

BAB II TINJAUAN PUSTAKA

2.1. Teori Terkait

Penelitian yang dilakukan oleh Maulia Usnaini dan rekannya membahas tentang perancangan sistem informasi inventaris aset. Tujuannya adalah untuk mengatasi permasalahan dan meningkatkan efisiensi dalam pengelolaan barang dengan mengembangkan aplikasi pencatatan inventaris berbasis web. Sistem ini diharapkan dapat memfasilitasi pengumpulan, pemantauan, dan pelaporan aset sekolah, serta mendukung proses pengambilan keputusan yang lebih baik serta akurat. Pada penelitian ini menerapkan metode *waterfall*[1].

I Komang Wiratama bersama rekannya membahas tentang penerapan sistem informasi inventarisasi di kantor desa Ketewel. Tujuannya yaitu untuk mengembangkan dan mengevaluasi sistem pencatatan inventaris yang berbasis website guna meningkatkan efisiensi dalam pengelolaan inventaris di organisasi, khususnya di sektor pendidikan. Aplikasi ini dirancang menggunakan *PHP* sebagai bahasa pemrograman, serta *MySQL* sebagai basis data, dan *framework CodeIgniter*[5].

Penelitian yang dilakukan oleh Doni Mahendra dan rekannya membahas tentang perancangan dan sistem inventaris dengan menggunakan teknik *web engineering*. Penelitian menggunakan teknik *web engineering* memiliki beberapa tahap, yaitu dari *costumer communication*,

communication, modeling, planning, construction, deployment dan implementasi aplikasi inventaris dirancang. Dengan memanfaatkan *PHP* sebagai bahasa pemrograman, sistem basis data *MySQL*, dan *framework CodeIgniter*[6].

Muhammad Saed Novendri dan rekannya melakukan penelitian yang membahas tentang aplikasi inventarisasi dikembangkan dengan bahasa pemrograman *PHP* dan *MySQL* sebagai sistem basis datanya. Tujuannya adalah untuk mengembangkan sistem informasi persediaan yang dapat menyederhanakan pengelolaan data barang yang sebelumnya dilakukan secara manual. Sistem informasi yang dirancang akan mencakup berbagai fungsi, seperti pengumpulan data lokasi penyimpanan, pencatatan barang masuk dan keluar, serta pencatatan laporan. Pengembangan pada sistem ini akan memanfaatkan bahasa pemrograman *PHP*, *CSS*, dan *HTML* untuk antarmuka pengguna, serta *MySQL* sebagai basis data[3].

Dalam referensi di atas, sudah banyak penelitian yang dilakukan seputar Website Inventaris barang. Namun, dalam penelitian tugas akhir ini dengan judul Sistem Informasi Inventaris Barang Dinas Komunikasi Informatika dan Statistik Kabupaten Brebes ini dilakukan untuk mengembangkan serta menyempurnakan aplikasi dari penelitian sebelumnya.

2.2. Landasan Teori

2.2.1. *Hypertext Preprocessor (PHP)*

Jenis bahasa pemrograman yang digunakan yang sering digunakan untuk pengembangan aplikasi website adalah *Hypertext Preprocessor* atau yang sering disingkat *PHP*. *PHP* adalah jenis bahasa pemrograman *scripting* dan juga berinteraksi dengan server. *PHP* menggunakan tambahan kode *HTML* untuk membuat aplikasi yang dapat memproses data secara efisien. Beberapa data yang diterima akan diolah melalui server, sementara sebagian lainnya akan ditransfer ke peramban (*browser*). Kode skrip khusus ini diinstal di server dan dieksekusi di sisi klien, yang merupakan *interface* pengguna browser, menampilkan hasil eksekusi tersebut[7].

Hal yang membedakan *PHP* dari produk sejenis lainnya adalah bahwa proses eksekusi kode dilakukan di sisi server. Dengan demikian, pengguna tidak memiliki akses untuk melihat struktur pemrograman yang digunakan dalam aplikasi[8].

2.2.2. Visual Studio Code

Visual Studio Code merupakan sebuah *software* pengedit kode sumber yang tersedia secara gratis dan dikembangkan oleh *Microsoft*, yang digunakan untuk platform *Windows*, *Linux*, dan *macOS*. *Software* ini dilengkapi dengan beragam kemampuan, termasuk fasilitas untuk pemecahan masalah (*debugging*), integrasi kontrol *Git*, serta kemampuan untuk terhubung dengan platform *GitHub*. Selain itu,

Visual Studio Code juga memungkinkan pengguna untuk menyesuaikan pengaturan sesuai preferensi mereka, seperti tema, pintasan *keyboard*, dan opsi preferensi lainnya. Pengguna juga dapat memperluas fungsionalitas editor dengan menginstal ekstensi tambahan yang tersedia[9]. Editor teks ini secara langsung mendukung pengembangan dalam bahasa pemrograman *JavaScript*, *Typescript*, dan *Node.js*. Selain itu, editor juga dapat diintegrasikan dengan bahasa pemrograman lainnya melalui plugin yang tersedia di toko aplikasi *Visual Studio Code* (seperti *C++*, *C#*, *Python*, *Go*, *Java*, dan lainnya)[10].

2.2.3. Laragon

Laragon merupakan *software* gratis yang kompatibel dengan berbagai sistem operasi dan juga memiliki fungsi sebagai server lokal atau *localhost*. Laragon menyediakan beragam macam layanan, alat, dan fitur seperti *Apache*, *MySQL*, *PHP Server*, *Memcached*, *Redis*, *Composer*, *Xdebug*, *PhpMyAdmin*, *Cmdr*, dan *Laravel*[11].

2.2.4 MySQL

MySQL ialah salah satu sistem manajemen basis data (SMBD) yang paling terkenal dan populer dalam komunitas pengembangan perangkat lunak. Sistem basis data ini memiliki fitur-fitur yang mendukung, seperti kemampuan *multithreading*, kemampuan untuk mendukung beberapa pengguna secara bersamaan, dan fungsionalitas

yang lengkap dalam hal manajemen basis data. *MySQL* dibangun guna memenuhi keperluan akan sistem basis data yang mampu memberikan performa yang cepat, keandalan yang tinggi, serta kemudahan penggunaan dengan menggunakan bahasa *Structured Query Language* (SQL)[12].

2.2.5 Database

Basis data atau *Database* merupakan suatu entitas yang terdiri dari kumpulan data terstruktur yang disimpan dan diakses dengan elektronik melalui sistem komputer. *Database* ini memiliki tingkat kompleksitas yang cukup tinggi dan terus berkembang sejalan dengan waktu. Semakin banyak data yang terdapat dalam basis data, semakin besar pula kebutuhan akan keamanan informasi agar tidak rentan terhadap serangan atau pencurian data[13].

Tujuan penggunaan database di suatu lembaga pada dasarnya adalah untuk memfasilitasi dan meningkatkan efisiensi dalam proses pengambilan dan penyimpanan data. Salah satu kelebihan dari penggunaan database adalah kemampuannya untuk mengorganisir data berdasarkan fungsinya dan jenisnya secara sistematis dan tepat. Ini bisa dilakukan dengan memisahkan data ke dalam beberapa file atau tabel terpisah, atau dengan mengelompokkan data dalam kolom atau bidang dalam setiap file atau tabel[14].

2.2.6 Laravel

Laravel merupakan sebuah kerangka kerja dari bahasa pemrograman PHP yang dirilis di bawah lisensi MIT, serta dirancang menggunakan konsep Model View Controller atau disingkat MVC. Kerangka kerja ini bertujuan guna memperbaiki mutu perangkat lunak dengan menghemat biaya pengembangan dan pemeliharaan pada tahap awal, sambil juga memperbaiki pengalaman pengguna dengan menyediakan sintaks yang ekspresif, jelas, dan efisien[15].

2.2.7 *Unified Modeling Language* (UML)

Bahasa pemodelan *software* yang telah distandardisasi untuk menyusun membuat cetak biru (*blueprint*) perangkat lunak disebut dengan *Unified Modeling Language* (UML). *Unified Modeling Language* memungkinkan visualisasi, spesifikasi, perancangan, dan dokumentasi berbagai komponen sistem dalam perangkat lunak. Dengan analogi yang sama, seperti seorang arsitek membuat rencana yang dapat digunakan oleh kontraktor saat membangun sebuah gedung, arsitek *software* membuat diagram UML yang membantu programmer dan pengembang dalam membangun *software*[16].

Terdapat empat model UML yang paling sering digunakan untuk menggambarkan suatu desain sistem *Usecase* diagram, *Class* diagram, *Behavioral State machine* diagram, dan juga *Sequence* diagram. Teknik-teknik pemodelan *Unified Modeling Language* ini disebut juga dengan

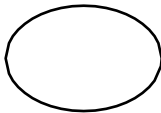


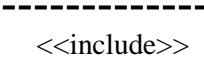
4 teknik dasar. Dalam proyek berorientasi objek, keempat teknik UML ini sangat mendominasi penggunaannya.


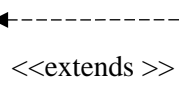
Dalam perancangan sistem terdapat *UML* yang sering digunakan sebagai berikut:

1. *Usecase* Diagram

Dalam pembuatan sistem informasi, *Usecase* Diagram dimanfaatkan untuk memodelkan perilaku atau tingkah laku sistem informasi dan bersifat statis. Gambaran simbol – simbol dalam penggunaan *Usecase* Diagram dapat ditemukan dalam Tabel 2.1

Tabel 2. 1 Simbol *Usecase* Diagram



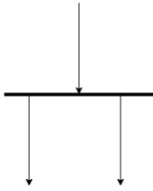

No	Simbol	Keterangan
1.		<i>Use Case</i> merupakan bentuk untuk menggambarkan interaksi antara sistem dengan <i>Actor</i> .
2.		<i>Actor</i> merupakan himpunan peran untuk berinteraksi dengan <i>Usecase</i> .
3.		<i>Association</i> merupakan garis yang menggambarkan hubungan antara satu objek dengan objek lainnya.
4.		<i>Include</i> , merupakan gambaran jika <i>use case</i> dipanggil oleh <i>usecase</i> lain.

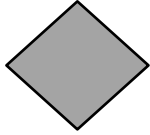
No	Simbol	Keterangan
5.		<i>Dependency</i> merupakan garis panah yang menunjukkan jika aktor berinteraksi secara pasif.
6.		<i>Extend</i> , merupakan gambaran jika memperluas <i>usecase</i> target.

2. Activity Diagram

Aktivitas yang terjadi pada sistem disebut *Activity* Diagram. Diagram ini berguna untuk memvisualisasikan urutan langkah alur kerja sistem. Simbol yang digunakan dalam *Activity* Diagram dapat dilihat pada Tabel 2.2.

Tabel 2. 2 Simbol *Activity* Diagram


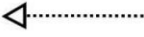

No	Simbol	Keterangan
1.		<i>End Point</i> atau <i>Final Node</i> adalah representasi dari hasil akhir dari suatu aktivitas.
2.		<i>Start Point</i> adalah awal dari suatu aktivitas yang ditempatkan di pojok kiri atas.
3.		<i>Fork</i> atau <i>Join</i> , digunakan untuk memarallelkan suatu kegiatan atau penggabungan 2 kegiatan paralel menjadi satu.
4.		<i>Activity</i> , merupakan gambaran dari suatu proses.

No	Simbol	Keterangan
5.		<i>Decision</i> , merupakan pilihan pengambilan suatu keputusan <i>false or true</i> .

3. Class Diagram

Diagram statis dari sebuah sistem disebut dengan *Class Diagram*, yang mencakup kelas, antarmuka, dan hubungan di antara mereka. Simbol yang digunakan *Class Diagram* ada pada Tabel 2.3.

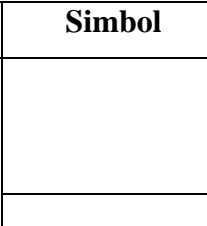
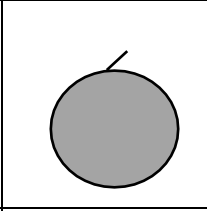
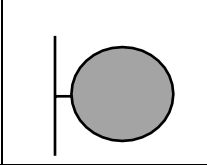
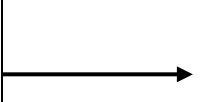
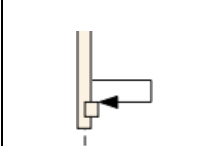
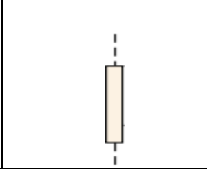

Tabel 2. 3 Simbol *Class Diagram*.

No	Gambar	Keterangan
1.		<i>Generallization</i> adalah ketika objek turunan mengadopsi perilaku dan struktur data dari objek induknya.
2.		<i>Class</i> , adalah kumpulan objek yang saling berbagi.
3.		<i>Nary Association</i> , digunakan untuk asosiasi terhindar dengan objek lainnya.
4.		<i>Collaboration</i> adalah representasi dari urutan aksi yang dijelaskan oleh suatu sistem yang memiliki dampak yang dapat diukur bagi aktor yang terlibat.
5.		Ini merupakan operasi yang dapat dilakukan secara sah oleh suatu objek.
6.		Ini merupakan garis panah yang menunjukkan interaksi pasif para aktor.
7.		<i>Association</i> merupakan garis yang menghubungkan objek satu dengan objek lainnya.

1. *Sequence Diagram*

Representasi visual dari interaksi yang terjadi antara objek dalam suatu sistem disebut dengan *Sequence Diagram*. Diagram ini termasuk pengiriman dan penerimaan pesan. Simbol yang digunakan dalam *Sequence Diagram* silakan lihat Tabel 2.4.

Tabel 2. 4 Simbol *Sequence Diagram*

No	Simbol	Keterangan
1.		<i>Entity Class</i> , merupakan bentuk deskripsi awal sistem dan berisi kumpulan kelas dalam bentuk unsur yang mendasari untuk membuat database.
2.		<i>ControlClass</i> , merupakan gambaran penghubung antara Boundaryclass dengan suatu tabel
3.		<i>Boundary Class</i> , merupakan gambaran dari penggambaran form.
4.		Pesan atau <i>message</i> , menunjukkan pengiriman pesan antar <i>class</i> .
5.		<i>Self Message</i> , menunjukkan pengiriman suatu pesan yang akan dikirim ke objek itu sendiri.
6.		<i>Activation</i> , menggambarkan suatu objek yang melakukan sebuah aksi/eksekusi operasi.
7.		<i>Lifeline</i> , Garis titik terkoneksi keobjek di sepanjang garis <i>lifeline</i> memiliki aktivasi.

2.2.8. Pengujian Sistem

Tahapan pengujian sangat penting dalam sistem informasi untuk memastikan aplikasi memenuhi persyaratan teknis dan bisnis yang diharapkan sebelum diberikan kepada pengguna (*customer*). Pengujian perangkat lunak adalah proses menjalankan program dengan tujuan menemukan *bug (error)* yang dapat menyebabkan kegagalan perangkat lunak. Pengujian dianggap berhasil jika mampu mengungkap kesalahan dalam program yang diuji. Pengujian perangkat lunak mencakup dua metode utama yaitu *black-box* dan *white-box*. Pengujian *black-box* berfokus pada fungsionalitas, memastikan aplikasi berfungsi sesuai dengan spesifikasinya. Sebaliknya, pengujian *white-box* mengevaluasi cara kerja internal aplikasi, dengan tujuan memastikan keakuratan metode yang digunakan[17].