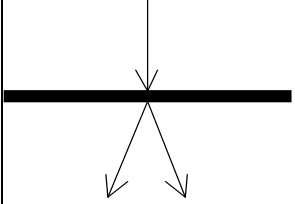


(Sumber : Ade Hendini ; 2016)

b. *Activity Diagram*

Activity diagram adalah diagram yang menggambarkan aliran kerja atau aktivitas dari sebuah sistem atau proses bisnis. Yang perlu diperhatikan adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan oleh aktor, jadi aktivitas yang dapat dilakukan oleh sistem. Jadi dengan kata lain, diagram ini menunjukkan bagaimana aktivitas-aktivitas tersebut bergantung satu sama lain. *Activity Diagram* menggambarkan *Work Flow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis (Ade Hendini, 2016).

Simbol-simbol yang digunakan dalam *Activity Diagram* dapat dilihat pada Tabel II.2

Tabel II.2. Simbol *Activity Diagram*

Gambar	Nama	Keterangan
	<i>Initial State / Start State</i>	<i>Start Point</i> diletakkan pada pojok kiri atas dan merupakan awal aktivitas.
	<i>Action State</i>	menggambarkan suatu proses atau kegiatan bisnis.
	<i>Transition</i>	digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.
	<i>Control Flow</i>	digunakan untuk menunjukkan kegiatan.
	<i>Decision</i>	Menggambarkan pilihan untuk pengambilan keputusan <i>true</i> atau <i>false</i> .

	<i>Final State</i>	Digunakan untuk mengakhiri aktivitas.
	<i>Swimlane</i>	Merupakan pembagian <i>Activity Diagram</i> untuk menunjukkan siapa melakukan apa.




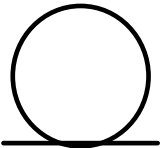
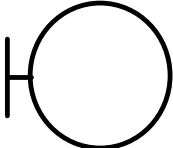
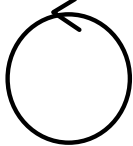
(Sumber: Ade Hendini ; 2016)

c. *Sequence Diagram*

Sequence Diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek (Ade Hendini, 2016).

Simbol-simbol yang digunakan dalam *Sequence Diagram* dapat dilihat pada Tabel II.3.

Tabel II.3. Simbol *Sequence Diagram*

Gambar	Nama	Keterangan
	<i>Activation</i>	Mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivasi sebuah operasi.
	<i>Lifeline</i>	garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i> .
	<i>Message</i>	simbol mengirim pesan antar <i>class</i> .
	<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>	Merupakan berisi kumpulan kelas yang menjadi <i>interfaces</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan <i>form entry</i> dan <i>form cetak</i>
	<i>Control class</i>	Merupakan suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas.

(Sumber: Ade Hendini ; 2016)

d. *Class Diagram*

Class Diagram 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 (Ade Hendini, 2016).

Class Diagram secara khas meliputi : Kelas (*Class*), *Relasi Assosiations*, *Generalitation* dan *Aggregation*, Atribut (*Attributes*), Operasi (*Operation / Method*) dan *Visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *Multiplicity* atau *Cardinality* (Ade Hendini, 2016).

Tabel II.4. *Multiplicity Class Diagram*

<i>Multiplicity</i>	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 maksimal 4.

(*Sumber: Ade Hendini ; 2016*)