

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Penelitian Terdahulu**

Penelitian yang dilakukan peneliti memiliki judul “Sistem Informasi Pengelolaan Perpustakaan Pada SDPN Setiabudi Bandung”. Penelitian terdahulu menjadi bekal ilmu bagi peneliti sehingga dapat memperkaya teori peneliti dalam mengkaji penelitian yang dilakukan. Dibawah ini adalah penelitian yang peneliti kaji sebelumnya.

Penelitian berjudul “Sistem Informasi Perpustakaan Sekolah Berbasis Web” yang dilakukan oleh saudari Diah Puspitasari tahun 2016. Tujuannya yaitu untuk menciptakan sebuah sistem informasi perpustakaan berbasis web agar memberikan informasi perpustakaan dengan cepat, tepat, dan akurat untuk anggota, petugas perpustakaan serta pimpinan [5].

Penelitian selanjutnya dilakukan oleh salah satu dosen UNIKOM yaitu ibu Mia Fitriawati bersama rekannya yaitu saudari Annisa Sari Saputri, dengan judul “Sistem Informasi Perpustakaan Pada SMP Negeri 52 Bandung”. Penelitian ini memiliki maksud yaitu untuk membuat suatu sistem informasi perpustakaan yang dapat membantu serta meningkatkan kinerja pada bagian perpustakaan, dan tujuan dari penelitian ini salah satunya untuk mengetahui permasalahan yang terdapat pada sistem informasi yang sedang berjalan pada perpustakaan SMP Negeri 52 Bandung [6].

Penelitian selanjutnya dilakukan oleh saudara Andi Rahman Putera dan Malik Ibrahim dengan judul “Rancang Bangun Sistem Informasi Peminjaman dan Pengembalian Buku Perpustakaan SMP Negeri 1 Madiun”. Tujuannya adalah untuk merancang dan membangun sistem informasi berbasis desktop pada perpustakaan SMPN 1 Madiun, dengan dibuatnya aplikasi tersebut mereka berharap dapat membantu petugas dalam mengelola administrasi data perpustakaan dan membantu para siswa dalam mencari dan melihat koleksi buku [7].

Penelitian selanjutnya dilakukan oleh saudari Nurul Alifah Rahmawati dan saudara Arif Cahyo Bachtiar. Penelitian ini berjudul “Analisis dan Perancangan Desain Sistem Informasi Perpustakaan Sekolah Berdasarkan Kebutuhan Sistem”, rancangan sistem informasi perpustakaan yang dibuat lebih sederhana daripada perpustakaan perguruan tinggi maupun perpustakaan pada daerah. Sistem informasi perpustakaan sekolah yang dibuat oleh peneliti dirancang dengan bahasa pemrograman phpMyAdmin dan basis data SQL. Aplikasi ini dapat melakukan peminjaman dan pengembalian serta pelaporan data denda [8].

Dari penelitian terdahulu yang dicantumkan diatas, penelitian yang dilakukan memiliki tujuan dan manfaat yang baik. Di era saat ini penggunaan internet menjadi solusi dalam menjalankan aktivitas agar lebih efektif dan efisien, hal ini juga dapat dimanfaatkan oleh para siswa untuk bijak menggunakan internet salah satunya yaitu mengakses sistem informasi perpustakaan tersebut.

Setelah mengkaji penelitian terdahulu tersebut dan telah mendapatkan sebuah gambaran, peneliti tetap menggali lagi guna menghasilkan solusi yang

baik juga. Maka dari itu peneliti mempunyai solusi yaitu membuat aplikasi sistem informasi perpustakaan untuk meningkatkan pelayanan agar lebih baik lagi dan mempermudah kinerja para pegawainya.

## **2. 2 Teori-teori Dasar**

Dalam melakukan penelitian ini perlu dipahami terlebih dahulu teori-teori dasar yang dicantumkan, agar peneliti mengetahui dan paham guna menghasilkan penelitian yang baik.

### **2.2.1 Website**

*Website* adalah keseluruhan halaman pada web yang terdapat dalam sebuah domain yang mengandung informasi. Sebuah website biasanya dibangun atas banyak halaman web yang saling berhubungan [9].

### **2.2.2 Aplikasi**

Aplikasi berasal dari kata *application* yang artinya penerapan, lamaran, penggunaan. Secara istilah aplikasi adalah program siap pakai yang direka untuk melaksanakan suatu fungsi bagi pengguna atau aplikasi lain dan dapat digunakan oleh sasaran yang dituju [10].

### **2.2.3 Pengelolaan Perpustakaan**

Pengelolaan adalah proses yang memberikan pengawasan pada semua hal yang terlibat dalam pelaksanaan dan pencapaian tujuan. Pengelolaan sebuah perpustakaan sesungguhnya dapat dilakukan dengan cara mengurai perpustakaan tersebut menjadi beberapa komponen-komponen kepentingan sebagai pendukung proses pendidikan di sekolah [11].

#### 2.2.4 Pemrograman Berorientasi Objek

Pemrograman berorientasi objek adalah sebuah konsep yang menggambarkan suatu pemrograman menyelesaikan masalah yang diibaratkan sebagai objek yang saling berinteraksi satu sama lain. Pada jenis pemrograman ini terdapat beberapa istilah atau konsep dasar, antara lain :

1. *Encapsulation*

*Encapsulation* adalah proses dalam menciptakan sebuah objek yang memiliki beberapa bagian atau *attribute* yang terbagi berdasarkan pada sifat yakni *public* (umum) dan *private* (khusus).

2. *Inheritance*

*Inheritance* merupakan hubungan antar dua objek atau lebih nantinya akan terdapat sebuah objek utama yang menurunkan *attribute* atau *method* yang dimiliki kepada objek yang lain, bisa diberikan keseluruhan ataupun hanya sebagian.

3. *Abstraction*

Abstraksi merupakan satu dari 3 prinsip sentral (*Encapsulation* dan *Inheritance*). Abstraksi yaitu proses dimana seorang programmer menyembunyikan semua data informasi dari suatu objek, data yang tidak relevan saja yang di sembunyikan. Tentu hal ini memiliki tujuan yaitu untuk meminimalisir kompleksitas dan meningkatkan efisiensi.

4. *Composition*

*Composition* adalah sebuah *class* terdiri dari beberapa bagian dari objek yang terpisah serta memiliki fungsi yang berbeda-beda satu sama lainnya [12].

### 2.2.5 Pengembangan Metode *Prototype*

Metode pengembangan *prototype* merupakan salah satu metode yang banyak dipakai. Karena dengan metode ini pengembang dan pelanggan dapat saling berkomunikasi selama proses dibuatnya suatu sistem. Agar sistem yang dibangun dengan metode *prototype* ini berhasil dengan baik yaitu harus dapat mendefinisikan aturan-aturan main pada saat awal, interaksi pengembang dan pelanggan harus setuju bahwa sistem dibangun untuk mendefinisikan sebuah kebutuhan. Terdapat tahapan-tahapan dalam metode ini, diantaranya :

1. Pengumpulan Kebutuhan

Pengembang dan pelanggan bersama-sama untuk menentukan format keseluruhan perangkat lunak, mengidentifikasi semua kebutuhan serta garis besar sistem yang akan dirancang.

2. Membangun *Prototyping*

Dalam tahapan ini membuat rancangan yang sementara berfokus pada proses misalnya membuat input dan format output kepada pelanggan.

3. Evaluasi *Prototyping*

Pelanggan melakukan evaluasi apakah *prototyping* yang sudah dirancang sesuai dengan keinginan atau tidak, jika tidak akan dilakukan revisi oleh pengembang.

4. Pengkodean Sistem

*Prototyping* yang telah menemukan kesepakatan antara pengembang dan pelanggan, selanjutnya kesepakatan tersebut diterjemahkan ke dalam bahasa pemrograman.

#### 5. Menguji Sistem

Setelah sistem sudah jadi dan siap pakai, perlu dilakukan pengujian terlebih dahulu diantaranya dengan melakukan pengujian dengan *White Box*, *Black Box*, *Basis Path*, dan lain-lain.

#### 6. Evaluasi Sistem

Pelanggan kembali melakukan evaluasi apakah sudah sesuai atau tidak sistem yang telah dibuat oleh pengembang, jika masih belum sesuai maka pengembang akan melakukan tahap revisi kembali.

#### 7. Menggunakan Sistem

Kondisi perangkat lunak sudah sesuai dengan keinginan pelanggan dan siap untuk digunakan oleh pelanggan [13].

### **2.2.6 Unified Modeling Language (UML)**

UML singkatan dari *Unified Modeling Language* yang berarti bahasa pemodelan standar. Chonoles (2003) mengatakan sebagai bahasa, yang berarti UML memiliki sintaks dan makna. UML bukan sekedar diagram, tetapi juga menjelaskan alur konteksnya. UML diaplikasikan untuk tujuan tertentu biasanya digunakan untuk :

1. Merancang sebuah perangkat lunak
2. Sarana interaksi antara perangkat lunak dengan proses bisnis

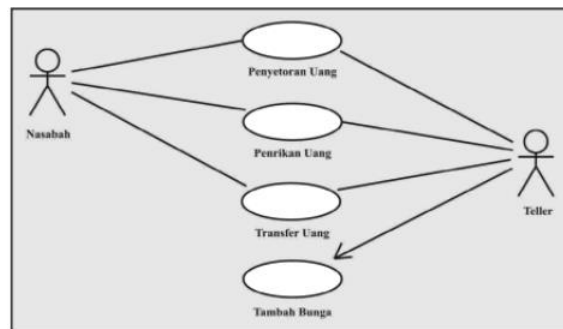
3. Menjabarkan sistem dengan rinci untuk menganalisis dan mencari apa yang dibutuhkan sistem.
4. Mendokumentasi sistem yang ada

UML memiliki diagram-diagram yang biasa digunakan untuk melakukan sebuah pemodelan, diantaranya seperti :

1. *Use Case Diagram*

*Use Case Diagram* adalah diagram yang menggambarkan hubungan antara aktor dengan sistem. *Use case* diagram bisa mendeskripsikan interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat. *Use case* diagram juga dapat digunakan untuk mengetahui fungsi apa saja yang berada di dalam sistem dan dapat juga mempresentasikan sebuah interaksi antara aktor dengan sistem. Pooley (2003:15) mengatakan bahwa model dari *use case* dapat dijelaskan dalam diagram *use case*, tapi perlu diingat, sebuah diagram tidak identik dengan model karena model lebih luas dari diagram [14]. Komponen pembentuk diagram *use case* adalah :

- a. Aktor (*actor*), memiliki fungsi untuk menggambarkan pihak-pihak yang mempunyai peran dalam sistem.
- b. *Use case*, merupakan gambaran fungsional dari suatu sistem.
- c. Hubungan (*link*), menjadi suatu gambaran aktor mana saja yang terlibat dalam *use case* tersebut.



**Gambar 2.1 Use Case Diagram**

**(Sumber : Analisa dan Perancangan Sistem Informasi Menggunakan Model Terstruktur dan UML [14, p.65])**

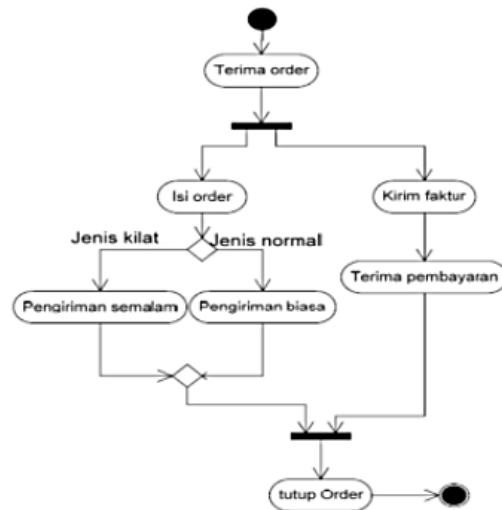
## 2. Use Case Skenario

Setiap *use case* yang ada akan dilengkapi dengan sebuah skenario, skenario merupakan alur jalannya proses *use case* dari sisi aktor dan sebuah sistem. Ada 2 jenis skenario antaranya skenario normal dan alternatif, skenario normal yaitu bila sistem berjalan tanpa ada kendala atau *error* sedangkan skenario alternatif yaitu jika sistem berjalan tidak sesuai harapan atau ditemukannya sebuah kendala atau *error* [15].

## 3. Activity Diagram

*Activity Diagram* atau diagram aktivitas adalah tipe khusus dari diagram status yang akan memperlihatkan aliran dari suatu satu aktivitas ke aktivitas yang lainnya didalam suatu sistem [15].



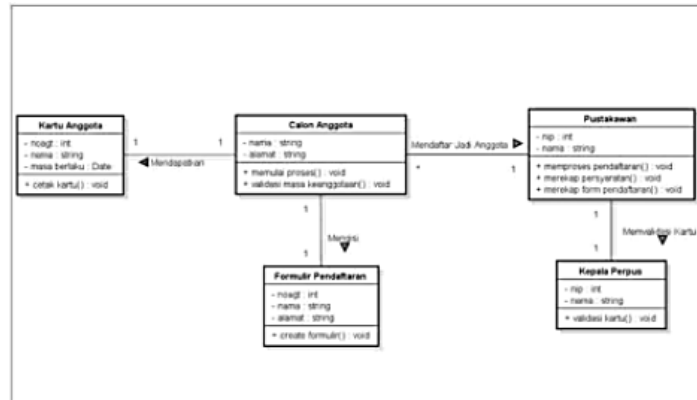


**Gambar 2.2 Activity Diagram**

(Sumber : Rekayasa Perangkat Lunak Berorientasi Objek [15, p.88])

#### 4. Class Diagram

*Class Diagram* merupakan diagram yang selalu terdapat dalam pemodelan sistem dengan berbasis objek. Diagram ini akan menghasilkan sebuah objek yang menjadi inti dari suatu pengembangan dan desain berorientasi objek. *Class Diagram* dapat menunjukkan relasi antar *class* didalam sistem yang saling berinteraksi untuk mencapai tujuan [16].

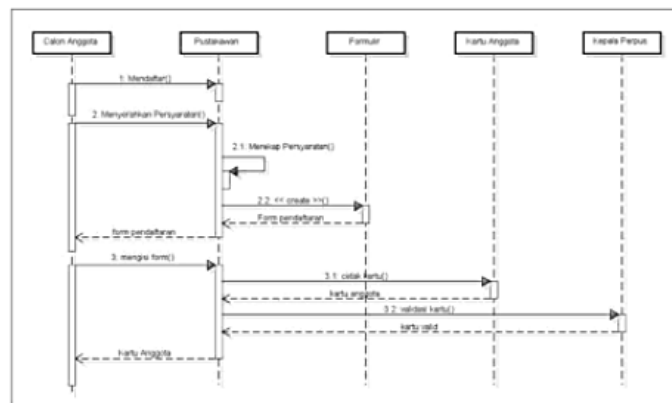


**Gambar 2.3 Class Diagram**

(Sumber : Aplikasi SPK Pemilihan Dosen Terbaik Metode Simple Additive Weighting (SAW) Dengan Java [16, p.32])

### 5. Sequence Diagram

*Sequence Diagram* menjadi salah satu jenis diagram yang terdapat pada UML yang berfungsi untuk menjelaskan interaksi yang terjadi antara objek dengan berdasarkan urutan waktu [16].



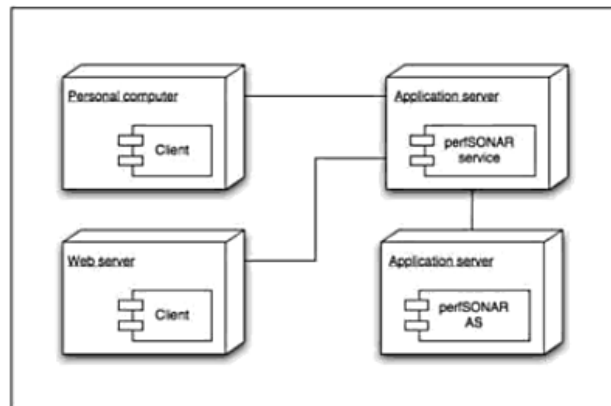
**Gambar 2.4 Sequence Diagram**

(Sumber : Aplikasi SPK Pemilihan Dosen Terbaik Metode Simple Additive Weighting (SAW) Dengan Java [16, p.32])

### 6. Deployment Diagram

*Deployment Diagram* yaitu diagram yang dapat menginterpretasi letak suatu sistem secara fisik, menjadi penunjuk bagian perangkat lunak yang terdapat dalam perangkat keras dan digunakan dalam

mengimplementasi suatu sistem dan hubungan antara komponen perangkat keras [16].



**Gambar 2.5 Deployment Diagram**

(Sumber : Aplikasi SPK Pemilihan Dosen Terbaik Metode Simple Additive Weighting (SAW) Dengan Java [16, p.34])

### 2.2.7 Testing Perangkat Lunak

Dalam merancang suatu sistem, diperlukan tahapan pengujian atau *testing* pada perangkat lunak, dengan tujuan agar sistem yang dibangun dapat bermanfaat dan sesuai dengan harapan serta kebutuhan pengguna.

Selain itu, tujuan dilakukannya pengujian pada perangkat lunak yaitu untuk menemukan kekurangan, *bug*, *error* yang terjadi pada sistem. Dengan melakukan pengujian maka akan meminimalisir hal tersebut. Untuk pengujian perangkat lunak terdapat 2 metode yang dapat digunakan, diantaranya :

#### 1. *Black Box Testing*

Pengujian dengan metode *black box* lebih berfokus terhadap unit program sudah memenuhi kebutuhan pada spesifikasi atau belum. Cara pengujian pada metode ini hanya sebatas menjalankan atau mengeksekusi modul, lalu menganalisa apakah sudah dapat berjalan

sesuai harapan dan jika tidak maka akan dilakukan pengujian dengan metode selanjutnya.

## 2. *White Box Testing*

Pengujian dengan metode *white box* merupakan cara pengujian yang lebih rinci daripada metode *black box*. Karena metode ini akan memeriksa pengkodean pada modul dan menganalisis apakah terdapat kesalahan atau tidak. Maka setiap baris program, *variable*, dan parameter yang terdapat pada modul akan dicek satu persatu [17].

### 2.3 *Software yang Digunakan*

Dalam melakukan penelitian ini, peneliti memanfaatkan beberapa *software* yang dapat membantu maksud dan tujuan penelitian yang dilakukan. *Software* yang digunakan menjadi penunjang peneliti untuk merampungkan sistem pengelolaan perpustakaan ini. Berikut *software* yang digunakan peneliti diantaranya :

#### 2.3.1. *Visual Studio Code*

*Visual Studio Code (VS Code)* yaitu sebuah *text editor* yang biasa digunakan programmer dalam melakukan pengkodean program. *Text editor* ini dirancang oleh Microsoft untuk sistem operasi *multiplatform* yang berarti tersedia untuk *Windows*, *Linux*, dan juga *Mac*. *Text editor* ini menjadi pilihan para programmer karena mendukung banyak bahasa pemrograman, seperti *JavaScript*, *Typescript*, dan *Node.js* dan dapat juga memasukkan bahasa pemrograman lainnya dengan bantuan *plugin* [18].

### 2.3.2. Codeigniter

Codeigniter adalah sebuah *framework* pada PHP yang memiliki konsep MVC (*Model, View, Controller*) didalamnya, yang dapat membantu pengembang untuk membangun aplikasi website dengan efisien.

Konsep MVC biasanya akan digunakan pada pemrograman dengan berorientasi objek. Konsep dari MVC dapat memisahkan antar *query* ke dalam *database (model)* dengan memiliki tampilan (*view*) dan diatur melalui logika pemrograman (*controller*). Terdapat kelebihan dan kekurangan pada *framework* codeigniter ini, seperti [19] :

#### 2.3.2.1 Kelebihan *Framework* Codeigniter

Kelebihan pada *framework* codeigniter diantaranya seperti :

1. Penggunaan MVC membantu pengembang, karena pengkodean menjadi terstruktur
2. Penggunaan pada \$\_GET diganti menjadi URL
3. Struktur serta *library* sudah disediakan oleh *framework*

#### 2.3.2.2 Kekurangan *Framework* Codeigniter

Kekurangan pada *framework* codeigniter diantaranya seperti :

1. Ketersediaan *library* terbatas, karena sangat sulit dalam mencari *plugin* yang resmi.
2. Tidak ditemukan *editor* khusus untuk codeigniter

### 2.3.3. XAMPP

Fungsi *software* ini adalah menjadi *localhost* atau *server* yang berdiri sendiri yang terdiri dari beberapa program lainnya di dalamnya seperti MySQL

*Database*, Apache HTTP Server dan PHP serta Perl. *Software* XAMPP dapat diunduh melalui website resminya, Penamaan XAMPP merupakan singkatan yang terdiri dari X (terdapat 4 sistem operasi), Apache, MySQL, PHP dan Perl. Dari masing-masing singkatannya memiliki arti, berikut arti dari setiap singkatannya :

a. X = *Cross Platform*

Menjadi kode bahwa *software* ini dapat berjalan di banyak sistem operasi.

b. A = Apache

Merupakan *web server* yang dapat diakses oleh siapapun tanpa dikenai biaya dan dikembangkan banyak orang.

c. M = MySQL

Berperan dalam mengedit, menghapus serta mengelola list daftar didalam *database*.

d. P = PHP

Menunjukkan huruf “P” disini yaitu bahasa pemrograman PHP yang biasanya digunakan dalam membuat website yang dinamis.

e. P = Perl

Menunjukkan bahasa pemrograman Perl yang biasanya digunakan untuk memenuhi berbagai kebutuhan, dan penggunaannya yang fleksibel.

Peneliti memanfaatkan *software* tersebut yaitu untuk digunakan sebagai *local server* ketika sedang membangun sistem. Dikarenakan ketika dalam tahapan

membangun sistem belum dilakukan *hosting*, maka penggunaan XAMPP membantu peneliti untuk menjalankan rancangan program.

#### **2.3.4. *Template Bootstrap***

*Template* yang satu ini bisa didapatkan dengan mendownloadnya melalui *website* resmi *bootstrap*. Peneliti menggunakan *template* ini karena mudah didapatkan dan tampilannya menarik. Dengan menggunakan *template* ini, tidak perlu membuat tampilan dari awal, karena dengan bantuan *template bootstrap* membantu membuat tampilan *website* menjadi lebih cepat.

#### **2.3.5. Pemrograman Dengan PHP**

PHP adalah salah satu bahasa pemrograman berbasis web yang diciptakan khusus untuk digunakan dalam membangun sesuatu aplikasi berbasis *website*. PHP mudah untuk dipelajari serta tersedia secara gratis, maka siapapun dapat mendapatkan dan menggunakan bahasa pemrograman ini. Jenis bahasa pemrograman PHP memerlukan logika dalam mengatur alur dari program yang akan dibangun yang biasanya disebut algoritma, maka dengan mempelajari terlebih dahulu mengenai struktur data dan algoritma akan membantu untuk menguasai bahasa pemrograman PHP [20].

#### **2.3.6. *phpMyAdmin***

*phpMyAdmin* merupakan salah satu aplikasi berbasis *website* yang berguna untuk mengelola *database* MySQL. Software ini semakin populer karena tersedia pilihan bahasa, hingga saat ini kurang lebih terdapat 65 bahasa yang dapat digunakan pada *phpMyAdmin*. Karena program yang dibangun peneliti

membutuhkan *database* untuk menyimpan data, peneliti menggunakan *phpMyAdmin* untuk mengelola databasenya.

### **2.3.7. CSS (*Cascading Style Sheet*)**

*Cascading Style Sheet* atau yang sering disebut CSS ini berguna untuk mengatur elemen tampilan dan berguna untuk memisahkan konten dari tampilan visual didalam situs. Pada tahun 1996 W3C (*World Wide Web Consortium*) mengembangkan CSS dengan alasan dahulu HTML tidak dibekali dengan *tags* yang berfungsi untuk memformat halaman situs. HTML dan CSS saling berkaitan, karena HTML pondasi situs dan CSS dapat memperbaiki semua aspek yang terkait dengan tampilan dari situs.