

BAB II

LANDASAN TEORI

2.1 Penelitian Terdahulu

Penelitian terdahulu digunakan untuk memberi suatu perbandingan referensi proyek yang telah dikerjakan, terdapat 4 contoh referensi dari penelitian terdahulu, yaitu

- a) Menurut Perdani (2012). Dalam tugas akhirnya mengungkapkan bahwa *sistem informasi* pengadaan dan penjualan suku cadang Yamaha Kebon Agung Motor, meliputi aktifitas yaitu proses pemesanan suku cadang, proses penerimaan suku cadang, proses penerimaan retur suku cadang, proses penjualan suku cadang dan proses pembuatan laporan. Temuan yang didapat dari penelitian ini yaitu sistem informasi tersebut masih belum mempunyai sistem katalog yang ditujukan untuk pembeli.
- b) Dalam sumber yang lain dikatakan bahwa, rancang bangun sistem informasi penjualan pada swalayan Koperasi Setia Bhakti Wanita Surabaya, meliputi aktifitas yaitu laporan berupa arus perputaran barang, laporan hasil penjualan pada swalayan, laporan perbandingan penjualan barang dan laporan kontribusi anggota swalayan (Putri, 2008). Temuan yang didapat dari penelitian ini yaitu kurangnya laporan keuntungan peritem yang ditentukan perhari, perbulan, dan pertahun, dengan menggunakan media *visualisasi* untuk membantu manajemen membaca laporan keuntungan dan *sistem informasi* tersebut masih belum mempunyai sistem katalog yang ditujukan untuk pembeli.

- c) Menurut Kuncono (2011), Tugas Akhir dengan judul rancang bangun sistem informasi penjualan pada CV. Konveksi Jaya, berdasarkan penelitian tugas akhir ini, meliputi aktivitas yaitu laporan presentase penjualan barang yang laku berdasarkan merk, jenis, dan ukuran pada periode sebelumnya, laporan penjualan berdasarkan jenis customer yang membeli, dan barang apa saja yang biasa dibeli. Temuan yang didapat dari tugas akhir ini yaitu tidak memiliki *sistem katalog berbasis komputerisasi* serta *visualisasi* untuk membantu mempermudah melihat keuntungan barang.
- d) Dalam sumber yang lain dikatakan bahwa rancang bangun *aplikasi client server transaksi* penjualan dan katalog berbasis multimedia di *Ice Cream House*, (Adriana, 2008). Berdasarkan penelitian tugas akhir ini, permasalahan yang terjadi di *Ice Cream House* adalah pengolahan *data transaksi* penjualan yang belum bisa menghasilkan laporan *transaksi* yang akurat. Selain itu para pembeli masih sulit mendapatkan informasi yang jelas mengenai *Ice Cream* yang dijual disana dari buku menu yang tersedia. Temuan dari tugas akhir ini yaitu sistem katalog yang dimiliki belum dapat dimengerti oleh pembeli tidak memiliki laporan-laporan pendapatan peritem dengan media *visualisasi* serta sistem *katalog* yang dapat memberikan *informasi* menu *Ice Cream* kepada pelanggan.

2.2 Administrasi

Menurut Silalahi (2006), administrasi secara sempit didefinisikan sebagai penyusunan dan pencatatan data dan informasi secara sistematis baik internal maupun eksternal dengan maksud menyediakan keterangan serta memudahkan

untuk memperoleh kembali baik sebagian maupun menyeluruh. Pengertian administrasi secara sempit ini lebih dikenal dengan istilah tata usaha.

2.3 Selling

Menurut Kartajaya, (2006). Selling cara menjalin relasi dengan pelanggan. Relasi ini tidak terbentuk begitu saja, tetapi dapat dibangun dengan beberapa langkah sistematis. Tahap pertama dengan meningkatkan *awareness* pelanggan terhadap merek. Tahap kedua, jadikan merek sebagai identitas bagi pelanggan. Tahap ketiga pelanggan memiliki hubungan jangka panjang dengan merek. Tahap keempat yaitu pelanggan menjadi bagian komunitas merek.

2.4 Penjualan

Definisi penjualan menurut Mulyadi (2008), penjualan merupakan kegiatan yang dilakukan oleh penjual dalam menjual barang atau jasa dengan harapan akan memperoleh laba dari adanya transaksi-transaksi tersebut dan penjualan dapat diartikan sebagai pengalihan atau pemindahan hak kepemilikan atas barang atau jasa dari pihak penjual ke pembeli. Serta penjualan adalah suatu kegiatan yang terdiri dari transaksi penjualan barang dan jasa, secara tunai maupun kredit. Penjualan dikelompokkan menjadi dua, yaitu penjualan tunai dan penjualan kredit. Penjualan tunai adalah penjualan yang pembayarannya diterima sekaligus. Penjualan kredit adalah penjualan yang pembayarannya tidak diterima sekaligus.

2.5 Retur Penjualan

Menurut Mulyadi (2008), transaksi retur penjualan terjadi ketika jika perusahaan menerima pengembalian produk dari pelanggan. Pengembalian produk oleh pelanggan harus diotorisasi oleh fungsi penjualan dan diterima oleh fungsi penerimaan.

2.6 Aplikasi

Menurut Jogiyanto (2005), aplikasi adalah penggunaan dalam suatu komputer, instruksi (*instruction*) atau pernyataan (*statement*) yang disusun sedemikian rupa sehingga komputer dapat memproses *input* menjadi *output*.

Dari defenisi di atas dapat disimpulkan bahwa aplikasi adalah suatu program komputer yang dibuat untuk mengerjakan dan melaksanakan tugas khusus dari pengguna. Aplikasi merupakan rangkaian kegiatan atau perintah untuk dieksekusi oleh komputer.

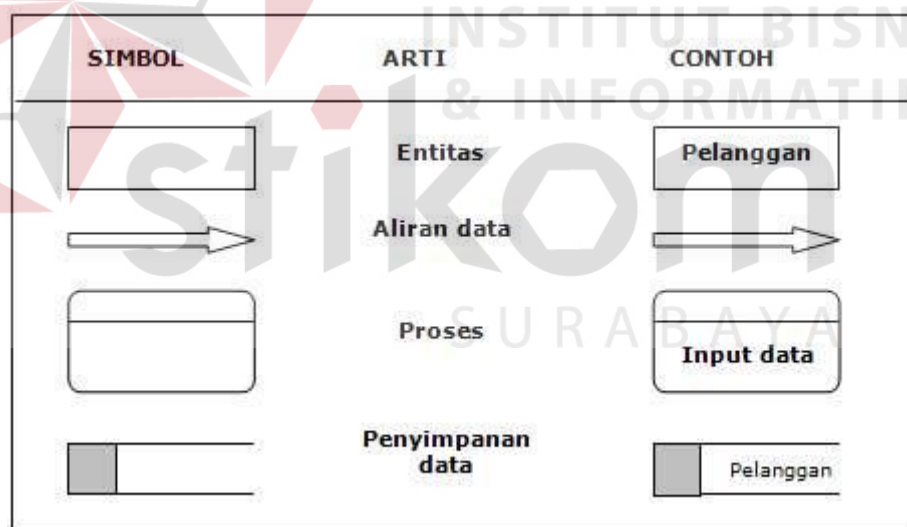
2.7 Data Flow Diagram

Menurut Kenneth dan Kendall (2003), *Data Flow Diagram* (DFD) adalah suatu diagram yang menggunakan notasi-notasi untuk menggambarkan arus dari data sistem, yang penggunaannya sangat membantu untuk memahami sistem secara logika, terstruktur dan jelas.

DFD merupakan alat bantu dalam menggambarkan atau menjelaskan proses kerja suatu sistem proses dan keluaran sistem yang berhubungan dengan masukan, proses dan keluaran serta mempresentasikan dan prosedur mendetail dalam sistem yang masukan, proses dan keluaran serta merepresentasikan dan

menganalisis prosedur-prosedur mendetail dalam sistem yang alir data juga mampu mengkonseptualisasikan bagaimana data-data berpindah lebih besar. Diagram mengkonseptualisasikan bagaimana data didalam organisasi. Pada aliran data menekankan logika yang didalam organisasi. Pada aliran data menekankan logika yang mendasari sistem sebuah sistem kontekstual data flow diagram pertama kali muncul adalah interaksi antara sistem dan entitas luar. DFD didesain untuk menunjukkan sebuah sistem yang terbagi-bagi menjadi suatu bagian sub-sistem yang lebih kecil dan untuk menggaris bawahi arus data antara kedua hal yang tersebut diatas. Diagram ini lalu dikembangkan untuk melihat lebih rinci sehingga dapat terlihat model-model yang terdapat di dalamnya.

Berikut ini simbol-simbol yang digunakan dalam sistem aliran data antara lain:



Gambar 2.1 Sistem alir data Kenneth dan Kendall (2003).

Berikut ini adalah keterangan dari gambar di atas Kenneth dan Kendall (2003):

1. Kotak rangkap dua digunakan untuk menggambarkan suatu *entitas eksternal* yang dapat mengirim data atau menerima data dari sistem.

2. Tanda panah menunjukkan perpindahan data dari suatu titik ke titik lain dengan kepala tanda panah mengarah ke tujuan data.
3. Bujur sangkar dengan sudut membulat digunakan untuk menunjukkan adanya proses transformasi.
4. Penyimpanan data menandakan penyimpanan manual, seperti lemari file atau sebuah file atau basis data terkomputerisasi. Karena penyimpanan data mewakili seseorang tempat atau sesuatu maka diberi nama dengan sebuah kata benda.

2.8 *Entity Relationship Diagram*

Menurut Supriyanto (2005), *Entity Relationship Diagram* (ERD) merupakan suatu model untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi. ERD untuk memodelkan struktur data dan hubungan antar data, untuk menggambarkannya digunakan beberapa notasi dan simbol.

Entity Relationship Diagram (ERD) merupakan teknik yang digunakan untuk memodelkan kebutuhan data dari suatu organisasi, biasanya oleh *system analyst* dalam tahap analisis persyaratan proyek pengembangan *system*. Sementara seolah-olah teknik diagram atau alat peraga memberikan dasar untuk *desain database* relasional yang mendasari sistem informasi yang dikembangkan. ERD bersama-sama dengan detail pendukung, merupakan model data yang pada gilirannya digunakan sebagai spesifikasi untuk *database*.

2.9 *System Development Life Cycle*

Menurut Pressman (2001), Model *Software Development Life Cycle* (SDLC) yaitu sebuah siklus hidup pengembangan perangkat lunak yang terdiri dari beberapa tahapan-tahapan penting dalam membangun perangkat lunak yang dilihat dari segi pengembangannya. Dengan siklus SDLC, proses membangun sistem dibagi menjadi beberapa langkah dan pada sistem yang besar, masing-masing langkah dikerjakan oleh tim yang berbeda. SDLC tidak hanya penting untuk proses produksi *software*, tetapi juga sangat penting untuk proses *maintenance software* itu sendiri dan terdapat 4 metodologi SDLC, di antara lain yaitu: “*Classic Life Cycle*” atau model *Waterfall* merupakan model yang paling banyak dipakai didalam *Software Engineering* (SE). Model ini melakukan pendekatan secara sistematis dan urut mulai dari level kebutuhan sistem lalu menuju ke tahap *analisis, desain, coding, testing / verification, dan maintenance*. Disebut dengan *waterfall* karena tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya dan berjalan berurutan. Model *waterfall* ini memiliki 6 tahapan, yaitu :

1. *Analysis*

Permodelan ini diawali dengan mencari kebutuhan dari keseluruhan sistem yang akan diaplikasikan ke dalam bentuk *software*. Hal ini sangat penting, mengingat *software* harus dapat berinteraksi dengan *elemen-elemen* yang lain seperti *hardware, database, dsb*. Tahap ini sering disebut dengan *Project Definition*.

2. *Requirements Specification*

Proses pencarian kebutuhan diintensifkan dan difokuskan pada *software*. Untuk mengetahui sifat dari program yang akan dibuat, maka para *software engineer* harus mengerti tentang domain informasi dari *software*, misalnya fungsi yang dibutuhkan, *user interface*, dan lain-lain. Dari 2 aktivitas tersebut (pencarian kebutuhan *sistem* dan *software*) harus didokumentasikan dan ditunjukkan kepada pelanggan.

3 *Design*

Proses ini digunakan untuk mengubah kebutuhan-kebutuhan diatas menjadi representasi ke dalam bentuk *software* sebelum *coding* dimulai. Desain harus dapat mengimplementasikan kebutuhan yang telah disebutkan pada tahap sebelumnya. Seperti 2 aktivitas sebelumnya, maka proses ini juga harus didokumentasikan sebagai konfigurasi dari *software*.

4. *Coding*

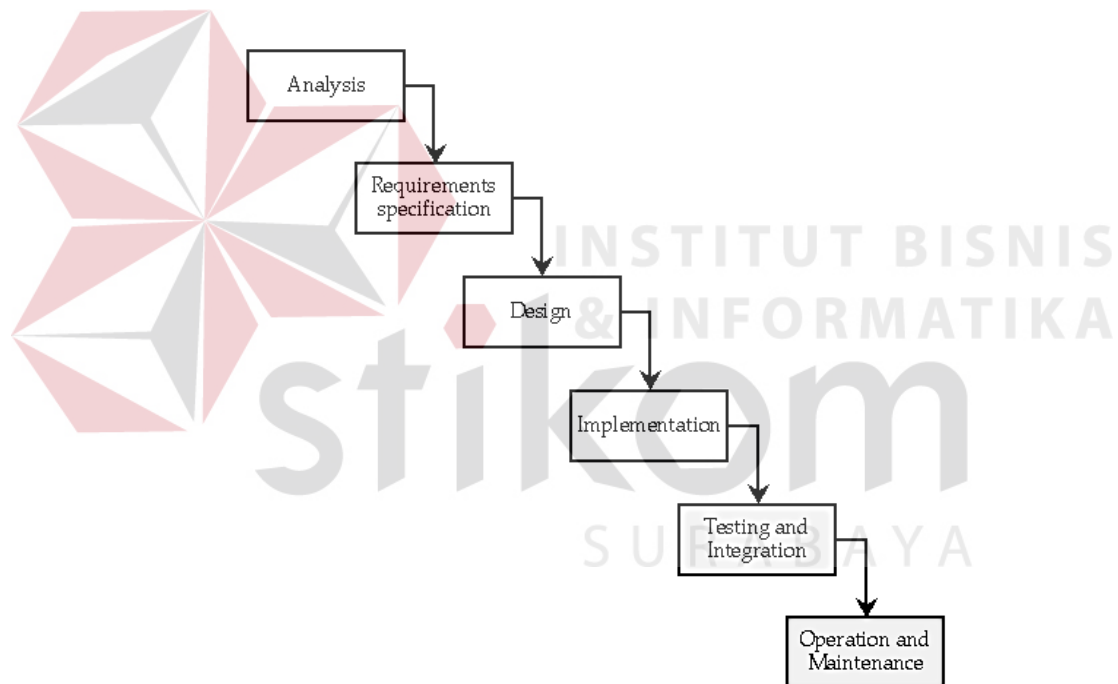
Desain yang telah dibuat kemudian diubah bentuknya menjadi bentuk yang dapat dimengerti oleh mesin, yaitu ke dalam bahasa pemrograman melalui proses *coding*. Tahap ini merupakan implementasi dari tahap desain yang secara teknis nantinya dikerjakan oleh programmer.

5. *Testing / Verification*

Sesuatu yang dibuat harus diujicobakan. Demikian juga dengan *software*. Semua fungsi-fungsi *software* harus diujicobakan, agar *software* bebas dari error, dan hasilnya harus benar-benar sesuai dengan kebutuhan yang sudah didefinisikan sebelumnya.

6. *Maintenance*

Pemeliharaan suatu *software* diperlukan, termasuk di dalamnya adalah pengembangan, karena *software* yang dibuat tidak selamanya hanya seperti itu. Ketika dijalankan mungkin saja masih ada error kecil yang tidak ditemukan sebelumnya, atau ada penambahan fitur-fitur yang belum ada pada *software* tersebut. Pengembangan diperlukan ketika adanya perubahan dari eksternal perusahaan seperti ketika ada pergantian *sistem operasi*, atau perangkat lainnya.



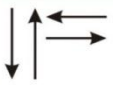















Gambar 2.2 Konsep *SDLC – Waterfall* (Pressman, 2005)

Keuntungan menggunakan teknik *waterfall*:

- a. Proses menjadi teratur.
- b. Estimasi proses menjadi lebih baik.
- c. Jadwal menjadi lebih menentu.

2.10 Bagan Alir Dokumen

Menurut Jogiyanto (2005), bagan alir dokumen atau di sebut bagan alir formulir atau *paperwork flowchart* merupakan bagan yang menunjukkan alir didalam program atau prosedur sistem secara logika dapat didefinisikan sebagai bagan yang menunjukkan arus pekerjaan secara keseluruhan dari sistem dengan menggunakan simbol seperti berikut pada tabel berikut:

	Flow Direction symbol Yaitu simbol yang digunakan untuk menghubungkan antara simbol yang satu dengan simbol yang lain. Simbol ini disebut juga connecting line.		Simbol Manual Input Simbol untuk pemasukan data secara manual on-line keyboard
	Terminator Symbol Yaitu simbol untuk permulaan (start) atau akhir (stop) dari suatu kegiatan		Simbol Preparation Simbol untuk mempersiapkan penyimpanan yang akan digunakan sebagai tempat pengolahan di dalam storage.
	Connector Symbol Yaitu simbol untuk keluar - masuk atau penyambungan proses dalam lembar / halaman yang sama.		Simbol Predefine Proses Simbol untuk pelaksanaan suatu bagian (sub-program)/prosedure
	Connector Symbol Yaitu simbol untuk keluar - masuk atau penyambungan proses pada lembar / halaman yang berbeda.		Simbol Display Simbol yang menyatakan peralatan output yang digunakan yaitu layar, plotter, printer dan sebagainya.
	Processing Symbol Simbol yang menunjukkan pengolahan yang dilakukan oleh komputer		Simbol disk and On-line Storage Simbol yang menyatakan input yang berasal dari disk atau disimpan ke disk.
	Simbol Manual Operation Simbol yang menunjukkan pengolahan yang tidak dilakukan oleh komputer		Simbol magnetik tape Unit Simbol yang menyatakan input berasal dari pita magnetik atau output disimpan ke pita magnetik.
	Simbol Decision Simbol pemilihan proses berdasarkan kondisi yang ada.		Simbol Punch Card Simbol yang menyatakan bahwa input berasal dari kartu atau output ditulis ke kartu
	Simbol Input-Output Simbol yang menyatakan proses input dan output tanpa tergantung dengan jenis peralatannya		Simbol Dokumen Simbol yang menyatakan input berasal dari dokumen dalam bentuk kertas atau output dicetak ke kertas.

Gambar 2.3 Flowchart Kenneth dan Kendall (2003).

2.11 Microsoft SQL Server

Menurut Marlinda (2004), *database* adalah suatu susunan/kumpulan data operasional lengkap dari suatu organisasi/perusahaan yang diorganisir/dikelola dan disimpan secara terintegrasi dengan menggunakan metode tertentu menggunakan komputer sehingga mampu menyediakan informasi optimal yang

diperlukan pemakainya. Penyusunan satu *database* digunakan untuk mengatasi masalah-masalah pada penyusunan data yaitu redundansi dan inkonsistensi data, kesulitan pengaksesan *data*, isolasi data untuk standarisasi, banyak pemakai (*multiple user*), masalah keamanan (*security*), masalah kesatuan (*integration*), dan masalah kebebasan data (*data independence*).

2.12 Testing Program *Black Box*

Menurut Pressman (2005), testing adalah proses eksekusi suatu program untuk menemukan kesalahan sebelum digunakan oleh pengguna akhir (*end-user*). Salah satu metode pengujian perangkat lunak adalah *Black-Box* Testing. *Black-box* Testing merupakan sebuah metode yang digunakan untuk menemukan kesalahan dan mendemonstrasikan fungsional aplikasi saat dioperasikan, apakah input diterima dengan benar dan output yang dihasilkan telah sesuai dengan yang diharapkan.

Fokus dari pengujian menggunakan metode *Black-Box* adalah pada pengujian fungsionalitas dan output dihasilkan aplikasi. Pengujian *Black-box* didesain untuk mengungkap kesalahan pada persyaratan fungsional dengan mengabaikan mekanisme *internal* atau komponen dari suatu program. pengujian perangkat lunak mempunyai beberapa level, untuk pengujian menggunakan metode *Black Box*, terdapat enam level yaitu *Integration*, *Functional*, *System*, *Acceptance*, *Beta*, dan *Regression*.

Salah satu dari pengujian *Black Box* yang dapat dilakukan oleh seorang penguji independen adalah *Functional testing*. Basis uji dari *functional testing* ini adalah pada spesifikasi dari komponen perangkat lunak yang akan diuji.

Functional testing memastikan bahwa semua kebutuhan-kebutuhan telah dipenuhi dalam sistem aplikasi. Dengan demikian fungsinya adalah tugas-tugas yang didesain untuk dilaksanakan sistem. *Functional testing* berkonsentrasi pada hasil dari proses, bukan bagaimana prosesnya terjadi.

