

BAB II

TINJAUAN PUSTAKA

2.1 Perangkat Lunak

Pressman mengemukakan bahwa perangkat lunak adalah instruksi-instruksi dalam hal ini adalah program komputer yang ketika dijalankan menyediakan fitur-fitur, fungsi-fungsi dan kinerja-kinerja yang dikehendaki pengguna, dalam perangkat lunak terdapat struktur data yang memungkinkan program-program dapat memanipulasi informasi, dan informasi tersebut tercetak dalam bentuk maya yang menggambarkan pengoperasian dan penggunaan program-program tersebut (Roger S. Pressman, 2012). Dari definisi di atas dapat disimpulkan bahwa perangkat lunak adalah suatu program yang dapat beroperasi untuk memanipulasi informasi, yang di dalamnya terdapat instruksi-instruksi berupa fitur atau fungsi tertentu sehingga informasi yang telah diperoleh dapat tercetak dalam bentuk maya.

Selanjutnya Sukamto dan Shalahuddin (2013:2), mendefinisikan bahwa “Perangkat Lunak (*Software*) adalah program komputer yang terasosiasi dengan dokumen perangkat lunak seperti dokumentasi kebutuhan, model desain, dan cara penggunaan (*user manual*)”.

Jadi, dari uraian definisi para ahli di atas, penulis dapat menyimpulkan bahwa perangkat lunak adalah istilah khusus untuk data yang diformat, dan disimpan secara digital, termasuk program komputer, dokumentasinya, dan berbagai informasi yang bisa dibaca, dan ditulis oleh komputer. Dengan kata lain, bagian sistem komputer yang tidak berwujud. Istilah ini menonjolkan perbedaan dengan perangkat keras komputer.

2.2 Manajemen

Menurut Lilis Sulastri, mengutip didalam bukunya Manajemen Sebuah Pengantar Sejarah, Tokoh, Teori, dan Praktik, mendefinisikan bahwa manajemen adalah

suatu seni mengatur yang melibatkan proses, cara, dan tindakan tertentu, seperti perencanaan, pengorganisasian, pengarahan dan pengendalian/ pengawasan, yang dilakukan untuk menentukan dan mencapai tujuan secara efisien dan efektif dengan dan melalui orang lain (Sulastri Lilis, 2014).

Sebagaimana dikutip oleh Pidarta, Johnson mengemukakan bahwa manajemen adalah proses mengintegrasikan sumber-sumber yang tidak berhubungan menjadi sistem total untuk menyelesaikan suatu tujuan. Dan juga sebagaimana dikutip oleh Handoko, Stoner menyebutkan bahwa “manajemen adalah proses perencanaan, pengorganisasian, pengarahan, dan pengawasan usaha-usaha para anggota dan penggunaan sumber daya-sumber daya organisasi lainnya agar mencapai tujuan organisasi yang telah ditetapkan. (Abdul Choliq, 2011:3)

Dari beberapa definisi di atas dapat disimpulkan bahwa manajemen adalah serangkaian kegiatan merencanakan, mengorganisasikan, menggerakkan, mengendalikan dan mengembangkan segala upaya dalam mengatur dan mendayagunakan sumber daya manusia, sarana dan prasarana untuk mencapai tujuan organisasi yang telah ditetapkan secara efektif dan efisien.

2.3 Definisi Barang Habis Pakai

Sesuai dengan PerPres Nomor 54 Tahun 2010 Pasal 1 Angka 14 dan perubahannya yang tercantum dalam PerPres Nomor 70 Tahun 2012 tentang Pedoman Pelaksanaan Pengadaan Barang/Jasa Pemerintah, disebutkan bahwa definisi barang adalah setiap benda baik berwujud maupun tidak berwujud, bergerak maupun tidak bergerak, yang dapat diperdagangkan, dipakai, dipergunakan atau dimanfaatkan oleh pengguna barang.

Penggolongan barang secara garis besar adalah sebagai berikut:

1. Barang Berwujud, yaitu barang yang mempunyai wujud tertentu. Contoh : buku, bolpoint, sepatu tas dan sebagainya.
2. Barang tak berwujud, yaitu barang yang tidak mempunyai wujud tertentu. Contoh : gas, air dan sebagainya.

3. Barang bergerak sendiri, yaitu barang yang dapat berpindah tempat dari satu tempat ke tempat yang lain, baik dengan cara berpindah sendiri atau dipindahkan. Contoh : kapal, perahu dan sebagainya.
4. Barang tidak bergerak, yaitu barang yang tidak dapat berpindah dari satu tempat ketempat yang lain. Contoh : tanah, pohon yang masih menancap di akarnya dan sebagainya.

Penggolongan barang berdasarkan penggunaannya, yaitu :

- 1) Barang habis pakai adalah barang/benda kantor yang penggunaannya hanya satu/ beberapa kali pakai atau tidak tahan lama. Contoh : kertas, tinta, karbon, klip, pensil dan pulpen.
- 2) Barang yang tidak habis pakai adalah barang/ benda kantor yang penggunaannya tahan lama. Contoh : stapler, perforator, cutter, dan gunting.

Sedangkan menurut Peraturan Menteri Keuangan No. 113/PMK.01/2016 tentang Pedoman Penatausahaan Persediaan di Lingkungan Departemen Keuangan, Barang Habis Pakai adalah barang yang merupakan bagian dari kekayaan negara yang menurut sifatnya dipakai habis untuk keperluan dinas atau jangka waktu pemakaiannya kurang dari satu tahun. Barang habis pakai yaitu barang berwujud, yang biasanya habis dikonsumsi dalam normal kurang dari satu tahun.

Pengelolaan stok barang habis pakai merupakan salah satu segi manajemen perusahaan sebagai pergerakan dari barang tersebut. Tujuan pengelolaan tersebut adalah agar bahan yang diperlukan selalu tersedia setiap dibutuhkan, dalam jumlah yang cukup dan mutu yang terjamin, untuk mendukung pelayanan yang bermutu (*good quality care*).

2.4 Web Mobile

Menurut Brian Fling (2016) *Web Mobile* merupakan situs *web* yang dirancang khusus untuk perangkat mobile. Situs *web mobile* sering memiliki desain yang sederhana dan biasanya bersifat memberikan informasi (Abdurahman 2016). Menurut Devi (2015) *Web Mobile* juga satu-satunya platform yang tersedia dan

mampu berjalan pada semua perangkat *mobile*, dan perancangan menggunakan standar dan protokol yang sama dengan *desktop web*. Untuk dapat mendesain aplikasi *web* untuk *mobile* harus diperhatikan betul bahwa karakteristik *web* untuk *mobile* berbeda dengan desktop (Indriasari, Adi, and Sidhi 2011).

Indonesian Journal on Networking and Security (2016:49), *web mobile* adalah lanjutan dari aplikasi tradisional menggunakan *smartphone*, *tablet*, atau komputer, desainnya akan beradaptasi dengan perangkat yang digunakan tersebut, dengan desain yang fleksibel ini memaksimalkan kemampuan *web browser* yang digunakan di perangkat *mobile*.

Kesimpulannya, *web mobile* adalah kumpulan halaman website yang dapat diakses pada perangkat *mobile* seperti *smartphone*, *tablet*, atau komputer, desainnya akan beradaptasi dengan perangkat yang digunakan tersebut.

2.5 Hypertext Preprocessor (PHP)

PHP (*Hypertext Preprocessor*) adalah suatu bahasa pemrograman yang digunakan untuk menerjemahkan baris kode program menjadi kode mesin yang dapat dimengerti oleh komputer yang berbasis *server-side* yang dapat ditambahkan ke dalam HTML (Supono dan Putratama, 2016). PHP sering dipakai para *programmer* untuk membuat situs *web* yang bersifat dinamis karena gratis (*open source*) dan berguna dalam merancang aplikasi *web*. Sedangkan, menurut Solichin (2016) mengemukakan bahwa “PHP merupakan salah satu bahasa pemrograman berbasis *web* yang ditulis oleh dan untuk pengembang *web*”.

Kumpulan kutipan diatas menerangkan bahwa *hypertext preprocessor* (PHP) merupakan bahasa pemrograman untuk membuat/mengembangkan aplikasi berbasis *web* dan bersifat *open soucre* dan ditanamkan ke dalam *script* HTML.

2.6 Basis Data (Database)

Himpunan kelompok data yang saling berhubungan yang diorganisasikan sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah merupakan definisi dari Basis Data. Secara lebih lengkap pemanfaatan basis data

dilakukan untuk memenuhi tujuan Kecepatan, kemudahan, efisiensi ruang penyimpanan, keakuratan, ketersediaan, kelengkapan, keamanan dan pemakaian bersama (Hidayatullah dan Kawistara, 2017).

Sedangkan menurut Rosa dan Shalahuddin (2015), “Basis Data adalah media untuk menyimpan data agar dapat diakses dengan mudah dan cepat”. Jadi dapat disimpulkan basis data adalah kumpulan data yang saling berhubungan agar dapat diakses oleh software tertentu.

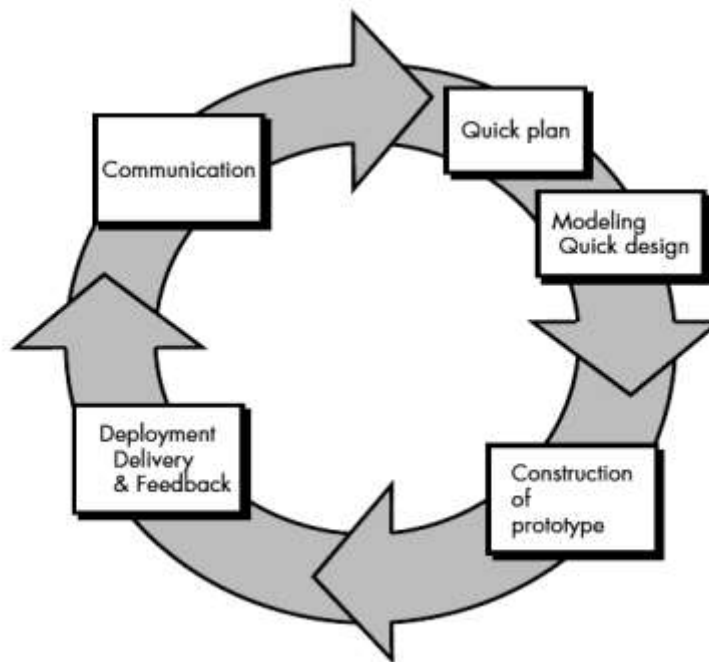
2.7 Metode Pengembangan Perangkat Lunak

Metode pengembangan yang digunakan dalam penelitian ini adalah *Prototype*. Menurut Pressman (2012:50), dalam melakukan perancangan sistem yang akan dikembangkan dapat menggunakan metode *prototype*. Metode ini cocok digunakan untuk mengembangkan sebuah perangkat yang akan dikembangkan kembali. Metode ini dimulai dengan pengumpulan kebutuhan pengguna, dalam hal ini pengguna dari perangkat yang dikembangkan adalah bagian Aset Fakultas Pertanian Universitas Lampung. Kemudian membuat sebuah rancangan kilat yang selanjutnya akan dievaluasi kembali sebelum diproduksi secara benar.

Prototype bukanlah merupakan sesuatu yang lengkap, tetapi sesuatu yang harus dievaluasi dan dimodifikasi kembali. Segala perubahan dapat terjadi pada saat *prototype* dibuat untuk memenuhi kebutuhan pengguna dan pada saat yang sama memungkinkan pengembang untuk lebih memahami kebutuhan pengguna secara lebih baik. Berikut adalah tahapan dalam metode *prototype*:

- 1) Komunikasi dan pengumpulan data awal, yaitu analisis terhadap kebutuhan pengguna (dalam hal ini adalah bagian Aset Fakultas Pertanian Universitas Lampung).
- 2) *Quick design* (desain cepat), yaitu pembuatan desain secara umum untuk selanjutnya dikembangkan kembali.
- 3) Pembentukan *prototype*, yaitu pembuatan perangkat *prototype* termasuk pengujian dan penyempurnaan.
- 4) Evaluasi terhadap *prototype*, yaitu mengevaluasi *prototype* dan memperhalus analisis terhadap kebutuhan pengguna.

- 5) Perbaiki *prototype*, yaitu pembuatan tipe yang sebenarnya berdasarkan hasil dari evaluasi *prototype*.
- 6) Produksi akhir, yaitu memproduksi perangkat secara benar sehingga dapat digunakan oleh pengguna.



Gambar 2.1 Model *Prototype* Paradigma

2.8 (*Unified Modeling Language*) UML

Menurut Nugroho (2015a), “*Unified Modeling Language* (UML) adalah alat bantu analisis serta perancangan perangkat lunak berbasis objek”. Sedangkan, menurut Rosa dan Shalahuddin (2014), berpendapat bahwa UML (*Unified Modeling Language*) adalah “Salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisa & desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek”. *Unified Modeling Language* (UML) adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. *Unified Modeling Language* (UML) adalah sebuah bahasa yang berdasarkan grafik atau



gambar untuk memvisualisasi, menspesifikasikan, membangun, dan pendokumentasian dari sebuah sistem pengembangan software berbasis OO (*Object-Oriented*). UML (*Unified Modeling Language*) memiliki diagram-diagram yang digunakan dalam pembuatan aplikasi berorientasi objek, diantaranya (Rosa dan Shalahuddin, 2014):

Dari pengertian di atas dapat disimpulkan bahwa UML merupakan suatu standarisasi pemodelan atau bahasa standar dalam merencanakan, memvisualisasikan, mendokumentasi dan merancang sebuah sistem perangkat lunak. Adapun jenis diagram UML antara lain:




2.8.1 Use Case Diagram

UsecaseDiagram menggambarkan deksripsi peringkat tertinggi dari sebuah fungsi tertentu dalam suatu sistem berupa komponen, kejadian atau kelas. Use case diartikan sebagai urutan langkah-langkah yang secara tindakan saling terkait (skenario), baik terotomatisasi maupun secara manual, untuk tujuan melengkapi satu tugas bisnis tunggal. Diagram use case bersifat statis, diagram ini memperlihatkan himpunan use case dan aktor-aktor (suatu jenis khusus dari kelas). Diagram ini terutama sangat penting untuk mengorganisasi dan memodelkan perilaku suatu sistem yang dibutuhkan serta diharapkan pengguna (Afrida dan Wibowo. 2018). Penjelasan simbol dapat dilihat pada tabel 2.1.

Tabel 2.1 Use Case Diagram

No	Gambar	Nama	Keterangan
1		Actor	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan use case.
2		Include	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (independent) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (independent).



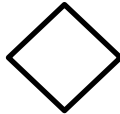
Tabel 2.1 Use Case Diagram (Lanjutan)

3		Association	Apa yang menghubungkan antara objek satu dengan objek lainnya.
4		System	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
5		Use Case	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.





2.8.2 Activity Diagram

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Perlu diperhatikan bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. Berikut adalah simbol-simbol yang ada pada diagram aktivitas (Rosa dan Shalahuddin, 2014).

Tabel 2.2 Simbol-Simbol Activity Diagram

Simbol	Deskripsi
Status Awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal
Aktifitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
Percabangan / <i>Decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu

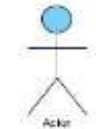


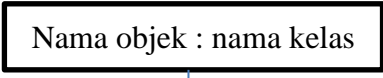



Tabel 2.2 Simbol-Simbol *Activity Diagram* (Lanjutan)

Transition / Asosiasi 	Transition untuk menghubungkan aktivitas selanjutnya setelah aktivitas sebelumnya
Penggabungan / <i>Join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
Status Akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
Swimlane 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi


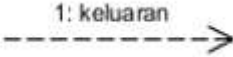

2.8.3 *Sequence Diagram*

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirim dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada *use case*. Banyaknya diagram sekuen yang harus digambar adalah minimal sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah didefinisikan interaksi jalannya pesan sudah dicakup pada diagram sekuen sehingga semakin banyak *usecase* yang didefinisikan maka diagram sekuen yang harus dibuat juga semakin banyak (Rosa dan Shalahuddin, 2019).

Tabel 2.3 Simbol-Simbol *Sequence Diagram*

Simbol	Deskripsi
Actor  Atau 	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi itu sendiri, jadi walaupun simbol dari aktor adalah orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda diawal frase nama aktor
Garis Hidup / <i>Lifeline</i> 	Menyatakan kehidupan suatu objek
Objek 	Menyatakan objek yang berinteraksi pesan
Waktu aktif 	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan didalamnya.
Pesan tipe <i>create</i> 	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat
Pesan tipe <i>call</i> 	Menyatakan suatu objek memanggil operasi / metode yang ada pada objek lain atau dirinya sendiri. Arah panah mengarah pada objek yang memiliki operasi / metode, karena ini memanggil operasi / metode maka operasi / metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi

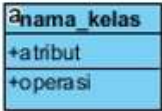

Tabel 2.3 Simbol-Simbol *Sequence Diagram* (Lanjutan)

Simbol	Deskripsi
Pesan tipe <i>send</i> 	Menyatakan bahwa suatu objek mengirimkan data / masukan / informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim
Pesan tipe <i>return</i> 	Menyatakan suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian
Pesan tipe <i>destroy</i> 	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada <i>create</i> maka ada <i>destroy</i>





2.8.4 Class Diagram

Class Diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi. Berikut adalah simbol-simbol yang ada pada diagram Kelas (Rosa dan Shalahuddin, 2013:146)

Tabel 2.4 Simbol-Simbol *Class Diagram*

Simbol	Deskripsi
Kelas 	Kelas pada struktur sistem
Antarmuka / <i>Interface</i>  Nama_Interface	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek

Tabel 2.4 Simbol-Simbol *Class Diagram* (Lanjutan)

Simbol	Deskripsi
Asosiasi / <i>assosiation</i> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
Asosiasi berarah / <i>Directed Association</i> 	Relasi antarkelas dengan makna kelas satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)
Agregasi / <i>Aggregation</i> 	Relas antar kelas dengan makna semuabagian (<i>whole-part</i>)

2.9 Black Box Testing

Black box testing adalah tipe *testing* yang memerlukan perangkat lunak yang tidak diketahui kinerja internalnya. Sehingga para tester memandang perangkat lunak seperti layaknya sebuah “kotak hitam” yang tidak penting dilihat isinya, tapi dikenal proses testing di bagian luar. Metode ujicoba *blackbox* memfokuskan pada keperluan fungsional dari *software*. Karena itu ujicoba *blackbox* memungkinkan pengembang *software* untuk membuat himpunan kondisi input yang akan melatih seluruh syarat-syarat fungsional suatu program (Wijayanto, 2014).

Pengujian kotak hitam dilakukan dengan membuat kasus uji yang bersifat mencoba semua fungsi dengan memakai perangkat lunak apakah sesuai dengan spesifikasi yang dibutuhkan.

2.10 Penelitian Terkait

Pada penelitian ini terdapat penelitian sejenis yang pernah dilakukan sebelumnya. Penulis akan membangun sebuah aplikasi manajemen pengelolaan barang habis pakai berbasis *web mobile*. Berikut ini merupakan beberapa penelitian yang pernah dilakukan sebelumnya.

Tabel 2.5 Penelitian Terkait

No	Nama	Judul	Keterangan	Sumber
1.	Ruli Supriati, Astri Wulan Sari	Aplikasi Sistem Pendataan Barang Habis Pakai Guna Meningkatkan Kualitas Stok Barang pada PT. Angkasa Pura II Tangerang	Pada penelitian ini membahas tentang pendataan dan pengelolaan barang habis pakai dan pelaporan barang kepada pimpinan	Jurnal SIMIKA Vol. 2 No. 2 Tahun 2019
2.	Rizki Putra Wicaksono1, Anteng Widodo	Sistem Informasi Persediaan Barang Berbasis Web pada CV. Patriot Kencana Medika Kudus	Penelitian ini membahas tentang pendataan barang secara terpusat dan pengecekan barang kosong yang dapat diakses melalui web	Jurnal Sistem Informasi & Manajemen Basis Data (SIMADA) Vol. 3 No. 1 Maret 2020

