



## PEMBUATAN APLIKASI KONTROL *STOCK OUT* MENGUNAKAN VISUAL BASIC.NET DAN SQL SERVER DI PERUSAHAAN MANUFAKTUR OTOMOTIF

Afianto, Hegaline Ferdi Endam

Konsentrasi Mekatronika Program Studi Teknik Produksi dan Manufaktur Politeknik Manufaktur Astra  
Jl.Gaya Motor Raya No. 8 Sunter II Jakarta 14330  
Telp. (021) 651 9555 ext.2880 Fax. (021) 651 9821  
E-mail: Afianto@polman.astra.ac.id; syahril.ardi@polman.astra.ac.id

**Abstrak :** *Part Tool Inventory Control* (PTIC) adalah bagian dari Divisi Maintenance yang bertugas untuk mengontrol *tool* dan part yang ada di PT XZ. Salah satu pekerjaan rutin di PTIC adalah *Stock Out*. Di dalam proses *Stock Out* ini terdapat proses yang sangat penting yaitu Input data *Stock Out*. Permasalahan yang terjadi adalah adanya penundaan input data yang seharusnya dilakukan bersamaan dengan proses *Stock Out* tetapi pada kondisi sesungguhnya input data dilakukan diluar proses *Stock Out* (*delay input*) Setelah dilakukan analisa ternyata hal tersebut disebabkan oleh waktu transaksi *Stock Out* yang tidak seimbang dengan waktu pelayanan yang disediakan. Jika input data dilakukan bersamaan dengan proses *Stock Out*, terdapat kemungkinan bahwa akan ada *customer* yang tidak terlayani. Untuk menanggulangi hal tersebut, dibuatlah program aplikasi kontrol *Stock Out* yang dilengkapi dengan Barcode scanner untuk proses input data *Stock Out*. Aplikasi ini akan secara langsung memperbaharui data dalam database dengan menggunakan input text berupa barcode, sehingga didapatkan waktu untuk input data adalah 26,03 detik, sehingga input data dapat dilakukan bersamaan dengan transaksi tanpa harus melakukan penundaan input data.

**Kata kunci** *Stock Out, Delay input, Input data Stock Out, aplikasi control Stock Out*

### 1. Pendahuluan

#### 1.1 Latar Belakang

PT XZ merupakan salah satu perusahaan yang bergerak di bidang manufaktur suku cadang otomotif. Berfokus pada pembuatan komponen otomotif baik roda empat (4W) maupun roda dua (2W), perusahaan ini sudah mempunyai beberapa Customer tetap.

PTIC (Part Tools Inventory Control) merupakan sebuah seksi dibawah maintenance yang bertugas bukan hanya untuk mengontrol ketersediaan spare part dan tool, tetapi juga untuk mengontrol aliran distribusi spare parts dan manajemen tiap spare parts nya.. Untuk mendukung kegiatan operator PTIC ini, telah dibangun sebuah program inventory yang berbasis WEB Application yaitu program Depas. Dimana bertujuan untuk menjalin komunikasi antara ketiga PTIC disetiap Plant.

Setiap transaksi yang dilakukan melalui program ini akan tersimpan dalam sebuah database. Database inilah yang kemudian akan menyimpan Stock terakhir yang akan dijadikan indikator ketersediaan part. Oleh karena itu dibutuhkan penginputan data out dan in secara teratur dan tepat waktu, agar persediaan di PTIC tetap terbaharui. Sehingga jika terjadi pengurangan Stock maka

langsung dapat diantisipasi apakah parts tersebut sudah harus kembali dipesan atau belum.

Permasalahan yang terjadi adalah proses input data yang disediakan memakan waktu yang relatif lama dan tidak seimbang dibandingkan dengan waktu service yang disediakan, sehingga Operator PTIC memutuskan untuk menunda input data.

#### 1.2 Perumusan Masalah

Berdasarkan latar belakang di atas, rumusan masalah pada penelitian ini adalah:

1. Bagaimana membuat aplikasi kontrol *Stock Out* menggunakan VB.NET yang dapat mempercepat *cycle time* proses *Stock Out* menjadi 400 detik untuk tiap transaksi?
2. Bagaimana cara membuat aplikasi kontrol *Stock Out* yang dapat melakukan operasi untuk membatalkan transaksi yang telah di-input?
3. Bagaimana cara membuat aplikasi kontrol *Stock Out* yang dapat memberikan peringatan otomatis saat terdapat persediaan yang telah mencapai *order point*?
4. Bagaimana cara membuat aplikasi kontrol *Stock Out* yang dapat melakukan operasi pemesanan melalui email?

## 2. Landasan Teori

### 2.2 Sistem Inventory

#### 2.2.1 Sistem

Menurut Herlambang (2005:116), definisi sistem dapat dibagi menjadi dua pendekatan, yaitu pendekatan secara prosedur dan pendekatan secara komponen. Berdasarkan pendekatan prosedur, sistem didefinisikan sebagai kumpulan dari beberapa prosedur yang mempunyai tujuan tertentu. Sedangkan berdasarkan pendekatan komponen, sistem merupakan kumpulan dari komponen-komponen yang saling berkaitan untuk mencapai tujuan tertentu.

#### 2.2.2 Persediaan (Inventory)

##### a.) Pengertian Persediaan (Inventory)

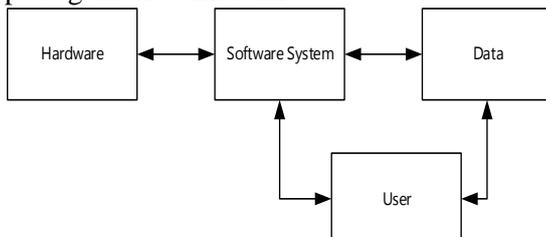
Persediaan (*inventory*) merupakan barang yang disimpan untuk digunakan atau dijual pada periode mendatang. Persediaan erat hubungannya dengan operasional perusahaan, baik perusahaan yang bergerak dalam bidang perdagangan maupun industri. Jika penanganan persediaan tidak dilaksanakan dengan baik maka akan mengakibatkan resiko terganggunya proses produksi atau tidak terpenuhinya pesanan pembelian, akibatnya dapat merugikan perusahaan.

Sifat atau batasan barang yang dapat diklasifikasikan sebagai persediaan adalah bervariasi sesuai dengan aktivitas perusahaan. Untuk mengetahui apakah pengertian persediaan itu, penulis akan menjelaskan batasan-batasannya.

### 2.3 Sistem Informasi

Pada dasarnya sistem informasi merupakan suatu sistem yang dibuat oleh manusia yang terdiri dari komponen-komponen dalam organisasi untuk mencapai suatu tujuan yaitu menyajikan data informasi. Sistem informasi menerima masukan data dan instruksi, mengolah data tersebut sesuai dengan instruksi, dan mengeluarkan hasilnya.

Komponen dari sistem informasi dapat dilihat pada gambar berikut ini :



Gambar 4.1 Komponen – Komponen dari Suatu Sistem

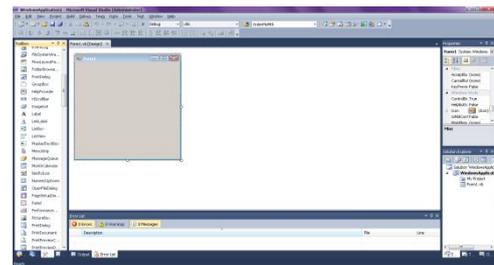
Dari definisi di atas terlihat bahwa sistem informasi merupakan satu kesatuan unsur (manusia

dan peralatan) yang bekerjasama untuk melaksanakan pengolahan informasi dari mulai mengumpulkan, pengolahan, penyimpanan sampai pendistribusiannya.

### 2.4 Perangkat Lunak

#### 2.4.1 Visual Studio 2010

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.

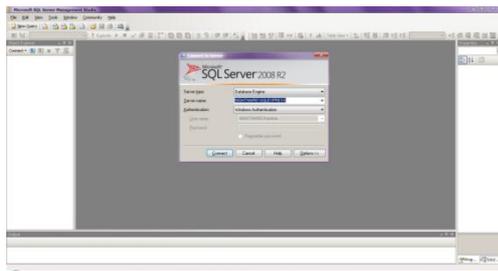


Gambar 4.2 Interface Visual Studio

Pembuatan program aplikasi menggunakan Visual Studio dilakukan dengan membuat tampilan aplikasi pada *form*, kemudian diberi *script* program di dalam komponen-komponen yang diperlukan. *Form* disusun oleh komponen-komponen yang berada di [*Toolbox*], dan setiap komponen yang dipakai harus diatur propertinya lewat jendela [*Property*].

#### 2.4.2 Microsoft SQL server 2008 R2

*Database* merupakan tempat penyimpanan data, yang memudahkan pengolahan dan pencarian data. SQL Server 2008 R2 merupakan DBMS (*Database Management System*) yang digunakan untuk mengolah data. Di samping memiliki *user interface* yang mudah digunakan, SQL Server 2008 R2 juga memiliki *fitur* tambahan yang bisa diimplementasikan untuk meningkatkan performa dan mengurangi tempat penyimpanan.



Gambar 4.3 Interface Microsoft SQL Server 2008 R2

Menurut Satzinger, Jackson dan Burd ( 2009, p488 ), *Database Management System* (DBMS) adalah sistem perangkat lunak yang mengelola dan mengontrol akses ke database.

Jadi kesimpulan dari pengertian *Database Management System* (DBMS) adalah sistem perangkat lunak yang digunakan untuk menambah, memperbarui, mengelola, mengakses dan menganalisis data dalam database.

## 2.5 Perangkat Keras

### 2.5.1 Barcode reader

#### a.) Pengertian Barcode

Barcode atau kode batang adalah sekumpulan data yang digambarkan dengan garis dan jarak spasi (ruang). Barcode menggunakan urutan garis batang vertikal dan jarak antar garis untuk mewakili angka atau simbol lainnya. Dengan demikian, setiap ketebalan garis batang dan jarak antara garis satu dengan yang lain selalu berbeda sesuai dengan isi data yang dikandung oleh kode batang atau Barcode tersebut.



Gambar 4.4 Barcode

#### b.) Barcode scanner

Barcode *Scanner* adalah alat yang digunakan untuk membaca kode-kode berbentuk garis-garis vertikal yang terdapat pada kebanyakan produk-produk *consumer good*. Penggunaan Barcode *scanner* ini mempunyai dua keuntungan tambahan. Pertama akan memperkecil kesalahan *input* yang disebabkan kesalahan operator komputer atau kasir. Kedua, penggunaan Barcode *scanner* mempercepat proses *entry* data sehingga mengurangi jumlah antrian yang panjang.

### 2.5.2 Computer

Istilah komputer<sup>1</sup> berasal dari bahasa latin “*computere*” yang artinya menghitung. Dalam bahasa Inggris disebut *to compute*. Komputer juga diterjemahkan sebagai sekumpulan alat elektronik yang saling bekerja sama, dapat menerima data(input),mengolah data(proses) dan memberikan informasi(output) sertaterkoordinasi dibawah control program yang tersimpan di memorinya

## 3. Pengumpulan Data

### 3.2 Analisa Kondisi Saat Ini (Kondisi Sebelum Perbaikan)

Kondisi yang ada pada proses *Stock Out* sekarang ini mempunyai waktu proses seperti yang ada pada tabel

Tabel 4.1 Tabel perbandingan waktu input

Proses	Manual	BHT	Standard
Menyerahkan BON	1	1	1
Mengambil Part *(+Scan Part)	360	360+102,2	360
Menyerahkan Part	2	2	2
Input data BON *(Transfer data)	132,8	20,45	36
Menyerahkan salinn BON	1	1	1

Dari tabel dapat dilihat bahwa untuk satu transaksi mempunyai kebutuhan waktu masing-masing yang terdiri dari *service time* dan *input time*. Saat operator PTIC memutuskan untuk menunda *input* data, maka waktu yang dibutuhkan untuk transaksi adalah 360 detik. Jika dibandingkan dengan *service time* yang diberikan maka,

Service time 1 =

4800 detik : 360 detik = **13 Customer**

Service time 2 =

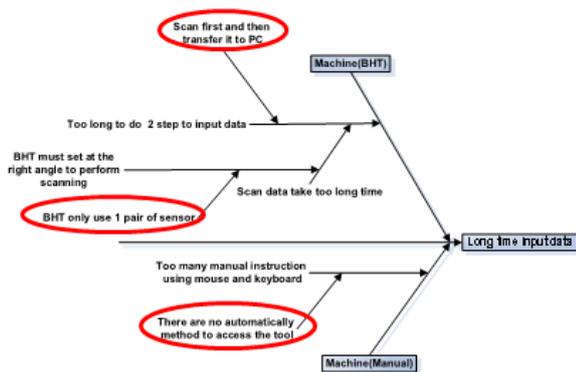
5400 detik : 360 detik = **15 Customer**

Terbukti dengan melakukan penundaan *input* maka seluruh *customer* dapat dilayani sesuai jadwal.

### 3.2.1 Analisa Sebab dan Akibat

Selanjutnya dilakukan analisa penyebab dan akibat dalam bentuk diagram *fishbone* ( Tulang Ikan ) berikut.

<sup>1</sup>Mahalik,Nitigor Pramchand.2004. MECHATRONICS in Principles,Concepts,and Applications.USA



Gambar 4.5 Analisa sebab akibat (Fishbone) problem di PTIC

Dari analisa sebab akibat di atas, telah ditemukan beberapa akar masalah yang kemudian dilakukan perencanaan untuk menanggulangi masalah tersebut

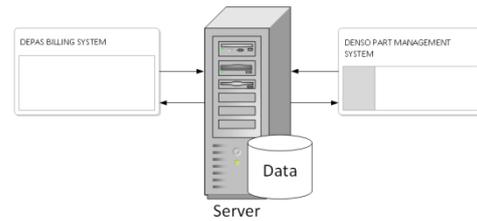
Tabel 4.2 Tabel perencanaan penanggulangan permasalahan

Faktor	Kondisi	Permasalahan	Penyelesaian	Target	PIC	Waktu
Mesin	Input data memakan terlalu banyak waktu	a) Hanya menggunakan 1 pasang IR b) Membutuhkan transfer data melalui terminal untuk input data	Mengganti menggunakan barcode omni directional dan mengkombinasikan dengan aplikasi kontrol stock out	Input time = 36 s	Jemu & Ega	28 February 2015
		Mamal input Terlalu banyak metode input mamal menggunakan mouse dan keyboard			Jemu & Ega	28 February 2015
Mesin	Ada perbedaan stock actual dan data karena pengembalian part	Program Depas tidak menyediakan operasi pembatalan transaksi	Membuat operasi untuk pembatalan transaksi	Dapat melakukan record untuk pembatalan transaksi	Jemu & Ega	28 February 2015
Mesin	Keterlambatan order part yang mencapai Order Point	Tidak ada peringatan apabila terdapat part yang telah mencapai OP	Membuat fitur auto reminder apabila terdapat part yang telah mencapai OP	Dapat menampilkan part mencapai OP otomatis pada waktu tertentu	Jemu & Ega	28 February 2015
Mesin	Pengumpulan file order memakan ruang di almari	Terjadi penumpukan order printout untuk machinery plant 2	Pembuatan metode order melalui email untuk part pembuatan oleh machinery plant 2	Dapat melakukan pengiriman man order melalui email	Jemu & Ega	28 February 2015

## 4. Perancangan Dan Pembuatan

### 4.1 Gambaran umum Program

Program Depas mempunyai *database* bernama DePASML\_U. Menggunakan database inilah program Kontrol Stock Out yang dibuat akan dihubungkan dengan program Depas. Secara diagram akan digambarkan pada gambar.

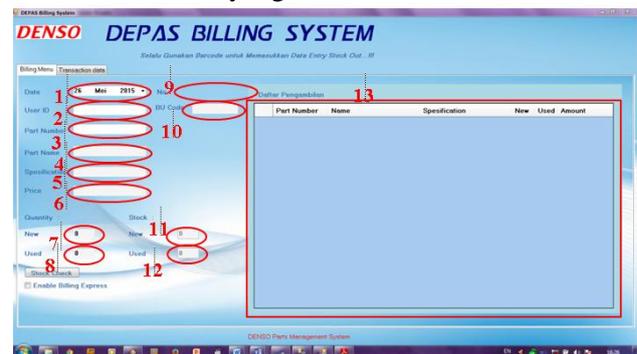


Gambar 4.1 Rancangan koneksi program Kontrol Stock Out ke Program Depas

Dari gambar dapat dijelaskan bahwa program Kontrol Stock Out akan mengambil data dari database depas. Database depas tersebut akan diakses dengan menghubungkan program Kontrol Stock Out dengan database tersebut. Cara yang akan ditempuh adalah dengan menggunakan nama server dan database depas tersebut. Setelah itu program Kontrol Stock Out ini akan mengakses data dalam database tersebut dengan mengatur *setting variable* sama dengan nama tabel dan kolom dari database yang akan diambil. Dengan demikian program Kontrol Stock Out ini dapat mengakses dan mengambil data dari database depas.

### 4.2 Perancangan dan pembuatan program Kontrol Stock Out dilengkapi barcode scanner dan printer.

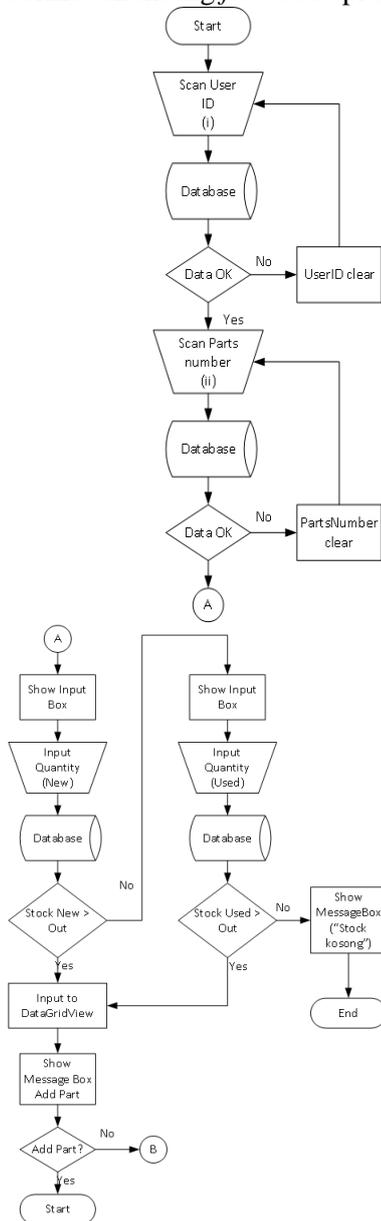
Program yang akan di bangun adalah suatu aplikasi berbasis desktop yang akan dilengkapi dengan barcode scanner sebagai perangkat. Sehingga aplikasi ini akan di desain untuk mengurangi campur tangan manusia dalam prosesnya. Penggunaan barcode reader juga akan mengurangi input manual oleh operator. Berikut tampilan windows aplikasi kontrol Stock Out yang dibuat.



Gambar 4.2 Tampilan jendela program Kontrol Stock Out

Selain menggunakan tool di atas, pada program ini juga akan memanfaatkan operasi menggunakan MessageBox dan InputBox. Hal ini bertujuan untuk mengurangi kerja operator untuk menggunakan mouse. Karena MessageBox dan inputbox ini akan

secara otomatis keluar dan operator hanya perlu memasukkan text/angka ataupun operasi menggunakan *keyboard* sehingga dapat mempercepat proses *input* data. Berdasarkan pada tampilan di atas maka sebagai dasar dalam pembuatan program, berikut dirancang *flowchart* program tersebut.

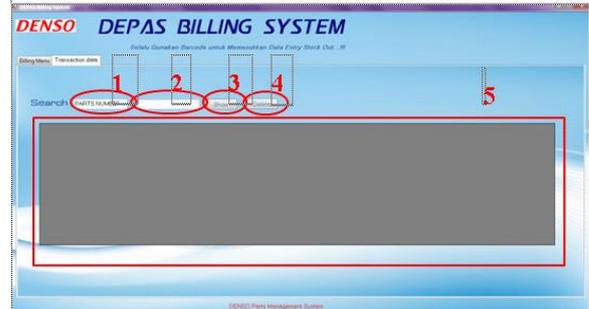


Gambar 4.3 Flowchart Program Kontrol *Stock Out*

Pembuatan *flowchart* program seperti di atas yang sebagian besar menggunakan proses oleh komputer ditujukan supaya meminimalisir kebutuhan akan tenaga *user* untuk melakukan operasi *Stock Out*. Dengan demikian operasi manual yang dilakukan oleh *user* akan menjadi lebih ditekan sehingga pada hasil akhir diharapkan *cycle time* untuk *input* data *Stock Out* menjadi lebih cepat.

### 4.3 Perancangan operasi pembatalan transaksi

Operasi pembatalan transaksi ini akan dijadikan salah satu tab menu pada program Kontrol *Stock Out*. Operasi ini akan dijalankan saat terdapat pengembalian part setelah *input* data. Berikut rancangan tampilan menu pembatalan transaksi



Gambar 4.4 Tampilan jendela pembatalan transaksi

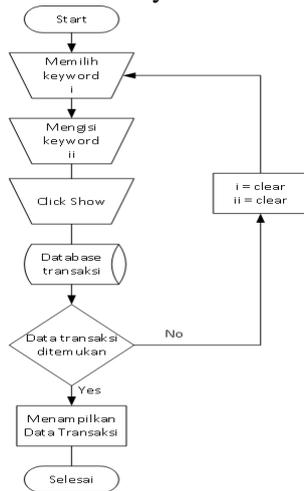
Permasalahan yang terjadi adalah belum adanya *database* yang secara khusus menyimpan *history* transaksi. Maka juga akan dibuat sebuah tabel baru dalam *database* DePASML\_U yang akan khusus menyimpan *history* transaksi. Table tersebut akan berisi tanggal, *partnumber*, *part name*, *Quantity* yang diambil, *Stock* di PTIC, harga tiap part, nomor seksi, *user id* dan total pengambilan dalam rupiah. Dengan PK (*Primary key*) nomor transaksi yang akan digunakan sebagai kata kunci pemanggilan transaksi. Berikut *design* tabelnya.

Tabel 4.1 Tabel *database* transaksi

Coloumn name	Data Type	Allow NULL
Date	Date	No
Partnumber	Varchar(50)	No
Partsname	Varchar(50)	No
QTY_New	int	No
QTY_Used	int	No
Stock_New	int	No
Stock_Used	int	No
Price	int	No
Section_Code	Varchar(50)	No
User_ID	Varchar(50)	No
No_TRX	Varchar(50)	No
No_key	int	No
Total	int	No
Status	Varchar(50)	No

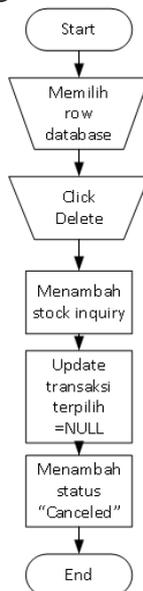
Pembatalan transaksi ini akan dilakukan dengan cara menghapus data transaksi, dan mengembalikan jumlah *Stock* pada *Stock inquiry*.

Sebelum memulai pembuatan program maka akan dibuat *flowchart* program. *Flowchart* ini akan digunakan sebagai pedoman dalam pembuatan program. Ada 2 buah *flowchart* yang dirancang dan 2 program yang akan dibuat yaitu:



Gambar 4.5 Flow chart menampilkan data transaksi

Program tersebut digunakan untuk menampilkan *history* transaksi yang pernah terjadi sesuai dengan kata kunci yang digunakan. Apakah *PartsNumber*, tanggal ataupun *No* transaksi. Yang kemudian akan ditampilkan pada *datagrid* view.



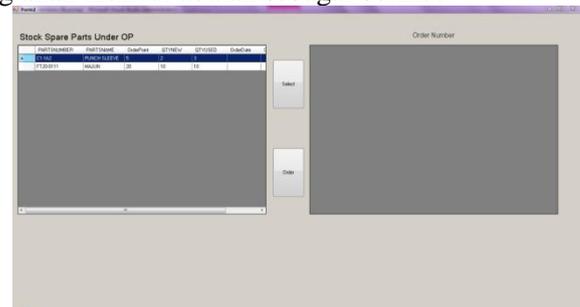
Gambar 4.6 Flowchart penghapusan data transaksi

Program di atas tidak digunakan untuk menghapus data transaksi yang dibatalkan.

Melainkan akan mengubah status transaksi menjadi "Canceled". Selain itu program yang diletakkan pada button delete ini akan menambah *Stock* pada *inquiry Stock* sesuai dengan jumlah part dari transaksi yang dibatalkan. Hal ini ditujukan untuk menyimpan data transaksi yang dibatalkan. Selain itu untuk mencegah adanya salah pilih transaksi maka apabila terpilih transaksi yang telah berstatus "Canceled" maka akan ditampilkan *messagebox* peringatan bahwa transaksi tersebut telah dibatalkan sebelumnya.

#### 4.4 Perancangan dan pembuatan program auto reminder

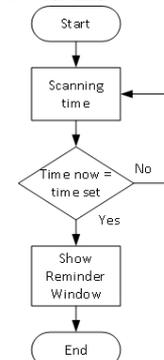
Pada program ini akan digunakan event otomatis yang akan di kontak menggunakan waktu. Jadi program ini akan menampilkan window yang di dalamnya berisi tentang parts yang stocknya berada dibawah *OP(Ordering Point)*. Window tampilan yang akan muncul adalah sebagai berikut:



Gambar 4.7 Tampilan Auto reminder

Pada tampilan ini tidak banyak menggunakan komponen. Hanya ada *DataGridView* dan label. Tetapi pada form Utama akan diletakkan *timer* yang berfungsi untuk melakukan fungsi otomatis menampilkan tampilan ini pada jam tertentu tiap harinya. Komponen tersebut akan dijelaskan pada tabel berikut :

Berdasarkan pada tampilan dan komponen yang digunakan maka akan dirancang program sesuai dengan *flowchart* dibawah ini :

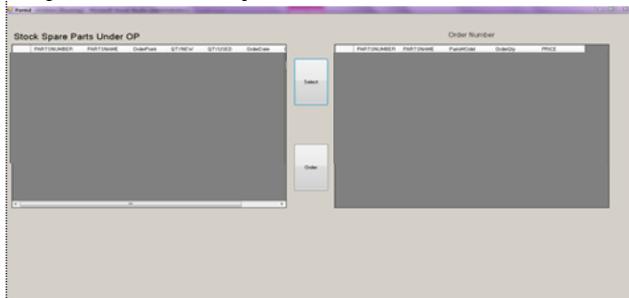


Gambar 4.8 Flowchart program auto reminder

Program yang dibuat adalah program yang digunakan untuk mencari data part yang telah mencapai OP maupun dibawah OP dari dalam database. Kemudian data yang terpilih itu akan ditampilkan dalam DataGridView. Data dalam datagridview ini kemudian akan dijadikan bahan untuk melakukan *order*.

#### 4.5 Perancangan *order* melalui email

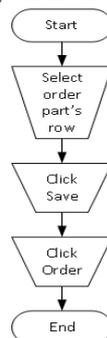
*Order* otomatis akan dibuat dengan menggunakan crystal report dalam format \*.pdf yang kemudian akan dijadikan *order* dalam bentuk lampiran email. *Order* ini akan menggunakan button yang tersedia pada Windows *Auto reminder*. Berikut tampilan windows nya



Gambar 4.9 Tampilan *Order* melalui email

Tampilan ini menjadi satu dengan tampilan *auto reminder*. Ini ditujukan supaya saat diketahui adanya part yang mencapai OP akan dapat dilakukan proses *order*. Berikut adalah *tool* yang digunakan pada tampilan *order via email*.

Dengan menggunakan tampilan di atas, berikut flowchart program yang akan dibuat.



Gambar 4.10 Flowchart program sending *order via email*

Program yang dibuat akan terbagi menjadi dua fungsi melalui 2 *button* yang tersedia yang pertama membuat *Order.rpt* dan yang kedua adalah send email plus attachment.

#### 5. Kesimpulan

Dari pembuatan program Kontrol *Stock Out* menggunakan vb.net dan ms sql server 2008 R2 ini, pembuatan aplikasi kontrol *Stock Out* menggunakan barcode honeywell ini dapat mempercepat proses *input* sehingga didapatkan waktu transaksi yang baru adalah 390,03 detik. Aplikasi kontrol *Stock Out* ini dapat melakukan operasi pembatalan transaksi dengan menggunakan sebuah tabel *database* yang baru. Tabel *database* ini akan mengakses data transaksi yang berjalan dan kemudian akan menghapus dan menambah quantity pada *inquiry Stock* sesuai dengan jumlah quantity pembatalan. Aplikasi kontrol *Stock Out* dapat melakukan operasi *Auto reminder* dengan menggunakan perintah *timer* untuk memunculkan jendela *auto reminder* secara otomatis. Di dalamnya ditampilkan part apa saja yang telah mencapai *order point* yang dimana data tersebut diambil dari *inquiry stock*. Aplikasi kontrol *Stock Out* yang dapat melakukan operasi pengiriman *order* melalui email dengan memanfaatkan *auto reminder*, yaitu dengan adanya data parts mencapai *order point* pada *auto reminder* maka akan secara langsung akan dikirim lewat email part yang sebelumnya belum *terorder*. Pengiriman email memanfaatkan akun google untuk mengirim email pada akun email perusahaan.

#### Referensi