

Aplikasi Terintegrasi Toko Swalayan

Elisabet Setiawan.¹, Erwin Suryana²

¹ *Staf Pengajar Jurusan S-1 Teknik Informatika*

² *Mahasiswa Jurusan S-1 Teknik Informatika*

Fakultas Teknologi Informasi

Universitas Kristen Maranatha

Jl. Prof. Drg. Suria Sumantri No. 65, BANDUNG 40164

*Email : elisabet.setiawan@eng.maranatha.edu,
erwin_suryan4@yahoo.com*

Abstract

Nowadays people become more and more aware of the fact that information technology has become a vital part in all aspects of life, especially in businesses. It has caused many business people to react fast in adopting a computerized system in advancing their businesses and competing with their competitors. Especially through internet, not only do businesses compete locally but also compete globally. Therefore, if businesses do not adopt the computerized system, sooner or later they will have a huge loss in terms of profit and efficiency.

Businesses, such as mini markets, also need a computerized system to deal with day-to-day activities, such as selling, ordering, and purchasing, so that adding, searching, editing, dan deleting data become easier and more efficient in terms of time and cost. This computerized application is intended to overcome many difficulties encountered by the manual system.

If this prototype will be implemented in the future, hopefully it can be useful for mini markets.

Keyword : mini markets, integrated application, computerized system, ordering, purchasing, selling

1. Pendahuluan

Sekarang ini, perkembangan teknologi informasi dan komunikasi telah sedemikian pesat sehingga mulai dirasakan bahwa keberadaan komputer dalam membantu kehidupan sehari-hari, terutama dalam dunia bisnis, sangat diperlukan. Dengan adanya sistem yang terkomputerisasi, setiap proses bisnis dapat dilakukan dengan lebih cepat, tepat, dan efisien. Data-data yang diperlukan juga dapat tersimpan dan terolah dengan lebih baik.

Aplikasi tersebut merupakan program *point of sale* (penjualan) yang memiliki fasilitas menu yang sederhana, khusus digunakan untuk mengelola administrasi stock, maupun keuangan secara komputerisasi pada toko kecil

atau menengah. Aplikasi tersebut juga dapat melakukan pencatatan dan perhitungan secara otomatis atas kegiatan transaksi - transaksi penjualan, pembelian, mengelola stock barang, data supplier, dan juga menyediakan laporan pendapatan. Diharapkan, aplikasi tersebut dapat semaksimal mungkin mengurangi proses kerja secara manual sehingga para pemilik toko swalayan berskala kecil-menengah dapat melakukan kegiatan bisnisnya dengan lebih efisien dalam segi waktu dan biaya, serta memudahkan dalam melakukan pemeriksaan terhadap stok barang dan transaksi pembelian/penjualan.

2. Software Yang Digunakan

Software yang digunakan dalam pembuatan aplikasi ini adalah :

a. Microsoft Office Access 2003

Digunakan untuk menyimpan data barang, stok, pegawai, dan transaksi penjualan/pembelian. Aplikasi tersebut menggunakan Microsoft Access, bukan SQL Server 2000, karena jumlah pemakai aplikasi tersebut terbatas sehingga pengaksesan ke database tidak terlalu banyak dan jumlah data yang disimpan masih dalam jumlah yang dapat diolah secara maksimal oleh Microsoft Access.

b. Microsoft Visual Studio .Net 2003 dengan bahasa pemrograman C# Windows Application

Aplikasi ini menggunakan bahasa pemrograman C# karena teknologi .Net Framework telah banyak (populer) digunakan dalam dunia bisnis dibandingkan dengan Java untuk saat ini.

3. Metode Analisis Sistem

Aplikasi ini dianalisis dengan menggunakan metode Unified Modelling Language (UML). Dalam UML terdapat banyak diagram yang digunakan untuk menganalisis suatu sistem aplikasi, antara lain:

a. Activity Diagram

Menggambarkan aliran kerja (workflow) bisnis dan operasional dari suatu sistem.

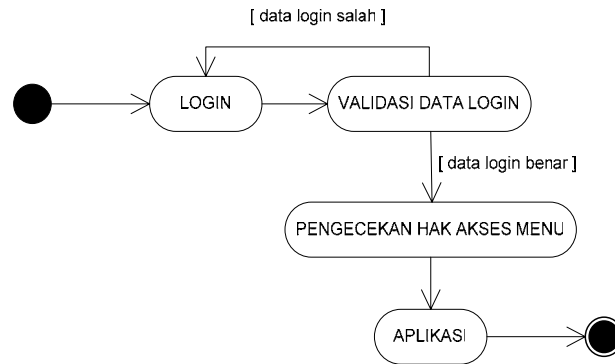
b. Sequence Diagram

Menggambarkan urutan pesan (messages) dari sistem. Tanda garis putus-putus vertikal menunjukkan garis waktu (timeline). Tanda panah menggambarkan pesan-pesan dari aktor atau objek ke objek lainnya.

c. Use Case Diagram

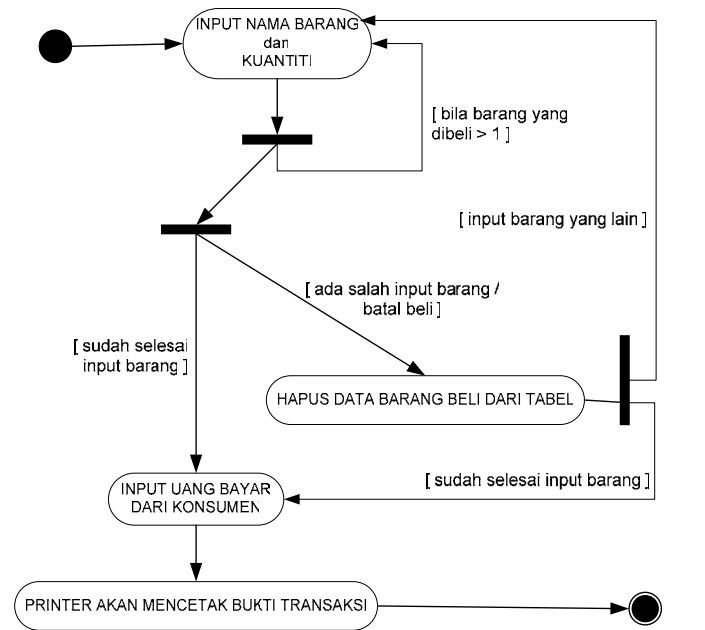
Use Case Diagram menggambarkan mengenai fungsionalitas dari suatu sistem yang sedang dibuat

4. Analisis dan Desain Sistem
4.1. Analisis UML
a. Activity Diagram



Gambar 1. Activity Diagram Login

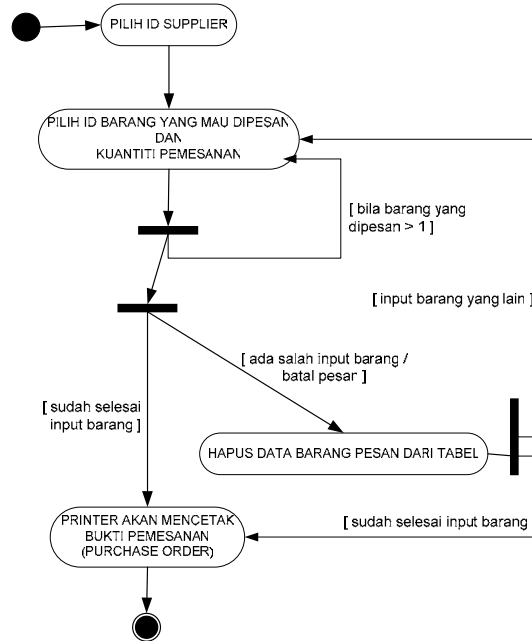
Untuk menggunakan aplikasi tersebut, *user* harus memasukkan data *login* (*username* dan *password*) yang valid. Apabila data *login* salah, *user* akan kembali ke proses *login*. Jika data *login* benar, sistem akan memeriksa hak akses menu yang dimiliki oleh *user* tersebut dan *user* masuk ke dalam aplikasi.



Gambar 2. Activity Diagram Penjualan

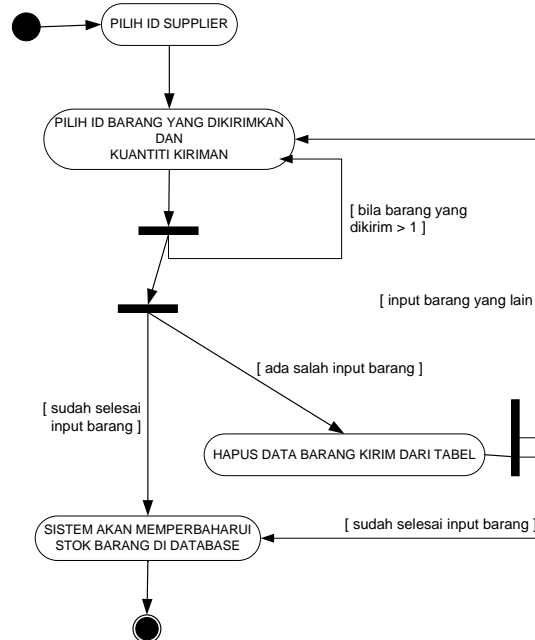
User (dalam hal ini kasir) memasukkan nama-nama barang beserta kuantitas yang dibeli oleh konsumen. Apabila kasir melakukan kesalahan dalam memasukkan data barang atau konsumen membatalkan pembelian suatu barang, maka data barang yang salah tersebut dapat dihapus.

Kemudian, kasir dapat menambah barang lain jika konsumen membeli lebih dari satu barang. Jika semua barang yang dibeli oleh konsumen sudah dimasukkan, kasir memasukkan jumlah uang yang dibayarkan oleh konsumen. Jumlah uang yang harus dikembalikan akan segera ditampilkan pada layar. Setelah itu printer akan mencetak bukti transaksi.



Gambar 3. Activity Diagram Pembelian (Purchase Order)

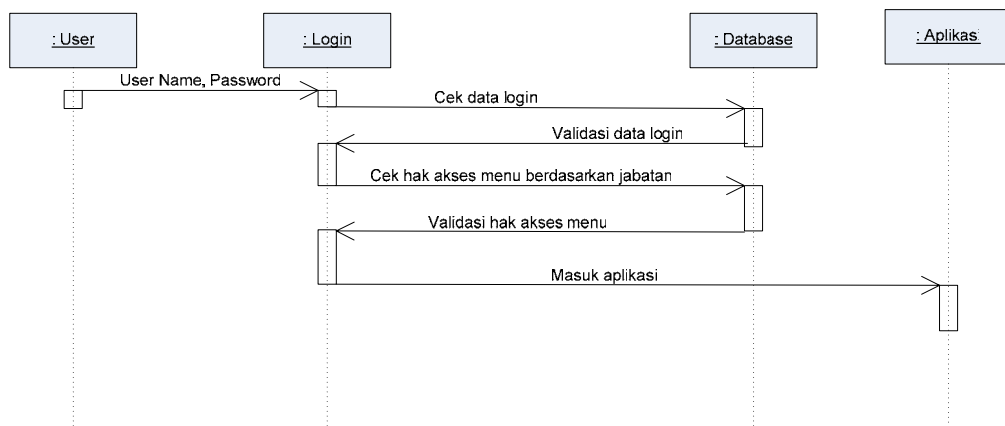
User memilih kode supplier yang mengirim barang lalu memilih kode barang yang akan dipesan beserta kuantitasnya masing-masing. Apabila user melakukan kesalahan dalam memasukkan data barang, maka data barang yang salah tersebut dapat dihapus. User dapat memasukkan data barang lain. Setelah semua data barang yang akan dipesan dimasukkan, user dapat mencetak P.O (Purchase Order) sebagai bukti pemesanan kepada supplier.



Gambar 4. Activity Diagram Penerimaan (Receive Order)

User memilih kode *Supplier* yang mengirimkan barang lalu memilih kode barang yang akan dikirimkan beserta kuantitasnya masing-masing. Apabila user melakukan kesalahan dalam memasukkan data barang, maka data barang yang salah tersebut dapat dihapus. User dapat memasukkan data barang lain. Terakhir, aplikasi akan memperbaharui jumlah stok barang dalam database.

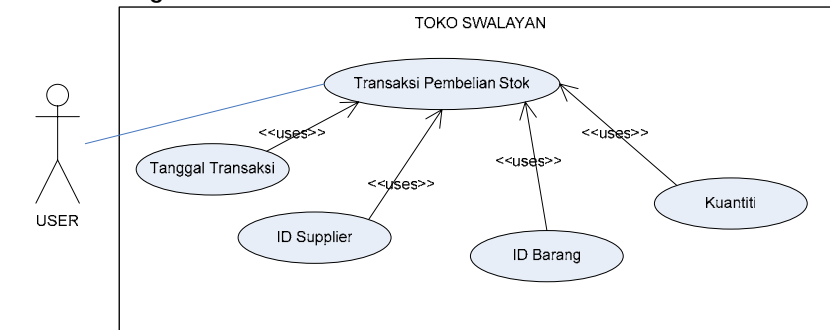
b. Sequence Diagram



Gambar 5. Sequence Diagram

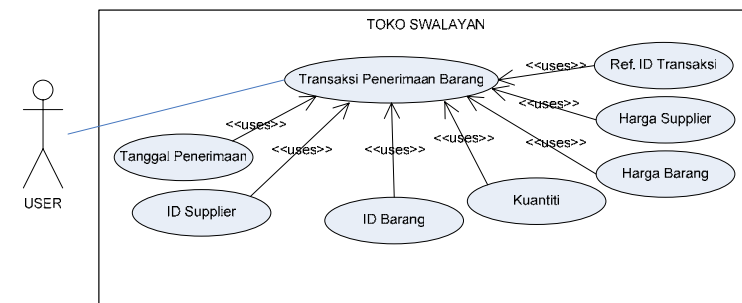
Untuk menggunakan aplikasi tersebut, *user* harus memasukkan *username* dan *password* yang valid. Sistem akan memeriksa data tersebut. Apabila data *login* benar, maka sistem akan memeriksa hak akses menu. Setelah itu, *user* baru dapat menggunakan aplikasi.

c. Use Case Diagram



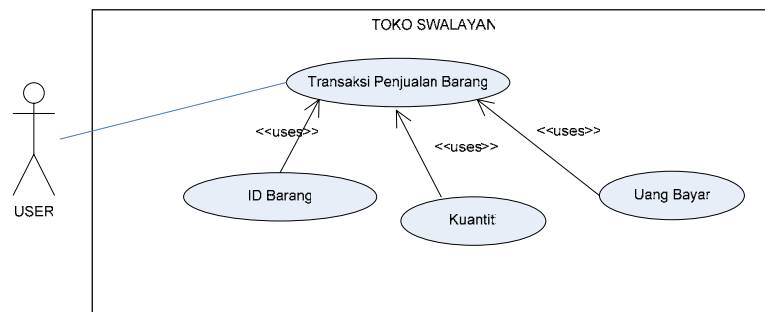
Gambar 6. Use Case Diagram Transaksi Pembelian Stok

Transaksi pembelian stok memerlukan data tanggal transaksi, kode supplier, kode barang, dan kuantitas pembelian.



Gambar 7. Use Case Diagram Transaksi Penerimaan Barang

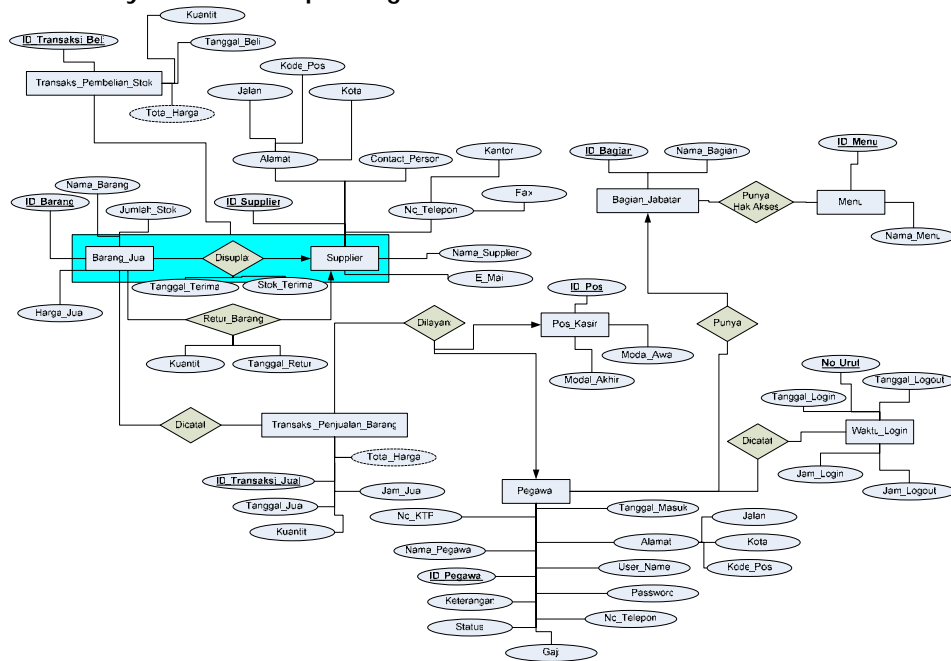
Transaksi penerimaan barang memerlukan data tanggal penerimaan, kode supplier, kode barang, kuantitas, harga barang, harga beli dari supplier, dan referensi kode transaksi.



Gambar 8. Use Case Diagram Transaksi Penjualan Barang

Transaksi penjualan barang memerlukan data kode barang, kuantitas, dan uang yang dibayarkan oleh konsumen.

4.2. Entity-Relationships Diagram



Gambar 6. E-R Diagram Aplikasi Terintegrasi Toko Swalayan

5. Perancangan Sistem

5.1. Transaksi Pembelian Stok

Form ini digunakan apabila mini market akan memesan barang kepada supplier. Setelah user memasukkan kode supplier, kode barang yang akan dipesan, maka sistem akan mencetak form Purchase Order (PO).

Gambar 7. Tampilan Form Aplikasi Transaksi Pembelian Stok

5.2. Transaksi Penerimaan Barang

Form ini digunakan apabila mini market menerima kiriman barang dari supplier. Setelah user memasukkan kode transaksi, kode supplier, kode barang yang dikirimkan, maka sistem akan memperbaharui jumlah stok di database.

Gambar 8. Tampilan Form Aplikasi Transaksi Penerimaan Barang

5.3. Transaksi Penjualan Barang

Form ini digunakan untuk transaksi penjualan barang kepada konsumen. Setelah *user* memasukkan kode barang-barang yang dibeli oleh konsumen, maka *user* harus memasukkan jumlah uang yang dibayarkan oleh konsumen. Terakhir, sistem akan mencetak bukti belanja dan memperbaharui jumlah stok barang di database.

Gambar 9. Tampilan Form Aplikasi Transaksi Penjualan Barang

6. Evaluasi Sistem

Hasil dari evaluasi menunjukkan bahwa aplikasi tersebut dapat menangani pekerjaan-pekerjaan yang dilakukan dalam sebuah mini market yang tidak memiliki cabang karena aplikasi ini bersifat stand-alone.

Simpulan

Untuk melakukan transisi sistem lama (manual) ke sistem baru (komputerisasi) terdapat 2 alternatif, yaitu :

- a. **Sistem lama dan sistem baru dijalankan bersama-sama sampai dengan tahap dimana sistem baru sudah bisa berdiri sendiri dan semua eror pada aplikasi sudah dapat terdeteksi.**
Cara ini dianjurkan bagi mini market yang sudah memiliki sistem komputerisasi tetapi masih terbatas dalam sisi *user interface* atau kemampuannya. Sistem baru harus mengadaptasi semua fungsionalitas yang sudah dijalankan pada sistem lama dan yang sudah menjadi standar dari toko swalayan tersebut. Data-data pada aplikasi lama harus dimigrasikan ke aplikasi baru.
- b. **Sistem baru langsung dipakai dan sistem lama langsung ditinggalkan**
Cara ini dianjurkan apabila mini market sebelumnya menggunakan cara manual (belum menggunakan sistem komputerisasi) dalam melakukan semua kegiatannya. Semua data yang ada, termasuk jumlah stok, data supplier, dan lain-lain dapat langsung dimasukkan ke dalam database.

Namun terdapat beberapa kendala dalam proses transisi ini, yaitu :

- c. **Penyesuaian jumlah stok barang pada sistem baru, dimana proses transaksi berjalan secara terus menerus sehingga jumlah stok akan selalu berubah-ubah.**
Solusi : Toko swalayan harus tutup selama 1 (satu) hari untuk menyesuaikan jumlah stok semua barang, sehingga keesokan harinya sistem baru dapat dioperasikan secara maksimal.
- d. **Pegawai yang sudah terbiasa dengan sistem lama.**
Solusi : Diadakan training yang intensif.
- e. **Spesifikasi sistem komputerisasi yang lama tidak memenuhi spesifikasi sistem komputerisasi yang baru.**
Solusi : Membeli spesifikasi komputer yang dibutuhkan oleh sistem komputerisasi yang baru. Jika spesifikasi komputer yang lama dapat di-*upgrade*, akan lebih baik.

DAFTAR PUSTAKA

Deitel, H. M., Deitel, P.J., Listfield, J., Nieto, T.R., Yaeger, C., & Zlatkina, M. (2002). *C# How To Program*. Upper Saddle River, NJ : Prentice Hall.

Febrian, Jack & Andayani, F. (2002). *Kamus Komputer dan Istilah Teknologi Informasi*. Bandung : Penerbit Informatika.

Nandu, S. *BookStock v2 - Samples to Show Simple OleDb.NET Connectivity*. (n.d). Retrieved May, 2005, from <http://www.masterC#.com>

Risal, L. (2002). *UML Tutor*. Bandung : Fakultas Teknologi Informasi, Universitas Kristen Maranatha.

Unified Modelling Language. (n.d). Retrieved May, 2005, from http://en.wikipedia.org/wiki/Unified_Modeling_Language