

BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1 Tinjauan Pustaka

Gozali, Gunadhi, dan Kurniawati merancang sistem informasi tentang “*Perancangan Sistem Informasi Penjualan Buku Pada PD. Restu Percetakan*”. Mereka merancang sistem informasi menggunakan metode *Sistem Traditional Development* (STD). Rancangan sistem informasi yang di bangun menghasilkan laporan pertahun atau perbulan dengan akurat. (Gozali, Gunadhi, and Kurniawati 2012).

Sudirman, Retnadi, dan Kurniawan mengembangkan aplikasi tentang “*Perancangan Program Aplikasi Transaksi Pembayaran SPP, UTS Dan UAS’. Menggunakan Metode Analisis Dan Desain Berorientasi Objek Model Unified Approach (UA)*”. Dalam pengembangan perangkat lunaknya Sudirman, Retnadi, dan Kurniawan menggunakan metodologi berorientasi object *Unified Approach* (UA) dikemukakan oleh Ali Bahrami (1999) yang terdiri dari tahapan-tahapan *Object Oriented Analysis* (OOA) dan *Object Oriented Design* (OOD) dengan menggunakan notasi grafis standar *Unified Modelling Language*(UML) untuk memodelkan kebutuhan sistemnya. (Sudirman, Retnadi, and Kurniawati 2012)

Galih Wisnu dan Rachmatullah mengembangkan aplikasi tentang “*Sistem Informasi Kasir di Hokky Waroeng Dimsum Surakarta*’. Mereka mengembangk aplikasi menggunakan metode *Waterfall*, perancangan sitem menggunakan *Java Netbeans 7.4*, *DFD*, *database MySql Server*, dan *ireport 3.7.2*. Hasil dari Sistem Informasi Kasir ini adalah Sistem Informasi Kasir mampu mengelola data kasir, yang meliputi *input* data penjualan, menjumlah dan mengurangi nominal uang transaksi, laporan penjualan, dan cetak nota pelanggan. (Galih Wisnu and Rachmatullah 2016)

2.2 Landasan Teori

2.2.1 Persediaan

Persediaan (*inventory*) adalah meliputi semua barang yang dimiliki perusahaan pada saat tertentu, dengan tujuan untuk dijual atau dikonsumsi dalam siklus operasi normal perusahaan. Aktiva lain yang dimiliki perusahaan, tetapi tidak untuk dijual atau dikonsumsi tidak termasuk dalam klasifikasi persediaan (Mina Sari and Muhammad Dahria 2010)

2.2.2 Penjualan

Penjualan adalah proses dimana penjual memberikan segala kebutuhan dan keinginan pembeli agar memberi manfaat yang baik untuk penjual maupun pembeli yang berkelanjutan dan dapat memberi keuntungan bagi kedua belah pihak (Winardi, 1991). Berikut merupakan beberapa jenis dari penjualan menurut (Basu Swasta 1997).

a. Trade Selling

Dapat terjadi bila produsen dan pedagang besar mempersilahkan pengecer untuk berusaha memperbaiki distributor produk-produk mereka. Hal itu melibatkan para penyalur dengan kegiatan promosi, peragaan, persediaan dan pengadaan produk baru, jadi titik beratnya pada penjualan melalui penyalur daripada penjualan ke pembeli akhir.

b. Missionary Selling

Dalam *missionary selling*, penjualan berusaha ditingkatkan dengan mendorong pembeli untuk membeli barang-barang dari penyalur perusahaan. Dalam perusahaan yang bersangkutan memiliki penyalur sendiri dalam pendistribusian produknya.

c. Technical Selling

Dalam *Technical Selling*, pengusaha berusaha meningkatkan penjualan dengan pemberian saran dan nasehat pada pembeli akhir dari barang dan jasanya dengan menunjukkan bagaimana produk dan jasa yang ditawarkan dapat mengatasi masalah yang ada.

d. *New Business Selling*

Dalam *Technical Selling*, pengusaha berusaha membuka transaksi baru dengan merubah calon pembeli menjadi pembeli. *New Business Selling* sering dipakai oleh perusahaan asuransi.

e. *Responsive Selling*

Dua jenis penjualan utama adalah *route driving* dan *retailing*. Jenis penjualan *responsive selling* tidak akan menciptakan penjualan yang terlalu besar meskipun layanan yang baik dan hubungan pelanggan yang menyenangkan dapat menjurus pada pembeli ulang.

2.2.3 Pembelian

Menurut Bodnar, George, dan Hopwood (2004), pembelian adalah proses pencarian sumber dan pemesanan bahan, jasa, dan perlengkapan. Tujuan utamanya adalah memperoleh bahan dengan biaya serendah mungkin yang konsisten dengan kualitas dan jasa yang dipersyaratkan.

Pembelian juga mempunyai prinsip-prinsip yang dijadikan pedoman. Prinsip pembelian adalah hal-hal pokok dalam pelaksanaan fungsi pembelian yang perlu dijadikan pokok atau acuan. Isi pokok dari prinsip pembelian itu adalah sebagai berikut (Suhendra, 2010).

a. *The Right Place*

Merupakan nilai suatu barang yang dinyatakan dalam mata uang yang layak atau yang umum berlaku pada saat dan kondisi pembelian dilakukan.

b. *The Right Quantity*

Jumlah yang tepat dapat dikatakan sebagai suatu jumlah yang benar-benar diperlukan oleh suatu perusahaan pada suatu saat tertentu.

c. *The Right Time*

The right time menyangkut pengertian bahwa barang tersedia setiap kali diperlukan. Dalam hal persediaan barang haruslah diperhitungkan karena jika ada persediaan barang tentunya ada biaya perawatan barang

d. *The Right Place*

Mengandung pengertian bahwa barang yang dibeli dikirimkan atau diserahkan pada tempat yang dikehendaki oleh pembeli.

e. *The Right Quality*

The right quality adalah mutu barang yang diperlukan oleh suatu perusahaan sesuai dengan ketentuan yang sudah dirancang.

f. *The Right Source*

The right source mengandung pengertian bahwa barang berasal dari sumber yang tepat. Sumber dikatakan tepat apabila memenuhi prinsip-prinsip yang lain yaitu *the right price, the right quantity, the right time, the right place, and the right quality*.

2.2.4 Software Development life Cycle

Dalam alur penelitian, metode yang digunakan adalah model SDLC (*Software Development Life Cycle*). SDLC adalah suatu kerangka yang menggambarkan beberapa kegiatan yang dilakukan melalui beberapa tahap dalam pembuatan sebuah *software* (Fatta 2007). Selain itu, SDLC juga penting untuk proses *maintenance software* itu sendiri.

Model SDLC yang dipakai dalam pengembangan aplikasi adalah model *Waterfall*. Menurut Sommerville (2011), *waterfall model* adalah sebuah contoh dari proses perencanaan dimana semua proses kegiatan harus terlebih dahulu direncanakan dan dijadwalkan sebelum dikerjakan. *Waterfall Model* atau *Classic Life Cycle* merupakan model yang paling banyak dipakai dalam *Software Engineering* (SE). Disebut *waterfall* karena tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya dan berjalan berurutan.

2.2.5 Black Box Testing

Dalam pengujian perangkat lunak ada dua yaitu *white box testing* dan *black box testing*. Dari kedua metode itu, yang digunakan pada skripsi adalah *black box testing* dengan alasan pemilik toko hanya menguji apakah fungsionalitas dalam aplikasi sudah sesuai dengan yang diharapkan.

Menurut Pressman (2012), *black box testing* berfokus pada persyaratan fungsional perangkat lunak yang memungkinkan *engineer* untuk memperoleh *input* yang sepenuhnya akan melaksanakan persyaratan fungsional untuk sebuah program. *Black box testing* berusaha untuk menemukan kesalahan dalam kategori berikut:

1. Fungsi yang tidak benar atau fungsi yang hilang.
2. Kesalahan antarmuka.
3. Kesalahan dalam struktur data atau akses *database* eksternal.
4. Kesalahan kinerja.
5. Kesalahan inisialisasi dan pemutusan kesalahan.

2.2.6 Teknologi Pengembangan Aplikasi

2.2.6.1 Unified Modeling Language (UML)

Unified Modeling Language (UML) merupakan bahasa pemodelan secara grafis yang digunakan untuk menspesifikasikan, memvisualisasikan, membangun, dan mendokumentasikan seluruh rancangan aplikasi perangkat lunak. Penggunaan model UML bertujuan untuk mengidentifikasi bagian-bagian yang termasuk dalam lingkup sistem di dalam aplikasi. Model UML yang dipakai dalam pengembangan aplikasi penjualan dan pembelian antara lain adalah *Use Case Diagram*, *Activity Diagram*, dan *Class Diagram*.

a. Use Case Diagram

Use Case Diagram mendeskripsikan sebuah interaksi antara satu atau lebih *actor* dengan sistem yang dibuat. Dapat dikatakan *Use Case* digunakan untuk mengetahui fungsi yang ada di dalam sistem dan siapa saja yang berhak menggunakan fungsi-fungsi yang dibutuhkan. Simbol-simbol yang digunakan dalam *use case diagram* dapat dilihat pada Tabel 2.1.

Tabel 2.1 Simbol-simbol dalam *Use Case Diagram*

NO	Gambar	Nama Gambar	Keterangan
1		<i>Use Case</i>	Merupakan fungsionalitas yang disediakan sistem sebagai unit yang bertukar pesan dengan <i>actor</i> .
2		<i>Actor</i>	Merupakan <i>abstraction</i> dari orang yang mengaktifkan fungsi dari target sistem dan merupakan orang yang berinteraksi dengan <i>use case</i> .
3		<i>Association</i>	Digambarkan dengan garis tanpa panah yang mengindikasikan siapa yang berinteraksi secara langsung dengan sistem.
4		<i>Generalization</i>	Mengindikasikan siapa yang berinteraksi secara pasif dengan sistem.
5		<i>Include</i>	Mengidentifikasi hubungan antar dua <i>use case</i> dimana satu <i>usecase</i> memanggil <i>usecase</i> yang lain.
6		<i>Extend</i>	Merupakan perluasan dari <i>use case</i> jika kondisi atau syarat terpenuhi.

b. *Activity Diagram*

Activity Diagram merupakan diagram yang digunakan untuk menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau

proses bisnis. Simbol-simbol yang digunakan dalam *activity diagram* dapat dilihat pada Tabel 2.2.

Tabel 2.2 Simbol-simbol dalam *Activity Diagram*

NO	Gambar	Nama Gambar	Keterangan
1		<i>Start Point</i>	Merupakan awal dalam aktifitas.
2		<i>End Point</i>	Merupakan akhir dalam aktifitas.
3		<i>Activities</i>	Menggambarkan suatu proses atau kegiatan bisnis
4		<i>Decision Point</i>	Menggambarkan pilihan untuk pengambilan keputusan dalam aktifitas.
5		<i>Swimlane</i>	Digunakan untuk pembagian <i>activity diagram</i> yang menunjukkan siapa yang melakukan aktifitas.

c. *Class Diagram*

Class Diagram merupakan diagram yang digunakan untuk menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan. *Class Diagram* secara khas meliputi : Nama Kelas (*Class Name*), Atribut (*Attributes*), Operasi (*Operations*), dan Relasi (*Relationships*).

2.2.6.3 Basis Data

Basis data merupakan salah satu teknologi yang harus dimiliki sebuah perusahaan, institusi, ataupun organisasi adalah teknologi yang dapat memproses data. Secara konsep basis data atau *database* adalah kumpulan dari data-data yang membentuk suatu berkas (*file*) yang saling berhubungan (*relation*) dengan tata cara tertentu untuk membentuk data baru atau informasi. Kumpulan dari data yang saling berhubungan (relasi) antara satu dengan lainnya yang diorganisasikan berdasarkan skema atau struktur tertentu (Supriyanto, 2005).

Untuk mengelola *database* diperlukan suatu perangkat lunak yang disebut DBMS (*Data Base Management System*). DBMS merupakan suatu sistem perangkat lunak yang memungkinkan pengguna untuk membuat, memelihara, mengontrol, dan mengakses *database*. Dengan DBMS, pengguna dapat mengontrol dan memanipulasi data yang ada.

2.2.6.4 *PhpMyAdmin*

PhpMyAdmin adalah perangkat lunak bebas yang ditulis dalam bahasa pemrograman PHP yang digunakan untuk menangani administrasi MySQL melalui Jejaring Jagat Jembar (*World Wide Web*). *phpMyAdmin* mendukung berbagai operasi MySQL, diantaranya (mengelola basis data, tabel-tabel, bidang (*fields*), relasi (*relations*), indeks, pengguna (*users*), perijinan (*permissions*), dan lain-lain). Pada dasarnya, mengelola basis data dengan MySQL harus dilakukan dengan cara mengetikkan baris-baris perintah yang sesuai (*command line*) untuk setiap maksud tertentu. Jika seseorang ingin membuat basis data (*database*), ketikkan baris perintah yang sesuai untuk membuat basis data.

Hal tersebut tentu saja sangat menyulitkan karena seseorang harus hafal dan mengetikkan perintahnya satu per satu. Saat ini banyak sekali perangkat lunak yang dapat dimanfaatkan untuk mengelola basis data dalam *MySQL*, salah satunya adalah *phpMyAdmin*. Dengan *phpMyAdmin*, seseorang dapat membuat *database*, membuat tabel, mengisi data, dan lain-lain dengan mudah, tanpa harus menghafal

baris perintahnya. *PhpMyAdmin* merupakan bagian untuk mengelola basis data *MySQL* yang ada di komputer (www. Wikipedia. org.2011).

2.2.6.5 MySQL

MySQL adalah sebuah implementasi dari sistem manajemen basis data relasional yang di distribusikan secara gratis dibawah lisensi GPL (*General Public License*). Setiap pengguna dapat secara bebas menggunakan *MySQL*, namun dengan batasan perangkat lunak tidak boleh dijadikan produk turunan yang bersifat komersial.

Berikut beberapa keistimewaan dari *MySQL* menurut beberapa sumber sebagai bahan pertimbangan:

1. Portabilitas. *MySQL* dapat berjalan stabil pada berbagai sistem operasi seperti *Windows, Linux, FreeBSD, Mac Os X Server, Solaris, Amiga*, dan masih banyak lagi.
2. Perangkat lunak sumber terbuka (*open source*). *MySQL* didistribusikan sebagai *open source* sehingga dapat digunakan secara gratis.
3. *Multi-user*. *MySQL* dapat digunakan oleh beberapa pengguna dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.
4. *Performance tuning*, *MySQL* memiliki kecepatan yang menakjubkan dalam menangani query sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.
5. Ragam tipe data. *MySQL* memiliki ragam tipe data yang sangat kaya, seperti *signed or unsigned integer, float, double, char, text, date, timestamp*, dan lain-lain.
6. Perintah dan Fungsi. *MySQL* memiliki operator dan fungsi secara penuh yang mendukung perintah *Select* dan *Where* dalam perintah (*query*).
7. Keamanan. *MySQL* memiliki beberapa lapisan keamanan seperti password yang terenkripsi.
8. Skalabilitas dan Pembatasan. *MySQL* mampu menangani basis data dalam skala besar, dengan jumlah *record* lebih dari 50 juta dan 60 ribu

tabel serta 5 milyar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.

9. Konektivitas. MySQL dapat melakukan koneksi dengan klien menggunakan protokol TCP/IP, *Unix Scket* (UNIX), atau named pipes (NT).
10. Lokalisasi. MySQL dapat mendeteksi pesan kesalahan pada klien dengan menggunakan lebih dari dua puluh bahasa. Meskipun demikian, bahasa Indonesia belum termasuk di dalamnya.
11. Antar Muka. MySQL memiliki antar muka (*interface*) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (*Application Programming Interface*).
12. Klien dan Peralatan. MySQL dilengkapi dengan berbagai *tool* yang dapat digunakan untuk administrasi basis data, dan pada setiap peralatan yang ada disertakan petunjuk *online*.
13. Struktur tabel. MySQL memiliki struktur tabel yang lebih fleksibel dalam menangani ALTER TABLE, dibandingkan basis data lainnya semacam PostgreSQL ataupun Oracle.

2.2.6.6 Bahasa Java

Java merupakan bahasa pemrograman yang bersifat umum atau non-spesifik (*general purpose*), dan secara khusus didisain untuk memanfaatkan dependensi implementasi seminimal mungkin.

Karena fungsionalitasnya yang memungkinkan aplikasi java mampu berjalan di beberapa *platform sistem operasi* yang berbeda, java dikenal pula dengan slogannya, "Tulis sekali, jalankan di mana pun".

Penulis memakai bahasa java dalam mengembangkan aplikasi karena java dapat dijalankan di beberapa *platform* atau *sistem operasi* komputer, sesuai dengan prinsip tulis sekali, jalankan di mana saja.

Java merupakan bahasa pemrograman yang paling populer digunakan, dan secara luas dimanfaatkan dalam pengembangan berbagai jenis perangkat lunak aplikasi ataupun aplikasi berbasis web.

1. Kelebihan Java

1. *Multiplatform*. Kelebihan utama dari Java ialah dapat dijalankan di beberapa *platform*. Dengan kelebihan java, *programmer* cukup menulis sebuah program Java dan dikompilasi (diubah, dari bahasa yang dimengerti manusia menjadi bahasa mesin) sekali lalu hasilnya dapat dijalankan di atas beberapa *platform* tanpa perubahan. Kelebihan java memungkinkan sebuah program berbasis java dikerjakan di atas operating system Linux tetapi dijalankan dengan baik di atas Microsoft Windows. *Platform* yang didukung adalah Microsoft Windows, Linux, Mac OS, dan Sun Solaris.
2. Perpustakaan kelas yang lengkap. Java terkenal dengan kelengkapan yang sangat memudahkan dalam penggunaan oleh para *programmer* untuk membangun aplikasinya. Kelengkapan perpustakaan ditambah dengan keberadaan komunitas java yang terus menerus membuat perpustakaan-perpustakaan baru untuk melingkupi seluruh kebutuhan pembangunan aplikasi.
3. Memiliki sintaks seperti bahasa pemrograman C++ sehingga menarik banyak *programmer* C++ untuk pindah ke Java.

2. Kekurangan Java

1. Masih ada beberapa hal yang tidak kompatibel antara platform satu dengan *platform* lain. Untuk J2SE, *SWT-AWT bridge* yang sampai sekarang tidak berfungsi pada Mac OS X.
2. Mudah didekompilasi. Dekompilasi adalah proses membalikkan dari kode jadi menjadi kode sumber. Java merupakan *bytecode* yang menyimpan banyak atribut bahasa tingkat tinggi, seperti nama-nama kelas, metode, dan tipe data. Hal yang sama juga terjadi pada Microsoft .NET Platform. Dengan demikian, algoritma yang digunakan program lebih sulit disembunyikan dan mudah dibajak.
3. Penggunaan memori yang banyak. Biasanya bukan merupakan masalah bagi pihak yang menggunakan teknologi terbaru (karena *trend* memori

terpasang makin murah), tetapi menjadi masalah bagi mereka yang masih harus berkuat dengan mesin komputer berumur lebih dari 4 tahun.

2.2.7 Perangkat Lunak Pendukung

2.2.7.1 Netbeans

NetBeans merupakan salah satu proyek open source yang disponsori oleh Sun Microsystems. Proyek ini berdiri pada tahun 2000 dan telah menghasilkan 2 produk, yaitu NetBeans IDE dan NetBeans Platform. NetBeans IDE merupakan produk yang digunakan untuk melakukan pemrograman baik menulis kode, meng-compile, mencari kesalahan dan mendistribusikan program. Sedangkan NetBeans platform adalah sebuah modul yang merupakan kerangka awal/pondasi dalam membangun aplikasi desktop yang besar. (Wahana Komputer 2010)

2.2.7.2 IReport

IReport merupakan sebuah aplikasi designer yang digunakan untuk membuat laporan yang didalamnya terdapat Library JasperReport. Library JasperReport sendiri merupakan JAR (*Java Library*) yang bersifat open source dan dirancang untuk menambahkan kemampuan pelaporan (*reporting capabilities*) pada aplikasi java. JasperReport memiliki sejumlah fitur, antara lain :

1. Layout dan desain laporan yang fleksibel.
2. Dapat menampilkan laporan dalam bentuk teks maupun gambar (*chart*).
3. Dapat menghasilkan report dalam berbagai format : html, pdf, rtf, xls, csv.
4. Dapat menerima data dari berbagai sumber data : JDBC, Bean Collection, ResultSet, CSV, XML, Hibernate.

2.2.8 Aplikasi

Menurut Anisyah (2000), aplikasi adalah penerapan, penggunaan atau penambahan. Dari pengertian aplikasi, dapat disimpulkan bahwa aplikasi merupakan *software* yang berfungsi untuk melakukan berbagai bentuk pekerjaan atau tugas-tugas tertentu seperti penerapan, penggunaan dan penambahan data.

Beberapa aplikasi yang digabung bersama menjadi suatu paket kadang disebut sebagai suatu paket atau suite aplikasi (*application suite*). Contohnya adalah *Microsoft Office* dan *OpenOffice.org* yang menggabungkan suatu aplikasi pengolah kata, lembar kerja, serta beberapa aplikasi lainnya.

Aplikasi dalam suatu paket biasanya memiliki antarmuka pengguna yang memiliki kesamaan sehingga memudahkan pengguna untuk mempelajari dan menggunakan tiap aplikasi. Sering kali mereka memiliki kemampuan untuk saling berinteraksi satu sama lain sehingga menguntungkan pengguna. Contohnya, suatu lembar kerja dapat dibenamkan dalam suatu dokumen pengolah kata walaupun dibuat pada aplikasi lembar kerja yang terpisah.

2.2.9 Aplikasi Dekstop

Pengertian Aplikasi menurut Jogiyanto (1999) adalah penggunaan dalam suatu komputer, instruksi (*instruction*) atau pernyataan (*statement*) yang disusun sedemikian rupa sehingga computer dapat memproses masukan menjadi keluaran.

Aplikasi adalah program komputer yang dibuat khusus untuk melaksanakan dan mengerjakan permintaan khusus dari pengguna. Aplikasi merupakan rangkian kegiatan untuk dieksekusi oleh komputer. Program merupakan *instruction set* yang akan dijalankan oleh pengguna, yaitu berupa software. Program inilah yang mengendalikan semua aktifitas yang ada pada pemroses. Isi dari program sebenarnya konstruksi logika yang dibuat oleh manusia dan diterjemahkan kedalam Bahasa mesin sesuai dengan format yang ada pada *instructionset*.