

Penerapan Economic Order Quantity dan SMS Gateway Pada Sistem Informasi CV. X

Maresha Caroline Wijanto

Jurusan S1 Teknik Informatika

Fakultas Teknologi Informasi, Universitas Kristen Maranatha

Jl. Prof. Drg. Suria Sumantri no. 65 Bandung 40164

email: maresha.caroline@yahoo.com

Abstract

The improvement in information technology has been increased day by day along with more requests for making life easier, more efficient, and secure. The transition of information system of CV. X from manual's system to a computerize will help the data management in CV. X. This application made for integrating data and help the owner to find an order's information with the use of EOQ (Economic Order Quantity) through SMS Gateway, so that the information directly known by the owner. With EOQ, we can also find how much an order that must be in order, how many times for an order in one period, dan the optimal's cost of an order. The report that made with an observation's method dan interview's technique to find out more about the requirement of this application, is hope can give the answer of all the questions that have not been answered and can be a reference for other application.

Keywords : data management, EOQ, SMS Gateway, observation, and interview.

1. Pendahuluan

Latar Belakang

CV. X merupakan sebuah toko komputer yang bergerak di bidang penjualan aksesoris komputer serta jasa “*service hardware*”. Sekarang ini, CV. X ini sedang berkembang pesat. Pada awalnya CV. X masih menggunakan sistem manual dalam pengolahan datanya, seperti data barang, data konsumen, data *supplier*, data transaksi, dan sebagainya. Seiring dengan berkembangnya teknologi dan semakin banyaknya data yang harus diolah, membuat pihak CV. X kesulitan menangani pengolahan sekaligus pencarian data-data yang ada.

Kurangnya integrasi antar data yang ada juga menyebabkan adanya kemungkinan stok barang habis tanpa diketahui sebelumnya. Hal ini dapat mengurangi keuntungan yang didapat toko. Selain itu juga, penambahan data yang masih manual membuat proses bisnis yang ada menjadi terhambat. Berdasarkan pertimbangan di atas, maka akan dibuatlah suatu aplikasi yang diharapkan bisa membantu penanganan masalah tersebut.

Rumusan Masalah

Berdasarkan latar belakang yang ada, maka rumusan masalah yang akan dibahas adalah:

1. Bagaimana penerapan *Economic Order Quantity* untuk menentukan jumlah persediaan barang?
2. Bagaimana membuat sistem yang cocok dengan proses bisnis di CV. X?
3. Bagaimana implementasi penambahan jumlah barang menggunakan PDA (*Personal Digital Assistant*)?

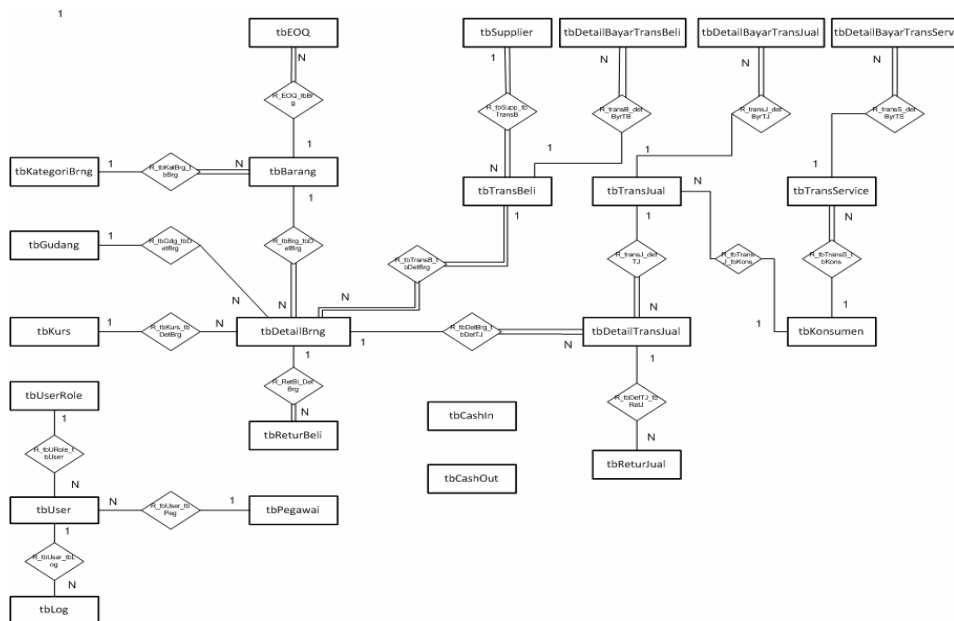
Tujuan Pembahasan

Sesuai dengan rumusan masalah yang ada, maka tujuan dari aplikasi ini adalah:

1. Mengatasi jumlah persediaan barang dengan menggunakan rumus *Economic Order Quantity* yang sudah disesuaikan dengan keadaan toko.
2. Membangun sebuah sistem informasi yang memudahkan pengguna dalam mengintegrasikan semua data yang ada.
3. Memudahkan proses penginputan jumlah barang melalui sistem informasi berbasis *mobile*.

2. Entity Relationship Diagram

Gambar untuk ERD berikut ini hanya terdiri dari entitas dan relasi sedangkan keterangan untuk atribut ada pada bagian bawah dan tidak digambarkan. Hal ini dikarenakan cukup kompleksnya atribut yang ada di dalam entitas yang tersedia.



Gambar 10 ERD

Berikut ini adalah skema dari ERD di atas:

- *tbUserRole* { id_role varchar(10), nama_role varchar(15), created_by varchar(10), created_date smalldatetime, modified_by varchar(10), modified_date smalldatetime ; PK : id_role }
- *tbPegawai* { id_pegawai varchar(10), nama_peg varchar(50), jns_klrmn varchar(10), alamat_peg varchar(50), jns_id char(10), no_id_peg char(20), tglLahir_peg smalldatetime, telp_peg char(20), gaji_peg money, rek_peg

- char(25), created_by varchar(10), created_date smalldatetime, modified_by varchar(10), modified_date smalldatetime ; PK : id_pegawai }
- *tbUser* { username varchar(10), password varchar(50), salt varchar(50), id_role varchar(10), id_pegawai varchar(10), created_by varchar(10), created_date smalldatetime, modified_by varchar(10), modified_date smalldatetime ; PK : username ; FK : id_role references tbUserRole, id_pegawai references tbPegawai }
 - *tbKurs* { id_kurs varchar(10), jns_kurs char(10), nilai_kurs money, created_by varchar(10), created_date smalldatetime, modified_by varchar(10), modified_date smalldatetime ; PK : id_kurs }
 - *tbKategoriBrng* { id_kat_brng varchar(10), nama_kat char(30), keterangan varchar(50), created_by varchar(10), created_date smalldatetime, modified_by varchar(10), modified_date smalldatetime ; PK : id_kat }
 - *tbGudang* { id_gudang varchar(10), nama_gdg varchar(30), alamat_gdg varchar(50), telp_gdg char(10) created_by varchar(10), created_date smalldatetime, modified_by varchar(10), modified_date smalldatetime ; PK : id_gudang }
 - *tbSupplier* {id_supplier varchar(10), nama_supp varchar(50), jns_klmn varchar(10), almt_supp varchar(50), telp_supp char(20), jns_id char(10), no_id_supp char(20), rek_supp char(25), prush_supp varchar(50), almt_prush varchar(50), telp_prush char(20), created_by varchar(10), created_date smalldatetime, modified_by varchar(10), modified_date smalldatetime ; PK : id_supplier }
 - *tbKonsumen* { id_konsumen varchar(10), nama_kons varchar(50), jns_klmn varchar(10), almt_kons varchar(50), telp_kons char(20), jns_id char(10), no_id_kons char(20), rek_kons char(25), prush_kons varchar(50), almt_prush varchar(50), telp_prush char(10), created_by varchar(10), created_date smalldatetime, modified_by varchar(10), modified_date smalldatetime ; PK : id_konsumen }
 - *tbCashOut* { id_cash_out varchar(10), kebutuhan varchar(50), grand_total money, tanggal smalldatetime, created_by varchar(10), created_date smalldatetime ; PK : id_trans_lain }
 - *tbCashIn* { id_cash_in varchar(10), kebutuhan varchar(50), grand_total money, tanggal smalldatetime, created_by varchar(10), created_date smalldatetime ; PK : id_trans_lain }
 - *tbTransBeli* { id_trans_beli varchar(10), id_supplier varchar(10), tanggal smalldatetime, grand_total money, status varchar(10), created_by varchar(10), created_date smalldatetime, modified_by varchar(10), modified_date smalldatetime ; PK : id_trans_beli ; FK : id_supplier references tbSupplier }
 - *tbDetailBayarTransBeli* { id_det_bayar_trans_beli varchar(10), id_trans_beli varchar(10), bayar money, created_by varchar(10), created_date smalldatetime; PK : id_det_bayar_trans_beli ; FK : id_trans_beli references tbTransBeli }
 - *tbTransJual* { id_trans_jual varchar(10), id_konsumen varchar(10), tanggal smalldatetime, grand_total money, status varchar(10), created_by varchar(10), created_date smalldatetime, modified_by varchar(10), modified_date

smalldatetime ; PK : id_trans_jual ; FK : id_konsumen references tbKonsumen }
}

- *tbDetailTransJual* { id_det_trans_jual varchar(10), id_trans_jual varchar(10), id_det_brng varchar(10), jumlah int, total_harga money, created_by varchar(10), created_date smalldatetime, modified_by varchar(10), modified_date smalldatetime ; PK : id_det_trans_jual ; FK : id_trans_jual references tbTransJual, id_det_brng references tbDetailBarang }
- *tbDetailBayarTransJual* { id_det_bayar_trans_jual varchar(10), id_trans_jual varchar(10), bayar money, created_by varchar(10), created_date smalldatetime; PK : id_det_bayar_trans_jual ; FK : id_trans_jual references tbTransJual }
- *tbTransService* { id_trans_serv varchar(10), id_konsumen varchar(10), jns_serv char(20), nm_brng varchar(25), grand_total money, tgl_skrng smalldatetime, tgl_sls smalldatetime, status varchar(10), created_by varchar(10), created_date smalldatetime, modified_by varchar(10), modified_date smalldatetime ; PK : id_trans_serv ; FK : id_konsumen references tbKonsumen }
- *tbDetailBayarTransServ* { id_det_bayar_trans_serv varchar(10), id_trans_serv varchar(10), bayar money, created_by varchar(10), created_date smalldatetime; PK : id_det_bayar_trans_serv ; FK : id_trans_serv references tbTransServ }
- *tbBarang* { id_barang varchar(10), id_kat_brng varchar(10), nama_barang varchar(30), hrg_jual money, hrg_jp money, stok int, created_by varchar(10), created_date smalldatetime, modified_by varchar(10), modified_date smalldatetime ; PK : id_barang ; FK : id_kat_brng references tbKategoriBrng }
- *tbDetailBarang* { id_det_brng varchar(10), id_barang varchar(10), id_trans_beli varchar(10), id_gudang varchar(10), id_kurs varchar(10), nama_det_brng varchar(30), jumlah int, hrg_dlr money, hrg_beli money, created_by varchar(10), created_date smalldatetime, modified_by varchar(10), modified_date smalldatetime ; PK : id_detail_brng ; FK : id_barang references tbBarang, id_trans_beli references tbTransBeli, id_gudang references tbGudang, id_kurs references tbKurs }
- *tbEOQ* { id_eoq varchar(10), id_barang varchar(10), pemesanan money, penyimpanan money, permintaan int, eoq int, jumlah_pemesanan int, lead_time int, rop int ; PK : id_eoq, FK : id_barang references tbBarang }
- *tbReturJual* { id_retur_jual varchar(10), id_det_trans_jual varchar(10), tanggal smalldatetime, alasan varchar(50), status bit, created_by varchar(10), created_date smalldatetime, modified_by varchar(10), modified_date smalldatetime ; PK : id_retur_jual ; FK : id_det_trans_jual references tbDetailTransJual }
- *tbReturBeli* { id_retur_beli varchar(10), id_tagihan varchar(10), tanggal smalldatetime, alasan varchar(50), status bit, created_by varchar(10), created_date smalldatetime, modified_by varchar(10), modified_date smalldatetime ; PK : id_retur_beli ; FK : id_tagihan references tbTagihan }
- *tbLog* { id_log varchar(10), username varchar(10), tanggal smalldatetime, ket varchar(50) ; PK : id_log }

3. Use Case

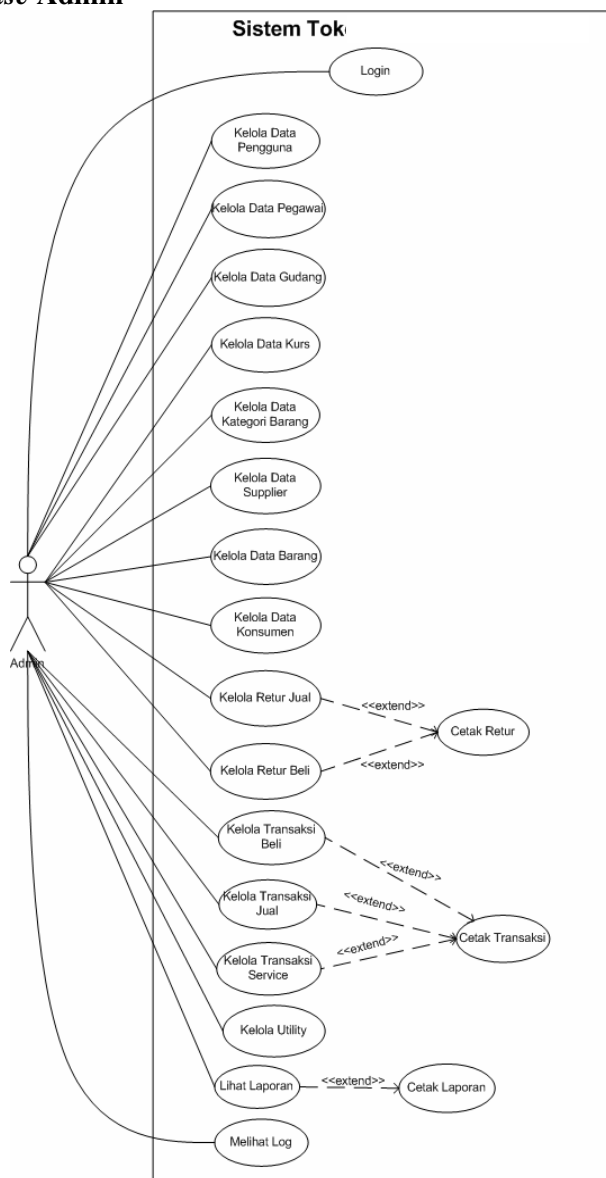
- **Use Case Kasir**



Gambar 11 Use Case Kasir

Kasir dapat melakukan *login*, ubah data pengguna, dan kelola data konsumen, *supplier*, retur jual, retur beli, transaksi jual, transaksi *service*, dan transaksi beli. Kelola data terdiri dari menambah, mengubah, dan melihat data. Untuk data transaksi perubahan hanya untuk pembayaran. Data retur dan transaksi dapat dicetak.

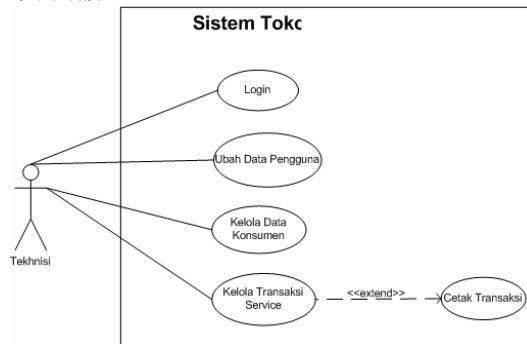
• Use Case Admin



Gambar 12 Use Case Admin

Admin dapat melakukan *login*, kelola data pengguna, pegawai, gudang, kurs, kategori barang, *supplier*, barang, konsumen, retur jual, retur beli, transaksi beli, transaksi jual, transaksi *service*, *utility*, lihat laporan, dan lihat *log*. Kelola data terdiri dari menambah, mengubah, dan melihat data. Untuk data transaksi pengubahan hanya untuk pembayaran. Untuk data retur dan utility tidak dapat diubah. Laporan, retur dan transaksi juga dapat dicetak.

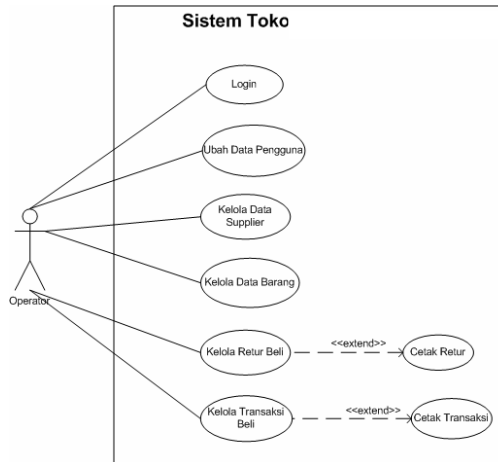
- **Use Case Tekhnisi**



Gambar 13 Use Case Tekhnisi

Tekhnisi dapat melakukan *login*, ubah data pengguna, kelola konsumen, dan transaksi *service*. Kelola data terdiri dari menambah, mengubah, dan menghapus. Untuk transaksi perubahan hanya untuk pembayaran. Data transaksi juga dapat dicetak.

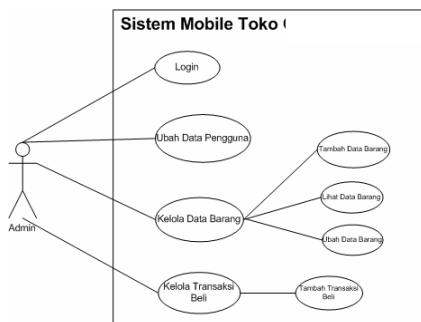
- **Use Case Operator**



Gambar 14 Use Case Operator

Operator dapat melakukan *login*, ubah data pengguna, kelola data *supplier*, barang, retur beli, dan transaksi beli. Kelola data terdiri dari menambah, mengubah, dan melihat data. Untuk transaksi perubahan hanya untuk pembayaran, sedangkan retur tidak dapat diubah. Retur dan transaksi juga dapat dicetak.

- **Use Case Mobile**



Gambar 15 Use Case Mobile

4. Proses *Economic Order Quantity (EOQ)* dan *SMS Gateway*

Aplikasi ini menggunakan EOQ untuk mengatur persediaan barang. Untuk barang-barang tertentu manajemen persediaannya diatur dengan menggunakan rumus EOQ. EOQ merupakan jumlah pembelian yang paling ekonomis untuk setiap kali pembelian/pemesanan. Yang dimaksud paling ekonomis adalah jumlah pembelian/pemesanan yang disertai dengan jumlah biaya yang paling rendah.

Notasi yang digunakan untuk memperoleh nilai EOQ, adalah:

- * Q = jumlah barang setiap pemesanan
- * Q* = jumlah optimal barang per pemesanan (EOQ)
- * D = permintaan tahunan barang persediaan dalam unit
- * S = biaya pemasangan atau pemesanan setiap pesanan
- * H = biaya penahan atau penyimpanan per unit per tahun

Berdasarkan notasi-notasi tadi, berikut adalah penentuan rumus EOQ:

1. Biaya pemasangan tahunan = $\frac{D}{Q} S$
2. Biaya penyimpanan tahunan = $\frac{D}{2} H$
3. Biaya pemasangan = biaya penyimpanan
 $\frac{D}{Q} S = \frac{D}{2} H$
4. Untuk mendapatkan Q*
 $2 D S = Q^2 H$
 $Q^2 = \frac{2DS}{H}$
 $Q^* = \sqrt{\frac{2DS}{H}}$
5. Jumlah pemesanan dalam satu tahun (N)
 $\frac{\text{Permintaan}}{\text{jumlah unit yang dipesan}} = \frac{D}{Q}$
6. Waktu antar pemesanan (T)
 $\frac{\text{jumlah hari kerja per tahun}}{\text{jumlah pemesanan dalam satu tahun}}$
7. Biaya total persediaan

= Biaya pemesanan + biaya penyimpanan

$$= \frac{D}{Q} S + \frac{Q}{2} H$$

8. Titik pemesanan ulang (*Reorder Point*)

$$ROP = d \times L$$

9. Jika ada stok pengaman atau *buffer* stok, maka:

ROP = (d x L) + *buffer* stock, dengan:

$$d = \text{permintaan per hari} = \frac{D}{\text{jumlah.hari kerja per tahun}}$$

L = *lead time*

Dengan rumus tersebut, dapat ditentukan batas minimum persediaan barang dan jumlah pemesanan yang optimal jika persediaannya telah mencapai batas tersebut.

Sistem akan mengirimkan sebuah pesan melalui SMS *Gateway* kepada *owner* untuk melakukan pemesanan sesuai jumlah yang telah ditentukan dengan rumus EOQ. Jika *owner* tidak berada di tempat, dapat melakukan pemesanan melalui aplikasi mobile yang disediakan juga.

5. Hasil Penelitian

Aplikasi ini dibuat dalam versi *desktop* dan *mobile*. Versi *desktop* dapat diakses oleh admin, kasir, *teknisi*, dan *operator*. Aplikasi ini hanya dapat digunakan di jaringan lokal toko.

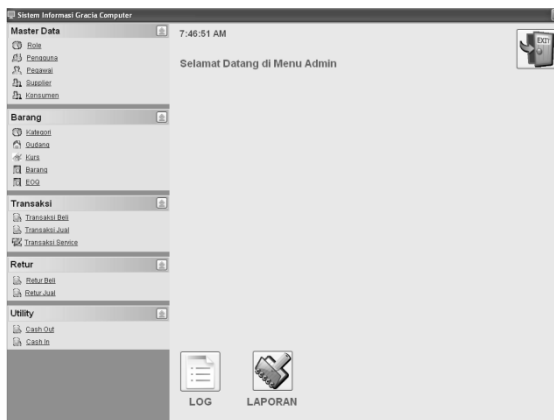
5.1 Aplikasi Desktop

Dalam aplikasi ini, pertama-tama admin atau kasir atau *teknisi* atau *operator* diharuskan untuk melakukan *login* terlebih dahulu, dengan mengisi kode pengguna dan kata kunci, sebelum masuk ke dalam aplikasi.



Gambar 16 Halaman *Login*

Jika proses *login* berhasil maka tampilan utama akan tergantung sesuai dengan hak akses pengguna. Tampilan utama tiap pengguna dapat dilihat di gambar berikut.



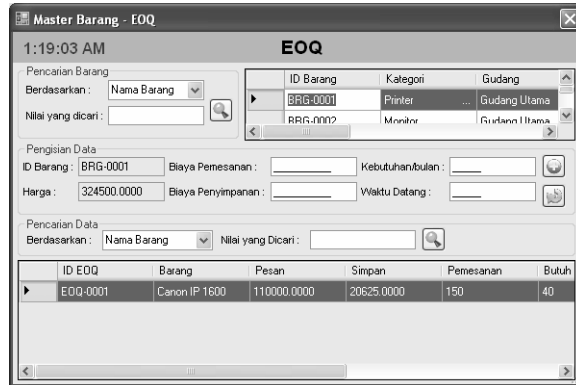
Gambar 17 Halaman Utama Admin

Gambar di atas adalah tampilan halaman utama untuk admin, untuk role lainnya seperti operator, kasir, dan teknisi hanya berbeda di pilihan menyanya saja.



Gambar 18 Master Data Konsumen

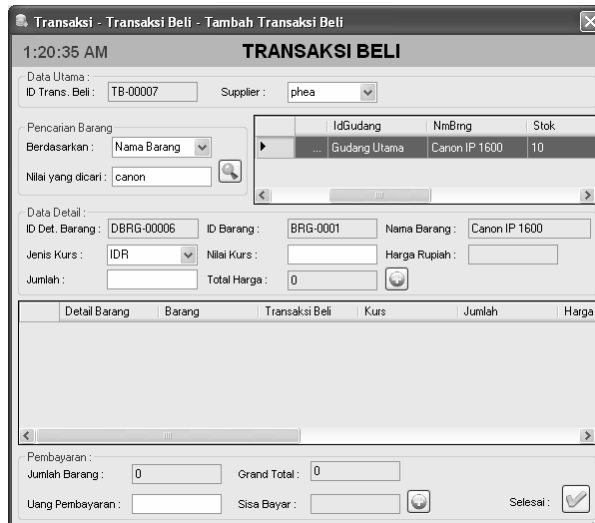
Gambar di atas adalah contoh dari master data konsumen. Bagian atas tengah terdapat judul menu, dan di sebelah kirinya terdapat jam *digital*. Di bagian tengah kiri selalu terdapat pencarian data berdasarkan kategori tertentu, disesuaikan dengan fitur tiap menu. Bagian bawahnya terdapat data yang sudah ada di *database*.



Gambar 19 EOQ

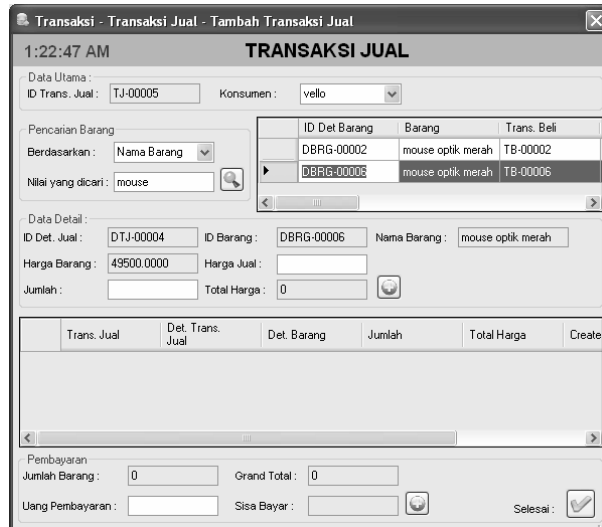
Di menu ini, pengguna dapat memilih barang mana saja yang akan diterapkan rumus *Economic Order Quantity* (EOQ) lalu pengaturan biaya yang juga diperlukan dalam perhitungan EOQ itu sendiri.

Di bawah ini akan ditampilkan transaksi yang dapat dilakukan, yaitu transaksi beli, transaksi jual, dan transaksi service. Khusus untuk transaksi pembelian dapat dilakukan dengan mata uang lain selain IDR yang telah ditentukan terlebih dahulu. Pengguna dapat melakukan pembayaran lebih dari satu kali untuk tiap transaksi, dan datanya semua tercatat.



Gambar 20 Transaksi Pembelian

Terdapat pencarian data barang berdasarkan kategori tertentu. Hasil pencarian akan tampil di bagian kanan pencarian dan data barang tadi dapat dipilih untuk menjadi detail transaksi pembelian.



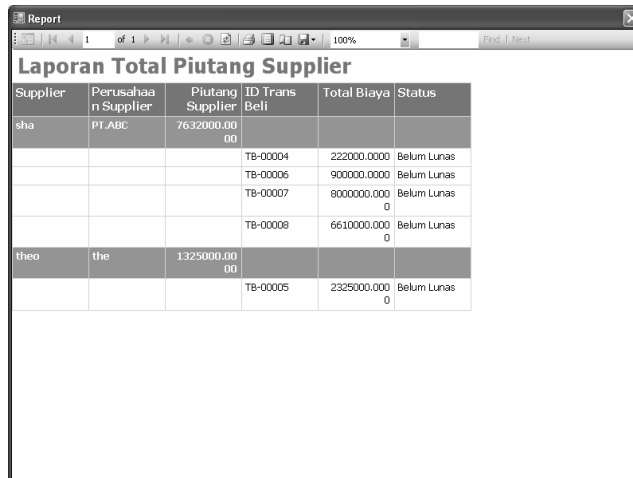
Gambar 21 Transaksi Penjualan

Sistem penjualannya bebas tergantung pengguna memilih barang yang mana, karena itu disediakan pencarian data barang. Dimana hasil pencarian akan tampil di bagian kanan dan data yang dipilih akan menjadi detail transaksi penjualan.



Gambar 22 Transaksi Service

Menu transaksi service ini merupakan gabungan dari tambah maupun pelunasan transaksi. Data detail pembayaran ada di bagian bawah jika salah satu data di bagian atas sudah terpilih.



Supplier	Perusahaan Supplier	Piutang Supplier	ID Trans Beli	Total Biaya	Status
sha	PT.ABC	7632000.00			
			TB-00004	222000.0000	Belum Lunas
			TB-00006	900000.0000	Belum Lunas
			TB-00007	8000000.0000	Belum Lunas
			TB-00008	6610000.0000	Belum Lunas
theo	the	1325000.00			
			TB-00005	2325000.0000	Belum Lunas

Gambar 23 Laporan Total Piutang Supplier

Di atas adalah contoh tampilan laporan yang dapat di generate oleh sistem.

5.2 Aplikasi Mobile

Untuk versi *mobile* hanya bisa diakses oleh admin. Versi ini tidak selengkap dengan versi *desktop* karena hanya untuk pengolahan data barang. Keuntungan aplikasi ini dapat diakses dari luar area lokal jaringan toko.



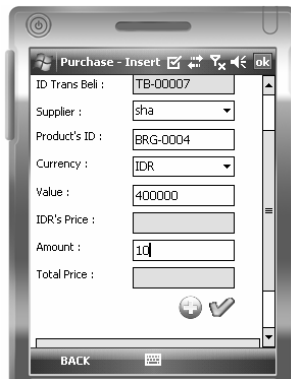
Gambar 24 Halaman Login - Mobile

Pengguna dapat memasukkan *username* dan *password* untuk dapat *login* ke dalam sistem ini.



Gambar 25 Halaman Utama - *Mobile*

Dari halaman utama, pengguna dapat memilih ingin melakukan fitur yang mana dengan menekan salah satu gambar yang ada.



Gambar 26 Transaksi Pembelian

Tampilan transaksi pembelian di menu mobile dibuat sesederhana mungkin untuk memudahkan pengguna.

6. Testing dan Evaluasi

- Pengguna memasukan data tidak lengkap saat memasukan maupun mengubah maka akan diberi pesan data tidak lengkap
- Pengguna memasukan nilai huruf atau angka negatif maka akan muncul pesan nilai harus angka dan bernilai positif
- Pengguna memasukan jumlah lebih besar dari stok yang dimiliki maka akan muncul pesan stok tidak mencukupi.
- Pengguna memasukan username yang sudah digunakan pengguna lain maka akan muncul pesan username sudah digunakan
- Pengguna memasukan password tidak sama dengan confirm password maka akan muncul pesan password tidak sama

7. Kesimpulan

Kesimpulan yang bisa diperoleh melalui beberapa evaluasi yang ada adalah:

1. Aplikasi ini dapat membantu *owner* untuk menentukan waktu pemesanan dan jumlah pemesanan yang terbaik bagi toko.
2. Aplikasi ini dapat menyatukan proses bisnis yang terjadi di toko, sehingga semua data yang ada menjadi lebih terorganisir.
3. Aplikasi ini dapat memudahkan proses penambahan barang melalui PDA. Aplikasi ini memberikan *error handling* yang dapat memberitahu pengguna jika salah memasukkan data sebelum data tersebut masuk ke *database*.

Daftar Pustaka

- [Dhe04] Dennis, Allan, & Haley, Barbara, W. (2004). *Systems Analysis & Design* (3rd ed.). USA : John Wiley & Sons, Inc.
- [Mar06] Marshall, D. (2006). *Programming Microsoft Visual C# 2005: The Language*. Microsoft: Washington.
- [Rog01] Roger.T, Burlton. (2001) . *Management Process Business - Profiting from Process*. Indiana : Sams Publishing.
- [Sch04] Schmuller, Joseph. (2004). *Sams Teach Yourself UML in 24 Hours* (3rd ed.). Indiana : Sams Publishing.
- [Vie07] Vieira, Robert. (2007). *Professional SQL Server 2005 Programming*. Indiana : Wiley Publishing, Inc.
- [Wil06] Williams, Laurie. 2006. *Testing Overview and Black-Box Testing Techniques*.