

Penerapan Kontrol Stok dalam Sistem Informasi Dagang Dengan Metode Perpetual Inventory System

¹Tony Wijaya dan ²Irawan Wingdes

STMIK Pontianak, Jl. Merdeka No. 372 Pontianak, Kalimantan Barat

^{1,2}Program Studi Sistem Informasi

e-mail: ¹mail.tonywijaya@gmail.com , ²irawan.wingdes@gmail.com

Abstrak

Kontrol stok merupakan fitur yang kritis dalam sebuah sistem informasi dagang. Pada prakteknya, stok sulit untuk dikontrol secara manual, terlebih lagi jika jumlah produk yang dijual cukup bervariasi. Adanya kekosongan stok, lamanya transportasi dan penundaan pengiriman merupakan beberapa hambatan yang menyebabkan keterlambatan dalam mengisi stok di gudang. Kekosongan stok ini merupakan kerugian bagi perusahaan karena tidak mampu mengimbangi permintaan konsumen. Tujuan penelitian ini adalah untuk menerapkan kontrol stok dalam sistem informasi dagang dengan metode Perpetual Inventory System. Perpetual Inventory System merupakan metode kontrol stok yang mengupdate stok setiap kali ada transaksi, baik itu penjualan, retur penjualan, pembelian, ataupun retur pembelian. Metode ini dapat melacak posisi stok yang terjual dan stok yang masih ada sehingga menghindari kekosongan stok. Perancangan program pada penelitian ini menggunakan metode Agile dengan pendekatan Extreme Programming dan diuji dengan metode black box untuk memastikan fitur yang dibutuhkan sudah sesuai. Dengan adanya kontrol stok, perusahaan dapat terhindar dari kerugian karena kekosongan stok, meningkatkan daya saing yang pada akhirnya meningkatkan laba perusahaan.

Kata kunci : *Kontrol stok, Sistem Informasi Dagang, Perpetual Inventory System.*

Abstract

Stock control is a critical feature needed in trade information system. In practice, it is difficult to control physical stock manually, especially with the complexity and variety of products. Stockouts, transportation delay or late shipments are several challenges in managing company's stock control. Stockouts means loss in revenue for company because unmet customer demands means there are lost opportunities. The purpose of this research is to implement stock control in trade information system with Perpetual Inventory System method. Perpetual Inventory System is a stock control method which updates stock level each time transaction happens, such as sales, return of sales, purchases, or return of purchases. This method can track stocks of sales and remaining stocks so it can avoid stockouts / out of stock situation. Software is designed using Agile Development Method with Extreme Programming approach and tested with blackbox to ensure all required feature is working. With stock control, company can avoid loss of revenues caused by stockouts or out of stock, increase competitiveness and therefore increases revenue.

Keywords: *Stock control, Trade Information System, Perpetual Inventory System.*

1. PENDAHULUAN

Manajemen inventory telah menjadi dasar keputusan di perusahaan-perusahaan dan telah diteliti dengan mendalam di lingkungan akademik maupun korporasi. Pertanyaan utama dalam pada topik ini adalah kapan memesan, berapa jumlah pesanan dan seberapa banyak harus disimpan sebagai stok cadangan [1]. Manajemen inventory mencakup beberapa keputusan yang bertujuan untuk menyelaraskan permintaan dengan penawaran produk pada waktu maupun tempat dengan biaya dan tingkat layanan sesuai tujuan perusahaan [2].

Manajemen inventory yang baik akan mendapatkan banyak manfaat berupa – carrying cost yang rendah, menggunakan teknik ilmiah untuk mengurangi biaya, rasio persediaan/penjualan yang rendah, material dalam pengerjaan yang lebih sedikit, pencatatan sistematis, perputaran persediaan tinggi, biaya akibat kekusangan, kelebihan, kerusakan produk yang menurun, biaya penanganan material yang rendah, waktu pengiriman yang pendek, sistem informasi manajemen yang lebih baik, serta kepuasan layanan bagi konsumen dan pemasok.

Manajemen material yang tidak efektif dapat mengakibatkan salahnya pembelian, kekurangan barang dan ketidakpastian persediaan. Hal tersebut mengakibatkan penundaan produksi yang dapat membuat tanggal pengiriman meleset. Pemesanan yang terlambat juga mengakibatkan hal yang sama. Jika perusahaan memperbaiki masalah dengan pemesanan mendadak, biaya akan meningkat karena mengadakan barang dari vendor yang bersedia dan pengiriman cepat yang biasanya berbiaya tinggi. [3]

Untuk mengatur inventory, manajer menggunakan beberapa dokumen, salah satunya yang sangat penting adalah pencatatan inventory dengan metode perpetual. Metode perpetual merupakan salah satu dari dua sistem (perpetual dan periodik) untuk mengatur permintaan konsumen [4].

Metode perpetual merupakan metode pencatatan inventory dengan data yang selalu diperbaharui, biasanya dengan menggunakan perangkat komputer. Sistem ini mempunyai kontrol yang lebih baik pada inventory. Pada umumnya sistem inventory perpetual modern mencatat beberapa hal berikut ini: unit yang dibeli dan biayanya, unit terjual dan biayanya, jumlah inventory yang masih ada [5]. Metode perpetual lebih superior dibanding periodic karena menyimpan catatan penjualan dan tingkat inventory yang mencegah kehabisan stok [6].

Sebuah sistem informasi dagang umumnya memiliki fitur penjualan, pembelian dan kontrol stok. Di samping penjualan dan pembelian, fitur kontrol stok merupakan fitur yang kritis. Fitur ini tidak hanya sekedar mengurangi stok ketika terjadi transaksi penjualan, atau menambah stok ketika terjadi transaksi pembelian. Kontrol stok berarti sistem ini mampu memberikan informasi kepada bagian purchasing kapan harus mulai melakukan pembelian sehingga tidak terjadi kekosongan stok. Fitur ini juga harus dapat memberikan informasi jumlah pembelian yang optimal sehingga tidak kelebihan stok sehingga menyebabkan biaya untuk pemeliharaan stok meningkat. Terlebih apabila stok tersebut merupakan produk yang memiliki tanggal kadaluarsa cepat seperti makanan atau minuman.

Tujuan penelitian ini adalah untuk menerapkan metode perpetual pada kontrol stok di dalam sistem informasi dagang. Fitur utamanya adalah adanya peringatan atau notifikasi untuk melakukan order stok apabila produk tertentu sudah mencapai batas kuantitas tertentu. Fitur lainnya adalah saran untuk kuantitas pembelian yang optimal sehingga perusahaan tidak melakukan order yang terlalu sedikit ataupun terlalu banyak.

2. METODE PENELITIAN

Dalam penelitian ini menggunakan metode perancangan perangkat lunak Agile (*Agile Software Development*) yang merupakan kerangka konseptual untuk mengembangkan perangkat lunak yang memperkenalkan beberapa tahapan perulangan (iterasi) selama siklus hidup sebuah

proyek [7]. Karakteristik dari metode agile adalah ringan, skala kecil hingga menengah, kebutuhan (*requirement*) yang samar dan/atau sering berubah, perancangan yang sederhana, hingga sistem yang minimal untuk menyelesaikan proyek. Nilai yang dijunjung oleh Agile adalah:

1. Mengedepankan individu dan interaksi daripada proses dan alat
2. Mengedepankan perangkat lunak yang berfungsi sesuai kebutuhan daripada dokumentasi yang lengkap
3. Mengedepankan kolaborasi dengan pelanggan daripada negosiasi kontrak
4. Mengedepankan respon terhadap perubahan daripada mengikuti rancangan

Extreme Programming (XP) merupakan pendekatan dari Agile yang paling luas digunakan di seluruh dunia. Tahapan pada proses XP yaitu [8]:

1. Planning, merupakan proses di mana individu atau tim pengembang perangkat lunak mendengarkan kebutuhan-kebutuhan (*requirements*) dari pemakai tentang proses bisnis berjalan dan mengetahui output apa yang dibutuhkan serta fitur dan fungsionalitas umum yang akan direkayasa.
2. Design, XP melakukan tahapan rancangan dengan cara KIS (*Keep It Simple*) atau dengan kata lain dengan cara yang sesederhana mungkin. Rancangan tersebut berisi panduan implementasi sebuah proyek. Tidak lebih dan tidak juga kurang.
3. Coding, XP merekomendasikan *pair programming* yaitu 2 orang yang berada pada 1 komputer untuk menyelesaikan permasalahan pemrograman secara real-time. Individu pertama akan fokus pada detail teknis pemrograman, sementara individu lainnya fokus pada logika dan standar yang harus dipenuhi supaya tetap berada pada jalur rancangan sehingga hasilnya sesuai dengan yang diharapkan.
4. Testing, dalam tahapan ini penulis menggunakan pengujian blackbox untuk memastikan fitur yang akan dirancang dapat berjalan dengan baik.

Microsoft Visual Studio 2015 merupakan sebuah rangkaian tool untuk melakukan rekayasa perangkat lunak, dari fase perancangan melalui tampilan visual, kode program, pengujian, perbaikan error, analisis kualitas kode program dan performa, implementasi ke pelanggan, serta pengumpulan data pemakaian program [9]. Aplikasi ini berbasis desktop yang dirancang, dikode, dan diuji menggunakan Visual Studio 2015. Bahasa pemrograman C# dan dikompilasi untuk .NET Framework versi 4.0.

Basis data yang digunakan untuk sistem informasi dagang adalah basis data MySQL. Basis data MySQL merupakan Relational DataBase Management System (RDBMS) yang dapat digunakan untuk aplikasi berbasis web, *cloud*, mobile dan aplikasi *embedded* [10].

3. HASIL DAN PEMBAHASAN

Pencatatan per transaksi merupakan kunci untuk mengimplementasikan metode Perpetual dalam kontrol stok. Pada penelitian ini, penulis menggunakan sistem informasi dagang pada PT. Cemerlang Andalan Nusantara di Pontianak. Data yang ditampilkan pada penelitian ini merupakan data sampel. Di dalam sistem informasi dagang, pencatatan ini dilakukan setiap kali terjadi transaksi penjualan dan pembelian. Pada sistem informasi dagang, transaksi penjualan melibatkan tabel basis data sebagai berikut:

1. **header_penjualan**, yang berisi informasi header nota / faktur penjualan seperti nama pelanggan, tanggal, termin, dan lain-lain.
2. **detail_penjualan**, yang berisi informasi barang yang dijual, kuantitas, harga satuan, diskon, dan lain-lain.
3. **setoran_penjualan**, yang berisi informasi setoran untuk penjualan yang bersangkutan.
4. **stok**, mengurangi stok untuk barang yang disebutkan pada tabel **detail_penjualan**.
5. **kartu_stok**, proses pencatatan risalah penjualan.

Sedangkan untuk transaksi pembelian, tabel basis data yang terlibat adalah:

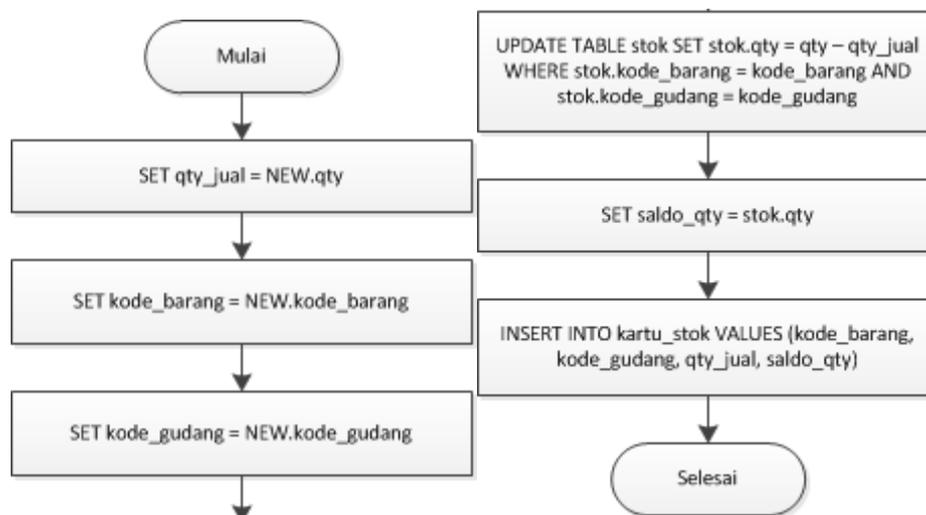
1. **header_pembelian**, yang berisi informasi header nota / faktur pembelian seperti nama supplier, tanggal, termin, dan lain-lain.
2. **detil_pembelian**, yang berisi informasi barang yang dibeli, kuantitas, harga satuan, diskon, dan lain-lain.
3. **setoran_pembelian**, yang berisi informasi setoran untuk pembelian yang bersangkutan.
4. **stok**, menambah stok untuk barang yang disebutkan pada table **detil_pembelian**.
5. **kartu_stok**, proses pencatatan risalah pembelian.

Pada sistem informasi dagang dengan menggunakan metode periodik, pencatatan belum dilakukan pada tabel **kartu_stok**. Tanpa pencatatan pada tabel tersebut di dalam basis data, sistem informasi dagang tidak dapat melakukan kontrol stok tepat setelah transaksi dilakukan. Untuk itu, penulis akan melakukan update pada sistem informasi dagang supaya memiliki fitur ini.

Adapun proses pencatatan pada tabel **kartu_stok** tidak membutuhkan perubahan pada coding program sistem informasi dagang. Proses ini dapat diotomatisasi dengan fitur *trigger* yang ada pada basis data MySQL. Sebuah *trigger* basis data adalah kode prosedural yang secara otomatis dieksekusi untuk merespon kejadian-kejadian tertentu pada tabel tertentu di dalam basis data [11]. Di dalam basis data, *trigger* didefinisikan pada tabel. Sebuah *trigger* dapat merespon pada beberapa kejadian berikut:

1. *BEFORE INSERT*, dieksekusi sebelum proses tambah record.
2. *AFTER INSERT*, dieksekusi setelah proses tambah record.
3. *BEFORE UPDATE*, dieksekusi sebelum proses ubah record.
4. *AFTER UPDATE*, dieksekusi setelah proses ubah record.
5. *BEFORE DELETE*, dieksekusi sebelum proses hapus record.
6. *AFTER DELETE*, dieksekusi setelah proses hapus record.

Untuk mencatat risalah transaksi penjualan ke dalam tabel **kartu_stok**, maka *trigger* yang diperlukan adalah *AFTER INSERT* yang didefinisikan pada tabel **detil_penjualan**. Berikut penulis sajikan flowchart untuk *trigger* tersebut:



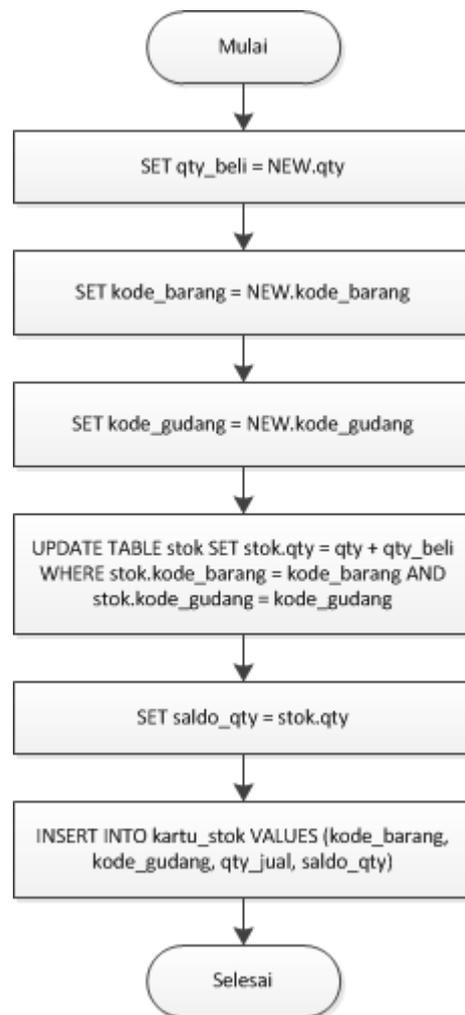
Gambar 1. Flowchart Trigger *AFTER_INSERT* pada tabel **detil_penjualan**.

Gambar 1 menunjukkan algoritma *trigger* basis data MySQL yang dieksekusi apabila terjadi transaksi penjualan. Proses tersebut dapat diuraikan sebagai berikut:

1. Mengambil kuantitas barang yang dijual dari record yang baru di-insert (*NEW.qty*) dan disimpan pada variabel *qty_jual*.

2. Mengambil kode barang dari record yang baru di-insert (`NEW.kode_barang`) dan disimpan pada variable `kode_barang`.
3. Mengambil kode gudang dari record yang baru di-insert (`NEW.kode_gudang`) dan disimpan pada variable `kode_gudang`.
4. Mutakhirkan stok di gudang, kurangi stok di gudang sebesar `qty_jual` untuk barang yang memiliki kode barang: `kode_barang` dan kode gudang: `kode_gudang`.
5. Mengambil stok barang setelah pengurangan pada nomor 4, dan disimpan pada variable `saldo_qty`.
6. Insert record baru ke dalam tabel `kartu_stok`.

Selain pada tabel **detil_penjualan**, *trigger* juga perlu didefinisikan pada tabel **detil_pembelian** untuk mencatat risalah transaksi pembelian ke dalam tabel `kartu_stok`. Flowchart untuk *trigger* tersebut relatif sama dengan flowchart untuk mencatat risalah transaksi penjualan, hanya saja pada transaksi pembelian menambah stok di gudang.

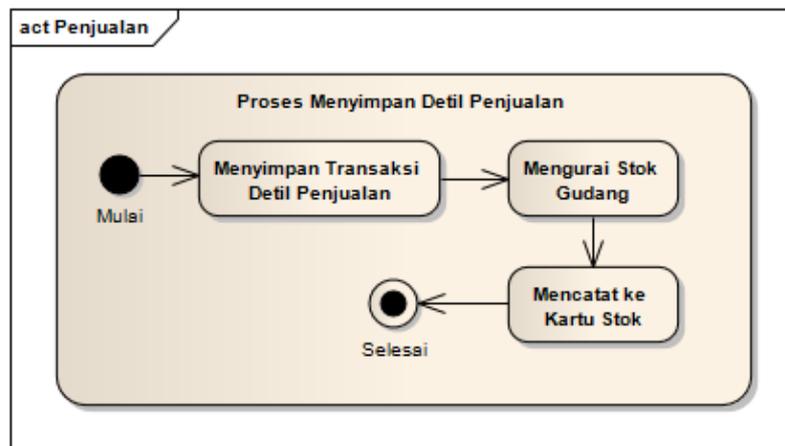


Gambar 2. Flowchart Trigger AFTER INSERT pada tabel `detil_pembelian`.

Gambar 2 menunjukkan algoritma trigger basis data MySQL yang dieksekusi apabila terjadi transaksi pembelian. Proses tersebut dapat diuraikan sebagai berikut:

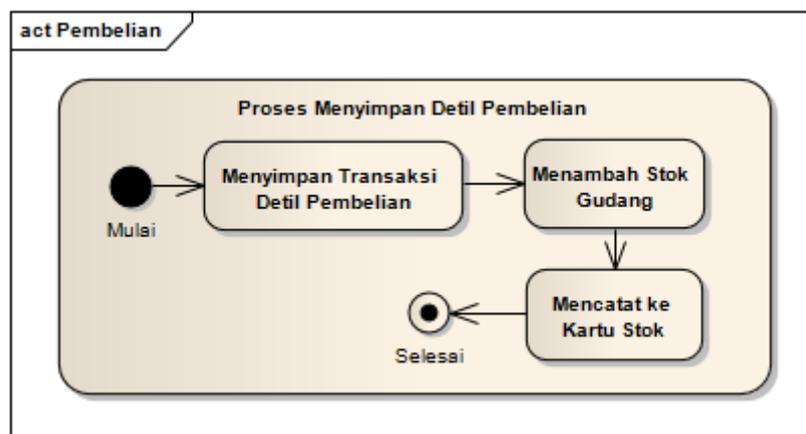
1. Mengambil kuantitas barang yang dibeli dari record yang baru di-insert (`NEW.qty`) dan disimpan pada variabel `qty_beli`.
2. Mengambil kode barang dari record yang baru di-insert (`NEW.kode_barang`) dan disimpan pada variable `kode_barang`.

3. Mengambil kode gudang dari record yang baru di-insert (NEW.kode_gudang) dan disimpan pada variable kode_gudang.
4. Mutakhirkan stok di gudang, tambahkan stok di gudang sebesar qty_beli untuk barang yang memiliki kode barang: kode_barang dan kode gudang: kode_gudang.
5. Mengambil stok barang setelah penambahan pada nomor 4, dan disimpan pada variable saldo_qty.
6. Insert record baru ke dalam tabel kartu_stok.



Gambar 3. Activity Diagram Proses Menyimpan Detil Penjualan.

Gambar 3 menunjukkan proses yang dilakukan pada saat menyimpan nota / faktur penjualan. Proses yang dilakukan tidak hanya menyimpan transaksi penjualan, tetapi juga mengurangi stok gudang dan mencatatnya ke dalam kartu stok. Kedua proses terakhir sudah dijelaskan pada gambar 1.

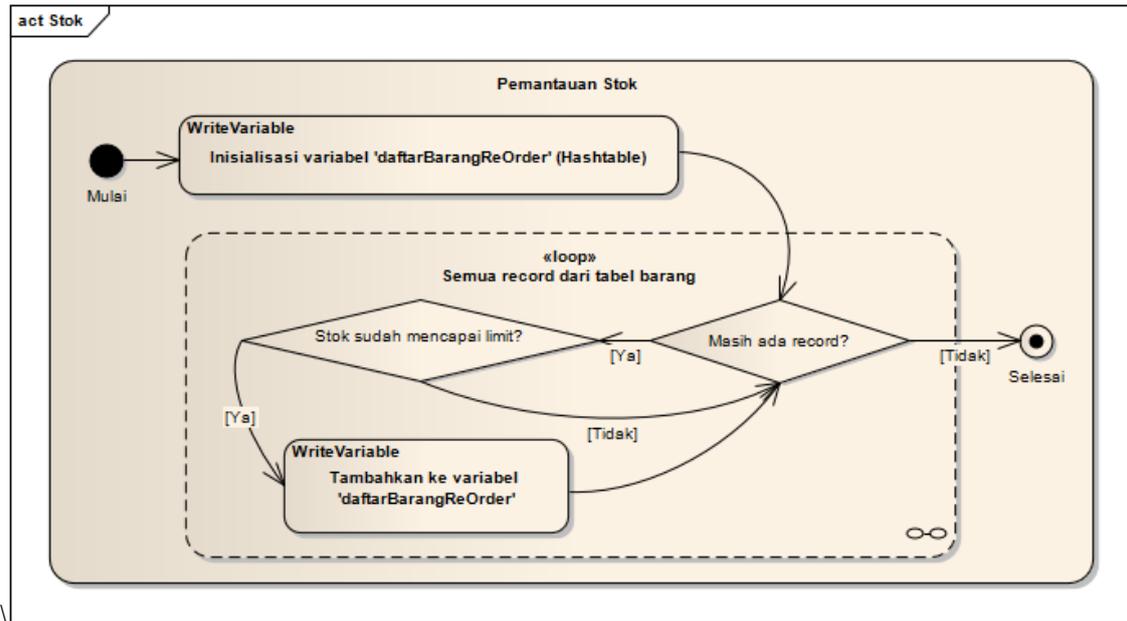


Gambar 4. Activity Diagram Proses Menyimpan Detil Pembelian.

Gambar 4 menunjukkan proses yang dilakukan pada saat menyimpan nota / faktur pembelian. Proses yang dilakukan tidak hanya menyimpan transaksi pembelian, tetapi juga menambah stok gudang dan mencatatnya ke dalam kartu stok. Kedua proses terakhir sudah dijelaskan pada gambar 2.

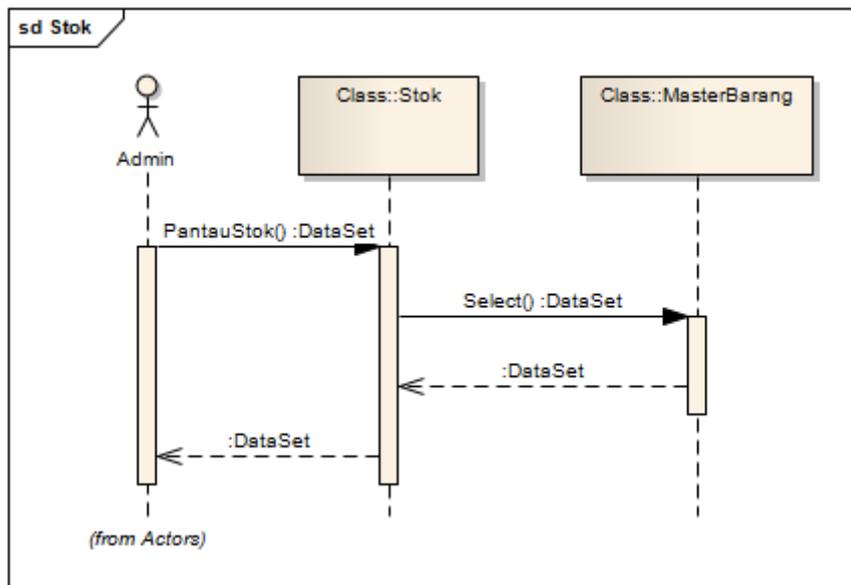
Setelah transaksi penjualan dan pembelian dicatat, maka selanjutnya adalah bagaimana sistem informasi dagang mengolah data tersebut menjadi informasi untuk keperluan kontrol stok. Pada sistem informasi dagang, penulis menambahkan sebuah modul untuk memantau kondisi stok setiap interval waktu tertentu. Interval ini dapat diatur oleh pemakai sesuai dengan kebutuhannya. Pada saat interval tersebut tercapai maka modul tersebut akan memeriksa stok produk apa yang telah mencapai limit bawah. Limit bawah yang dimaksud adalah posisi stok di

mana produk tersebut sudah harus dilakukan re-order. Activity diagram di bawah ini (Gambar 5) akan menjelaskan proses tersebut.



Gambar 5. Activity Diagram proses Pemantauan Stok.

Activity diagram proses pemantauan stok pada gambar 5 menjelaskan logika dan alur pemrograman untuk memantau produk mana yang sudah mencapai limit terbawah. Produk yang telah mencapai limit akan dimasukkan ke dalam variable 'daftarBarangReOrder' yang nantinya akan ditampilkan ke dalam peringatan.

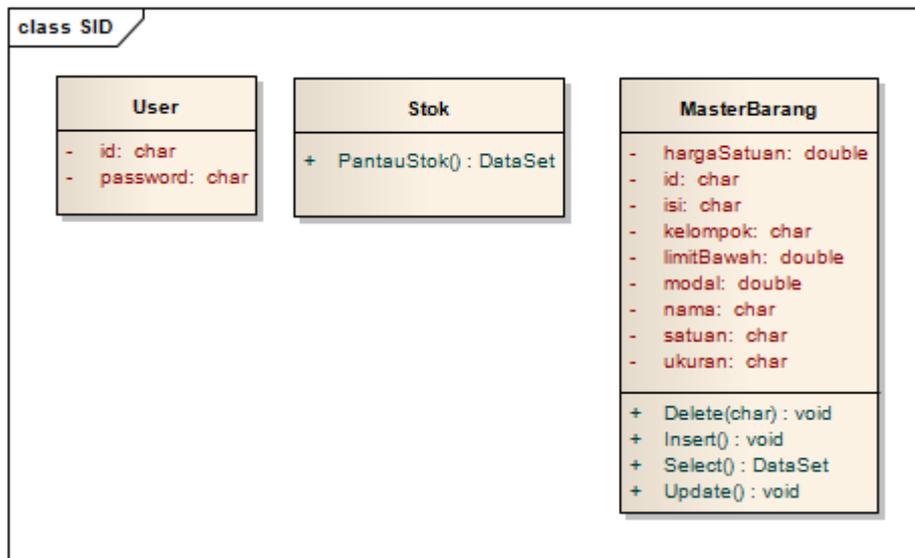


Gambar 6. Sequence Diagram Pemantauan Stok.

Pada gambar 6, penulis menyajikan Sequence Diagram yang menunjukkan interaksi antara admin dengan 2 class yaitu class Stok dan class MasterBarang. Urutan prosedur juga dapat dilihat pada diagram ini. Yang pertama yaitu faktoris memanggil fungsi PantauStok () tanpa parameter karena mengambil semua data barang / produk yang ada. Tahap selanjutnya, class Stok akan meminta data barang kepada class MasterBarang melalui fungsi Select () yang

mengembalikan nilai (*return value*) berupa *DataSet*. *DataSet* merupakan kumpulan record dalam .NET Framework yang berisi record-record dari basis data. *DataSet* tersebut kemudian diproses oleh class Stok melalui fungsi PantauStok (). Rincian proses pada fungsi ini sudah dijelaskan pada diagram Activity (gambar 5). Fungsi ini juga mengembalikan nilai dalam bentuk *DataSet*.

Class diagram digunakan untuk menampilkan beberapa kelas yang ada dalam sistem informasi dagang. *Class diagram* mendeskripsikan jenis-jenis objek dalam sistem. *Class diagram* menunjukkan properti dan operasi sebuah kelas dan batasan-batasan yang terdapat dalam hubungan-hubungan objek tersebut. *Class diagram* memiliki 3 bagian utama yaitu attribute, operation, dan name. (Gambar 7).



Gambar 7. Class Diagram Sistem Informasi Dagang.

Pada gambar 7, dapat dilihat semua class beserta atribut dan fungsi yang digunakan pada gambar 6 (Sequence Diagram). Nilai kembali juga dijabarkan secara jelas sehingga koneksi antara gambar 6 dan gambar 7 menjadi nyata.

Penerapan kontrol stok dengan metode Perpetual Inventory System ke dalam sistem informasi dagang menggunakan merupakan program berbasis desktop yang dirancang dengan menggunakan bahasa pemrograman C# dari .NET Framework 4.0. Tools yang digunakan adalah Visual Studio 2015. Berikut merupakan hasil rancangan form stok yang harus dilakukan re-order (Gambar 8).

Kode Barang	Nama Barang	Stok	Limit Bawah
BJM-W5W 13961 24V CP	SIGNALING - LAMPU SENJA, INTERIOR, PLAT	0	0
DLG 12825 WLED 12V X1	LED DAY LIGHT GUIDE	28	30
DRL 12830 WLED 12V X1	LED DAYTIME RUNNING LIGHT	97	100
FESTOON 30mm LED 12941 10000K X1	X-TREME ULTINON LED - LAMPU INTERIOR, PLAT, BAGASI	0	3
FESTOON 30mm LED 12941 6700K X1	X-TREME ULTINON LED - LAMPU INTERIOR, PLAT, BAGASI	0	2
GS N120	AKI GS N120 ASTRA	0	1
H1 12258 DV 12V 55W P14,5S S2	DIAMOND VISION - LAMPU DEPAN	0	1
H1 12258 PR 12V 55W P14,5S C1	PREMIUM VISION - LAMPU DEPAN	0	10
H1 12258 XV 12V 55W S2	X-TREME VISION - LAMPU DEPAN	0	2
H1 13258 24V 70W C1	STANDARD - LAMPU DEPAN	1	30
H11 12362 12V 55W C1	STANDARD - LAMPU DEPAN, FOG LAMP	0	25
H11 C 12362 XV 12V 55W PGJ19-2 S2	X-TREME VISION - LAMPU DEPAN, FOG LAMP	0	2
H16 12366 12V 19W PGJ19-3 C1	STANDARD - LAMPU DEPAN, FOG LAMP	0	1
H3 12336 PR 12V 55W PK22S C1	PREMIUM VISION - LAMPU DEPAN, TEMBAK	0	40
H3 13336 ML 24V 70W C1	MASTER LIFE - LAMPU DEPAN, TEMBAK	0	20
H3 13628 RA 24V 100W PK22S C1	RALLY - LAMPU DEPAN, TEMBAK	1	100
H3FIT 12455 RA 12V 100W PK22S C1	RALLY - LAMPU DEPAN, TEMBAK	0	0
H4 12342 DV 12V 60/55W S2	DIAMOND VISION - LAMPU DEPAN	0	0

Gambar 8. Hasil Rancangan Form Peringatan Barang yang Harus Dilakukan Re-Order.

Gambar 8 menunjukkan daftar barang-barang yang perlu dilakukan re-order segera oleh bagian purchasing karena stok untuk barang tersebut sudah mencapai limit bawah. Limit bawah dapat diatur berbeda-beda untuk setiap barang karena penjualan untuk tiap barang berbeda-beda. Limit bawah ini ditentukan dengan angka tertentu supaya jarak antara proses pemesanan hingga barang sampai di gudang tidak terjadi kekosongan stok.

Pengujian sistem dilakukan penulis untuk menguji 2 hal. Yang pertama adalah apakah penerapan metode Perpetual Inventory System sudah sesuai dengan kebutuhan fungsional atau belum. Pengujian kedua adalah seberapa besar dampak yang dirasakan oleh kehadiran middleware ini dibandingkan tanpa adanya middleware.

Pengujian pertama dilakukan dengan metode blackbox testing, yaitu menguji apakah fitur yang menjadi kebutuhan sistem sudah dapat dipenuhi oleh middleware yang dihasilkan. Pengujian akan dilakukan terhadap proses penjualan dan pembelian. Hal ini dimaksudkan untuk menguji apakah kedua proses tersebut dapat menambah / mengurangi stok di gudang sekaligus mencatat ke dalam risalah transaksi stok. Tampilan kondisi stok sebuah barang sebelum penjualan dan pembelian dapat dilihat pada gambar 9. Dari gambar juga dapat dilihat bahwa penulis menggunakan modul Kartu Stok yang sudah ada pada sistem informasi dagang untuk menampilkan stok sebuah produk dengan kode “H4 12342 PR 12V 60/55W P43T-38 C1”. Produk tersebut memiliki saldo terakhir sebanyak 11,016 buah di gudang.

Kartu Stok: Tampil, Ubah, Hapus

Form Record

1 of 1

Cabang: Pusat (Pontianak) Kode Barang: H4 12342 PR 12V 60/55W P43T-38 C1 Tahun: 2017 Bulan: 3 Gudang: Gudang

Kode Transaksi	Tgl Transaksi	Tgl Waktu Input	Keterangan	Tipe	Subjek	Mutasi Qty	Saldo Qty	Id
00128/2017	02/03/2017	02/03/2017 - 13:03:30	Insert	Jual	KENCANA MOTOR	-10	11,016	apue

Gambar 9. Kondisi Stok Sebelum Penjualan dan Pembelian.

Nota penjualan: Tambah

Form

Nota Terakhir:

Cabang: Pusat (Pontianak) Kode / Nama Pelanggan: CASH CASH

No Nota / FP: 00131/2017 010.000-17.69688245 Plafon / Piutang: 0.00 0.00

Tanggal: 13/03/2017 Sales: -

C/K: Tunai 0 hari No Kendaraan / Supir: KB

Tanggal JT: 13/03/2017 Ekspedisi: -

Tanggal LS: 13/03/2017 Keterangan: -

No.	Kode Barang	Nama barang	Harga @	Harga @ + PPn	Qty	Gudang
1	H4 12342 PR 12V 60/55W P43T-38 C1	PREMIUM VISION - LAMPU DEPAN	24,545.45	27,000.00	6.00	Gudang

Gambar 10. Pengujian Penjualan.

Kartu Stok: Tampil, Ubah, Hapus

Form Record

1 of 2

Cabang: Pusat (Pontianak) Kode Barang: H4 12342 PR 12V 60/55W P43T-38 C1 Tahun: 2017 Bulan: 3 Gudang: Gudang

Kode Transaksi	Tgl Transaksi	Tgl Waktu Input	Keterangan	Tipe	Subjek	Mutasi Qty	Saldo Qty	Id
00128/2017	02/03/2017	02/03/2017 - 13:03:30	Insert	Jual	KENCANA MOTOR	-10	11,016	apue
00131/2017	13/03/2017	13/03/2017 - 13:43:35	Insert	Jual	CASH	-6	11,010	tony

Gambar 11. Kondisi Stok Setelah Penjualan.

Pembelian: Tambah

Form

Pembelian Terakhir:

Cabang: Pusat (Pontianak) Kode / Nama Supplier: 8 PD. BUMI JAYA INDAH

Kode: BL.CAN.17-00001 Keterangan: -

Tanggal: 13/03/2017 Valuta / Kurs: IDR 1.00

C/K: Tunai 0 hari

Tanggal JT / LS: 13/03/2017 13/03/2017

No	Acc	Tgl Acc	Kode Barang	Harga @	Qty	Gudang	Modal	Tipe Diskon
1.			H4 12342 PR 12V 60/55W P43T-38 C1	15,000	5	Gudang	14,168.2	nominal

Gambar 12. Pengujian Pembelian.

Kartu Stok: Tampil, Ubah, Hapus

Form Record

1 of 3

Cabang: Pusat (Pontianak) Kode Barang: H4 12342 PR 12V 60/55W P43T-38 C1 Tahun: 2017 Bulan: 3 Gudang: Gudang

Kode Transaksi	Tgl Transaksi	Tgl Waktu Input	Keterangan	Tipe	Subjek	Mutasi Qty	Saldo Qty	Id
00128/2017	02/03/2017	02/03/2017 - 13:03:30	Insert	Jual	KENCANA MOTOR	-10	11,016	apue
00131/2017	13/03/2017	13/03/2017 - 13:43:35	Insert	Jual	CASH	-6	11,010	tony
BL.CAN.17-00001	13/03/2017	13/03/2017 - 14:01:22	Insert	Beli	PD. BUMI JAYA INDAH	5	11,015	tony

Gambar 13. Kondisi Stok Setelah Pembelian.

Pada gambar 10, penulis melakukan penjualan terhadap produk dengan kode “H4 12342 PR 12V 60/55W P43T-38 C1” sebanyak 6 buah. Pada gambar 11, terbukti bahwa stok berkurang sebanyak 6 buah dan risalah transaksi tercatat secara otomatis pada kartu stok. Hal ini menunjukkan bahwa pencatatan stok metode Perpetual Inventory System berjalan dengan baik. Gambar 12 menunjukkan pengujian pembelian terhadap produk yang sama sebanyak 5 buah. Gambar 13 menunjukkan kondisi produk yang sudah bertambah sebanyak 5 buah dan hal tersebut tercatat dengan sempurna di kartu stok.

Hal ini menunjukkan bahwa proses pencatatan stok metode Perpetual Inventory System berjalan dengan baik. Setelah pencatatan dilakukan dengan sempurna, hal selanjutnya kembali kepada modul peringatan stok yang telah dijelaskan sebelumnya. Dengan adanya proses pencatatan ini, maka modul peringatan juga dapat berfungsi dengan baik.

4. KESIMPULAN

Berdasarkan hasil dan pembahasan, maka dapat diambil kesimpulan bahwa implementasi perpetual inventory pada sistem informasi dagang dapat mengatasi masalah kekosongan stok karena stok sudah dapat dilacak dan dikontrol dengan lebih cepat menggunakan perpetual inventory system.

Implementasi perpetual inventory system juga mengatasi masalah daya saing perusahaan dari sisi efisiensi kerja dimana efisiensi ditingkatkan dengan pencatatan dan pengendalian stok yang lebih teliti. Masalah pada pencatatan periodik dimana stok barang tidak dapat diketahui sebelum adanya cek fisik / opname berhasil diatasi dengan menggunakan perpetual inventory sistem.

Pada akhirnya implementasi perpetual inventory system juga dapat meningkatkan laba perusahaan dengan mengurangi masalah keuntungan yang hilang karena kekosongan stok, efisiensi dalam pelacakan stok membuat pekerjaan lebih mudah bagi pengguna sehingga sumber daya tidak terbuang percuma untuk melakukan stok opname dengan hasil perhitungan biaya yang kurang akurat.

Modul peringatan juga berhasil diterapkan sehingga manajemen tidak perlu mengingat ataupun melakukan pencarian dengan catatan manual pada saat pengorderan stok. Modul ini akan sangat membantu manajemen dalam mengefisienkan proses pengaturan stok barang.

5. SARAN

Penulis menyadari bahwa aplikasi ini masih bisa dikembangkan lagi dengan menambahkan algoritma khusus untuk menghitung berapa stok minimal yang paling ekonomis. Pada penelitian selanjutnya dapat diterapkan algoritma baru untuk menghitung berapa stok minimal yang dapat diorder dan kapan harus mengorder serta berapa besaran order dengan biaya yang paling ekonomis sesuai dengan jarak, packing dan biaya pengorderan serta gudang.

DAFTAR PUSTAKA

- [1] Namit, K., J. Chen, 1999, Solutions to the inventory model for gamma lead-time demand. *International Journal of Physical Distribution & Logistics Management* vol 29 no 2 hal 138-154
- [2] Wanke, P., 2011, Product, operation, and demand relationships between manufacturers and retailers. *Transportation Research: Part E: Logistics and Transportation Review* vol 48 no 1, hal 340-354.

- [3] Mgbemena., C., E., Godwin, H. C., 2013, Re-engineering Materials Management System in the Oil and Gas Service Industry, Asian Journal of Engineering and Technology vol 1 no 1, hal 17-25
- [4] Tersine R.J., Principles of inventory and materials management, 1994.
- [5] Horngren, C. T., Harrison, W. T., Oliver, M. S., Financial and Managerial Accounting, Prentice Hall, New Jersey, 2012
- [6] Chołodowicz, E., Orłowski., P., 2016, Comparison of a Perpetual and PD Inventory Control System with Smith Predictor and Different Shipping Delays Using Bicriterial Optimization and SPEA2, Pomiary Automatyka Robotyka no 3, hal 5-12
- [7] N., Dr. Rajkumar, J., Selvakumar, B., Ranjith Babu, Extended Scrum Method of Agile Practice for Small Scale Project Development (Online), *International Journal of Innovative Research in Science, Engineering and Technology*, ISSN (Online): 2319-8753, Vol 3, Special Issue 1, 2014, Hal 374.
- [8] Pressman, R., *Software Engineering: a Practitioner's Approach*. Sixth Edition. McGraw-Hill., Singapore. 2005.
- [9] Microsoft Developer Network (MSDN), Visual Studio IDE (Visual Studio 2015) (Online), (<https://msdn.microsoft.com/en-us/library/dn762121.aspx>, diakses 14 November 2016).
- [10] MySQL: About MySQL (Online), (<https://www.mysql.com/about/>, diakses 24 November 2016).
- [11] Bester, K., Chandler, A. T., Shewell, M. A., & Yates, S. J. (2016). U.S. Patent No. 20,160,132,543. Washington, DC: U.S. Patent and Trademark Office.