



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB III

PELAKSANAAN KERJA MAGANG

3.1. Kedudukan dan Koordinasi

Dalam pelaksanaan kerja magang ini, penulis berkedudukan sebagai *programmer* pada Divisi IT. Sebagai pembimbing lapangan dan sekaligus **Head of IT Department** adalah Bapak Mudarwi.

Bersama dengan **Head Office Manager**, beliau memberikan informasi dan desain sistem secara garis besar, menjabarkan kendala-kendala yang terdapat pada sistem yang ada, memaparkan kebutuhan untuk mekanisme yang ingin dibangun, dan ikut serta dalam pengujian modul-modul yang dibuat untuk memastikan bahwa pengerjaan sesuai dengan *timeline* yang telah ditentukan.

3.2. Tugas yang Dilakukan

Selama melaksanakan kerja magang di PT Arta Dwi Mitra, pengembangan dilakukan pada sistem logistik khususnya untuk modul *Sales Order* dan *Delivery Order*. Banyak modul yang dibuat untuk mengembangkan mekanisme ini. Modul-modul yang berhasil dibuat untuk pengembangan mekanisme ini adalah modul SO (*Stock Opname*), modul DO (*Delivery Order*), modul DO (*Delivery Order*) Checker, modul Cancel Koli, dan modul Koli Tes.

Modul SO (*Stock Opname*), digunakan untuk memasukkan data produk baru sekaligus sebagai penanda bahwa produk tersebut memang ada dan dapat didistribusikan. Modul DO (*Delivery Order*), digunakan untuk memasukkan data produk yang akan didistribusikan ke toko. Modul DO (*Delivery Order*) Checker, digunakan untuk memeriksa apakah ada kesalahan pada *Delivery Order*. Modul Cancel Koli, digunakan untuk membatalkan pengiriman suatu koli. Modul Koli Tes, digunakan untuk menghindari kecurangan yang dapat dilakukan oleh petugas di gudang sehingga menjamin bahwa produk yang dikirim akan sama jumlahnya dengan yang diterima oleh toko.

3.3. Uraian Pelaksanaan Kerja Magang

Tabel 3.1. Realisasi Kerja Magang

Minggu	Kegiatan	
1	a	Instalasi dan <i>setup software</i> yang dibutuhkan (Visual Studio 2008, WinCE Emulator)
	b	Mempelajari konsep dan aplikasi yang ingin dikembangkan
	c	Pengenalan <i>Portable Data Terminal</i> HT680
2	a	Membuat form <i>login</i> , form menu, form SO, form <i>reset</i>
	b	Melakukan <i>testing</i> di <i>emulator</i> dan <i>Portable Data Terminal</i>
3	a	Mempelajari konsep DO (Delivery Order)
	b	Membuat form DO
	c	Perbaikan form <i>login</i> , form menu, form SO, form <i>reset</i>
	d	Melakukan <i>testing</i> di <i>Portable Data Terminal</i>
4	a	Membuat form DO checker dan cancel koli
	b	Perbaikan form DO
	c	Melakukan <i>testing</i> di <i>Portable Data Terminal</i>
5	a	Membuat form koli tes
	b	Melakukan <i>testing</i> dan penyelesaian versi 1
	c	Mempelajari konsep dan melakukan perancangan untuk versi 2 (memanfaatkan sinyal wifi)
6	a	Membuat form <i>login</i> , form menu, form SO, form cancel koli versi 2
	b	Melakukan <i>testing</i> di <i>Portable Data Terminal</i>
7	a	Membuat form DO, DO checker, Koli Tes versi 2
	b	<i>Debugging</i> dan <i>enhancing</i> versi 1
	c	Melakukan <i>testing</i> di <i>Portable Data Terminal</i>
8	a	Perbaikan form DO versi 1 dan versi 2
	b	<i>Debugging</i> dan <i>enhancing</i> versi 2
	c	Penyelesaian versi 1 dan versi 2

Pada minggu pertama periode magang, kegiatan yang dilakukan adalah melakukan instalasi dan *setup software* yang dibutuhkan. Karena pembuatan modul menggunakan bahasa pemrograman C#, maka digunakan aplikasi Visual Studio 2008. Kemudian sebagai pengganti *portable data terminal*, digunakan aplikasi WinCE Emulator. Setelah berhasil melakukan instalasi aplikasi-aplikasi yang dibutuhkan, mulai dipelajari tentang konsep mekanisme yang sedang digunakan, bagaimana mekanisme ini ingin dikembangkan, mekanisme seperti apa yang ingin dibangun, dan modul apa saja yang dibutuhkan. Kemudian juga diperkenalkan dengan alat dimana mekanisme diimplementasikan, yaitu *portable data terminal* HT680.

Pada minggu kedua periode magang, mulai dilakukan pengembangan berdasarkan konsep yang sudah disepakati. Pertama dibuat form *login* yang

berguna sebagai pengatur agar hanya pengguna yang mengetahui *login ID* yang dapat masuk dan menggunakan aplikasi. Dilanjutkan dengan membuat form menu yang digunakan untuk menampilkan semua pilihan modul. Kemudian mulai dikerjakan modul-modul yang dibutuhkan. Modul pertama yang dibuat adalah modul SO (*Stock Opname*) dengan membuat form SO. Modul SO (*Stock Opname*), digunakan untuk memasukkan data produk baru sekaligus sebagai penanda bahwa produk tersebut memang ada dan dapat didistribusikan. Kemudian dibuat juga form *reset* yang digunakan untuk mengulang suatu proses. Setelah itu dilakukan uji coba dengan *portable data terminal* untuk semua form yang telah berhasil dibuat.

Pada minggu ketiga periode magang, mulai dipelajari tentang konsep DO (*Delivery Order*) seperti apa itu DO (*Delivery Order*), bagaimana cara kerja DO (*Delivery Order*), perancangan form DO, dan bagaimana form DO ingin dibuat. Setelah mempelajari dan mengumpulkan informasi, dibuatlah form DO untuk menampilkan modul DO (*Delivery Order*). Modul DO (*Delivery Order*), digunakan untuk memasukkan data produk yang akan didistribusikan ke toko. Dalam minggu ketiga juga dilakukan perbaikan pada form *login*, form menu, form SO, dan form *reset* yang telah berhasil dibuat. Setelah itu dilakukan uji coba dengan menggunakan *portable data terminal* dan *emulator* untuk semua form yang telah berhasil dibuat dan diperbaiki.

Pada minggu keempat periode magang, pengembangan dilakukan dengan membuat form DO checker dan cancel koli yang masing-masing digunakan untuk menampilkan modul DO (*Delivery Order*) Checker dan modul Cancel Koli. Modul DO (*Delivery Order*) Checker, digunakan untuk memeriksa apakah ada kesalahan pada DO (*Delivery Order*). Modul Cancel Koli, digunakan untuk membatalkan pengiriman suatu koli. Dalam minggu keempat juga dilakukan perbaikan pada form DO (*Delivery Order*) yang telah berhasil dibuat. Setelah itu dilakukan uji coba dengan menggunakan *portable data terminal* dan *emulator* untuk semua form yang telah berhasil dibuat dan diperbaiki.

Pada minggu kelima periode magang, dibuat form koli tes untuk menampilkan modul Koli Tes. Modul Koli Tes, digunakan untuk menghindari

kecurangan yang dapat dilakukan oleh petugas di gudang sehingga menjamin bahwa produk yang dikirim akan sama jumlahnya dengan yang diterima oleh toko. Setelah itu dilakukan uji coba dengan menggunakan *portable data terminal* dan *emulator* untuk semua form yang telah berhasil dibuat dan diperbaiki. Setelah 5 modul telah berhasil dibuat dan diuji coba, dinyatakan bahwa mekanisme pertama berhasil diselesaikan dan diberi penanda versi 1. Setelah itu mulai dipelajari konsep dan melakukan perancangan untuk membangun mekanisme versi 2, yaitu mekanisme dengan memanfaatkan sinyal wifi (*wireless*).

Pada minggu keenam periode magang, dibuat form *login*, form menu, form SO, dan form cancel koli yang masing-masing digunakan untuk menampilkan halaman *login*, halaman menu, modul SO (*Stock Opname*), dan modul Cancel Koli pada mekanisme versi 2. Setelah itu dilakukan uji coba dengan menggunakan *portable data terminal* dan *emulator* untuk semua form yang telah berhasil dibuat.

Pada minggu ketujuh periode magang, mekanisme versi 2 kembali dikembangkan dengan membuat form DO, DO checker, dan koli tes untuk masing-masing menampilkan modul DO (*Delivery Order*), DO (*Delivery Order*) Checker, dan Koli Tes. Pada minggu ketujuh juga dilakukan *debugging* untuk mekanisme versi 1. Setelah itu dilakukan uji coba dengan menggunakan *portable data terminal* dan *emulator* untuk semua form yang telah berhasil dibuat dan diperbaiki.

Pada minggu kedelapan periode magang, dilakukan perbaikan pada form DO untuk versi 1 dan versi 2. Setelah itu kembali dilakukan uji coba dan *debugging* untuk semua modul yang telah dibuat dan diperbaiki, baik pada versi 1 dan juga versi 2. Setelah dilakukan uji coba dan *debugging*, maka dinyatakan bahwa mekanisme versi 2 berhasil diselesaikan. Waktu yang tersisa selama periode magang kembali digunakan untuk melakukan perbaikan dan uji coba pada kedua mekanisme baru yang berhasil dibangun, versi 1 dan versi 2.

3.3.1. Proses Pelaksanaan

Pengembangan modul yang dikerjakan dalam kerja magang ini menggunakan metode *waterfall*. Metode pengembangan ini dipilih karena pengguna sudah mengerti kebutuhan secara jelas dan risiko pengembangan bisa dikatakan sangat kecil (Pressman, 2005).

A. Analisis dan Persiapan Pengembangan

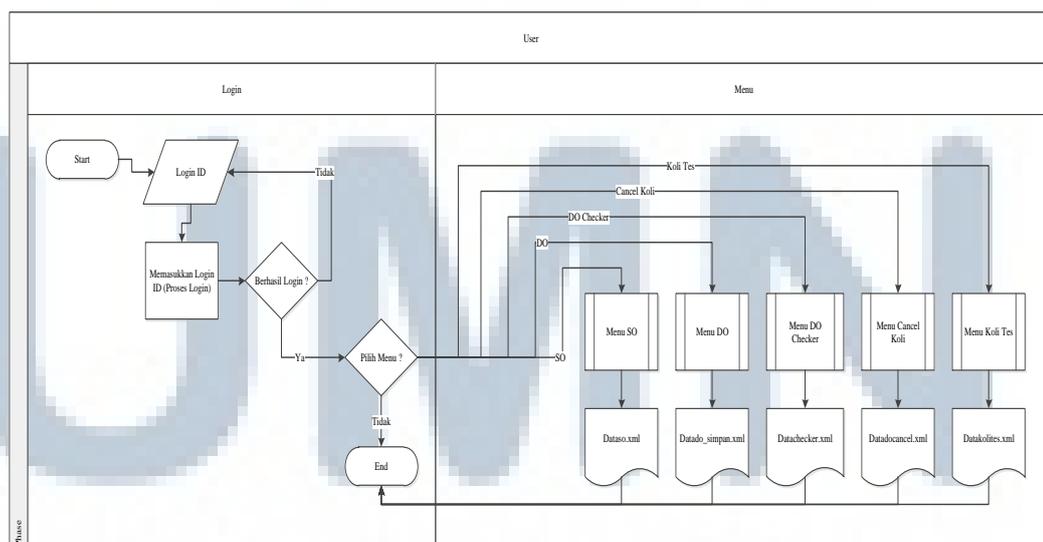
Pada tahap pertama dimulai dengan melakukan analisis terhadap mekanisme yang ingin dikembangkan, seperti bagaimana mekanisme akan dibangun, modul-modul apa saja yang harus dibuat, aplikasi apa saja yang dibutuhkan, serta mulai melakukan perancangan-perancangan. Setelah itu dapat dilakukan pengumpulan data, informasi, dan studi literatur yang berkaitan dengan topik yang ingin dikerjakan. Hal ini tetap dilakukan selama pengembangan modul dan kerja magang apabila ditemukan permasalahan dalam pengembangan.

B. Perancangan

1. Flowchart

Sistem flow untuk mekanisme ini ditunjukkan pada gambar dibawah ini.

a. Form *Login* dan Menu



Gambar 3.1. Flowchart form *Login* dan Menu

Keterangan halaman login

1) Pengguna memasukkan *login ID* ketika ingin menggunakan aplikasi.

a) Tidak

Jika tidak berhasil *login*, pengguna dapat mengulang dengan memasukkan *login ID*.

b) Ya

Jika berhasil *login*, pengguna dapat masuk ke halaman menu dan melakukan pilihan.

1. Modul SO (*Stock Opname*)

Jika memilih SO (*Stock Opname*), maka pengguna akan masuk ke dalam modul SO (*Stock Opname*).

2. Modul DO (*Delivery Order*)

Jika memilih DO (*Delivery Order*), maka pengguna akan masuk ke dalam modul DO (*Delivery Order*).

3. Modul DO (*Delivery Order*) Checker

Jika memilih DO (*Delivery Order*) Checker, maka pengguna akan masuk ke dalam modul DO (*Delivery Order*) Checker.

4. Modul Cancel Koli

Jika memilih Cancel Koli, maka pengguna akan masuk ke dalam modul Cancel Koli.

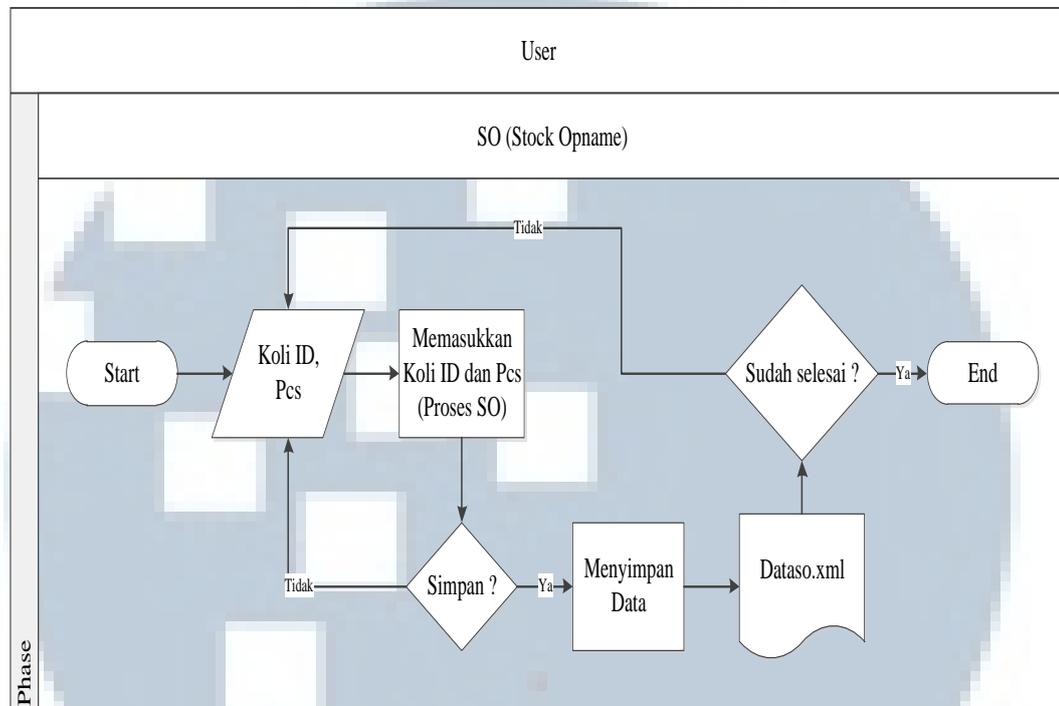
5. Modul Koli Tes

Jika memilih Koli Tes, maka pengguna akan masuk ke dalam modul Koli Tes.

6. Tidak

Jika tidak memilih modul apapun, pengguna dapat keluar dari aplikasi.

b. Modul SO (*Stock Opname*)



Gambar 3.2. Flowchart modul SO (*Stock Opname*)

Keterangan modul SO (*Stock Opname*)

1) Ketika pengguna memasukkan Koli ID dan Pcs saat menjalankan modul SO (*Stock Opname*), apakah data ingin disimpan ?

a) Tidak

Jika tidak disimpan, pengguna dapat mengulang dengan memasukkan Koli ID dan Pcs.

b) Ya

Jika ya, data akan disimpan ke dataso.xml. Kemudian apakah sudah selesai melakukan SO (*Stock Opname*) ?

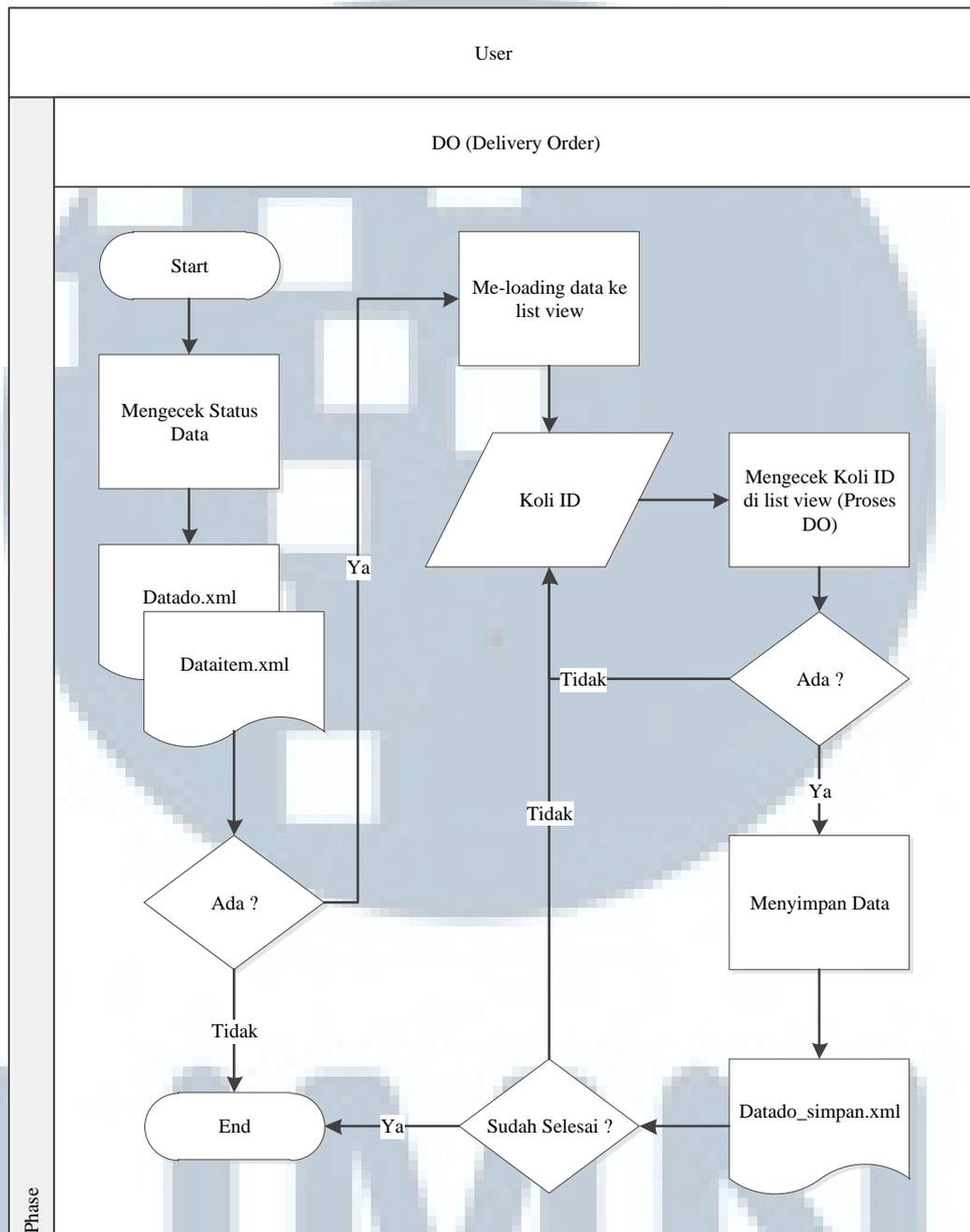
1. Tidak

Jika tidak, pengguna dapat melanjutkan dengan memasukkan Koli ID dan Pcs.

2. Ya

Jika ya, maka pengguna dapat keluar dari aplikasi.

c. Modul DO (*Delivery Order*)



Gambar 3.3. Flowchart modul DO (*Delivery Order*)

Keterangan modul DO (*Delivery Order*)

- 1) Modul akan mengecek apakah terdapat datado.xml dan dataitem.xml.

a) Tidak

Jika tidak ada, modul tidak dapat dijalankan dan pengguna akan secara otomatis keluar dari aplikasi.

b) Ya

Jika ada, modul akan melakukan *loading* data dan menampilkannya ke list view.

2) Setelah data sudah ditampilkan ke list view, pengguna dapat memasukkan Koli ID

3) Modul akan mengecek apakah Koli ID yang dimasukkan ada di list view.

a) Tidak

Jika tidak ada, pengguna dapat mengulang dengan memasukkan Koli ID.

b) Ya

Jika ada, data akan disimpan ke `datado_simpan.xml`. Kemudian apakah sudah selesai melakukan DO (*Delivery Order*) ?

1. Tidak

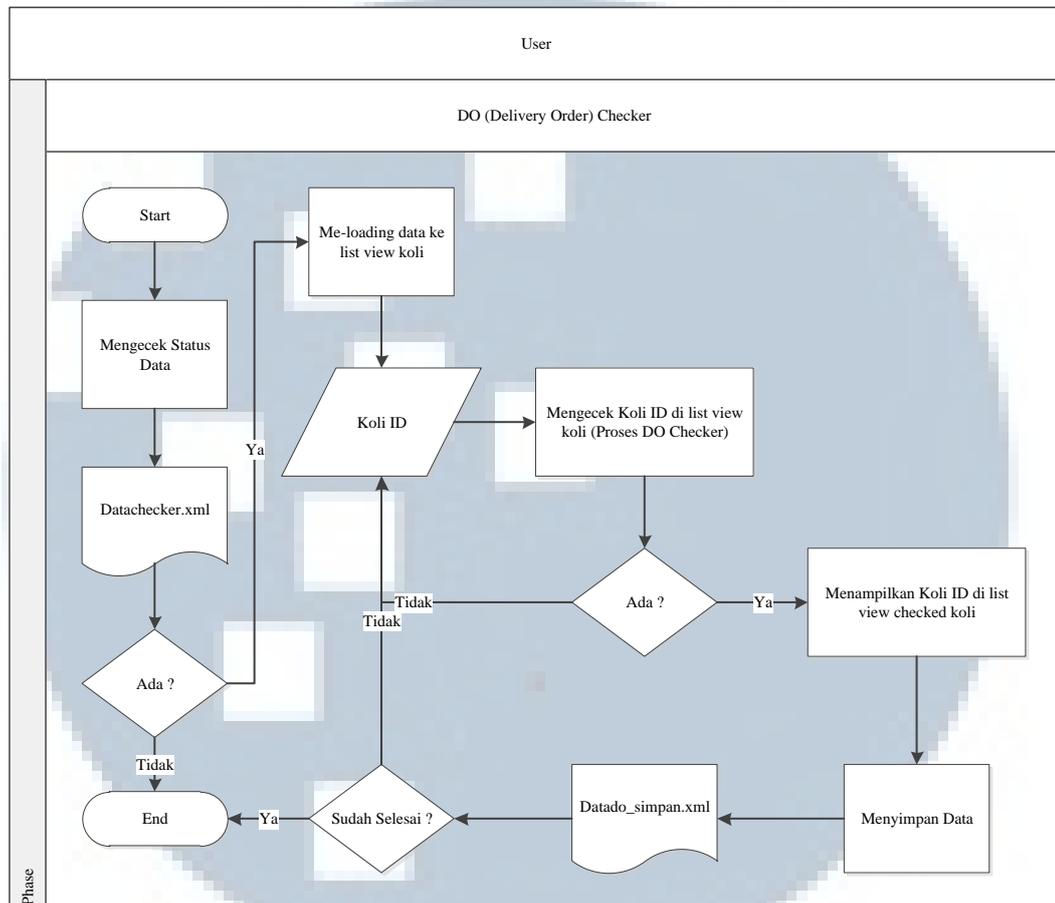
Jika tidak ada, pengguna dapat melanjutkan dengan memasukkan Koli ID.

2. Ya

Jika ya, maka pengguna dapat keluar dari aplikasi.

U
M
M
N

d. Modul DO (*Delivery Order*) Checker



Gambar 3.4. Flowchart modul DO (*Delivery Order*) Checker

Keterangan modul DO (*Delivery Order*) Checker

1) Modul akan mengecek apakah terdapat datachecker.xml.

a) Tidak

Jika tidak ada, modul tidak dapat dijalankan dan pengguna akan secara otomatis keluar dari aplikasi.

b) Ya

Jika ada, modul akan melakukan *loading* data dan menampilkannya ke list view koli.

2) Setelah data sudah ditampilkan ke list view koli, pengguna dapat memasukkan Koli ID

3) Modul akan mengecek apakah Koli ID yang dimasukkan ada di list view koli.

a) Tidak

Jika tidak ada, pengguna dapat mengulang dengan memasukkan Koli ID.

b) Ya

Jika ada, datachecker.xml akan di-*update*. Kemudian apakah sudah selesai melakukan DO (Delivery Order) Checker?

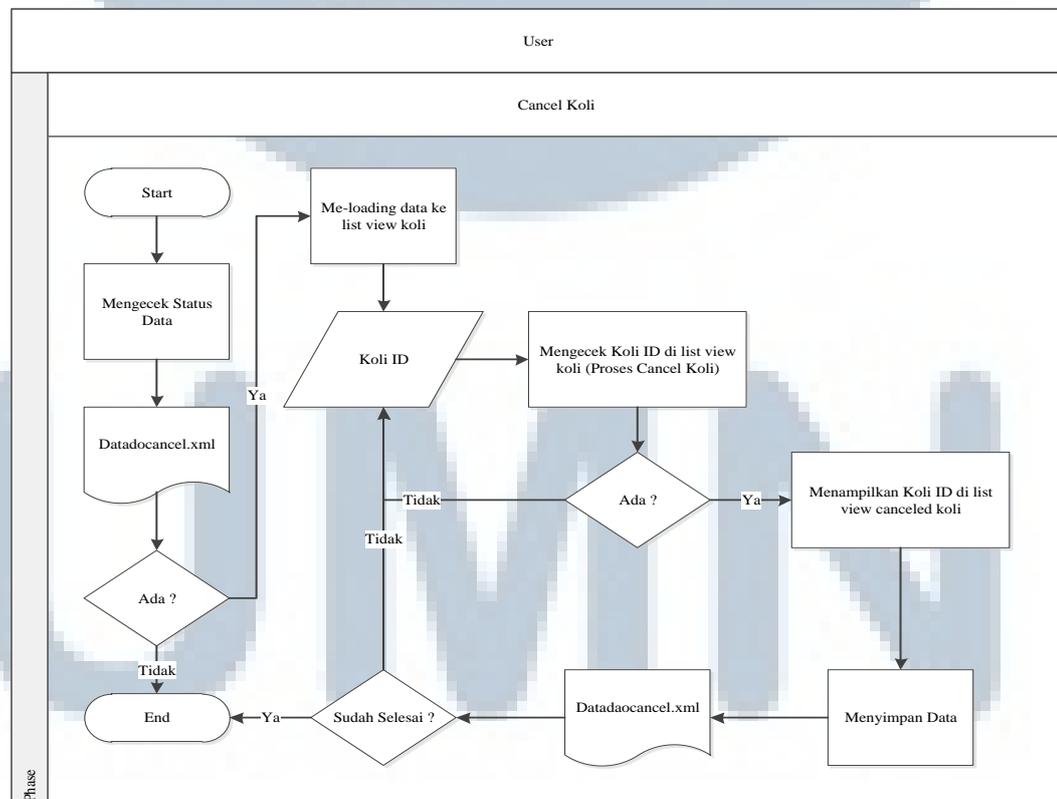
1. Tidak

Jika tidak ada, pengguna dapat melanjutkan dengan memasukkan Koli ID.

2. Ya

Jika ya, maka pengguna dapat keluar dari aplikasi.

e. Modul Cancel Koli



Gambar 3.5. Flowchart modul Cancel Koli

Keterangan modul Cancel Koli

1) Modul akan mengecek apakah terdapat `datadocancel.xml`.

a) Tidak

Jika tidak ada, modul tidak dapat dijalankan dan pengguna akan secara otomatis keluar dari aplikasi.

b) Ya

Jika ada, modul akan melakukan *loading* data dan menampilkannya ke list view koli.

2) Setelah data sudah ditampilkan ke list view koli, pengguna dapat memasukkan Koli ID

3) Modul akan mengecek apakah Koli ID yang dimasukkan ada di list view koli.

a) Tidak

Jika tidak ada, pengguna dapat mengulang dengan memasukkan Koli ID.

b) Ya

Jika ada, `datadocancel.xml` akan di-*update*. Kemudian apakah sudah selesai melakukan Cancel Koli ?

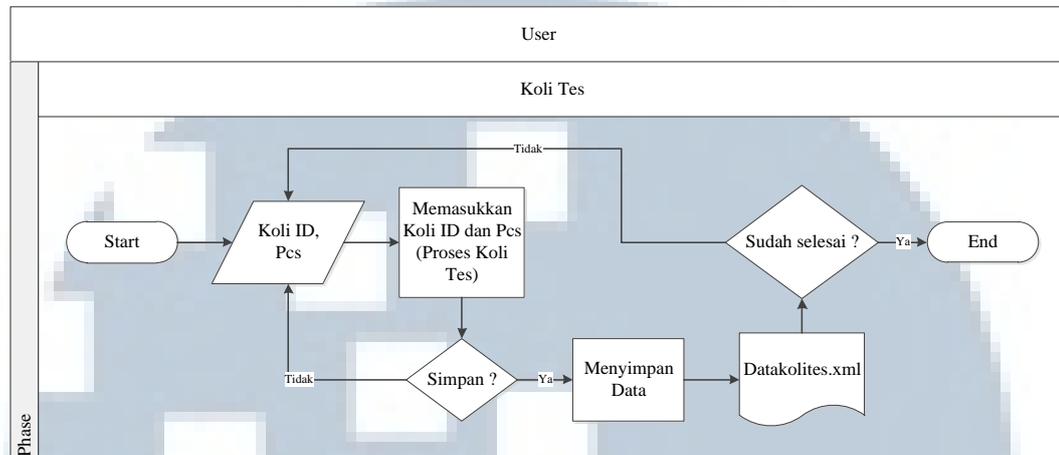
1. Tidak

Jika tidak ada, pengguna dapat melanjutkan dengan memasukkan Koli ID.

2. Ya

Jika ya, maka pengguna dapat keluar dari aplikasi.

f. Modul Koli Tes



Gambar 3.6. Flowchart modul Koli Tes

Keterangan modul Koli Tes

1) Ketika pengguna memasukkan Koli ID dan Pcs saat menjalankan modul Koli Tes, apakah data ingin disimpan ?

a) Tidak

Jika tidak disimpan, pengguna dapat mengulang dengan memasukkan Koli ID dan Pcs.

b) Ya

Jika ya, data akan disimpan ke datakolites.xml. Kemudian apakah sudah selesai melakukan Koli Tes ?

1. Tidak

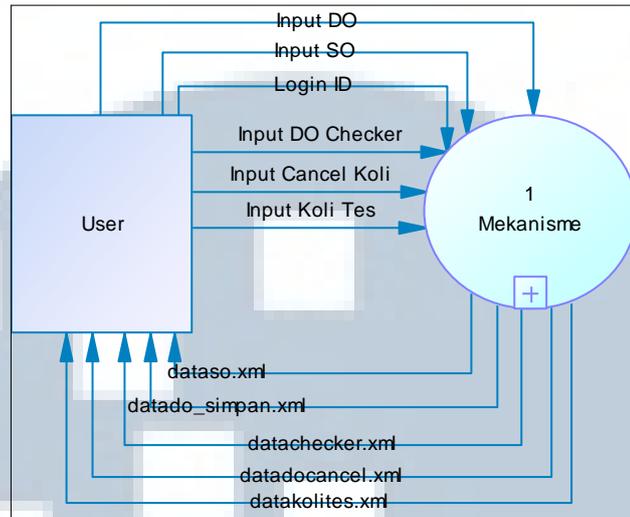
Jika tidak, pengguna dapat melanjutkan dengan memasukkan Koli ID dan Pcs.

2. Ya

Jika ya, maka pengguna dapat keluar dari aplikasi.

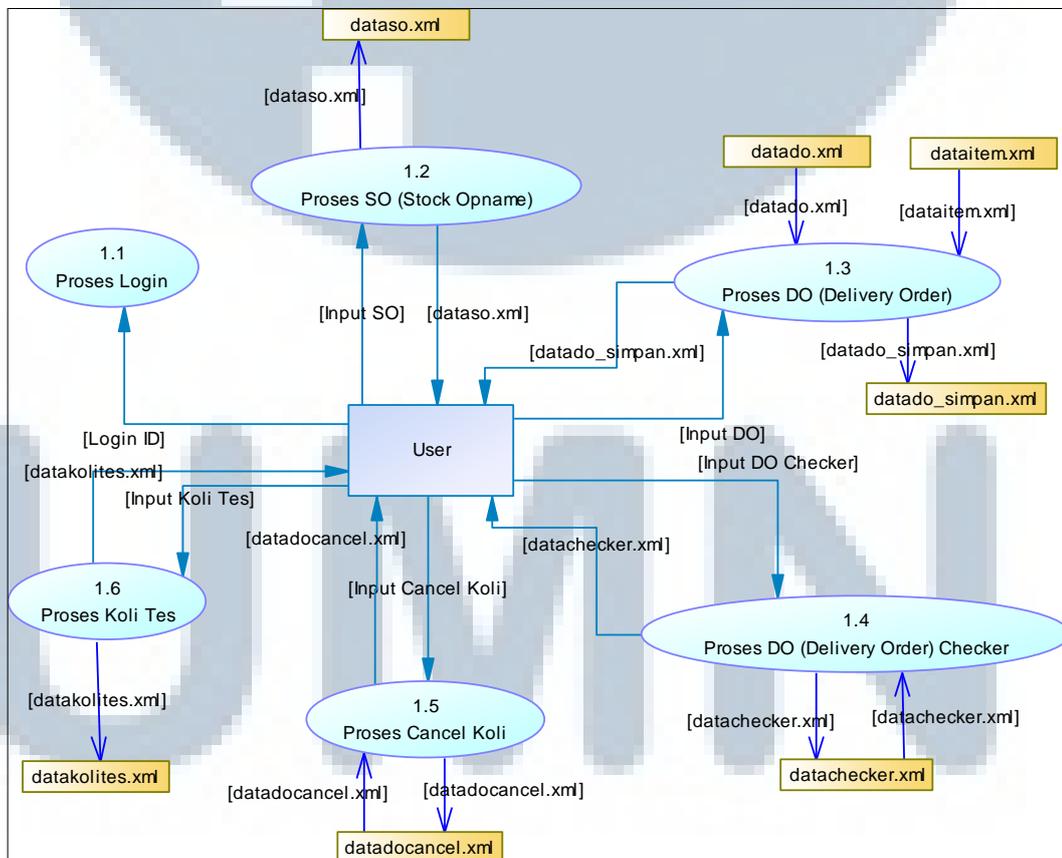
2. Data Flow Diagram

Mekanisme yang dikembangkan memiliki 5 modul, yaitu modu SO (*Stock Opname*), modul DO (*Delivery Order*), modul DO (*Delivery Order*) Checker, modul Cancel Koli, dan modul Koli Tes.



Gambar 3.7. Context Diagram Mekanisme

Dari *context diagram* diatas, terlihat hanya terdapat seorang user. Hal ini hanya ada satu orang user yang berinteraksi langsung dengan mekanisme. User memiliki hak penuh atas semua modul yang ada di dalam mekanisme.



Gambar 3.8. DFD Level 1 Mekanisme

Berdasarkan pada gambar DFD level 1, terlihat 6 proses yang dapat dilakukan oleh pengguna.

a. Proses *login*

Pada proses *login*, pengguna harus memasukkan *login ID* agar dapat menggunakan aplikasi

b. Proses SO (*Stock Opname*)

Pada proses SO (*Stock Opname*) pengguna dapat memasukkan *input* SO, yaitu Koli ID dan Pcs. Setelah modul selesai dijalankan, pengguna akan mendapatkan *dataso.xml*.

c. Proses DO (*Delivery Order*)

Pada proses DO (*Delivery Order*) pengguna dapat memasukkan *input* DO, yaitu Koli ID dan Pcs. Modul akan membaca data dari *datado.xml* dan *dataitem.xml* untuk menampilkan data. Setelah modul selesai dijalankan, pengguna akan mendapatkan *datado_simpan.xml*.

d. Proses DO (*Delivery Order*) Checker

Pada proses DO (*Delivery Order*) Checker pengguna dapat memasukkan *input* DO, yaitu Koli ID. Modul akan membaca *datachecker.xml* untuk menampilkan data. Setelah modul selesai dijalankan, pengguna akan mendapatkan *datachecker.xml*.

e. Proses Cancel Koli

Pada proses Cancel Koli pengguna dapat memasukkan *input* Cancel Koli, yaitu Koli ID. Modul akan membaca *datadocancel.xml* untuk menampilkan data. Setelah modul selesai dijalankan, pengguna akan mendapatkan *datadocancel.xml*.

f. Proses Koli Tes

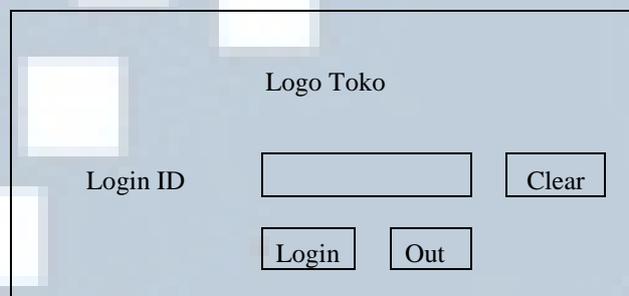
Pada proses Koli Tes pengguna dapat memasukkan *input* Koli Tes, yaitu Koli ID dan Pcs. Setelah modul selesai dijalankan, pengguna akan mendapatkan *datakolites.xml*.

3. Rancangan Antarmuka

Berdasarkan diskusi dan analisis tentang pengembangan mekanisme yang akan dilakukan, maka didapatkan persetujuan dan masukan untuk perancangan antarmuka.

a. Halaman *Login*

Halaman *login* adalah halaman yang pertama kali dilihat oleh user ketika menjalankan aplikasi. Pada halaman ini terdapat beberapa navigasi yang dapat dilakukan, yaitu mengisi *login ID*, menghapus *login ID*, melakukan *login*, dan keluar dari aplikasi.



Logo Toko

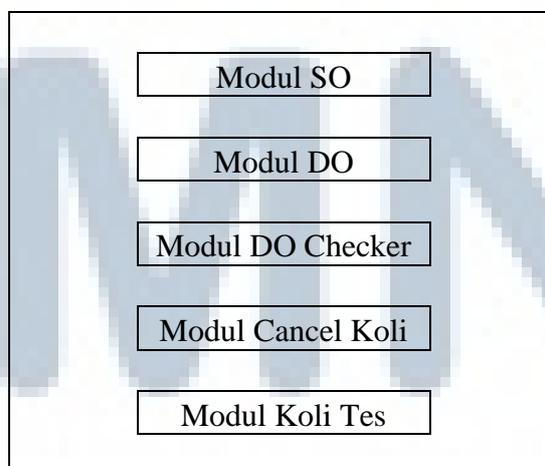
Login ID Clear

Login Out

Gambar 3.9. Halaman *login*

b. Halaman Menu

Halaman menu adalah halaman yang diakses setelah pengguna berhasil melakukan *login*. Halaman ini dirancang untuk menampilkan pilihan modul yang dapat digunakan untuk proses *scanning*.



Modul SO

Modul DO

Modul DO Checker

Modul Cancel Koli

Modul Koli Tes

Gambar 3.10. Halaman Menu

c. Modul SO (*Stock Opname*)

Modul SO (*Stock Opname*), digunakan untuk memasukkan data produk baru sekaligus sebagai penanda bahwa produk tersebut memang ada dan dapat didistribusikan. Pada *textbox* Koli ID, akan muncul hasil dari *scan*. Kemudian, pengguna juga dapat mengisi jumlah produk pada *textbox* Pcs. Apabila ingin melanjutkan *scanning* pengguna harus menekan tombol 'Continue' dan tombol 'Clear' jika ingin menghapus hasil *scan*. *Textbox* Jumlah Koli akan otomatis menghitung jumlah koli yang sudah di-*scan*. Untuk keluar dari modul, pengguna harus menekan tombol 'Finish'. Tombol 'Reset' digunakan untuk mengulang proses *scanning* yang telah dilakukan.

Koli ID	<input type="text"/>
Pcs	<input type="text"/>
	<input type="button" value="Continue"/> <input type="button" value="Clear"/>
SEPARATOR	
Jumlah Koli	<input type="text" value="Auto Generated"/>
	<input type="button" value="Finish"/> <input type="button" value="Reset"/>

Gambar 3.11. Modul SO (*Stock Opname*)

d. Modul DO (*Delivery Order*)

Modul DO (*Delivery Order*), digunakan untuk memasukkan data produk yang akan didistribusikan ke toko. Pada modul ini pengguna hanya dapat mengisi *textbox* Koli ID dengan melakukan *scan*. Tombol 'OK' untuk mengonfirmasi dan tombol 'Clear' menghapus hasil *scan*. Untuk keluar dari modul, pengguna harus menekan tombol 'Finish'. Tombol 'Reset' digunakan untuk mengulang proses *scanning* yang telah dilakukan.

The screenshot shows a software interface for the Delivery Order (DO) module. It contains several input fields and buttons. At the top, there are three rows of fields: 'ID Toko' with 'Auto Generated', 'PLU' with 'Auto Generated', and 'Pcs' with 'Auto Generated'. To the right of these is a 'Jumlah Scan' field with 'Auto Generated'. Below these is a 'Koli ID' text box, followed by 'OK' and 'Clear' buttons. A large box labeled 'Auto Generated List View' is in the center. At the bottom, there are 'Tot. Pcs' and 'Tot. CBM' fields, both with 'Auto Generated' values, and 'Finish' and 'Reset' buttons.

Gambar 3.12. Modul DO (*Delivery Order*)

Berikut daftar informasi yang ditampilkan secara otomatis dari data yang ada didalam modul DO (*Delivery Order*).

Tabel 3.2. Informasi yang ditampilkan dalam modul DO

Informasi	Deskripsi
Jumlah Scan	Jumlah produk yang sudah di- <i>scan</i>
ID Toko	ID Toko dimana DO akan dikirim
PLU	<i>Product Label Unit</i> /Kode produk
Pcs	Jumlah produk
List View	Menampilkan informasi produk yang bisa di DO
Tot. Pcs	Total produk
Tot. CBM	Total <i>volume</i>

e. Modul DO (*Delivery Order*) Checker

Modul DO (*Delivery Order*) Checker, digunakan untuk memeriksa apakah ada kesalahan pada *Delivery Order*. Pada modul ini pengguna hanya dapat mengisi *textbox* Koli ID dengan melakukan *scan*. Tombol

‘OK’ untuk mengonfirmasi dan tombol ‘Clear’ menghapus hasil *scan*. Untuk keluar dari modul, pengguna harus menekan tombol ‘Finish’. Tombol ‘Reset’ digunakan untuk mengulang proses *scanning* yang telah dilakukan.

The screenshot shows a software interface for a Delivery Order (DO) checker. It features several input fields and buttons. The 'ID Toko' field is pre-filled with 'Auto Generated'. Below it is an empty 'Koli ID' field. There are two buttons: 'OK' and 'Clear'. Below these are two more buttons: 'Auto Generated Koli List view' and 'Auto Generated Checked List view'. At the bottom, there are two more buttons: 'Finish' and 'Reset'. The 'Checked Total Koli' field is also pre-filled with 'Auto Generated'.

Gambar 3.13. Modul DO (*Delivery Order*) Checker

Berikut daftar informasi yang ditampilkan secara otomatis dari data yang ada didalam modul DO Checker.

Tabel 3.3. Informasi yang ditampilkan dalam modul DO Checker

Informasi	Deskripsi
ID Toko	ID Toko dimana DO akan dikirim
Koli List View	Menampilkan informasi koli hasil DO yang dapat di <i>check</i>
Checked List View	Menampilkan informasi koli hasil DO yang sudah di <i>check</i>
Checked	Jumlah koli yang telah di <i>check</i>
Total Koli	Total koli

f. Modul Cancel Koli

Modul Cancel Koli, digunakan untuk membatalkan pengiriman suatu produk. Pada modul ini pengguna hanya dapat mengisi *textbox* Koli ID dengan melakukan *scan*. Tombol ‘OK’ untuk mengonfirmasi dan tombol ‘Clear’ menghapus hasil *scan*. Untuk keluar dari modul, pengguna harus menekan tombol ‘Finish’. Tombol ‘Reset’ digunakan untuk mengulang proses *scanning* yang telah dilakukan. Apabila pengguna menekan tombol ‘Final Save’, maka proses tidak dapat di-*reset*.

Gambar 3.14. Modul Cancel Koli

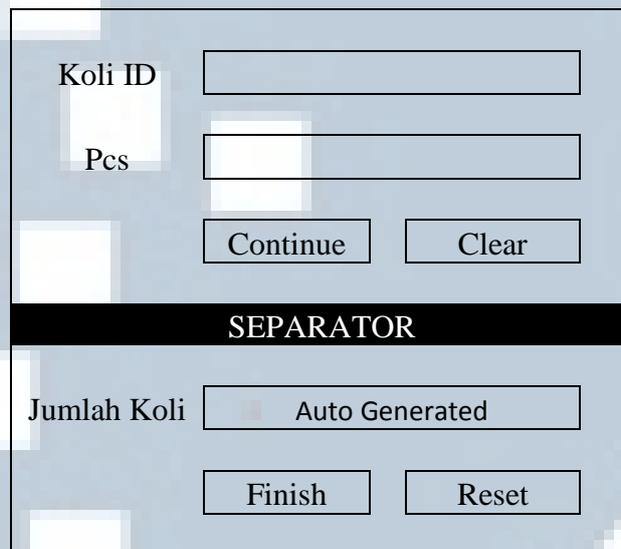
Berikut daftar informasi yang ditampilkan secara otomatis dari data yang ada didalam modul Cancel Koli.

Tabel 3.4. Informasi yang ditampilkan dalam modul Cancel Koli

Informasi	Deskripsi
ID Toko	ID Toko dimana DO akan dikirim
Koli List View	Menampilkan informasi koli hasil DO yang dapat di <i>cancel</i>
Canceled List View	Menampilkan informasi koli hasil DO yang sudah di <i>cancel</i>

g. Modul Koli Tes

Modul Koli Tes, digunakan untuk menghindari kecurangan yang dapat dilakukan oleh petugas di gudang sehingga menjamin bahwa produk yang dikirim akan sama jumlahnya dengan yang diterima oleh toko. *Layout* dan fungsi yang dimiliki sama persi dengan modul SO (*Stock Opname*), hanya saja data yang dihasilkan berbeda.



Koli ID	<input type="text"/>
Pcs	<input type="text"/>
	<input type="button" value="Continue"/>
	<input type="button" value="Clear"/>
SEPARATOR	
Jumlah Koli	<input type="checkbox"/> Auto Generated
	<input type="button" value="Finish"/>
	<input type="button" value="Reset"/>

Gambar 3.15. Modul Koli Tes

C. Implementasi

Setelah melakukan perancangan antarmuka, tahap selanjutnya dilakukan implementasi rancangan kedalam program. Berikut adalah hasil *print screen* dari pembuatan aplikasi.

U
M
M
N

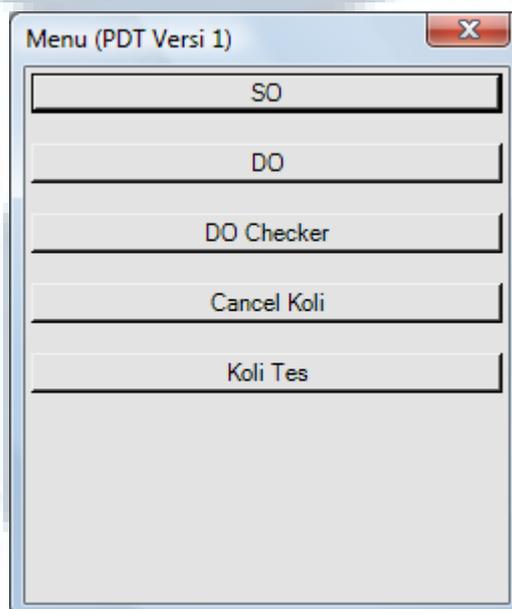
1. Halaman *Login*



Gambar 3.16. Implementasi *Login*

Pada halaman *login*, pengguna harus memasukkan *login ID* agar dapat masuk kedalam aplikasi.

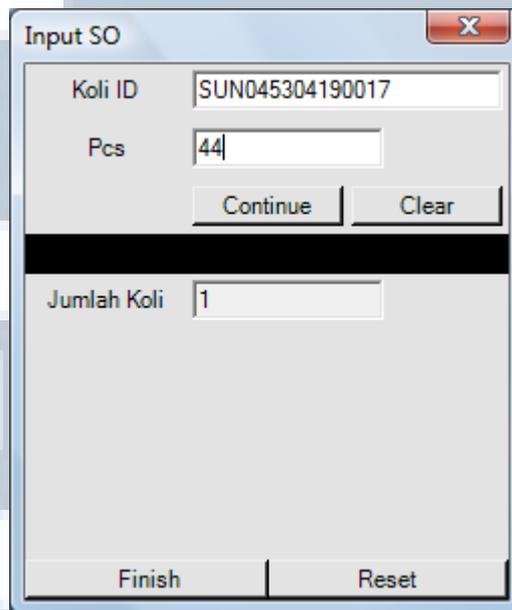
2. Halaman *Menu*



Gambar 3.17. Implementasi *Menu*

Setelah berhasil *login*, akan muncul halaman menu dimana pengguna dapat memilih modul. Dari gambar, terlihat bahwa terdapat modul SO (*Stock Opname*), modul DO (*Delivery Order*), modul DO (*Delivery Order*) Checker, modul Cancel Koli, modul Koli Tes.

3. Modul SO (*Stock Opname*)



Koli ID	SUN045304190017
Pcs	44
Jumlah Koli	1

Gambar 3.18. Implementasi Modul SO (*Stock Opname*)

Pada proses SO (*Stock Opname*) pengguna dapat memasukkan *input* SO, yaitu Koli ID dan Pcs. Setelah modul selesai dijalankan, pengguna akan mendapatkan *dataso.xml*.

4. Modul DO (*Delivery Order*)

No	Koli	PLU	Pcs
1	SUN005008180006	00009098	36
2	SUN005008250025	00005852	24
3	SUN005008250026	00009011	36
4	SUN005008250027	00013316	24
5	SUN005008250028	00013317	36
6	SUN005008250029	00013870	36
7	SUN005008250030	00013871	36

Tot. Pcs 36 Tot. CBM 0.0126

Gambar 3.19. Implementasi Modul DO (*Delivery Order*)

Pada proses DO (*Delivery Order*) pengguna dapat memasukkan *input* DO, yaitu Koli ID dan Pcs. Modul akan membaca data dari *datado.xml* dan *dataitem.xml* untuk menampilkan data. Setelah modul selesai dijalankan, pengguna akan mendapatkan *datado_simpan.xml*.

5. Modul DO (*Delivery Order*) Checker

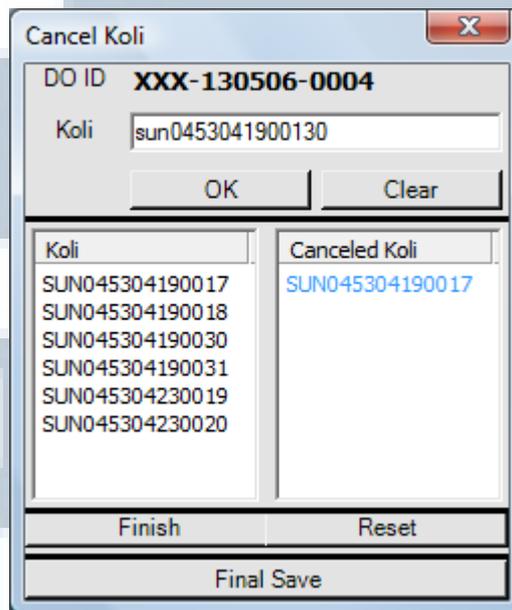
Koli	Checked Koli
SUNXA1307220001	SUNXA1307220001
SUNXA1307220005	
SUNXA1307220007	

Checked 1 dari Total Koli 3

Gambar 3.20. Implementasi Modul DO (*Delivery Order*) Checker

Pada proses DO (*Delivery Order*) Checker pengguna dapat memasukkan *input* DO, yaitu Koli ID. Modul akan membaca *datachecker.xml* untuk menampilkan data. Setelah modul selesai dijalankan, pengguna akan mendapatkan *datachecker.xml*.

6. Modul Cancel Koli



Koli	Canceled Koli
SUN045304190017	SUN045304190017
SUN045304190018	
SUN045304190030	
SUN045304190031	
SUN045304230019	
SUN045304230020	

Gambar 3.21. Implementasi Modul Cancel Koli

Pada proses Cancel Koli pengguna dapat memasukkan *input* Cancel Koli, yaitu Koli ID. Modul akan membaca *datadocancel.xml* untuk menampilkan data. Setelah modul selesai dijalankan, pengguna akan mendapatkan *datadocancel.xml*.

7. Modul Koli Tes

Gambar 3.22. Implementasi Modul Koli Tes

Pada proses Koli Tes pengguna dapat memasukkan *input* Koli Tes, yaitu Koli ID dan Pcs. Setelah modul selesai dijalankan, pengguna akan mendapatkan *datakolites.xml*.

3.3.2. Kendala yang Ditemukan

Berikut adalah kendala yang ditemukan dalam pengembangan modul Sales Order dan Delivery Order yang dilakukan dalam kurun waktu dua bulan.

- a. Pada saat melakukan instalasi dan *setup software* yang dibutuhkan, antara SDK WinCE Emulator, versi Visual Studio, dan versi Operating Sistem tidak saling *compatible*. Oleh karenanya pengembangan tidak dapat dikerjakan.
- b. Ketika sebuah modul telah berhasil dibuat dan ingin dilakukan uji coba, mekanisme selalu membutuhkan *portable data terminal*. Tidak bisa diuji coba secara langsung. Hal ini cukup memakan waktu karena harus memasang dan menghubungkan *portable data terminal* ke komputer.
- c. Sulitnya konfigurasi WinCE Emulator untuk dihubungkan dengan Visual Studio 2008.

- d. Seringnya terjadi perubahan kebutuhan, baik dari sisi rancangan maupun dari sisi pengerjaan. Hal ini membuat waktu menjadi sedikit lebih lama dari yang direncanakan.

3.3.3. Solusi Atas Kendala yang Ditemukan

Berdasarkan kendala-kendala yang ditemukan tersebut, berikut adalah solusi yang dapat digunakan untuk mengatasinya.

- a. Mengumpulkan informasi dari internet. Kemudian mulai mengganti aplikasi serta mengatur agar aplikasi dan sistem dapat saling *compatible*. Dengan begitu, proses pengerjaan dan pengembangan dapat dilakukan.
- b. Menggunakan sebuah aplikasi *emulator* sebagai pengganti *portable data terminal*, sehingga pada saat ingin melakukan uji coba tidak perlu untuk memasang dan menghubungkan *portable data terminal* ke komputer.
- c. Mencari informasi konfigurasi yang tepat untuk menghubungkan aplikasi WinCE Emulator dengan Visual Studio 2008 dari internet.
- d. Melakukan komunikasi dan konsultasi dengan pihak-pihak terkait. Kemudian melakukan persiapan dan perancangan dengan baik serta pengumpulan kebutuhan secara lengkap sebelum memulai tahap pembangunan.

U M N