

BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1 Tinjauan Pustaka

Fandi Halim (2016), aplikasi *website e-commerce "Pengembangan Website e-Commerce "TOMcell".*", tujuan mengembangkan *website e-commerce* pada TOMcell adalah untuk memperluas jangkauan pemasaran, meningkatkan pelayanan, mendatangkan pelanggan baru dan menjaga pelanggan yang loyal, serta menyediakan promosi yang lebih efektif, efisien, serta up-to-date. Metodologi pengembangan sistem yang digunakan adalah *Object-Oriented Analysis and Design (OOAD)* dan *tool* yang digunakan adalah *Unified Modeling Language (UML)*, serta pengembangan aplikasi menggunakan bahasa pemrograman *web PHP* dan basis data *PostgreSQL*.

Akbar Ricky (2016), mengembangkan *aplikasi "Penerapan Sistem Informasi Penjualan dan Inventori HANDPHONE serta Aksesorisnya Menggunakan Enterprise Resource Planning (ERP) (Studi Kasus: Toko Nahda Cell)."* Dalam pengembangannya aplikasi ERP ini dapat dirancang sendiri dan dapat juga menggunakan aplikasi yang *open source*. Penerapan ERP pada Toko Nahda Cell menggunakan ERP yang bersifat *Open Source Software (OSS)*. Tahapan yang dilakukan dalam penerapannya dimulai dengan studi pendahuluan, selanjutnya studi literatur digunakan untuk mencari landasan teori dan penelitian terdahulu tentang penerapan sistem ERP, mengidentifikasi proses bisnis *inventory* dan penjualan barang yang sedang berjalan kemudian membuatkan usulan sistem secara terkomputerisasi, yang digambarkan dengan menggunakan *Business Process Model Notation (BPMN)*, serta penggambaran model kerja sistem yang akan diterapkan menggunakan *use case diagram*.

Wyanaputra dan Fery Wongso Johan (2016). Mereka mengembangkan aplikasi *"Perancangan Sistem Pemesanan Barang Berbasis WEB Di Toko ZENITH KOMPUTER Di Pekanbaru."* Dalam pengembangannya menggunakan PHP (Personal Home Page) dan rancangan databasenya menggunakan *Xamp Server*. Hasil dari perancangan aplikasi Sistem Informasi Pemesanan berbasis web

menunjukkan bahwa peranan aplikasi Komputer dalam sistem informasi sangat penting sebagai penunjang dalam meningkatkan kualitas kegiatan Pemesanan dan pelayanan di lingkungan toko Zenith Komputer.

Dari ketiga penelitian sebelumnya, sistem informasi penjualan yang akan dibuat adalah sistem informasi yang dapat mengatur tentang penjualan dan pembelian pada *Counter* Bakoelan *HP* terhadap pelanggan yang meliputi transaksi penjualan dan pembelian itu sendiri dan transaksi pembayaran yang dilakukan pelanggan. Berdasarkan penelitian yang sudah ada, maka penulis menambah beberapa spesifikasi yang menjadi perbedaan dengan system yang dikembangkan.

Perbedaan tersebut adalah sebagai berikut:

1. Aplikasi dapat mencatat transaksi penjualan pada setiap pelanggan.
2. Aplikasi dapat otomatis membuat laporan yang dapat ditentukan oleh pemilik toko baik itu harian atau bulanan.
3. Aplikasi dapat menampilkan laporan penjualan dan pembelian untuk *inventory* toko.

2.2 Landasan Teori

2.2.1 Pengenalan Sistem Aplikasi Penjualan

Sistem Aplikasi Penjualan Handphone, aksesoris dan pulsa ini menyajikan transaksi penjualan handphone, aksesoris dan pulsa, menginput laporan transaksi pembelian barang, menginput laporan transaksi penjualan barang, menginput data pelanggan. Kelayakan operasional menyangkut beberapa aspek. Untuk disebut layak secara operasional, usulan kebutuhan sistem harus benar-benar bisa menyelesaikan masalah yang ada di sisi pemesan sistem informasi. Di samping itu, informasi yang dihasilkan oleh sistem harus merupakan informasi yang benar-benar dibutuhkan oleh pengguna tepat pada saat pengguna menginginkannya. Dalam sistem ini, kebutuhan pengguna akan akuratnya laporan penjualan akan teratasi, kecepatan transaksi akan meningkat, dan me-manage database akan terasa lebih mudah. Pengguna juga dihadapkan pada form yang *user-friendly*.

2.3 Spesifikasi Software Yang Digunakan

2.3.1 Pengertian *SQL Server Management*

Menurut Mujiono (2015), *SQL Server Management* adalah sistem manajemen *database* relasional (RDBMS) yang dirancang untuk aplikasi dengan arsitektur *client/server*. Istilah *client*, *server*, dan *client/server* dapat digunakan untuk merujuk kepada konsep yang sangat umum atau hal yang spesifik dari perangkat keras atau perangkat lunak. Pada level yang sangat umum, sebuah *client* adalah setiap komponen dari sebuah sistem yang meminta layanan atau sumber daya (*resource*) dari komponen sistem lainnya. Sedangkan sebuah *server* adalah setiap komponen sistem yang menyediakan layanan atau sumber daya ke komponen sistem lainnya.

Kelebihan Microsoft SQL Server

Dengan kemampuannya untuk mengolah data yang besar maka DBMS ini sangat cocok untuk perusahaan mikro, menengah hingga perusahaan besar sekalipun.

- DBMS jenis ini memiliki kelebihan memmanage user data serta masing-masing user dapat diatur hak aksesnya terhadap pengaksesan data base oleh DBA
- Mempunyai tingkat keamanan data yang sangat baik
- Dapat melakukan back up, recovery dan rollback data dengan mudah
- Mempunyai kelebihan untuk membuat data base mirroring dan clustering

Kekurangan Microsoft SQL Server

- DBMS jenis ini hanya dapat berjalan pada sistem operasi / *platform windows* saja.
- Software ini mempunyai lisensi dari *microsoft* sehingga pemakaiannya membutuhkan biaya yang cukup mahal.

2.3.2 Definisi Sistem Informasi

Menurut O'Brien (2010), sistem informasi adalah suatu kombinasi terartur apapun dari *people* (orang), *hardware* (perangkat keras), *software* (piranti

lunak), *computer networks and data communications* (jaringan komunikasi), dan *database* (basis data) yang mengumpulkan, mengubah dan menyebarkan informasi di dalam suatu bentuk organisasi.

Secara umum Sistem informasi dapat didefinisikan sebagai suatu sistem di dalam suatu organisasi yang merupakan kombinasi dari orang-orang, fasilitas, teknologi, media prosedur-prosedur dan pengendalian yang ditujukan untuk mendapatkan jalur komunikasi penting, memproses tipe transaksi rutin tertentu, memberi sinyal kepada manajemen dan yang lainnya terhadap kejadian-kejadian internal dan eksternal yang penting dan menyediakan suatu dasar informasi untuk pengambilan keputusan.

Suatu sistem informasi pada dasarnya terbentuk melalui suatu kelompok kegiatan operasi yang tetap, yaitu:

1. Mengumpulkan data
2. Mengelompokkan data
3. Menghitung
4. Menganalisa
5. Menyajikan laporan

Sasaran sistem informasi adalah:

1. Meningkatkan penyelesaian tugas.
2. Pemakai harus lebih produktif agar menghasilkan keluaran yang memiliki mutu yang tinggi.
3. Meningkatkan efektifitas secara keseluruhan.
4. Sistem harus mudah dan sering digunakan.
5. Meningkatkan efektifitas ekonomi.
6. Keuntungan yang diperoleh dari sistem harus lebih besar dari biaya yang dikeluarkan.

2.3.3 Aplikasi Desktop

Menurut Rachmad Hakim S (2016). Aplikasi dalam bahasa awam sering disebut sebagai sebuah kumpulan program atau *script*. Aplikasi Desktop yang dibangun dengan menggunakan *Struts framework* terdiri dari komponen-komponen individual yang digabungkan menjadi satu aplikasi. Aplikasi tersebut dapat diinstal dan dieksekusi oleh aplikasi container. Komponen-komponen tersebut dapat digabungkan karena mereka terletak dalam sebuah konteks aplikasi yang sama, yang menjadikan mereka bergantung satu dengan yang lainnya, baik secara langsung ataupun tidak langsung.

2.3.4 Desktop Base

Menurut Achung Khoiry (2011). *Desktop based application* adalah suatu aplikasi yang dapat berjalan sendiri atau independen tanpa menggunakan *browser* atau koneksi Internet di suatu *computer otonom*, dengan *operating system* atau *flatfom* tertentu. Contoh: *Microsoft Office Word 2007* dan aplikasi *office* lainnya, *Windows Media Player*, dll. Keunggulan *desktop based application* dibandingkan dengan *web based application* adalah Dapat berjalan dengan independen, tanpa perlu menggunakan *browser*. Tidak perlu koneksi Internet, karena semua *file* yang diperlukan untuk menjalankan aplikasinya sudah terinstall sebelumnya. Dapat dengan mudah memodifikasi settingannya. Prosesnya lebih cepat.

2.3.5 Microsoft Visual Studio

Menurut Anwar Anshary (2011). *Microsoft Visual Studio* merupakan sebuah perangkat lunak lengkap (*suite*) yang dapat digunakan untuk melakukan pengembangan aplikasi, baik itu aplikasi bisnis, aplikasi personal, ataupun komponen aplikasinya, dalam bentuk aplikasi *console*, aplikasi *Windows*, ataupun aplikasi *Web*. *Visual Studio* mencakup kompiler, *SDK*, *Integrated Development Environment* (IDE), dan dokumentasi (umumnya berupa *MSDN Library*). Kompiler yang dimasukkan ke dalam paket *Visual Studio* antara lain *Visual C++*, *Visual C#*, *Visual Basic*, *Visual Basic .NET*, *Visual InterDev*, *Visual J++*, *Visual J#*, *Visual FoxPro*, dan *Visual SourceSafe*.

Microsoft *Visual Studio* dapat digunakan untuk mengembangkan aplikasi dalam *native code* (dalam bentuk bahasa mesin yang berjalan di atas *Windows*) ataupun *managed code* (dalam bentuk *Microsoft Intermediate Language* di atas *.NET Framework*). Selain itu, *Visual Studio* juga dapat digunakan untuk mengembangkan aplikasi Silverlight, aplikasi *Windows Mobile* (yang berjalan di atas *.NET Compact Framework*).

Visual Studio kini telah menginjak versi *Visual Studio 9.0.21022.08*, atau dikenal dengan sebutan *Microsoft Visual Studio 2008* yang diluncurkan pada 19 November 2007, yang ditujukan untuk platform *Microsoft .NET Framework 3.5*. Versi sebelumnya, *Visual Studio 2005* ditujukan untuk platform *.NET Framework 2.0* dan *3.0*. *Visual Studio 2003* ditujukan untuk *.NET Framework 1.1*, dan *Visual Studio 2002* ditujukan untuk *.NET Framework 1.0*. Versi-versi tersebut di atas ini dikenal dengan sebutan *Visual Studio .NET*, karena memang membutuhkan *Microsoft .NET Framework*. Sementara itu, sebelum muncul *Visual Studio .NET*, terdapat *Microsoft Visual Studio 6.0 (VS1998)*.

2.3.6 Microsoft Visual C#

Menurut Mira Astin (2012). *Microsoft Visual C#* adalah sebuah program alat bantu pemrograman (*Rapid Application Development tool*) yang dibuat oleh *Microsoft Corporation* dan dapat digunakan untuk membuat program 11 berbasis grafis dengan menggunakan bahasa pemrograman mirip *C++*. Program ini telah dimasukkan ke dalam produk *Microsoft Visual Studio*, bersama-sama dengan *Visual C++*, *Visual Basic*, *Visual FoxPro* serta *Visual J#*. Sejauh ini, program ini merupakan program yang paling banyak digunakan oleh para *programmer* untuk membuat program dalam bahasa *C#*.

Microsoft Visual C# adalah sebuah bahasa pemrograman yang bersifat OOP (*Object Oriented Programming*) yang dikembangkan oleh *Microsoft Corp* guna menggantikan *visual J#* yang terhalah lisensi dengan pemilik *java* pada saat itu *Sun Microsystems*, yang akhirnya membuat *Microsoft* untuk merancang sebuah bahasa pemrograman baru yang lebih dinamis.

Microsoft Visual C# menghilangkan beberapa hal yang bersifat kompleks yang terdapat dalam beberapa macam bahasa pemrograman seperti *Java* dan *C++*, termasuk diantaranya menghilangkan *macro*, *templates*, *multiple inheritance* dan *virtual base classes*. Hal-hal tersebut yang dapat menyebabkan kebingungan pada saat menggunakannya, dan juga berpotensi dapat jadi masalah bagi para programmer *C++*. Jika kita pertama kali belajar bahasa *C#* sebagai bahasa pemrograman, maka hal-hal tersebut diatas tidak akan membuat waktu anda terbuang terlalu banyak untuk mempelajarinya.

Microsoft visual C# bersifat sederhana, karena bahasa ini kepada bahasa *C* dan *C++*. Jika anda familiar, seperti *statement*, *expression*, *operators*, dan beberapa fungsi yang diadopsi langsung dari *Java*, *C* dan *C++*, tetapi dengan berbagai perbaikan yang membuat bahasanya menjadi lebih sederhana.

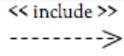
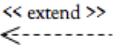
2.3.7 UML (Unified Modeling Language)

Menurut Munawar (2005). UML - *Unified Modeling Language* (UML) adalah tujuan umum, perkembangan, bahasa pemodelan di bidang rekayasa perangkat lunak, yang dimaksudkan untuk menyediakan cara standar untuk memvisualisasikan desain sistem.

UML awalnya termotivasi oleh keinginan untuk membakukan sistem notasi yang berbeda dan pendekatan untuk desain perangkat lunak yang dikembangkan oleh *Grady Booch*, *Ivar Jacobson* dan *James Rumbaugh* di *Rational Software* di 1994-1995, dengan pengembangan lebih lanjut yang dipimpin oleh mereka melalui tahun 1996.

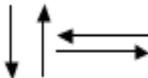
Macam-Macam UML Diagram

a. Use Case Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		Actor	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan use case.
2		Include	Menspesifikasikan bahwa use case sumber secara eksplisit.
3		Extend	Menspesifikasikan bahwa use case target memperluas perilaku dari use case sumber pada suatu titik yang diberikan.
4		Association	Apa yang menghubungkan antara objek satu dengan objek lainnya.
5		System	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
6		Use Case	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor

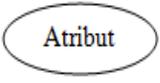
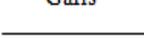
Gambar 2.1. Simbol-Simbol Use Case Diagram

b. Activity Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		Activity	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		Action	State dari sistem yang mencerminkan eksekusi dari suatu aksi
3		Initial Node	Bagaimana objek dibentuk atau diawali.
4		Activity Final Node	Bagaimana objek dibentuk dan diakhiri
5		Decision	Digunakan untuk menggambarkan suatu keputusan / tindakan yang harus diambil pada kondisi tertentu
6		Line Connector	Digunakan untuk menghubungkan satu simbol dengan simbol lainnya

Gambar 2.2. Simbol-Simbol Activity Diagram

c. ER Diagram

Notasi	Keterangan
	Entitas adalah suatu objek yang dapat diidentifikasi dalam lingkungan pemakai.
	Relasi menunjukkan adanya hubungan di antara sejumlah entitas yang berbeda.
	Atribut berfungsi mendeskripsikan karakter entitas (atribut yang berfungsi sebagai <i>key</i> diberi garis bawah).
	Garis sebagai penghubung antara relasi dan entitas atau relasi dan entitas dengan atribut.

Gambar 2.3. *Simbol-Simbol ER Diagram*

d. Class Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
2		<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4		<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
5		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
6		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan memengaruhi elemen yang bergantung padanya elemen yang tidak mandiri
7		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya

Gambar 2.4. *Simbol-Simbol Class Diagram*

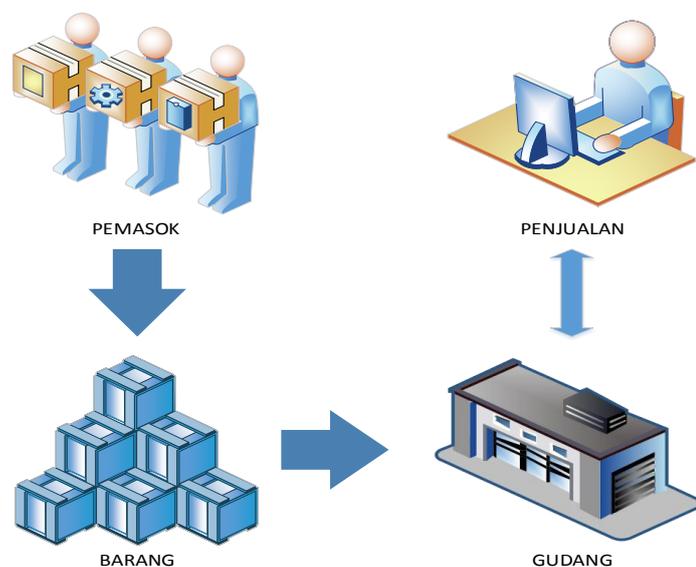
2.4 Software Development Life Cycle

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*. 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 *ClassicLife Cycle* merupakan model yang paling banyak dipakai dalam *SoftwareEngineering* (SE). Disebut *waterfall* karena tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya dan berjalan berurutan.

2.5 Alur Proses

2.4.1 Alur Proses Penjualan

Secara garis besar proses penjualan terdiri dari 4 proses yang dapat dilihat pada Gambar 2.5.



Gambar 2.5. Alur Proses Penjualan

Proses penjualan yang berjalan pada *counter* adalah sebagai berikut:

a. Penerimaan Barang

Dalam proses penerimaan barang pemasok mengirimkan barang ke *counter* Bakoelan HP, kemudian semua barang di simpan ke dalam gudang Bakoelan HP.

b. Penerimaan Barang Di Gudang

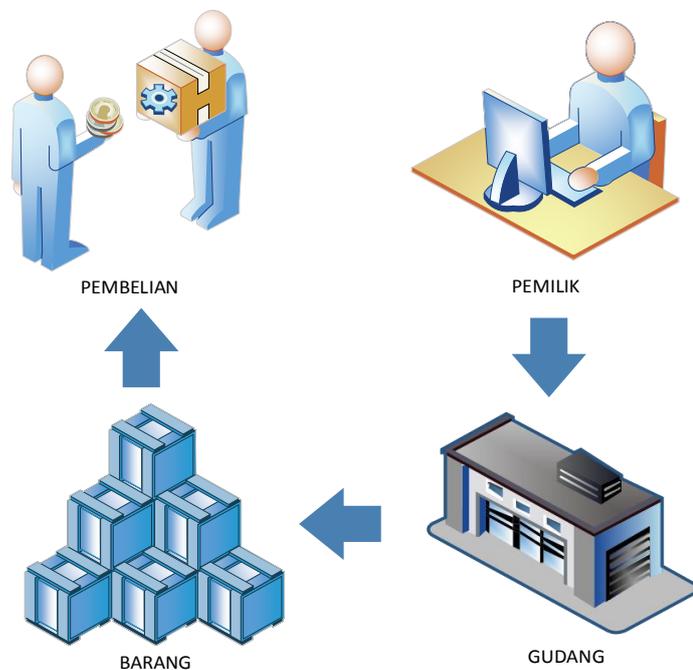
Yang dimaksudkan dengan penerimaan barang di gudang adalah gudang penyimpanan barang-barang yang dijual ke pelanggan.

c. Proses Penjualan

Pada proses penjualan dilakukan kegiatan penjualan barang dari gudang kepada pelanggan, setiap barang yang terjual secara otomatis masuk ke dalam laporan penjualan sesuai kategori barang.

2.4.2. Alur Proses Pembelian

Secara garis besar proses pembelian terdiri dari 4 proses yang dapat dilihat pada Gambar 2.6.



Gambar 2.6. Alur Proses Pembelian

Proses pembelian yang berjalan di *counter* adalah sebagai berikut:

a. Pengecekan Barang

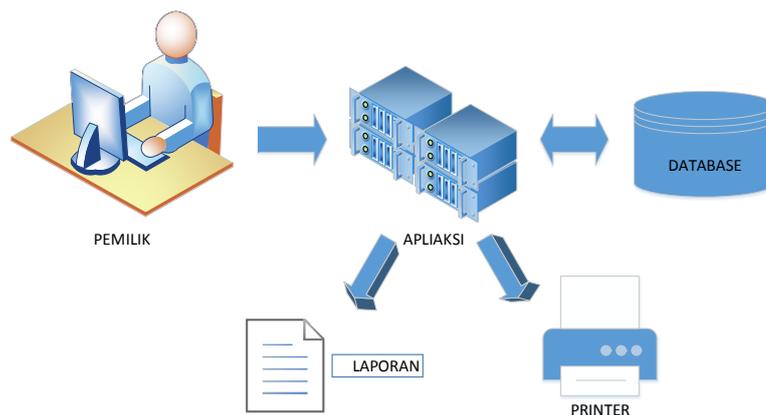
Dalam proses pengecekan barang, pemilik akan memeriksa barang-barang yang stoknya sudah habis.

b. Proses Pembelian

Pada proses pembelian, dilakukan kegiatan pembelian barang ke pemasok barang, setiap barang yang dibeli secara otomatis masuk kedalam laporan pembelian barang.

2.4.3 *Arsitektur Sistem*

Dalam mengembangkan aplikasi diperlukan perancangan arsitektur perangkat lunak yang bertujuan untuk menggambarkan bagaimana sistem dikembangkan dan dijalankan. Arsitektur perangkat lunak pada aplikasi dapat dilihat pada Gambar 2.7.



Gambar 2.7. *Arsitektur Sistem*

Berikut adalah penjelasan tentang Gambar 2.7:

a. Pihak *Counter*

Pihak *Counter* merupakan orang yang mengoperasikan aplikasi. *User* adalah pegawai dari Bakoelan *HP* yang biasa melakukan proses pendataan dan transaksi.

b. Komputer

Komputer merupakan sebuah perangkat keras yang digunakan oleh pegawai Bakoelan HP sebagai alat untuk menjalankan banyak aplikasi. Komputer digunakan saat sedang menjalankan aplikasi. Sebelum dijalankan, aplikasi harus diinstal terlebih dahulu di dalam komputer.

c. Perangkat Lunak

Melalui aplikasi, pemilik dapat melakukan beberapa aktivitas, yaitu menginputkan data, menghapus data, mencari data, melakukan transaksi penjualan dan pembelian, dan mencetak laporan.

d. Basis Data

Basis data digunakan untuk menampung data-data yang telah dimasukkan oleh pihak *counter*.

e. Laporan

Laporan merupakan hasil (*output*) dari aplikasi. *Report* yang diharapkan dari aplikasi dapat berupa arsip transaksi simpanan, laporan transaksi penjualan dan transaksi pembelian yang dapat otomatis diarsipkan pada setiap bulan.