

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Penelitian Terkait**

Perkembangan teknologi informasi khususnya komputer semakin berkembang pesat di berbagai bidang pekerjaan dan jasa. Kebanyakan orang berpendapat bahwa dengan adanya pemanfaatan komputer pada bidang pekerjaannya, maka pekerjaan mereka akan semakin mudah dan cepat. Hal ini juga tidak terlepas dari pemanfaatan komputer pada bidang jasa bimbingan belajar.[1]

Pemahaman akan keberhasilan pendidikan siswa di sekolah masih diukur dari dapat atau tidaknya siswa tersebut berhasil mencapai nilai prestasi *akademis* yang telah ditetapkan sekolah. Parameter pencapaian nilai akademis di Indonesia dapat dilihat dari standar nilai kelulusan yang tinggi yang harus dicapai siswa sebagai syarat untuk masuk ke jenjang pendidikan yang lebih tinggi.[2]

Saat ini dengan perkembangan teknologi informasi dan komunikasi (*TIK*) memungkinkan kita belajar tanpa menuntut kehadiran fisik kita di suatu tempat yaitu yang disebut *e-learning*. Kehadiran *e-learning* dalam dunia pendidikan merupakan suatu paradigma baru. Dengan *e-learning* kita dapat belajar di mana saja dan kapan saja. Saat ini kalau kita membuka internet akan banyak kita temukan website atau portal yang dirancang untuk

*e-learning*. [3]

Untuk memenuhi kebutuhan akan sebuah sistem baru yang lebih aman, mudah dan efisien dalam mengelola penerimaan siswa baru, tentunya adalah sebuah sistem terkomputerisasi serta berbasis web yang pada akhirnya pendaftaran siswa baru dapat dilakukan secara online. [4]

Dari permasalahan tersebut menyebabkan pemilik merasa kesulitan, dalam proses pendataan siswa, serta kemungkinan kesalahan yang terjadi dalam menginput data. Data yang dibuat dalam sistem informasi siswa terdiri dari nama, jenis kelamin, agama, alamat, kelas, penjadwalan jam bimbingan belajar, video pembelajaran serta rekap nilai dari masing masing siswa yang akan diberikan kepada wali murid. [5]

## 2.2 Landasan Teori

### 2.2.1 Waterfall

Metode penelitian *waterfall* adalah salah satu pendekatan dalam proses pengembangan perangkat lunak metode penelitian *waterfall* menggambarkan proses pengembangan perangkat lunak sebagai serangkaian tahapan yang dilakukan secara berurutan dan linier. Tahapan-tahapan ini termasuk analisis kebutuhan, perencanaan, desain, implementasi, pengujian, dan pemeliharaan. [6]

### 2.2.2 Reservasi

Reservasi sistem merujuk pada proses atau sistem yang digunakan untuk memesan atau mengamankan layanan, fasilitas, atau produk tertentu untuk digunakan di masa mendatang. Reservasi sistem umumnya terkait dengan industri layanan seperti perhotelan, transportasi, restoran, dan hiburan, di mana pengguna dapat melakukan pemesanan secara terencana dan terjadwal.[7]

Konsep reservasi sistem melibatkan pengguna dalam melakukan pemesanan dengan menggunakan suatu platform atau sistem, baik secara langsung di lokasi fisik maupun melalui media elektronik seperti situs web, aplikasi ponsel, atau sistem reservasi telepon. Informasi yang biasanya diminta dalam proses reservasi meliputi waktu, tanggal, jumlah orang, jenis layanan atau fasilitas yang dipesan, dan detail pembayaran.

### 2.2.3 CodeIgniter



Gambar 2. 1 Logo CodeIgniter

CodeIgniter adalah kerangka kerja (*framework*) pengembangan aplikasi web berbasis PHP yang bersifat sumber

terbuka (*open source*) CodeIgniter dirancang untuk memudahkan pengembangan aplikasi web dengan menyediakan seperangkat alat dan aturan yang sudah disiapkan sehingga pengembang dapat fokus pada logika bisnis aplikasi mereka tanpa harus memulai dari awal.[8]

#### 2.2.4 Visual Studio Code



Gambar 2. 2 Logo Visual Studio Code

Visual Studio Code (*VS Code*) adalah sebuah editor kode sumber yang sangat populer dan kuat yang dikembangkan oleh Microsoft. Visual Studio Code sangat populer di kalangan pengembang perangkat lunak karena kecepatan, kehandalan, dan fleksibilitasnya. Dengan dukungan yang kuat dari komunitas dan pembaruan reguler dari Microsoft, VS Code terus berkembang dan menjadi pilihan utama untuk banyak pengembang.[9]

### 2.2.5 Bootstrap



Gambar 2. 3 Logo Bootstrap

Bootstrap adalah kerangka kerja (*framework*) front-end yang sumber terbuka (*open source*) yang digunakan untuk mempermudah pengembangan aplikasi web dan desain responsif. Diciptakan oleh Twitter, Bootstrap menyediakan seperangkat alat dan komponen berbasis HTML, CSS, dan JavaScript yang dirancang untuk mempercepat proses pengembangan dan memastikan konsistensi dalam tampilan dan perilaku antarmuka pengguna.[10]

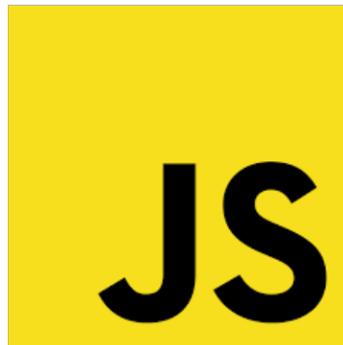
### 2.2.6 HTML



Gambar 2. 4 Logo HTML

HTML adalah bahasa markup untuk membuat halaman web. Kepanjangan dari HTML adalah *Hypertext Markup Language*. Artinya, HTML menggunakan symbol tertentu (tag) yang nantinya akan diterjemahkan oleh browser ke halaman web.[11]

### 2.2.7 Java Script



Gambar 2. 5 Logo Javascript

JavaScript adalah bahasa pemrograman yang digunakan untuk pengembangan website agar lebih dinamis. Ibarat kata, JavaScript memberikan “kehidupan” dalam website sehingga terciptanya interaksi antara pengunjung dengan situs tersebut.[12]

Website dinamis yang dimaksud berarti konten di dalamnya dapat bergerak atau mengubah apapun yang tampak di layar tanpa harus dimuat ulang secara manual. Misalnya seperti konten gambar animasi, maps, slideshow, polling, dan sebagainya.

### 2.2.8 XAMPP



Gambar 2. 6 Logo Xampp

XAMPP adalah singkatan dari *Cross-platform*, Apache, MySQL, PHP, dan Perl. Ini adalah sebuah perangkat lunak yang menyediakan lingkungan server yang lengkap dan siap pakai untuk pengembangan dan pengujian aplikasi web secara lokal pada komputer pribadi.[13]

XAMPP memungkinkan pengguna untuk menjalankan server web Apache, basis data MySQL, serta bahasa pemrograman PHP dan Perl pada komputer lokal tanpa perlu menginstalnya secara terpisah. Ini sangat berguna bagi para pengembang web yang ingin mengembangkan dan menguji situs atau aplikasi web sebelum mempublikasikannya secara online.

### 2.2.9 MYSQL



Gambar 2. 7 Logo MySQL

MySQL adalah sebuah database manajemen sistem (*DBMS*) populer yang memiliki fungsi sebagai relational database manajemen sistem (*RDBMS*). Selain itu MySQL software merupakan suatu aplikasi yang sifatnya open source serta server basis data MySQL memiliki kinerja sangat cepat, reliable, dan mudah untuk digunakan serta bekerja dengan arsitektur client server atau embedded sistem.[14]

MySQL adalah program database server yang mampu menerima dan mengirimkan data dengan sangat cepat, multi user serta menggunakan perintah standar SQL (*Structured Query Language*) dan baik digunakan sebagai client maupun server.

Alat yang dirancang terintegrasi dengan database kehadiran mahasiswa pada suatu matakuliah sehingga dapat berperan sebagai pengganti sistem absensi manual. Database kehadiran dibuat dengan MySQL XAMPP.

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

*Unified Modeling Language* atau lebih sering dikenal dengan sebutan *UML*, adalah salah satu metode dalam teknik rekayasa perangkat lunak yang digunakan untuk menggambarkan alur dan cara kerja sistem, fungsi, tujuan dan mekanisme kontrol sistem tersebut. Dalam teknik rekayasa perangkat lunak bidang analisis dan perancangan sistem informasi, saat ini lebih banyak menggunakan gabungan dari konsep pemrograman berorientasi objek dengan teknik pembuatan perangkat lunak, dimana suatu sistem dilihat sebagai objek tersendiri yang sudah mencakup data dan proses atau dapat bekerja secara mandiri dalam satu set sistem (*package*).[15]

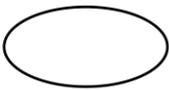
Dalam teknik perancangan sistem informasi, terdapat 4 model *UML* yang paling efektif penggunaannya untuk menggambarkan desain sistem, yaitu: *Use Case diagram*, *Class diagram*, *Sequence diagram*, dan *Behavioral State Machine diagram*. Empat teknik pemodelan *UML* ini juga disebut sebagai 4 teknik dasar (*core*) pemodelan berbasis *UML*. Keempat teknik pemodelan *UML* ini telah mendominasi penggunaannya dalam proyek-proyek berorientasi objek.

Dalam Perancangan berorientasi objek berbasis *UML* menggunakan alat bantu sebagai berikut:

### 1. *Use Case Diagram*

*Use case diagram* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *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* bisa dilihat pada tabel dibawah.

Tabel 2. 1 Simbol Use Case

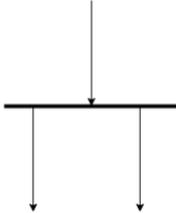
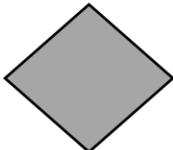
No	Simbol	Keterangan
1		<i>Use Case</i> : Abstraksi dan interaksi antara sistem dan aktor.
2		<i>Actor</i> : Mewakili peran orang, sistem yang lain, atau alat ketika berkomunikasi dengan <i>use case</i> .
3		<i>Asosiasi</i> antara aktor dan <i>use case</i> , digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi

		secara langsung dan bukannya mengindikasikan data.
4		Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem.
5		<i>Include</i> , Menunjukkan spesialisasi actor untuk dapat berpartisipasi dengan use case.
6		<i>Extend</i> , merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.

## 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* bisa dilihat pada tabel dibawah.

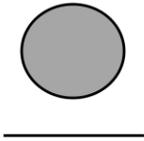
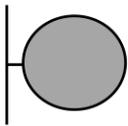
Tabel 2. 2 Simbol Activity Diagram

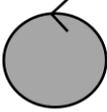
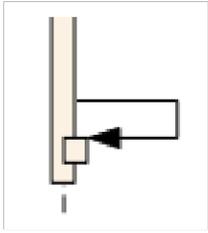
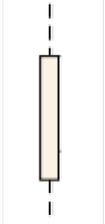
No	Simbol	Keterangan
1		<i>Start Point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktivitas.
2		<i>End Point</i> , akhir aktivitas.
3		<i>Activities</i> , menggambar kan suatu proses/kegiatan bisnis.
4		<i>Fork/percabangan</i> digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.
5		<i>Decision Points</i> , menggambarkan pilihan untuk pengambilan keputusan, <i>true</i> atau <i>false</i> .
6		<i>Swimlane</i> , pembagian <i>activity diagram</i> untuk menunjukkan siapa melakukan apa.

### 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* Bisa dilihat tabel di bawah.

Tabel 2. 3 Sequence Diagram

No	Simbol	Keterangan
1		<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.
2		<i>Boundary Class</i> , 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> .

No	Simbol	Keterangan
3		<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.
4		<i>Message</i> , simbol mengirim pesan antar <i>class</i> .
5		<i>Recursive</i> , menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.
6		<i>Activation</i> , mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivasi sebuah operasi.
7		<i>Lifeline</i> , garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i> .

#### 4. Diagram Kelas (*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. *Class Diagram* secara khas meliputi : Kelas (*Class*), Relasi *Associations*, *Generalisation* 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*.

Tabel 2. 4 Class Diagram

<b><i>Multiplicity</i></b>	<b>Keterangan</b>
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih
1..*	Satu atau lebih
0..1	Boleh tidak ada, maksimal 1
n..n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimal 4