BAB II

LANDASAN TEORI

Dalam merancang dan membangun aplikasi, sangatlah penting untuk mengetahui terlebih dahulu dasar-dasar teori yang digunakan. Dasar-dasar teori tersebut digunakan sebagai landasan berpikir dalam melakukan pembahasan lebih lanjut sehingga terbentuk suatu aplikasi yang sesuai dengan tujuan awal.

2.1 Penelitian Terdahulu

Penelitian terdahulu digunakan untuk memberi suatu perbandingan referensi proyek yang telah dikerjakan, terdapat 2 contoh referensi dari penelitian terdahulu, yaitu:

a) Menurut Putri (2008). Dalam tugas akhirnya mengungkapkan bahwa *sistem informasi* penjualan pada swalayan Koperasi Setia Bhakti Wanita Surabaya, meliputi aktifitas yaitu laporan perbandingan penjualan barang dan laporan kontribusi anggota swalayan. Temuan yang didapat dari penelitian ini yaitu kurangnya laporan keuntungan peritem yang ditentukan perhari, perbulan, dan pertahun, dengan menggunakan media *visualisasi* untuk membantu managemen membaca laporan keuntungan dan *sistem informasi* tersebut masih belum mempunyai sistem katalog yang ditujukan untuk pembeli.

b) 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 persentase 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 di beli. Temuan yang didapat dari tugas akhir ini yaitu tidak memiliki *sistem katalog* berbasis *komputerisasi* serta *visualisasi* untuk membantu mempermudah melihat keuntungan barang.

2.2 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 dikelompokan 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.

2.3 Aplikasi

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

Dari definisi di atas dapat disimpulkan bahwa aplikasi adalah suatu program komputer yang dibuat untuk mengerjakan dan melaksanakan tugas

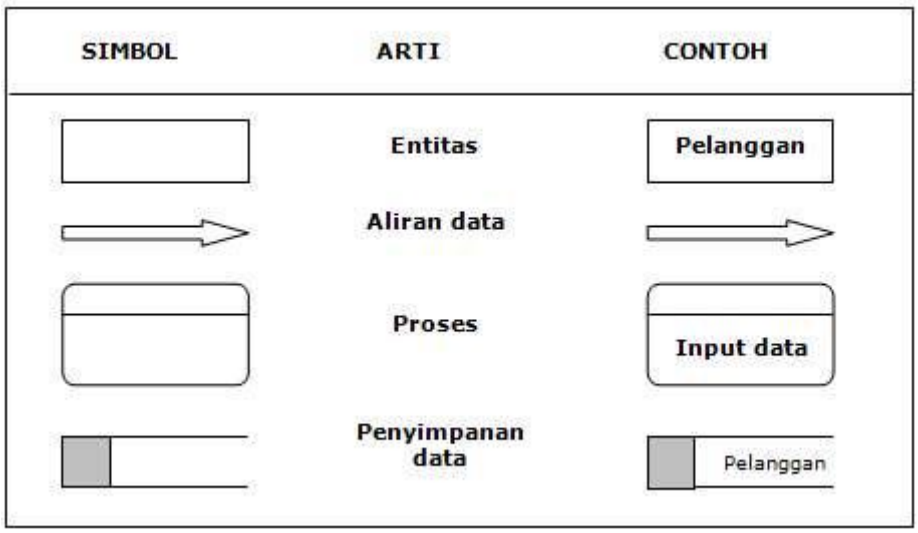
khusus dari pengguna. Aplikasi merupakan rangkaian kegiatan atau perintah untuk dieksekusi oleh komputer.

2.4 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 dam keluaran serta mempresentasikan dan prosedur mendetail dalam sistem yang masukan, proses dam keluaran serta merepresentasikan dan menganalisis prosedur-prosedur mendetail dalam sistem yang alir data juga mampu mengkoseptualisasikan bagaimana data-data berpindah lebih besar. Diagram mengkoseptualisasikan 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.

2.5 *Entity Relationship Diagram*

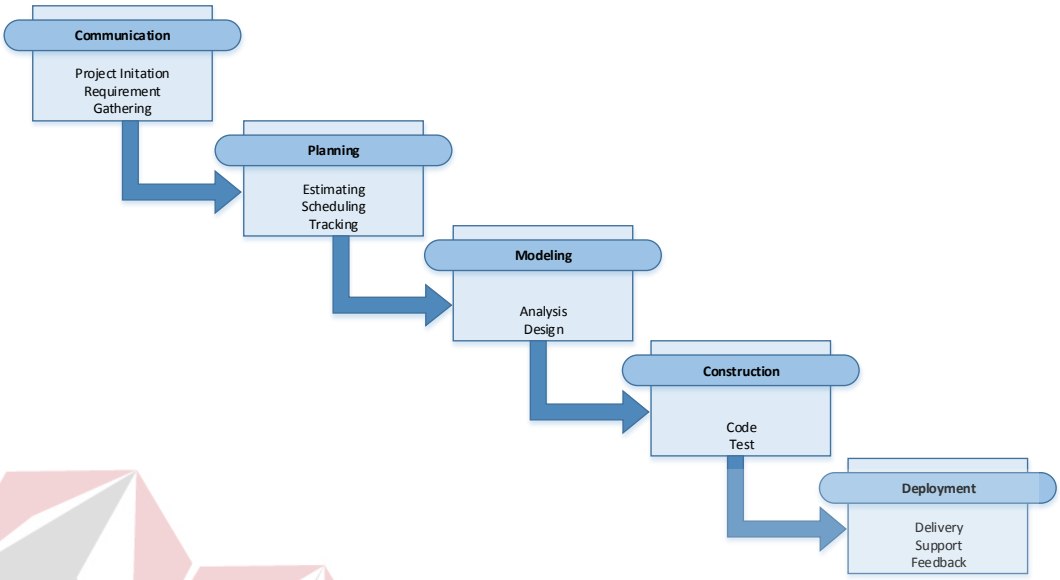
Menurut Supriyanto (2005), Entity Relationship Diagram (ERD) merupakan suatu model untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi. ERD untuk memodelkan struktur data dan hubungan antar data, untuk menggambarkannya digunakan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

2.6 System Development Life Cycle

Menurut Pressman (2015), *System Develoment Life Cycle* (SDLC) ini biasanya disebut juga dengan model *waterfall*. Menurut Pressman (2015), nama lain dari Model *Waterfall* adalah Model Air Terjun kadang dinamakan siklus hidup klasik (*classic life cy*le), dimana hal ini menyiratkan pendekatan yang sistematis dan berurutan (sekuensial) pada pengembangan perangkat lunak. Pengembangan perangkat lunak dimulai dari spesifikasi kebutuhan pengguna dan berlanjut melalui tahapan-tahapan perencanaan (*planning*), pemodelan (*modeling*), konstruksi (*construction*), serta penyerahan sistem perangkat lunak ke

para pelanggan/pengguna (*deployment*), yang diakhiri dengan dukungan berkelanjutan pada perangkat lunak yang dihasilkan.



Gambar 2.2. *Software Development Life Cycle*

(Sumber: Pressman, 2015)

Gambar 2.2 menunjukkan tahapan umum dari model proses *waterfall*. Model ini disebut dengan waterfall karena tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya dan berjalan berurutan. Akan tetapi, Pressman (2015) memecah model ini meskipun secara garis besar sama dengan tahapan-tahapan model *waterfall* pada umumnya.

Model ini merupakan model yang paling banyak dipakai dalam *Software Engineering*. Model ini melakukan pendekatan secara sistematis dan urut mulai dari level kebutuhan sistem lalu menuju ke tahap *Communication*, *Planning*, *Modeling*, *Construction*, dan *Deployment*. Berikut ini adalah penjelasan dari tahap-tahap yang dilakukan di dalam Model *Waterfall* menurut Pressman (2015):

a. *Communication*

Langkah pertama diawali dengan komunikasi kepada konsumen/pengguna. Langkah awal ini merupakan langkah penting karena menyangkut pengumpulan informasi tentang kebutuhan konsumen/pengguna.

b. *Planning*

Setelah proses *communication* ini, kemudian menetapkan rencana untuk pengerjaan software yang meliputi tugas-tugas teknis yang akan dilakukan, risiko yang mungkin terjadi, sumber yang dibutuhkan, hasil yang akan dibuat, dan jadwal pengerjaan.

c. *Modeling*

Pada proses *modeling* ini menerjemahkan syarat kebutuhan ke sebuah perancangan perangkat lunak yang dapat diperkirakan sebelum dibuat coding. Proses ini berfokus pada rancangan struktur data, arsitektur *software*, representasi *interface*, dan detail (algoritma) prosedural.

d. *Construction*

Construction merupakan proses membuat kode (*code generation*). Coding atau pengkodean merupakan penerjemahan desain dalam bahasa yang bisa dikenali oleh komputer. *Programmer* akan menerjemahkan transaksi yang diminta oleh user. Tahapan inilah yang merupakan tahapan secara nyata dalam mengerjakan suatu *software*, artinya penggunaan komputer akan dimaksimalkan dalam tahapan ini. Setelah pengkodean selesai maka akan dilakukan *testing* terhadap sistem yang telah dibuat. Tujuan *testing* adalah menemukan kesalahan-kesalahan terhadap sistem tersebut untuk kemudian bisa diperbaiki.

e. *Deployment*

Tahapan ini bisa dikatakan *final* dalam pembuatan sebuah *software* atau sistem. Setelah melakukan analisis, desain dan pengkodean maka sistem yang sudah jadi akan digunakan *user*. Kemudian *software* yang telah dibuat harus dilakukan pemeliharaan secara berkala.

2.7 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*).

2.8 Testing

Testing adalah proses pemantapan kepercayaan akan kinerja program atau sistem sebagaimana yang diharapkan. Testing *software* adalah proses mengoperasikan *software* dalam suatu kondisi yang dikendalikan untuk verifikasi, mendeteksi *error* dan validasi. Verifikasi adalah pengecekan atau pengetesan entitas-entitas, termasuk *software*, untuk pemenuhan dan konsistensi dengan melakukan evaluasi hasil terhadap kebutuhan yang telah ditetapkan. Validasi

adalah melihat kebenaran sistem apakah proses yang telah ditulisan sudah sesuai dengan apa yang dibutuhkan oleh pengguna. Deteksi *error* adalah testing yang berorientasi untuk membuat kesalahan secara intensif, untuk menentukan apakah suatu hal tersebut terjadi bilamana tidak seharusnya terjadi atau suatu hal tersebut tidak terjadi. *Test cas*e merupakan suatu tes yang dilakukan berdasarkan pada suatu inisialisasi, masukan, kondisi ataupun hasil yang telah ditentukan sebelumnya (Romeo, 2003). Adapun kegunaan dari *test case* ini, adalah sebagai berikut:

1. Untuk melakukan testing kesesuaian suatu komponen terhadap desain *White Box Testing.*
2. Untuk melakukan testing kesesuaian suatu komponen terhadap spesifikasi *Black Box Testing*.

2.8.1 White Box Testing

White box testing adalah suatu metode desain *test case* yang menggunakan struktur kendali dari desain prosedural. Seringkali *white box testing* diasosiasikan dengan pengukuran cakupan tes, yang mengukur persentase jalur-jalir dari tipe yang dipilih untuk dieksekusi oleh *test cases*. *White box testing* dapat menjamin semua struktur *internal* data dapat dites untuk memastikan validitasnya (Romeo, 2003).

Cakupan pernyataan, cabang dan jalur adalah suatu teknik *white box testing* yang menggunakan alur logika dari program untuk membuat *test cases* alur logika adalah cara dimana suatu bagian dari program tertentu dieksekusi saat menjalankan program. Alur logika suatu program dapat direpresentasikan dengan *flow graph*.

2.8.2 Black Box Testing

Black box testing dilakukan tanpa adanya suatu pengetahuan tentang detail struktur internal dari sistem atau komponen yang dites, juga disebut sebagai functional testing. Black box testing berfokus pada kebutuhan fungsional pada software, berdasarkan pada spesifikasi kebutuhan dari software (Romeo, 2003).

Dengan adanya black box testing, perekayasa software dapat menggunakan kebutuhan fungsional pada suatu program. Black box testing dilakukan untuk melakukan pengecekan apakah sebuah software telah bebas dari error dan fungsi-fungsi yang diperlukan telah berjalan sesuai dengan yang diharapkan.