

## **BAB II**

### **TINJAUAN PUSTAKA DAN LANDASAN TEORI**

#### **2.1. Tinjauan Pustaka**

Penelitian sebelumnya dilakukan oleh Rizal Wahyuono Fakultas Teknologi dan Informatika Universitas Dinamika dengan judul “Rancang Bangun Aplikasi Pencatatan dan Tindak Lanjut Audit Internal SPI Berbasis Web Pada PT.PELINDO MARINE SERVICE”. Tujuan dari penelitian ini adalah membuat aplikasi pencatatan dan tindak lanjut audit internal SPI yang dapat membantu divisi SPI. Penelitian ini menggunakan metode System Development Life Cycle (SDLC) dalam mengembangkan aplikasi pencatatan dan tindak lanjut audit internal SPI [4].

Penelitian sebelumnya juga pernah dilakukan oleh Ahmad Miftahul Khoiri dan Dodik Arwin Dermawan Fakultas Teknik Universitas Negeri Surabaya dengan judul “Rancang Bangun Sistem Informasi Audit Internal Berbasis Web (Studi Kasus: Balai Riset dan Standardisasi Industri Surabaya)”. Balai Riset dan Standardisasi Industri Surabaya, sebuah lembaga yang bergerak di bidang sertifikasi di bawah kementerian perindustrian RI, dalam melaksanakan audit internal selama ini masih dilakukan secara manual. Belum tersedianya media perantara bagi auditor dan auditee dalam menjalankan masing-masing tugasnya, membuat pelaksanaan audit internal menjadi kurang efektif dan efisien. Untuk mengatasi permasalahan tersebut, maka diperlukan sebuah sistem yang dapat memfasilitasi semua proses di saat audit internal dilaksanakan. Sistem ini akan dibangun pada platform Web dengan menggunakan framework Laravel dan database mySQL. Proses pengembangan sistem menerapkan metode Waterfall karena memiliki tahapan yang runtut mulai dari perencanaan hingga perawatan sistem yang dikembangkan [5].

Penelitian sebelumnya juga pernah dilakukan oleh Dwi Rani Febriani dan Hendri Irawan, Fakultas Teknologi Informasi, Universitas Budi Luhur dengan judul “Penerapan Sistem Informasi Audit Mutu Internal Berbasis Web Guna Meningkatkan Efisiensi Kerja Studi Kasus: Lembaga Penjaminan Mutu Universitas Budi Luhur”. Dalam menjalani audit internal, masih terdapat sejumlah permasalahan yang terjadi di Lembaga Penjaminan Mutu Universitas Budi Luhur. Mulai dari masih ditemukan ketidaksesuaian dalam menentukan jenis temuan audit oleh auditor, hingga penyelesaian laporan audit mutu internal yang tidak dapat diselesaikan tepat waktu. Penelitian yang dilakukan bertujuan untuk membangun sebuah sistem informasi berorientasi objek untuk audit mutu

internal berbasis web menggunakan bahasa pemrograman PHP dan XAMPP yang digunakan sebagai aplikasi web server apache yang terintegrasi dengan MySQL sebagai databasenya. Hasil penelitian ini dapat membantu efisiensi dalam proses audit internal seperti daftar periksa, dan dapat membantu ketepatan waktu dalam penyelesaian laporan audit mutu internal pada Lembaga Mutu Universitas Budi Luhur [6].

Pada penelitian ini, akan dibangun aplikasi untuk mempermudah pihak SPI Politeknik Negeri Cilacap dalam melaksanakan dan mengelola audit berupa kertas data audit, berita acara, serta berkas lain yang berkaitan dengan pelaksanaan proses auditing barang milik negara di Politeknik Negeri Cilacap. Tahap pengembangan sistem informasi ini menggunakan metode SDLC (*System Development Life Cycle*) model *Waterfall* yang dibangun menggunakan bahasa pemrograman PHP (*Hypertext Preprocessor*) dan *MySQL* sebagai aplikasi *database* dengan penambahan fitur notifikasi *email*.

## **2.2. Landasan Teori**

### **2.2.1. Audit**

Adalah sebuah proses pengumpulan serta pemeriksaan bukti mengenai informasi guna menentukan dan membuat laporan terkait tingkat kesesuaian antara informasi dan kriteria yang ditetapkan.[7].

### **2.2.2. Aplikasi**

Aplikasi adalah penggunaan dalam suatu komputer, instruksi atau pernyataan yang disusun sedemikian rupa sehingga komputer dapat memproses input menjadi output [8].

### **2.2.3. Barang Milik Negara**

Menurut Peraturan Pemerintah Nomor 27 Tahun 2014 tentang Pengelolaan Barang Milik Negara (BMN) menyatakan bahwa Barang Milik Negara adalah semua barang yang dibeli atau diperoleh atas beban Anggaran Pendapatan dan Belanja Negara atau berasal dari perolehan lainnya yang sah [1].

### **2.2.4. Website**

Website adalah kumpulan halaman yang menampilkan informasi data teks, data gambar diam atau gerak, data animasi, suara, video dan atau gabungan dari semuanya, baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait dimana masing- masing dihubungkan dengan jaringan-jaringan halaman[9].

### 2.2.5. Framework

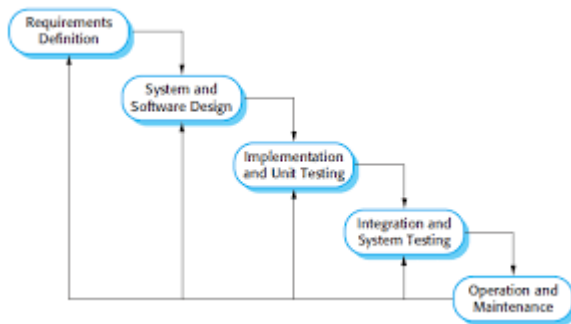
Framework merupakan sebuah kerangka kerja yang dibuat untuk memudahkan pembuatan sebuah website. Framework memiliki komponen dan variabel yang dibutuhkan para perancang website untuk memudahkan pembacaan kode, perencanaan, pengujian dan pemeliharaan[10].

### 2.2.6. Laravel

Laravel merupakan framework berbasis PHP (PHP: Hypertext Preprocessor) yang telah memiliki kerangka sistematis dengan menggunakan konsep MVC (Model View Controller). Laravel dirilis pada tanggal 5 Juni 2011 dibawah lisensi MIT License dengan menggunakan GitHub sebagai tempat berbagi kode. Setelah mengalami banyak perbaikan dan pengembangan, Laravel stabil mulai dirilis pada tanggal 6 Juni 2016 dengan versi Laravel 5.2.36 dan dibangun dengan bahasa pemrograman PHP serta sistem operasi cross platform[10].

### 2.2.7. Metode Pengembangan Sistem

Waterfall merupakan salah satu model dalam pengembangan perangkat lunak dimana semua proses kegiatan harus direncanakan dan dijadwalkan sebelum dikerjakan[12]. Proses dari metode Waterfall model antara lain[16]:



**Gambar 2.1** Model *Waterfall*

a. Analisis Kebutuhan Perangkat Lunak

Proses pengumpulan kebutuhan dilakukan secara intensif untuk menspesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh user. Spesifikasi kebutuhan perangkat lunak pada tahap ini perlu untuk didokumentasikan.

b. Desain

Desain perangkat lunak adalah proses multi langkah yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak representasi antar muka, dan prosedur pengodean. Tahap ini mentranslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya. Desain perangkat lunak yang dihasilkan pada tahap ini juga perlu didokumentasikan.

c. Pembuatan Pengkodean Program

Desain harus ditranslasikan ke dalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain.

d. Pengujian

Pengujian fokus pada perangkat lunak secara dari segi logik dan fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (error) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.

e. Pemeliharaan

Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirim ke user. Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru. Tahap pendukung atau pemeliharaan dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak yang sudah ada, tapi tidak untuk membuat perangkat lunak baru.



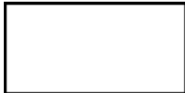
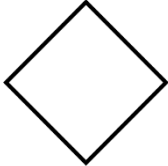

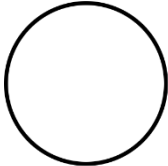
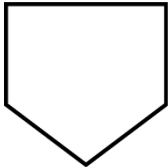
### 2.2.8. Notifikasi *Email*

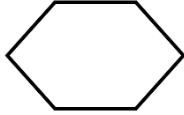
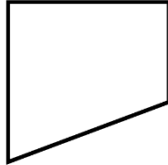
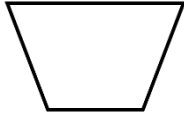

Adalah pesan yang dikirim ke yang bersangkutan secara otomatis untuk berbagai kepentingan. Setelah itu, penerima akan menerima status pesan notifikasi yang dikirimkan.

### 2.2.9. *Flowchart*

*Flowchart* atau bagan alir adalah penggambaran secara grafik dari langkah-langkah dan urutan prosedur dari suatu program. Simbol *flowchart* dipakai sebagai alat untuk membantu dalam mendefinisikan sebuah proses. Proses dalam *flowchart* digambarkan dari halaman atas ke bawah dan dari kiri ke kanan. Aktifitas yang digambarkan kemudian didefinisikan dengan jelas sehingga dimengerti oleh pembacanya [15].

Tabel 2.1 Simbol Flowchart

No	Gambar Simbol	Nama	Keterangan
1.		Terminal	Menyatakan permulaan atau akhir suatu program.
2.		Arus/ <i>Flow</i>	Untuk menyatakan jalanya arus suatu proses.
3.		Proses	Menunjukkan sebuah kegiatan atau proses yang dilakukan sistem.
4.		<i>Decision</i> /Logika	Menunjukkan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan (ya/tidak).
5.		<i>Input</i> <i>output</i>	Menyatakan proses <i>input/output</i> tanpa tergantung dengan jenis peralatannya.
6.		<i>Connector</i>	Menyatakan sambungan dari suatu proses ke proses lainnya dalam halaman/lembar yang sama.
7.		<i>Offline</i> <i>Connector</i>	Menyatakan sambungan dari suatu proses ke proses lainnya dalam halaman/lembar yang berbeda.

8.		<i>Predefined Proses</i>	Menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberi harga awal.
9.		Manual <i>input</i>	Memasukan data secara manual dengan menggunakan <i>online keyboard</i> .
10.		Manual	Menyatakan suatu Tindakan (proses) yang tidak dilakukan oleh komputer (manual).
11.		<i>Document</i>	Mencetak laporan ke printer.

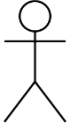
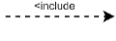



### 2.2.10. UML (*Unified Modeling Language*)

*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 *blueprint* dimana didalamnya termasuk sebuah bisnis proses, penulisan kelas-kelas dalam sebuah bahasa yang spesifik [15].

#### A. *Use Case Diagram*

Use case digunakan untuk menggambarkan fungsi dasar dari sebuah sistem informasi. Use case mendeskripsikan cara sistem bisnis berinteraksi dengan lingkungannya[12]. Berikut adalah elemen-elemen dari use case diagram:


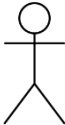
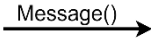
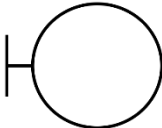
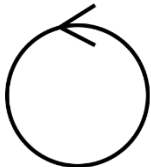
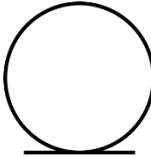
**Tabel 2.2** Simbol *Use Case Diagram*

No	Simbol	Fungsi
1.		Actor/role adalah orang atau sistem lain yang berinteraksi dengan sistem saat ini.
2.		Memasukkan satu use case dalam use case lainnya. Perilaku (behavior) yang harus terpenuhi agar sebuah event dapat terjadi, di mana kondisi ini sebuah use case adalah bagian dari use case lainnya. Tanda panah mengarah dari base use case (pusat) menuju ke use case yang di-includ.
3		Menggambarkan navigasi antar class, berupa banyak objek lain yang berhubungan dengan satu objek, dan apakah suatu class menjadi bagian dari class lainnya.
4.		Berisi nama dari sistem yang diletakkan di dalam atau di bagian atas boundary. Mewakili ruang lingkup sistem. Actor berada di luar ruang lingkup sistem.
5.		Use case adalah bagian utama dari fungsionalitas sistem. Bisa extend (memperluas) use case lainnya. Ditempatkan di dalam system boundary (batasan sistem). Dilabeli dengan kata kerja – frase kata benda.

### **B. Sequence Diagram**

Sequence diagram menunjukkan pesan yang lewat di antara objek untuk use case tertentu dari waktu ke waktu. Sequence diagram mengilustrasikan objek-objek yang berpartisipasi di dalam suatu use case [12]. Berikut adalah elemen-elemen dari sequence diagram.

**Tabel 2.2** Simbol *Sequence* Diagram

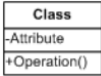
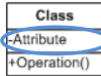
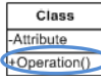
No	Simbol	Fungsi
1.		Merupakan persegi panjang sempit panjang yang ditempatkan di atas lifeline. Menunjukkan kapan suatu objek mengirim atau menerima pesan.
2..		Adalah orang atau sistem yang memperoleh manfaat dari dan berada di luar sistem. Berpartisipasi dalam suatu urutan dengan mengirim dan / atau menerima pesan. Ditempatkan di bagian atas diagram.
3.		Spesifikasi dari komunikasi antar objek yang memuat informasi – informasi tentang aktivitas yang terjadi.
4.		Menggambarkan tampilan program.
5.		Menggambarkan controller.
6.		Digunakan untuk menggambarkan hubungan kegiatan yang akan dilakukan.



### C. Class Diagram

Class diagram adalah model statis yang menunjukkan kelas dan hubungan di antara kelas yang tetap konstan dalam sistem dari waktu ke waktu. Class diagram menggambarkan kelas, yang meliputi perilaku dan keadaan, dengan hubungan antar kelas[12]. Berikut adalah elemen-elemen dari class diagram:

**Tabel 2.3** Simbol *Class Diagram*

No	Simbol	Fungsi
1.		Class, Mewakili orang, tempat, atau hal-hal yang dibutuhkan sistem untuk menangkap dan menyimpan informasi. Memiliki nama yang diketik dengan huruf tebal dan berpusat di bagian atas kotak. Memiliki daftar atribut di kotak tengahnya. Memiliki daftar operasi di kotak bawahnya. Tidak secara eksplisit menunjukkan operasi yang tersedia untuk semua kelas..
2.		Attribute, merupakan properti yang menggambarkan keadaan suatu objek. Dapat diturunkan dari atribut lain, ditampilkan dengan menempatkan garis miring sebelum nama atribut.
3		Operation, mewakili tindakan atau fungsi yang dapat dilakukan oleh kelas. Dapat diklasifikasikan sebagai konstruktor, permintaan, atau operasi pembaruan. Termasuk tanda kurung yang mungkin berisi parameter atau informasi yang diperlukan untuk melakukan operasi.

#### 2.2.11. Basis Data (*Database*)

Basis data adalah suatu kumpulan data terhubung yang disimpan secara bersama-sama pada suatu media, tanpa adanya suatu kerangkapan data, sehingga mudah untuk digunakan kembali, dapat digunakan oleh satu atau lebih program aplikasi secara optimal, data disimpan tanpa mengalami ketergantungan pada program yang akan menggunakannya, data disimpan sedemikian rupa sehingga apabila ada penambahan, pengambiiian dan modifikasj data dapat dilakukan dengan mudah dan terkontrol [13].

**~ Halaman ini sengaja dikosongkan ~**