

BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1 Tinjauan Pustaka

Beberapa penelitian yang serupa mengenai inventaris barang seperti diantaranya oleh Nurul Huda dan Rahayu Amalia (2020) pada PT. PLN (Persero) WS2JB Cabang Palembang yang terletak di Sumatera Selatan dimana perusahaan mengalami kendala seperti misalnya kesulitan dalam menyajikan laporan data barang sehingga menyebabkan laporan dilaksanakan secara manual sehingga laporan seringkali terlambat dan sering terjadi duplikasi. Sehingga diperlukan sistem informasi yang menunjang perusahaan tersebut yang dapat membantu mengatasi masalah yang ada. Sistem ini dibangun menggunakan metode *waterfall* dengan perangkat lunak XAMPP, bahasa pemrograman PHP dan *database* MySQL. Hasil dari penelitian ini efektif untuk menyajikan data inventaris barang sesuai dengan divisi yang ada[5].

Penelitian lainnya dilakukan oleh Albert Anotona Hary Saputra Gulo (2021) dimana PT. Quantum Mitra Sinergi yang merupakan perusahaan dalam bidang IT mengalami kendala dalam mencatat barang masuk dan keluar secara *mobile* sehingga diperlukan sebuah sistem informasi yang dapat membantu pekerja dalam mencatat barang masuk dan keluar dalam gudang perusahaan. Metode yang digunakan dalam penelitian ini adalah metode *System Development Life Cycle (SDLC) waterfall* yang dibangun berbasis website dan aplikasi yang dapat diinstal langsung ke perangkat Android. Hasil dari penelitian ini memudahkan pekerja dalam melakukan pendataan barang karena adanya fitur *scan barcode* pada aplikasi yang dikembangkan untuk perangkat berbasis sistem operasi Android[6].

Penelitian selanjutnya telah dilakukan oleh Victor Marudut Mulia Siregar (2018) dimana SMA Negeri 4 Pematangsiantar merupakan lembaga pendidikan yang terletak di Provinsi Sumatera

Utara yang memiliki masalah terkait dengan pencatatan data tempat inventaris barang, pencatatan barang masuk dan keluar, serta laporan data barang sehingga diperlukan sistem yang dapat mempermudah pekerjaan pencatatan inventaris barang. Sistem ini dibangun menggunakan rancangan *Data Flow Diagram* (DFD) dan berbasis perangkat lunak *desktop* yang dikembangkan berdasarkan bahasa pemrograman Visual Basic NET dan *database* MySQL. Hasil dari penelitian ini pekerjaan pendataan inventaris barang yang ada di SMA Negeri 4 Pematangsiantar dapat dilaksanakan lebih cepat dan akurat[7].

Penelitian lain juga telah dilakukan oleh Zulfiqri Nuryadi Ahmad (2020) dimana PT. KFC di bagian gudang memiliki kendala dalam kegiatan keluar dan masuknya barang dimana pencatatan masih kurang efektif dan efisien dalam melaksanakan kegiatan pendataan barang masuk dan keluar dikarenakan rekapitulasi masih menggunakan tulis tangan dan diinput melalui Microsoft Excel serta memerlukan penyimpanan data yang lebih banyak sehingga diperlukan suatu sistem yang dapat mempercepat penghitungan data barang masuk dan keluar. Sistem yang dibangun menggunakan bahasa pemrograman Java dengan perangkat lunak Netbeans 8.2 dan *database* MySQL. Hasil dari penelitian ini efektif dalam meningkatkan kinerja karyawan dalam menghadapi revolusi industri 4.0 dan manajemen inventaris barang menjadi lebih maju daripada sebelumnya[8].

Selain itu, penelitian lainnya juga telah dilakukan oleh Asep Deddy Supriatna, Sri Rahayu, dan Adam Fakhrol Rozi (2021) dimana Pondok Pesantren Hidayatussalam Garut memiliki masalah dalam mengelola inventaris barang yang ada dikarenakan pendataan masih menggunakan buku catatan inventaris barang. Proses pelaporan data juga dilaksanakan secara manual sehingga tidak dapat secara langsung diakses ketika diperlukan sewaktu-waktu sehingga dibangun suatu sistem informasi yang dapat menunjang proses inventaris barang. Sistem yang dibangun berbasis website dengan metode *Rapid Application Development* (RAD) menggunakan *Entity Relationship Diagram* (ERD) sebagai penggambaran definisi dari

hubungan antar entitas. Hasil dari penelitian ini memudahkan pondok pesantren dalam menyajikan laporan secara akurat dan informasi yang ada sesuai dengan kebutuhan[9].

Berdasarkan penelitian yang telah dilakukan sebelumnya dan melakukan perbandingan ditemukan suatu perbedaan yaitu sistem akan terdapat fitur grafik kondisi barang baik, rusak sedang maupun rusak berat. Selain itu sistem dibangun menggunakan metode *System Development Life Cycle (SDLC)* dengan metode *prototyping* dan dikembangkan berbasis website menggunakan *framework* Laravel dan MySQL sebagai media penyimpanan *database*.

2.2 Landasan Teori

2.2.1 Sistem Informasi

Sistem informasi adalah beberapa komponen yang berhubungan antara yang satu dengan yang lain agar dapat mencapai tujuan. Adanya sistem informasi maka pekerjaan yang ada dapat dilakukan dengan mudah melalui komputer[10]. Sistem informasi umumnya berfungsi untuk mengumpulkan, memproses, menyimpan, menganalisis, dan membagikan informasi yang diperlukan[11]. Komponen yang ada di dalam sistem informasi adalah sebagai berikut[12] :

1. Komponen *input*, yaitu data yang masuk ke dalam sistem informasi.
2. Komponen model, yaitu beberapa gabungan dari langkah, logika dan model matematika yang melakukan proses data yang tersimpan untuk menghasilkan *output* yang diperlukan.
3. Komponen *output*, yaitu hasil informasi yang berkualitas dan dokumentasi yang berguna untuk seluruh tingkatan manajemen dan pengguna sistem.
4. Komponen teknologi, yaitu alat yang digunakan untuk menerima *input*, menyimpan data, mengakses data, menghasilkan data dan menerima *output* dan memantau pengendalian sistem.

5. Komponen basis data, yaitu suatu kumpulan data yang berhubungan satu sama lain yang disimpan ke dalam komputer dengan menggunakan *software database*.
6. Komponen kontrol, yaitu komponen yang mengendalikan gangguan yang ada terhadap sistem informasi.

2.2.2 Rekayasa Website

Rekayasa Website adalah pengembangan, operasi, dan pemeliharaan aplikasi berbasis website yang merupakan subdisiplin dari Rekayasa Perangkat Lunak untuk merancang, mengembangkan, memelihara, dan melibatkan aplikasi berbasis website. Meskipun rekayasa website dibuat menggunakan beberapa bahasa pemrograman dan mengadopsi pengembangan perangkat lunak, pengembangan website berbeda dari rekayasa perangkat lunak[13].

A. Metode Pengembangan Sistem

Metode yang digunakan dalam pengembangan Sistem Informasi Inventaris Barang dan Aset Desa di Kantor Desa Kedawung ini yaitu *System Development Life Cycle* (SDLC) yang merupakan proses pengembangan dan mengubah sistem perangkat lunak dengan model dan metodologi yang digunakan untuk mengembangkan sistem[14]. *System Development Life Cycle* (SDLC) yang digunakan dalam pengembangan sistem adalah metode *Prototyping.a*

Metode *Prototyping* adalah metode pengembangan sistem yang menghasilkan prototipe dari suatu perangkat lunak yang digunakan oleh pengembang untuk dapat berinteraksi dengan pengguna untuk mendemonstrasikan tampilan awal dan mencari solusi yang tepat sehingga pengguna sistem dapat lebih siap dalam perubahan yang akan terjadi dalam suatu sistem[15].

B. Metode Pengujian Sistem

Metode pengujian sistem yang digunakan adalah metode *Blackbox Testing*. *Blackbox Testing* merupakan metode pengujian perangkat lunak yang melakukan pengetesan fungsi dan struktur

kerja perangkat lunak. Berbeda daripada *Whitebox Testing* yang melakukan pengujian menggunakan kode program. *Blackbox Testing* mendeskripsikan eksternal *hardware* seperti spesifikasi, persyaratan, dan desain[16].

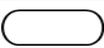
2.2.3 Framework

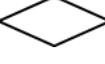
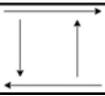
Framework adalah suatu kumpulan perintah yang dikumpulkan dalam *class and function* yang berguna untuk memudahkan pengembang website dalam memanggilnya tanpa harus menuliskan sintaks program yang berulang untuk dapat menghemat waktu[17]. Ada berbagai macam *framework* untuk pengembangan website diantaranya adalah Laravel. Laravel adalah *framework* yang dikembangkan oleh Taylor Otwell yang memiliki banyak fungsi-fungsi kode yang disediakan untuk kemudian diinstal ke Laravel agar pengembang dapat membuat proyek pengembangan website dari skala rendah hingga menengah[18].

2.2.4 Flowchart

Flowchart merupakan bagan yang berisi prosedur atau logika kerja suatu sistem. *Flowchart* digunakan untuk menjelaskan mengenai tahap-tahap penyelesaian suatu masalah dengan menggunakan simbol-simbol yang mudah dipahami. Tujuan dari *flowchart* sendiri yaitu untuk membuat gambaran tahapan penyelesaian suatu masalah sederhana dan rapi menggunakan simbol standar sehingga dapat dengan mudah dipahami oleh *programmer*[19]. Simbol-simbol *flowchart* dapat dilihat pada Tabel 2.1 dan Tabel 2.2.

Tabel 2.1 Simbol-Simbol *Flowchart*[20]

Simbol	Nama	Keterangan
	Terminal	Berfungsi untuk menggambarkan

		pemulaian dan pengakhiran program
	Proses	Menggambarkan proses penghitungan atau pengolahan suatu data
	Predefined Process	Melakukan permulaan sub program atau proses pengolahan data
	Preparation	Menggambarkan proses inialisasi data
	Input-Output	Menggambarkan masukan data atau penunjukkan hasil dari proses
	Decision	Menggambarkan suatu keputusan untuk memilih proses dari suatu kondisi yang ada
	Flow	Menggambarkan hubungan antara simbol yang satu dengan yang lain

Tabel 2.2 Simbol-Simbol Aliran Sistem Informasi[21]

Simbol	Nama	Keterangan
	<i>Document</i>	Menunjukkan dokumen input dan output untuk manual maupun mekanik atau komputer
	<i>Connector</i>	Penghubung bagian-bagian <i>flowchart</i> yang berada pada satu halaman

	<i>Offline Connector</i>	Penghubung bagian-bagian <i>flowchart</i> yang berada pada halaman yang lain
	<i>Manual Operation</i>	Menunjukkan pengolahan yang dilakukan secara manual
	<i>Manual Input</i>	Memasukan data secara manual <i>on-line keyboard</i> .

2.2.5 Unified Modelling Language (UML)

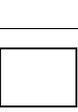
Unified Modelling Language (UML) adalah suatu alat yang berfungsi untuk melakukan visualisasi dan dokumentasi dari analisa dan desain dalam memodelkan sistem secara visual dan merupakan suatu pemodelan dalam menggambarkan sistem yang terkait dengan objek[22]. Berikut ini adalah jenis-jenis UML :

a. *Use Case Diagram*

Use Case Diagram adalah model untuk *behavior* sistem informasi yang akan dibuat. *Use Case Diagram* menjelaskan interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat[23]. *Use Case Diagram* digunakan agar mengetahui fungsi yang ada dalam sistem dan menentukan pengguna dalam sistem itu sendiri. Simbol dalam *Use Case* terdapat pada Tabel 2.3.

Tabel 2.3 Simbol-Simbol Diagram *Use Case*[24]

Simbol	Nama	Deskripsi
	Aktor	Menjelaskan pengguna ketika berinteraksi dengan <i>use case</i>
	<i>Use Case</i>	Menjelaskan terkait urutan aksi yang ditampilkan sistem sehingga menghasilkan suatu hasil yang terukur

	<i>Association</i>	Menjelaskan hubungan objek yang satu dengan yang lain
	<i>Include</i>	Menjelaskan bahwa <i>use case</i> sumber eksplisit
	<i>Extend</i>	Menjelaskan bahwa <i>use case</i> target memperluas perilaku dari usecase pada titik yang diberikan
	Sistem	Menjelaskan paket yang menampilkan sistem secara terbatas

b. *Sequence Diagram*

Sequence Diagram yaitu model untuk menggambarkan *behavior* dari satu skenario. *Sequence Diagram* menunjukkan beberapa objek yang terlibat dalam suatu proses dan menjelaskan mengenai urutan penyampaian informasi antar objek tersebut[25]. Simbol dalam *Sequence Diagram* digambarkan pada Tabel 2.4.

Tabel 2.4 Simbol *Sequence Diagram*[26]

Simbol	Nama	Deskripsi
	Aktor	Menggambarkan orang yang berinteraksi dengan sistem
	<i>Focus Control / Lifeline</i>	Menggambarkan tempat mulai dan berakhirnya <i>message</i>

	<i>Message</i>	Menggambarkan pengiriman pesan
	<i>Boundary Class</i>	Menggambarkan suatu gambaran dari <i>foem</i>
	<i>Control Class</i>	Menggambarkan penghubung antara <i>boundary</i> dengan tabel
	<i>Entity Class</i>	Menggambarkan hubungan yang akan dilakukan

2.2.6 Basis Data (*Database*)

Basis data adalah suatu sistem terkomputerisasi yang bertujuan untuk menyimpan data yang diolah dan membuat informasi tersedia ketika diperlukan. Secara sederhana, basis data adalah media untuk menyimpan suatu data agar dapat diakses dengan mudah dan cepat secara *real time*[27].

a. *Database Management System (DBMS)*

Database Management System (DBMS) adalah suatu kumpulan data yang saling terhubung dengan perangkat lunak untuk dikelola. DBMS terdiri dari perangkat keras dan perangkat lunak pendukung pengelola data yang berfungsi untuk melihat, menambahkan, menghapus, dan mengubah suatu data[28].

b. *Structure Query Language (SQL)*

Structure Query Language (SQL) adalah bahasa yang digunakan untuk melakukan pengolahan *Relation Database Management System* (RDBMS) seperti memberikan perintah menampilkan, menambah, menghapus, dan mengubah suatu data yang tersimpan[29]. Salah satu dari perangkat lunak dari SQL yaitu MySQL. MySQL adalah jenis *database* server yang populer dan digunakan untuk mengakses *database* yang mengandung satu atau beberapa tabel[30].

c. *Class Diagram*

Class Diagram merupakan diagram yang diperlukan untuk menampilkan kelas atau paket pada sistem yang akan digunakan. *Class Diagram* menggambarkan sistem atau relasi yang ada pada sistem tersebut sehingga dapat memperlihatkan gambaran struktur kelasnya dan hubungannya secara logika[31]. Simbol dalam *Class Diagram* digambarkan pada Tabel 2.5.

Tabel 2.5 Simbol *Class Diagram*[32]

Simbol	Nama	Keterangan
	Kelas	Kelas pada struktur sistem
	Antarmuka / <i>Interface</i>	Sama dengan konsep didalam pemrograman berorientasi objek (PBO)
	Asosiasi / <i>Association</i>	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
	Asosiasi Berarah / <i>Directed Association</i>	Relasi antarkelas dengan makna antara kelas yang satu

		dengan yang lain digunakan dalam kelas lain, Biasanya disertai dengan <i>multiplicity</i>
	Generalisasi	Relasi antarkelas dengan makna generalisasi-spesialisasi (umum-khusus)
	Kebergantungan / <i>Dependency</i>	Relasi antarkelas saling bergantung satu sama lain
	Agresiasi / <i>Aggregation</i>	Relasi antarkelas dengan makna keseluruhan bagian (<i>whole-part</i>)

d. *Entity Relationship Diagram (ERD)*

Entity Relationship Diagram (ERD) merupakan diagram yang digambarkan dalam bentuk grafis dalam pembuatan *database* yang menghubungkan hubungan antar data yang satu dengan data yang lain. Fungsi *Entity Relationship Diagram* adalah untuk membantu pembuatan *database* dan memberikan gambaran terhadap kerja *database* yang akan dibuat[33]. Simbol dalam *Entity Relationship Diagram (ERD)* digambarkan pada Tabel 2.6.

Tabel 2.6 Simbol *Entity Relationship Diagram (ERD)*[34]

Simbol	Nama	Keterangan
	<i>Entity</i>	<i>Entity</i> , simbol yang menyatakan elemen yang didokumentasi menggunakan data
	<i>Attribute</i>	<i>Attribute</i> , simbol yang digunakan untuk

		menunjukkan nama atribut pada suatu entitas
	<i>Primary Key Attribute</i>	<i>Primary Key Attribute</i> , simbol yang digaris bawah berfungsi untuk menjadi key diantara beberapa nama atribut yang ada pada suatu entitas
	<i>Relationship</i>	<i>Relationship</i> , simbol yang menyatakan relasi berfungsi untuk menunjukkan hubungan antara entitas yang satu dengan entitas yang lain
	<i>Link</i>	<i>Link</i> , simbol yang digunakan untuk menghubungkan relasi entitas yang satu dengan entitas dengan antribut yang lain

2.2.7 Inventaris Barang dan Aset

Inventaris merupakan daftar dari semua fasilitas yang ada di seluruh bagian termasuk gedung dan isinya. Inventarisasi barang atau aset memiliki tujuan untuk memberikan tanda pengenal bagi barang atau aset yang ada. Inventaris diharuskan mengandung informasi yang jelas dan mudah dimengerti sehingga dapat memperlancar pekerjaan. Dengan demikian perawatan atas barang atau aset akan menjadi lebih mudah[35]. Penentuan kondisi barang dan aset diperlukan untuk mengukur suatu kinerja dari aset maupun barang. Barang atau aset diharuskan untuk dapat digunakan dengan aman dan efektif. Oleh karena itu diperlukan suatu pemeliharaan agar dapat digunakan dan dimanfaatkan dengan baik[36]. Kondisi barang dan aset dikelompokkan sebagai berikut mengacu pada Peraturan Menteri Keuangan Republik Indonesia Nomor 181/PMK.06/2016 Tentang Penatausahaan Barang Milik Negara[37] :

Peralatan, Mesin, dan Aset Tetap Lainnya

1. Baik, yaitu apabila kondisi barang tersebut masih dalam keadaan utuh dan berfungsi dengan baik.
2. Rusak Ringan, yaitu apabila kondisi barang tersebut masih dalam keadaan utuh, tetapi kurang berfungsi dengan baik. Untuk berfungsi dengan baik memerlukan perbaikan ringan dan tidak memerlukan penggantian bagian utama/komponen pokok.
3. Rusak Berat, yaitu apabila kondisi barang tersebut tidak utuh dan tidak berfungsi lagi atau memerlukan perbaikan besar/penggantian bagian utama/komponen pokok, sehingga tidak ekonomis lagi untuk diadakan perbaikan/rehabilitasi.

Jalan

1. Baik, yaitu apabila kondisi jalan dalam keadaan utuh dan berfungsi dengan baik
2. Rusak Ringan, yaitu apabila kondisi jalan dalam keadaan utuh namun memerlukan perbaikan ringan agar dapat digunakan sebagaimana mestinya
3. Rusak Berat, yaitu apabila kondisi jalan dalam keadaan tidak utuh/tidak berfungsi dengan baik dan memerlukan perbaikan dengan biaya besar

Tanah

1. Baik, yaitu apabila kondisi tanah tersebut siap dipergunakan dan/atau dimanfaatkan sesuai dengan peruntukannya.
2. Rusak Ringan, yaitu apabila kondisi tanah tersebut karena suatu sebab tidak dapat dipergunakan dan/atau dimanfaatkan dan masih memerlukan pengolahan/perlakuan (contoh: pengeringan, pengurukan, perataan, dan pemadatan) untuk dapat dipergunakan sesuai dengan peruntukannya.
3. Rusak Berat, yaitu apabila kondisi tanah tersebut tidak dapat lagi dipergunakan dan/atau dimanfaatkan sesuai dengan peruntukannya karena adanya bencana alam, erosi, dan sebagainya.

- HALAMAN INI SENGAJA DIKOSONGKAN -