

BAB II

LANDASAN TEORI

2.1. Tinjauan Pustaka

2.1.1. Pengertian Sistem

Menurut Sutabri (2012:6) “Sistem pada dasarnya adalah sekelompok unsur yang erat hubungannya satu dengan yang lain, yang berfungsi bersama-sama untuk mencapai tujuan tertentu”.

Menurut Yakub (2012:1) “Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang berhubungan, terkumpul bersama-sama untuk melakukan suatu kegiatan atau tujuan tertentu”.

2.1.2. Pengertian Informasi

Menurut Sutabri (2012:22) Informasi adalah data yang telah diklasifikasikan atau diolah atau diinterpretasikan untuk digunakan dalam proses pengambilan keputusan. Sistem pengolahan informasi akan mengolah data menjadi informasi atau mengolah data dari bentuk tak berguna menjadi berguna bagi yang menerimanya.

Menurut Sutarman (2012:14) “Informasi adalah sekumpulan fakta (data) yang diorganisasikan dengan cara tertentu sehingga mereka mempunyai arti bagi penerima”.

2.1.3. Pengertian Sistem Informasi

Menurut Sutabri (2012:38) Sistem Informasi adalah suatu sistem di dalam organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan laporan-laporan yang diperlukan oleh pihak luar tertentu.

Menurut Sutarman (2012:14) “Sistem Informasi adalah sistem yang dapat didefinisikan dengan mengumpulkan, memproses, menyimpan, menganalisis, menyebarkan informasi untuk tujuan tertentu. Seperti sistem lainnya, sebuah sistem informasi terdiri atas input (data, instruksi) dan output (laporan, kalkulasi)”.

2.1.4. Pengertian Layanan

Menurut kamus Besar Bahasa Indonesia “Layanan berasal dari kata layan yang kata kerjanya adalah melayani yang mempunyai arti membantu menyiapkan (mengurus) apa-apa yang diperlukan seseorang”.

2.1.5. Pengertian Penatu

Menurut kamus Besar Bahasa Indonesia “Penatu adalah usaha atau orang yang bergerak dibidang pencucian (penyetrikaan) pakaian”.

2.1.6. Pengertian Transaksi

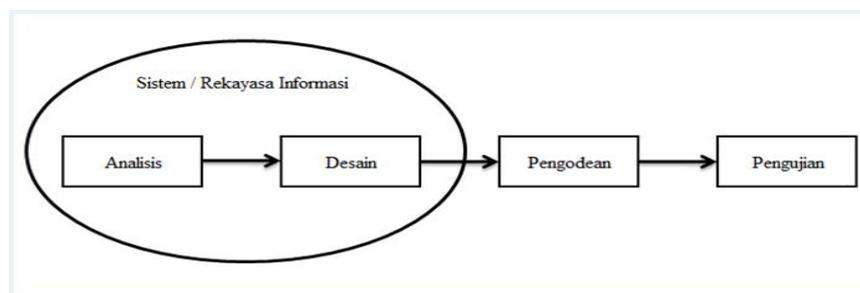
Menurut Skousen (2007:71) mengemukakan bahwa “Transaksi adalah pertukaran barang dan jasa antara (baik individu, perusahaan-perusahaan dan organisasi lain) kejadian lain yang mempengaruhi ekonomi atas bisnis”.

Menurut Bastian (2007:27) “Transaksi adalah pertemuan antara dua belah pihak (penjual dan pembeli) yang saling menguntungkan dengan adanya data/bukti/dokumen pendukung yang dimasukkan kedalam jurnal setelah melalui pencatatan”.

2.1.7. Konsep Dasar Model Pengembangan Sistem

Model pengembangan sistem adalah model yang digunakan untuk mengembangkan suatu sistem informasi, yaitu suatu urutan proses standar yang diikuti untuk melaksanakan seluruh langkah yang diperlukan untuk menganalisis, merancang, mengimplementasi dan memelihara sistem. Model yang digunakan oleh peneliti adalah Model SDLC (*Sistem Development Life Cycle*) atau sering disebut sebagai pendekatan air terjun (*Waterfall*).

Menurut Sukamto dan Shalahuddin (2013:28) Model SDLC air terjun (*waterfall*) sering juga disebut model sekuensial linier (*Sequential linear*) atau alur hidup klasik (*classic life cycle*). Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengodean, pengujian dan tahap pendukung atau (*support*).



Sumber : Sukamto dan Shalahuddin (2013:28)

Gambar II.1.

Ilustrasi Model *Waterfall*

1. 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 didokumentasikan.

2. Desain

Desain perangkat lunak adalah proses multi langkah yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka 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.

3. Pembuatan kode program (*Code Generation*)

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

4. Pengujian (*Testing*)

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.

5. Pendukung (*Support*)

Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirimkan ke *user*. Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru.

2.1.8. Konsep Dasar Pemrograman

Metodologi berorientasi objek adalah suatu strategi pembangunan perangkat lunak yang mengorganisasikan perangkat lunak sebagai kumpulan objek yang berisi data dan operasi yang diberlakukan terhadapnya. Metodologi berorientasi objek merupakan suatu cara bagaimana sistem perangkat lunak dibangun melalui pendekatan objek secara sistematis. Metode berorientasi objek didasarkan pada penerapan prinsip-prinsip pengelolaan kompleksitas. Metode berorientasi objek meliputi rangkaian aktivitas analisis berorientasi objek, perancangan berorientasi objek, pemrograman berorientasi objek dan pengujian berorientasi objek.

Menurut Sukamto dan Shalahuddin (2015:103); pendekatan berorientasi objek merupakan suatu teknik atau cara pendekatan dalam melihat permasalahan dan sistem (sistem perangkat lunak, sistem informasi, atau sistem lainnya). Pendekatan berorientasi objek akan memandang sistem yang akan dikembangkan sebagai suatu kumpulan objek yang berkorespondensi dengan objek-objek dunia nyata. Ada banyak cara untuk mengabstraksikan dan memodelkan objek. Objek tersebut, mulai dan abstraksi objek, kelas, hubungan antarkelas sampai abstraksi sistem. Saat mengabstraksikan dan memodelkan objek, data dan proses-proses yang dipunyai oleh objek akan dienkapsulasi (dibungkus) menjadi satu kesatuan.

Berikut ini adalah beberapa konsep dasar yang harus dipahami tentang metodologi berorientasi objek :

1. Kelas (*Class*)

Kelas adalah kumpulan objek-objek dengan karakteristik yang sama. Kelas merupakan definisi statik dan himpunan objek yang sama yang mungkin lahir atau diciptakan dan kelas tersebut. Sebuah kelas akan mempunyai sifat (atribut). Kelakuan (operasi/metode), hubungan(*relationship*) dan arti.

Suatu kelas dapat diturunkan dan kelas lain, dimana atribut dan kelas semula dapat diwariskan ke kelas yang baru.

2. Objek (*Object*)

Objek adalah abstraksi dan sesuatu yang mewakili dunia nyata seperti benda, manusia, satuan organisasi, tempat, kejadian, struktur, sataus, atau hal-hal lain yang bersifat abstrak. Objek merupakan suatu entitas yang mampu menyimpan informasi (status) dan mempunyai operasi (kelakuan) yang dapat diterapkan atau dapat berpengaruh pada status objeknya. Objek mempunyai siklus hidup yaitu diciptakan, dimanipulasi, dan dihancurkan.

3. Metode (*Method*)

Operasi atau metode atau method pada sebuah kelas hampir sama dengan fungsi atau prosedur pada metodologi struktural.sebuah kelas boleh memiliki lebih dari satu metode atau operasi. Metode atau operasi yang berfungsi untuk memanipulasi objek itu sendiri. Operasi atau metode merupakan fungsi atau transformasi yang dapat dilakukan terhadap objek atau dilakukan oleh objek

4. Atribut (*Attribute*)

Atribut dari sebuah kelas adalah variabel global yang dimiliki sebuah kelas. Atribut dapat berupa nilai atau elemen-elemen data yang dimiliki oleh objek dalam kelas objek. Atribut dipunyai secara individual oleh sebuah objek, misalnya berat, jenis, nama, dan sebagainya. Atribut sebaiknya bersifat privat untuk menjaga konsep enkapsulasi.

5. Abstraksi (*Abstraction*)

Prinsip untuk merepresentasikan dunia nyata yang kompleks menjadi satu bentuk model yang sederhana dengan mengabaikan aspek-aspek lain yang tidak sesuai dengan permasalahan.

6. Enkapsulasi (*Encapsulation*)

Pembungkusan atribut data dan layanan (operasi-operasi) yang dipunyai objek untuk menyembunyikan implementasi dan objek sehingga objek lain tidak mengetahui cara kerjanya.

7. Pewarisan (*Inheritance*)

Mekanisme yang memungkinkan satu objek mewarisi sebagian atau seluruh definisi dan objek lain sebagai bagian dan dirinya.

8. Antarmuka (*Interface*)

Antarmuka atau *interface* sangat mirip dengan kelas, tapi tanpa atribut kelas dan memiliki metode yang dideklarasikan tanpa isi. Deklarasi metode pada sebuah *interface* dapat diimplementasikan oleh kelas lain. Sebuah kelas dapat mengimplementasikan lebih dari satu antarmuka dimana kelas ini akan mendeklarasikan metode pada antarmuka yang dibutuhkan oleh kelas itu sekaligus mendefinisikan isinya pada kode program kelas itu. Metode pada antarmuka yang diimplementasikan pada suatu kelas harus sama persis biasanya digunakan agar kelas yang lain tidak mengakses langsung ke suatu kelas, mengakses antarmukanya.

9. *Reusability*

Pemanfaatan kembali objek yang sudah didefinisikan untuk suatu permasalahan lainnya yang melibatkan objek tersebut. Misalkan dalam

sebuah aplikasi peminjaman buku diperlukan kelas Anggota, maka ketika membuat penyewaan VCD, kelas anggota ini bisa digunakan kembali dengan sedikit perubahan untuk aplikasi penyewaan VCD tanpa harus membuat dari awal kembali.

10. *Generalisasi dan spesialisasi*

Menunjukkan hubungan antar kelas dan objek yang umum dengan kelas dan objek yang khusus. Misalnya kelas yang lebih umum (*generalisasi*) adalah kendaraan darat dan kelas khususnya (*spesialisasi*) adalah mobil, motor, dan kereta.

11. Komunikasi antar objek

Komunikasi antar-objek dilakukan lewat pesan (*message*) yang dikirim dan satu objek ke objek lainnya.

12. Polimorfisme (*Polymorphism*)

Kemampuan suatu objek untuk digunakan di banyak tujuan yang berbeda dengan nama yang sama sehingga menghemat baris program.

13. *Package*

Package adalah sebuah kontainer atau kemasan yang dapat digunakan untuk mengelompokkan kelas-kelas sehingga memungkinkan beberapa kelas yang bernama sama disimpan dalam *package* yang berbeda.

2.2. Peralatan Pendukung Sistem (*Tools System*)

2.2.1. *Unified Modeling Language (UML)*

Menurut Prabowo (2011:10) Beberapa literature menyebutkan bahwa UML menyediakan Sembilan jenis diagram, yang lain menyebutkan delapan karena ada beberapa diagram yang digabung, misalnya diagram

komunikasi, diagram urutan dan diagram pewaktuan digabung menjadi diagram interaksi.

Namun demikian model-model itu dapat dikelompokkan berdasarkan sifatnya yaitu statis dan dinamis. Jenis diagram itu antara lain:

1. *Class Diagram*

Diagram ini memperlihatkan himpunan kelas-kelas, antarmuka-antarmuka, kolaborasi-kolaborasi, serta relasi-relasi. Diagram ini umum dijumpai pada pemodelan sistem berorientasi objek. Meskipun bersifat statis diagram kelas juga sering memuat kelas-kelas aktif.

2. *Package Diagram*

Diagram ini memperlihatkan kumpulan kelas-kelas, merupakan bagian dari diagram komponen.

3. *Use Case Diagram*

Diagram ini memperlihatkan himpunan-himpunan use case dan actor-aktor (suatu jenis khusus dari kelas). Diagram ini terutama sangat penting untuk mengorganisasi dan memodelkan perilaku suatu sistem yang dibutuhkan serta diharapkan pengguna.

4. Diagram Interaksi dan *Sequence* (urutan)

Diagram ini adalah interaksi yang menekankan pada pengiriman pesan dalam suatu waktu tertentu.

5. *Communication Diagram*

Diagram ini sebagai pengganti diagram kolaborasi UML 1.4 yang menekankan organisasi structural dari objek-objek yang menerima serta mengirim pesan.

6. *Statechart* Diagram

Diagram status memperlihatkan keadaan-keadaan pada sistem, memuat status (state), transisi, kejadian serta aktivitas.

7. *Activity* Diagram

Diagram aktivitas adalah tipe khusus dari diagram status yang memperlihatkan aliran dari suatu aktivitas ke aktivitas lainnya dalam suatu sistem. Diagram ini terutama penting dalam pemodelan fungsi-fungsi suatu sistem dan memberi tekanan pada aliran kendali antar objek.

8. *Component* Diagram

Diagram komponen ini memperlihatkan organisasi serta keberuntungan sistem/perangkat lunak pada komponen-komponen yang telah ada sebelumnya.

9. *Deployment* Diagram

Diagram ini memperlihatkan konfigurasi saat aplikasi dijalankan (runtime). Memuat simpul-simpul beserta komponen-komponen yang ada didalamnya.

2.2.2. *Database Management System (DBMS)*

Menurut Sukanto dan Shalahuddin (2015:44) ”DBMS (*Database Management System*) atau dalam bahasa Indonesia sering disebut sebagai sistem manajemen basis data adalah suatu sistem aplikasi yang digunakan untuk menyimpan mengelola, dan menampilkan data”.

2.2.3. XAMPP

Menurut Wicaksono (2008:7) menjelaskan bahwa “XAMPP adalah sebuah software yang berfungsi untuk menjalankan website berbasis PHP dan menggunakan pengolahan data MYSQL di komputer lokal”.

2.2.4. Processor Hypertext Protocol (PHP)

Menurut Wahyono (2009:35), “PHP(dulu: *personal home page*, sekarang PHP: *Hypertext preprocessor*) merupakan *script* untuk membuat suatu aplikasi yang akan terintegrasi ke dalam halaman HTML, sehingga suatu halaman web tidak lagi bersifat statis, namun menjadi bersifat dinamis”.

Menurut Sibero (2011:49) PHP adalah pemrograman interpreter yaitu proses penerjemahan baris kode mesin yang dimengerti computer secara langsung pada saat baris kode dijalankan atau sering disebut suatu bahasa dengan hak cipta terbuka atau yang juga dikenal dengan istilah *Open Source* yaitu pengguna dapat mengembangkan kode-kode fungsi PHP sesuai dengan kebutuhannya.

2.2.5. Adobe Dreamweaver CS3

Menurut Wahana Komputer (2010:2) “*Adobe Dreamweaver* merupakan salah satu program aplikasi yang digunakan untuk membangun sebuah *website*, baik secara grafis maupun dengan menuliskan kode sumber secara langsung”.

2.2.6. MySQL

Menurut Wahana Komputer (2010:11) “MySQL merupakan suatu aplikasi yang digunakan untuk manajemen suatu data dan banyak digunakan khalayak diseluruh dunia”.

2.2.7. Entity Relationship Diagram (ERD)

Menurut Sukamto dan Shalahuddin (2015:50) Pemodelan awal basis data yang paling banyak digunakan adalah menggunakan *Entity Relationship Diagram (ERD)*. ERD dikembangkan berdasarkan teori himpunan dalam bidang matematika. ERD digunakan untuk pemodelan basis data relasional. Sehingga jika penyimpanan basis data menggunakan OODBMS maka perancangan basis data tidak perlu menggunakan ERD. ERD memiliki beberapa aliran notasi seperti notasi Chen (dikembangkan oleh Peter Chen), Barker (dikembangkan oleh Richard Barker, Ian Palmer, Harry Ellis), notasi Crow's Foot, dan beberapa notasi lain. Namun yang banyak digunakan adalah notasi dari Chen.

2.2.8. Black Box Testing

Menurut Rizky (2011:264), berpendapat bahwa "*Black Box Testing* adalah tipe testing yang meperlakukan perangkat lunak yang tidak diketahui kinerja internalnya. Sehingga para tester memandang perangkat lunak seperti layaknya sebuah "kotak hitam" yang tidak penting dilihat isinya, tapi cukup dikenai proses testing di bagian luar".

2.3. Penelitian Terkait

Menurut Rusdina (2016:230) dijelaskan bahwa Arba Smile Laundry Kiloan Plus adalah perusahaan yang bergerak di bidang jasa pencucian. Dalam kegiatannya Arba Smile Laundry Kiloan Plus sehari-hari tidak terlepas dari transaksi penjualan yang dilakukan. Tetunya, untuk mengelola data-data tersebut akan memerlukan energy yang cukup banyak disertai waktu yang lumayan lama.

Menurut Khoirunnissa, dkk (2016:93) dijelaskan bahwa aplikasi strategi bisnis ini dapat dimanfaatkan kepada para pelaku bisnis di bidang jasa seperti halnya jasa *laundry*. Pada umumnya, perhitungan laba rugi, laporan keuangan dan pendataan keluar masuk pakaian masih dilakukan dengan menggunakan kertas seperti menuliskan laporan keuangan pada sebuah buku serta menghitungnya satu per satu secara manual sehingga kemungkinan terjadi kesalahan manusia sangat besar.