



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, penulis berkedudukan sebagai *programmer* pada divisi IT yang dikoordinasi langsung oleh Sony Raditya Suryawijaya selaku direktur dan pembimbing lapangan penulis. Tugas yang diberikan kepada tim penulis untuk membuat sistem informasi yang bisa dikustomisasi sesuai keinginan agar bisa diimplementasikan ke perusahaan baru yang akan dibuat nantinya.

3.2 Tugas yang Dilakukan

Pembangunan sistem yang dikerjakan selama kerja magang menggunakan *System Development Life Cycle Waterfall*. *Waterfall* merupakan siklus yang terdiri dari analisis dan definisi persyaratan, perancangan sistem dan perangkat lunak, implementasi dan pengujian unit, integrasi dan pengujian sistem, operasi dan pemeliharaan. Namun, dalam kerja magang ini penulis hanya melakukan sampai dengan tahap pengujian unit. Sebagai gambaran berikut tabel yang menunjukkan kegiatan yang dikerjakan penulis selama kerja magang setiap minggunya.

Tabel 3.1 Jadwal Kerja Magang

No	Kegiatan	Juni	Juli				Agustus			
		4	1	2	3	4	1	2	3	4
1	Studi literature									
2	Analisa kebutuhan									
3	Perancangan sistem									
4	Pembuatan sistem									
5	Pengujian sistem									

3.2.1 Studi Literatur

Kerja magang penulis diawali dengan mempelajari dan menganalisa kebutuhan sistem dengan tim penulis. Penulis bersama dengan timnya mencatat kebutuhan sistem dari user dan menganalisa kebutuhan-kebutuhan tersebut. Bersamaan dengan itu penulis melakukan studi literatur dengan membaca materi tentang bahasa pemrograman C#, serta database Microsoft SQL Server 2008 dan 2012. Hal tersebut diperlukan karena program yang akan dibuat menggunakan bahasa pemrograman C# serta terintegrasi dengan database Microsoft SQL Server 2008 yang berada di internet. Materi tersebut dicari oleh penulis melalui internet.

Untuk bahasa pemrograman C#, penulis mempelajari beberapa hal tentang C# dari materi perkuliahan tetapi penulis merasa pada materi perkuliahan belum mencakup banyak hal sehingga penulis merasa perlu untuk mendalami C# sebelum membuat program tersebut. Penulis juga mempelajari materi Microsoft SQL Server 2008 dari internet. Penulis mempunyai pengalaman menggunakan Microsoft SQL Server sebelumnya sehingga pengalaman tersebut sangat membantu penulis dalam pengerjaan program ini.

Studi literatur juga penulis lakukan sepanjang kerja magang untuk mengantisipasi jika terdapat masalah atau error dalam proses pembuatan sistem. Oleh karena itu penulis juga mencari informasi melalui internet, terutama literatur C# dan SQL Server 2008.

3.2.2 Analisa Kebutuhan

Tahapan kerja magang yang selanjutnya yaitu analisa kebutuhan, dimana penulis menganalisa kebutuhan-kebutuhan apa saja yang diperlukan dalam sistem tersebut. Kebutuhan-kebutuhan tersebut antara lain sebagai berikut.

1. *Login* memiliki fungsi untuk login ke dalam sistem yang akan dibuat serta fungsi untuk logout dari sistem yang akan dibuat. *User* yang akan *login* akan dibagi menjadi 2 jenis yaitu *user* dan *admin*. *User* yang akan *login* akan dibatasi sesuai tipe *user*. Setelah login *user* akan diberikan menu tersendiri sesuai tipe *user*. Untuk menjaga sekuritas maka ada maksimal *login* yang tidak berhasil

untuk mencegah serangan *brute force* dan kemudian *id* akan dikunci apabila hal tersebut terjadi.

2. *User Management* yang memiliki fungsi untuk melihat, menambahkan, merubah data user yang ada. Selain itu juga harus dapat untuk merubah *password user* apabila *user* lupa *password*. Bisa juga menghilangkan *block* pada *user*.
3. *User Type Management* yang memiliki fungsi untuk melihat, menambahkan, merubah data tipe *user* yang ada. Datanya juga bisa dinonaktifkan atau diaktifkan kembali.
4. *Plant Management* yang memiliki fungsi untuk melihat, menambahkan, merubah data *plant*. *Plant* disini adalah gedung tempat produksi barang. Datanya juga bisa dinonaktifkan atau diaktifkan kembali.
5. *Production Line Management* yang memiliki fungsi untuk melihat, menambahkan, merubah data *production line*. *Production Line* adalah lini produksi barang. Datanya juga bisa dinonaktifkan atau diaktifkan kembali.
6. *Storage Location Management* yang memiliki fungsi untuk melihat, menambahkan, merubah data *storage location*. *Storage Location* adalah tempat untuk menyimpan barang. Datanya juga bisa dinonaktifkan atau diaktifkan kembali.
7. *Storage Location Type Management* yang memiliki fungsi untuk melihat, menambahkan, merubah data tipe *storage location*. *Storage Location Type* ini berfungsi untuk menentukan jenis gudang. Datanya juga bisa dinonaktifkan atau diaktifkan kembali.
8. *Base of Measure Management* yang memiliki fungsi untuk melihat, menambahkan, merubah data satuan dasar. Satuan dasar ini dipakai untuk setiap macam item, misal sentimeter. Datanya juga bisa dinonaktifkan atau diaktifkan kembali.
9. *Item Management* yang memiliki fungsi untuk melihat, menambahkan, dan merubah data barang. Datanya juga bisa dinonaktifkan atau diaktifkan kembali.

10. *Item Type Management* yang memiliki fungsi untuk melihat, menambahkan, dan merubah data tipe barang. Datanya juga bisa dinonaktifkan atau diaktifkan kembali.
11. *Status Item Management* yang memiliki fungsi untuk melihat, menambahkan, dan merubah data status barang. Status barang contohnya barang jadi atau bahan baku. Datanya juga bisa dinonaktifkan atau diaktifkan kembali.
12. *Goods Movement* yang berfungsi untuk memindahkan barang antar tempat penyimpanan barang (*Storage Location*). Datanya juga bisa dinonaktifkan atau diaktifkan kembali.
13. *Stock Item Management* untuk melihat stok barang yang tersedia dan informasi tentang barang tersebut. Stok yang ada harus bisa dilacak sehingga apabila ada masalah bisa diketahui sumbernya dari mana.
14. Modul *Procurement* yang berfungsi untuk membeli barang dari supplier.
15. *Goods Receipt* yang berfungsi untuk menerima barang yang dikirim dari *supplier* dan memasukkannya ke gudang.
16. *Shipment* yang berfungsi untuk mengirimkan barang ke pembeli.
17. *Customer Management* yang memiliki fungsi untuk melihat, menambahkan, dan mengubah data pembeli. Datanya juga bisa dinonaktifkan atau diaktifkan kembali.
18. *Supplier Management* yang memiliki fungsi untuk melihat, menambahkan, dan mengubah data penjual. Datanya juga bisa dinonaktifkan atau diaktifkan kembali.
19. *Database Management* yang memiliki fungsi untuk melihat, menambahkan, dan mengubah data *database*. *Database* disini dipakai apabila ingin mengalihkan user ke *database* tertentu. Datanya juga bisa dinonaktifkan atau diaktifkan kembali.
20. Modul produksi untuk memproduksi barang.
21. *Overview* untuk melihat data produksi dengan singkat. Misalnya data produksi per minggu dalam grafik.

Oleh karena banyaknya modul di atas, maka penulis memfokuskan ke modul yang merupakan inti dari sub sistem yang dikerjakan penulis yaitu sebagai berikut:

- *Login*
- *Goods Receipt*
- Modul Produksi

3.2.3 Perancangan Sistem

Tahapan setelah analisa kebutuhan yaitu perancangan sistem. Pada tahap perancangan ini penulis membuat *entity relationship diagram*, *data flow diagram*, dan rancangan antarmuka sistem yang akan dibuat nanti. Hal-hal tersebut diperlukan karena menjadi gambaran dari hubungan antar tabel berdasarkan *entity relationship diagram* dan juga alur data yang diperoleh dari *data flow diagram*. Rancangan antarmuka diperlukan untuk menjadi suatu acuan tampilan dari sistem yang akan dibuat nantinya.

Pada pembangunan sistem ini diperlukan beberapa perangkat keras yang dapat menunjang pembuatan. Berikut perangkat keras yang dibutuhkan oleh penulis.

1. Laptop Asus N43SL, dengan spesifikasi sebagai berikut
 - a. *Processor* Intel i7 2.00GHz
 - b. *Hard disk* dengan kapasitas total 640GB
 - c. RAM dengan kapasitas 8GB
 - d. VGA Nvidia GeForce 540M CUDA 2GB
2. *Mouse* Steelseries Xai, yang berfungsi untuk memudahkan navigasi kursor.

3.2.4 Pembuatan Sistem

Dalam pembuatan sistem diperlukan beberapa perangkat lunak untuk menunjang dalam kerja magang ini. Berikut penjabaran dari perangkat lunak yang digunakan.

1. Sistem operasi Windows 7 Ultimate 64bit
2. Visual Studio 2012 Ultimate sebagai *IDE* untuk mengembangkan program ini.

3. Team Foundation Server 2012 Express sebagai perangkat lunak untuk mengatur pengerjaan tim. Perangkat lunak ini berfungsi untuk mengoptimisasi mengembangkan sistem secara tim.
4. SQL Server Management Studio, sebagai perangkat lunak untuk mengakses *database* yang ada di *internet*.
5. Google Chrome Versi 31.0.1650.57 sebagai browser untuk membuka web management untuk *database* yang ada di *internet*.

3.2.5 Pengujian Sistem

Tahap pengujian sistem merupakan suatu tahapan dimana sistem yang telah dibuat dijalankan dan diuji dengan menggunakan data sementara yang sengaja dibuat untuk mengisi database. Pengujian sistem ini dilakukan untuk mengetahui sistem yang telah dibuat bisa berjalan dan berfungsi dengan baik atau tidak, serta apakah masih terdapat masalah-masalah dalam sistem tersebut. Adapun spesifikasi minimal dan rekomendasi perangkat lunak maupun perangkat keras ialah sebagai berikut.

1. .NET Framework 4.5 sebagai framework untuk menjalankan program C#.
2. Sistem operasi Windows 7.
3. *Processor* Intel Core Duo.
4. RAM sebesar 2 GB.

3.2.6 Penulisan Laporan

Penulisan laporan dilakukan pada saat memasuki menyelesaikan kerja magang. Sebagai gambaran dalam pelaksanaan kerja magang, tabel berikut menunjukkan realisasi kerja magang selama menjalankan kerja magang pada PT Golden Sun Indonesia.

Tabel 3.2 Realisasi Kerja Magang

Minggu ke	Kegiatan
1	<ul style="list-style-type: none"> - Menerima <i>requirement</i> program - Menentukan <i>design flow</i> program - Merancang <i>user interface</i> dasar - Instalasi program yang diperlukan seperti Team Foundation Server - Perancangan <i>database</i>
2	<ul style="list-style-type: none"> - Pembuatan <i>design user interface</i> awal - Pembuatan basis kode - Pembuatan <i>data access layer</i>
3	<ul style="list-style-type: none"> - Pengerjaan Data Access Layer bagian <i>create, read, update, disable, enable</i>. - Pengecekan <i>bug</i>
4	<ul style="list-style-type: none"> - Finishing DAL - Pembuatan <i>core logic</i> program serta UI Program
5	<ul style="list-style-type: none"> - Pembuatan modul logic untuk pembelian dan penerimaan barang

Tabel 3.2 Realisasi Kerja Magang (Lanjutan)

Minggu ke	Kegiatan
6	<ul style="list-style-type: none"> - Perubahan desain UI dan penambahan fitur program - Memulai modul stock
7	<ul style="list-style-type: none"> - Pembuatan modul logic transformation (Perubahan barang) serta Shipment
8	<ul style="list-style-type: none"> - Penyelesaian modul stock serta pengecekan program secara keseluruhan
9	<ul style="list-style-type: none"> - Pembuatan modul overview serta dokumentasi - Pengecekan terakhir program

3.3 Uraian Pelaksanaan Kerja Magang

Pelaksanaan kerja magang ini dapat diuraikan menjadi tiga yaitu proses pelaksanaan, kendala yang ditemukan dan solusi atas kendala yang ditemukan. Berikut ini merupakan penjelasan dari pelaksanaan kerja magang.

3.3.1 Proses pelaksanaan

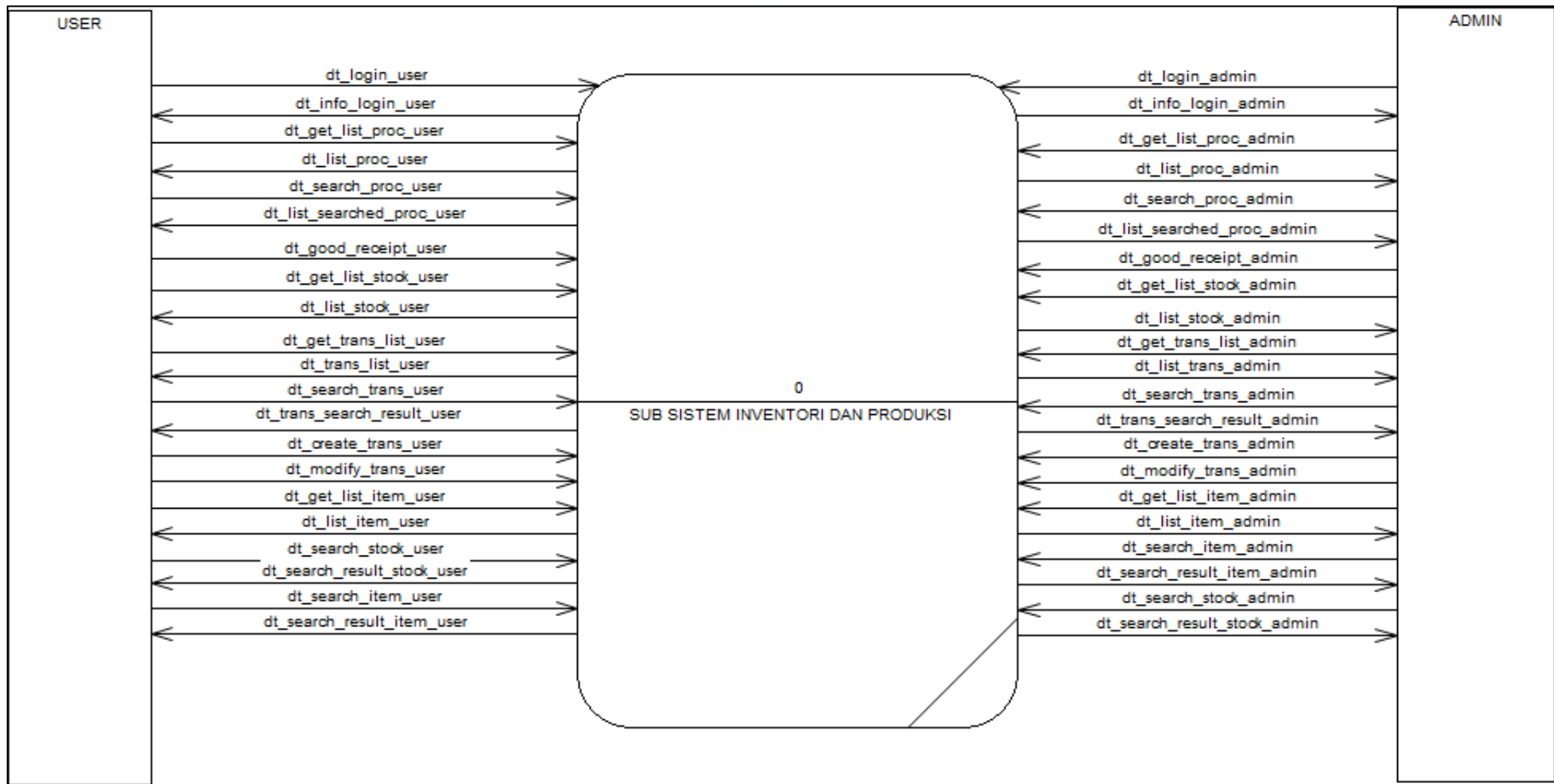
Ada beberapa hal yang dilakukan pada waktu melaksanakan kerja magang, mulai dari perancangan sistem yang meliputi *Data Flow Diagram*, *entity relationship diagram* dan juga struktur tabel, lalu dilanjutkan dengan desain antarmuka, dan terakhir adalah implementasi. Berikut penjelasan dari masing-masing hal tersebut.

A. Perancangan Sistem

Model yang digunakan pada perancangan sistem ini adalah dengan metode prosedural sehingga yang perlu dibuat adalah *Data Flow Diagram*, dan *Entity Relationship Diagram*.

A.1 Data Flow Diagram

Main proses dari sub sistem inventori dan produksi ini adalah *Sub Sistem Inventori dan Produksi*, dimana *Context diagram* ini terdapat dua entitas yaitu *User* serta *Admin* dengan *data flow* menuju ke proses yang berarti terdapat suatu input yang diberikan kepada proses ataupun *data flow* yang mengarah kepada entitas yang berarti terdapat data yang ditunjukkan atau *feedback* kepada entitas tersebut.

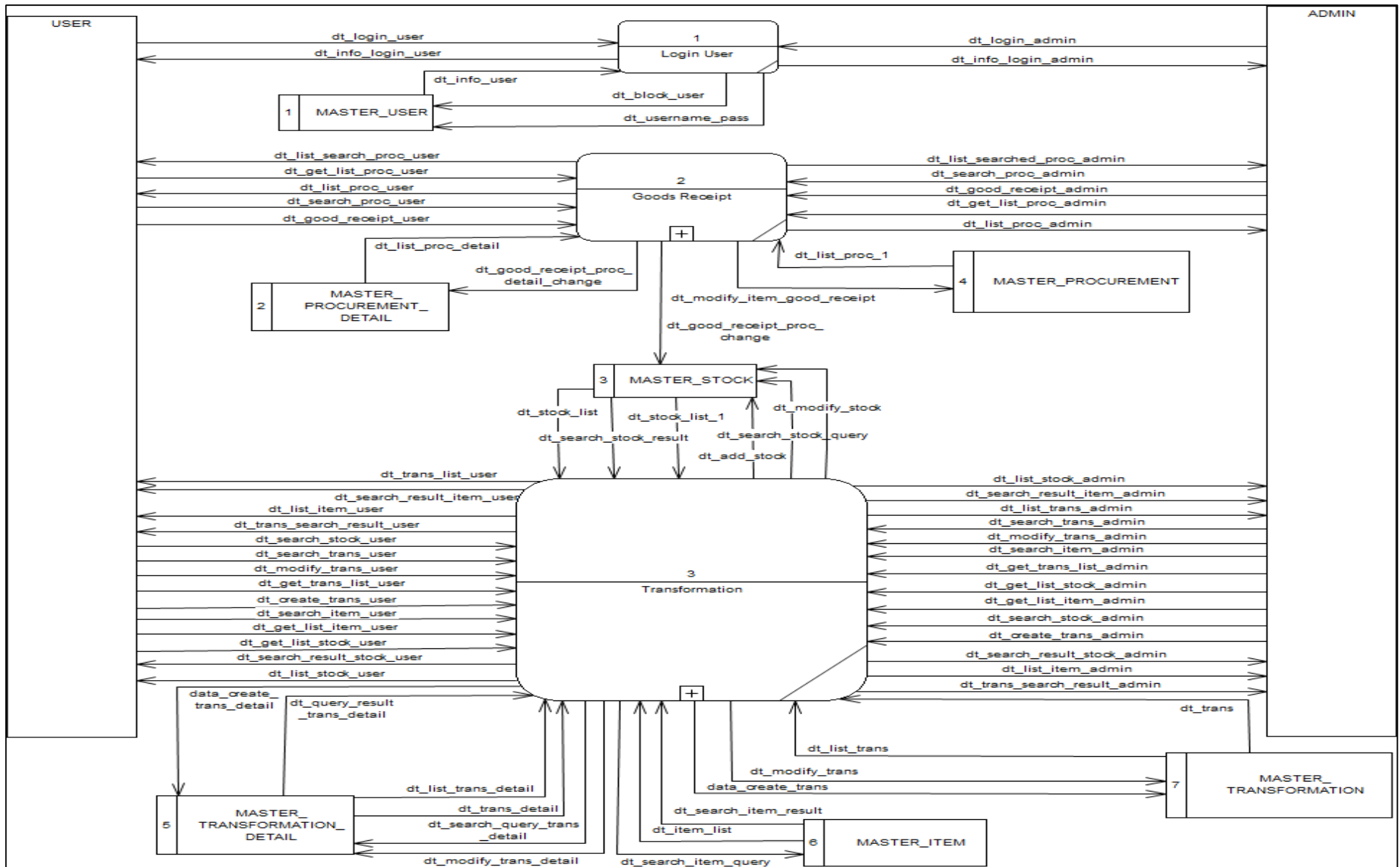


Gambar 3.1 DFD *Context Diagram* Sub Sistem Inventori dan Produksi

Dilanjutkan dengan level 1 dimana terdapat tiga subproses yang terdapat dalam *main* proses tersebut yaitu Proses *Login User*, Proses *Goods Receipt*, dan Proses *Transformation*. Di dalam level 1 ini terdapat tujuh *table* yang digunakan yaitu *User*, *Stock Table*, *Procurement Detail*, *Procurement*, *Transformation*, *Transformation Detail*, dan *Item*. Selain itu terdapat data *flow* yang sudah terincikan arahnya ke subproses-subproses yang terdapat dalam DFD level 1 ini, baik itu menuju *table* maupun entitas. Proses *Goods Receipt* berfungsi untuk menerima barang yang masuk ke gudang sedangkan proses *Transformation* berfungsi untuk memproduksi barang atau merubah dari satu barang menjadi barang lainnya.

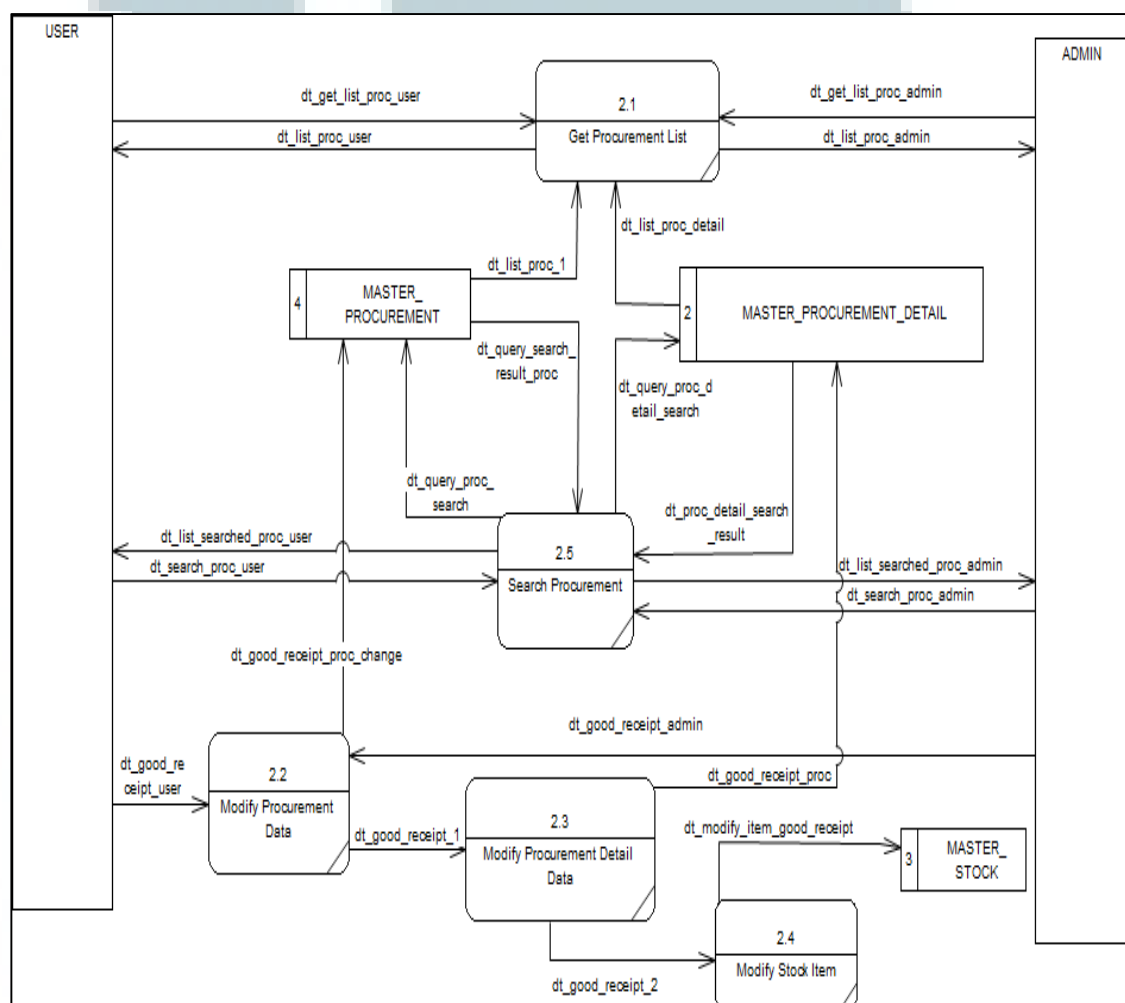


UMN



Gambar 3.2 DFD level 1 Sub Sistem Inventori dan Produksi

Pada level satu diatas terdapat dua subproses yang terbagi kembali yaitu proses *Goods Receipt* dan proses *Transformation*. Pada level dua untuk proses *Goods Receipt* terdapat empat subproses yang terdiri dari proses *Get Procurement List*, proses *Modify Procurement Data*, proses *Modify Procurement Detail Data*, proses *Modify Stock Item*, dan proses *Search Procurement*. Di proses ini juga menggunakan tiga table yaitu *Procurement*, *Stock*, dan *Procurement Detail*. Berikut gambar level 2 Proses *Goods Receipt*.

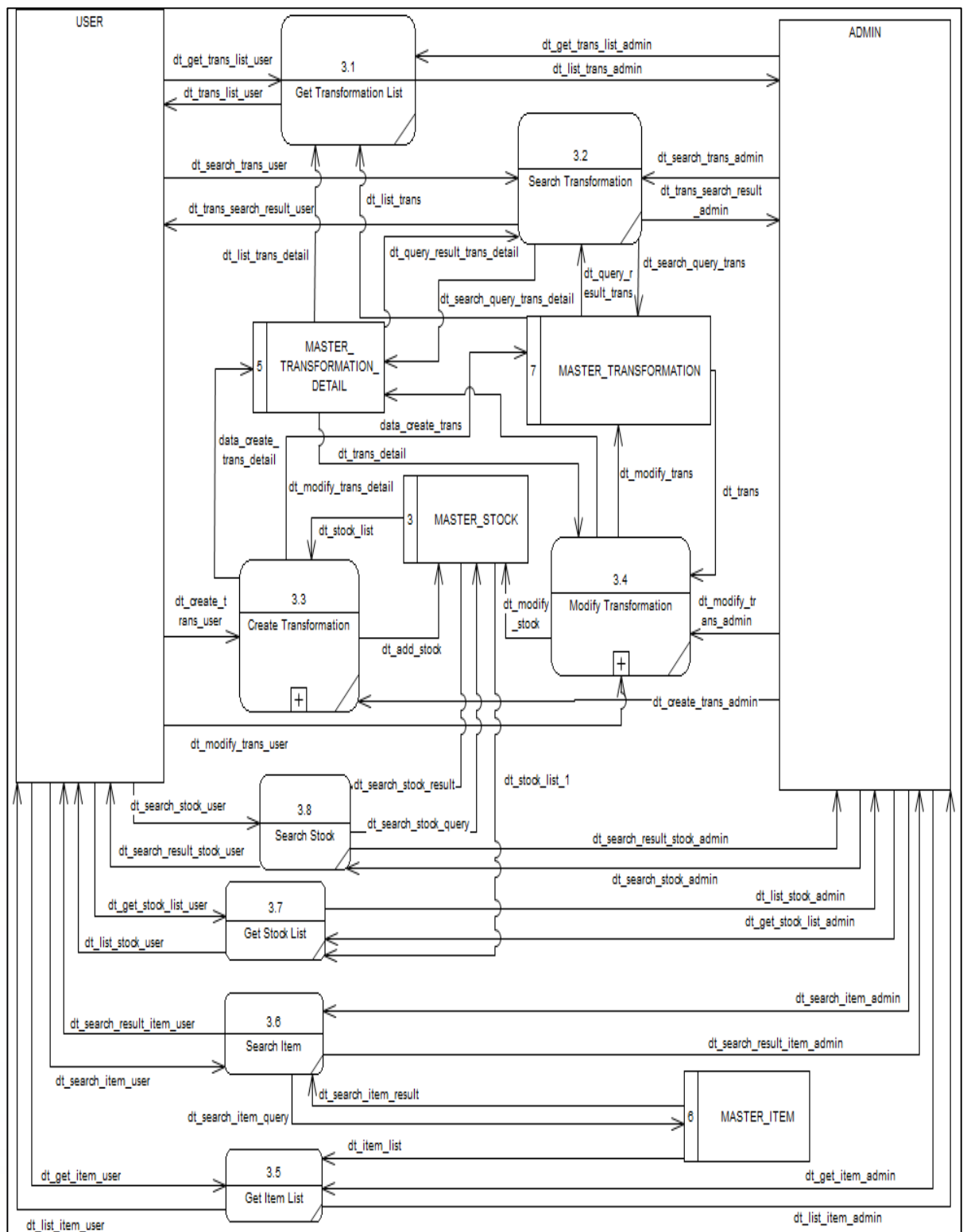


Gambar 3.3 DFD level 2 Proses *Goods Receipt*

Pada Proses *Transformation* terdapat 8 proses yaitu proses *Get Transformation List*, proses *Search Transformation*, proses *Create Transformation*, proses *Modify Transformation*, proses *Get Item List*, proses *Search Item*, proses *Get Stock List*, proses *Search Stock*. Proses ini juga menggunakan empat table yaitu *Item Table*, *Stock Table*, *Transformation Table*, dan *Transformation Detail Table*. Berikut gambar level 2 Proses *Transformation*.

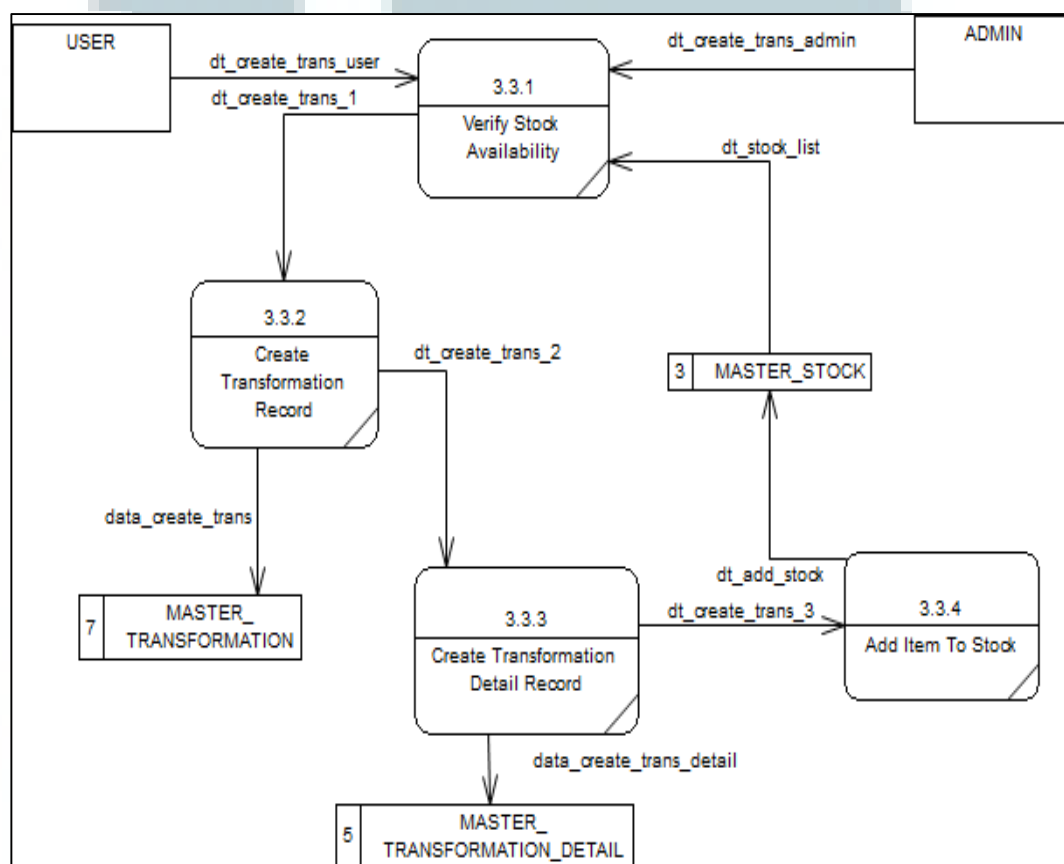


U
M
N



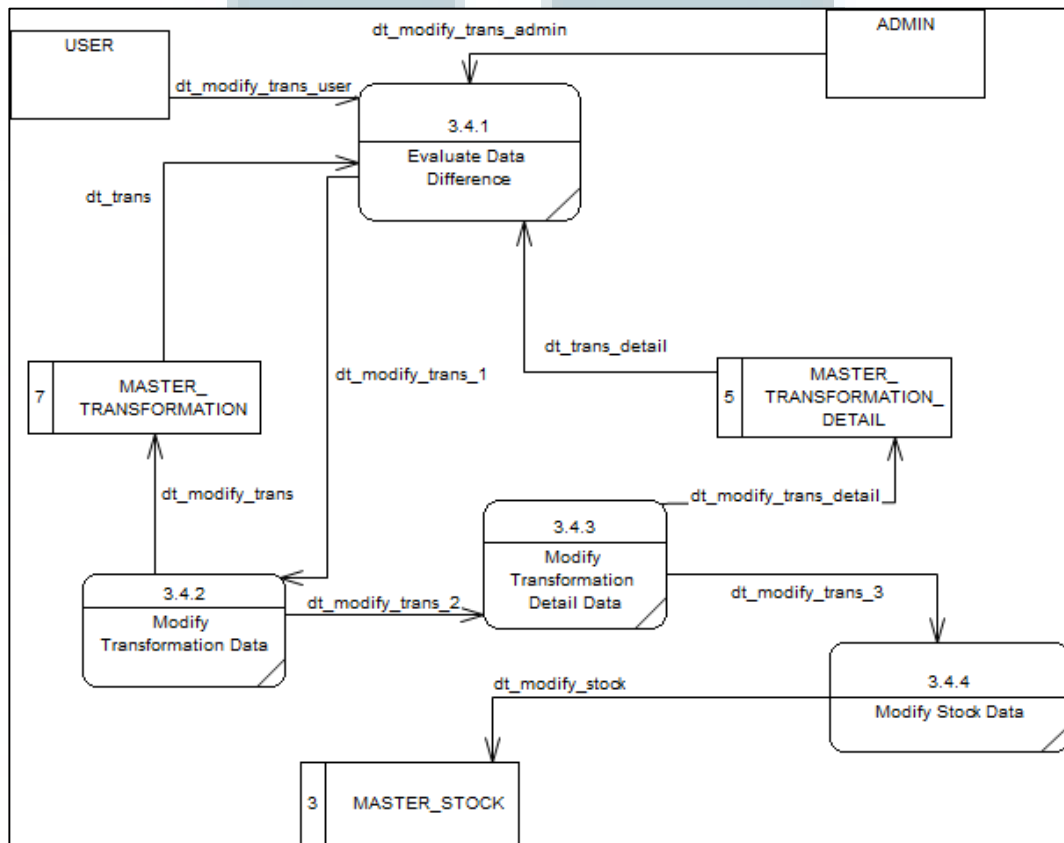
Gambar 3.4 DFD level 2 Proses Transformation

Pada DFD level dua proses *Transformation* tersebut masih ada dua subproses yang terbagi lagi yaitu proses *Create Transformation* dan *Modify Transformation*. Proses *Create Transformation* hanya mempunyai empat buah proses yaitu proses *Verify Stock Availability*, proses *Create Transformation Record*, proses *Create Transformation Detail Record*, dan proses *Add Item To Stock*. Proses tersebut juga menggunakan *table Transformation*, *Transformation Detail*, dan *Stock*. Berikut gambar level 3 Proses *Create Transformation*.



Gambar 3.5 DFD level 3 Proses *Create Transformation*

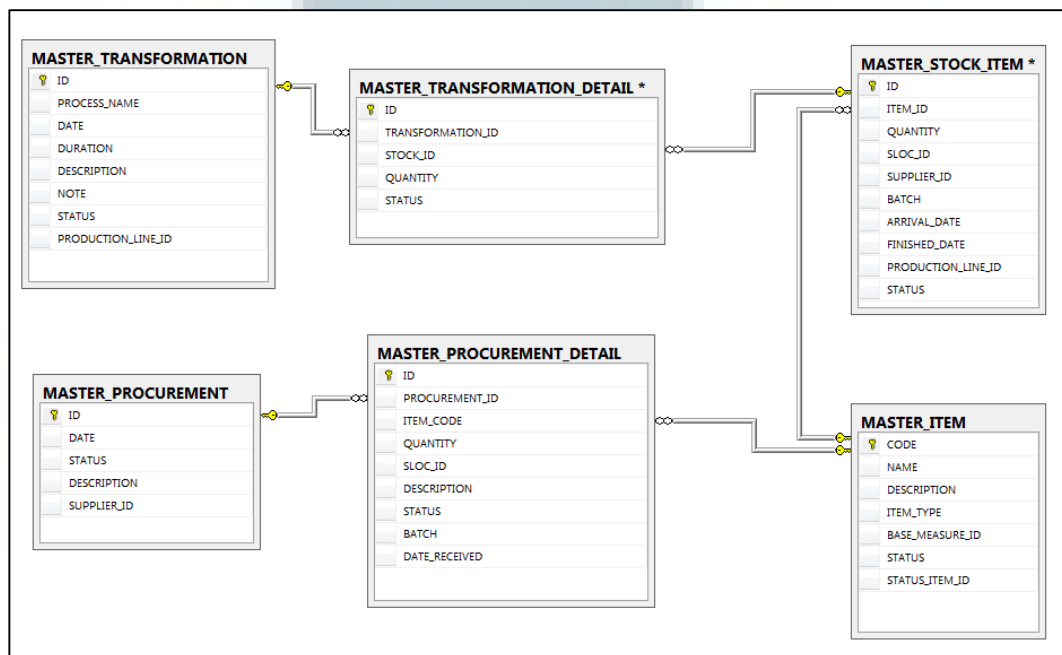
Pada Proses *Modify Transformation* juga terdapat empat proses yaitu proses *Evaluate Data Difference*, proses *Modify Transformation Data*, proses *Modify Transformation Detail Data*, dan proses *Modify Stock Data*. Proses ini juga menggunakan table *transformation*, *transformation detail*, dan *stock*. Berikut gambar level 3 Proses *Modify Transformation*.



Gambar 3.6 DFD level 3 Proses *Modify Transformation*

A.2. Entity Relationship Diagram

Berikut ini adalah ERD dari Sub Sistem Inventori dan Produksi yang dibuat untuk PT Golden Sun Indonesia.



Gambar 3.7 ERD Sub Sistem Inventori dan Produksi

Pada ERD tersebut tabel *master_transformation* berhubungan dengan *master_transformation_detail*. Hubungan pada *primary key id* dari tabel *master_transformation* dengan *transformation_id* pada *master_transformation_detail* di mana setiap transformasi bisa mempunyai satu atau lebih detail. Kemudian *primary key id* pada tabel *master_stock_item* terhubung dengan *stock_id* pada tabel *master_transformation_detail*. Hubungannya merupakan hubungan *one to one* karena setiap detail transformasi menghasilkan *record* baru di tabel *master_stock_item*. Setelah itu *primary key code* pada *master_item* mempunyai hubungan *one to one* dengan *item_id* pada *master_stock_item* dimana setiap *record* pada *master_stock_item* mempunyai satu *item_id*. Kemudian *primary key code* pada *master_item* juga mempunyai hubungan *one to one* dengan *item_code* pada *master_procurement_detail*. Setiap detail

pembelian pasti mempunyai satu *item_code*. Hubungan yang terakhir yaitu *primary key id* pada *master_procurement* dengan *procurement_id* pada *master_procurement_detail* yang berupa hubungan *one to many*. Setiap *procurement* bisa mempunyai banyak detail.

Terdapat perbedaan penamaan pada *table* yang dibuat dengan rancangan DFD karena pada implementasi dianggap lebih mudah untuk menambahkan *prefix* dan *suffix* tertentu pada setiap *table*.

Dalam sub sistem inventori dan produksi ini menggunakan tujuh tabel, namun satu tabel lainnya yaitu tabel *master_user* merupakan tabel yang berdiri sendiri dan tidak memiliki hubungan dengan tabel-tabel yang lainnya, sehingga tidak termasuk dalam ERD.

A.3 Struktur Table

Database yang digunakan dalam sistem ini adalah Microsoft SQL SERVER 2008. Berikut adalah struktur-struktur tabel yang digunakan.

Nama tabel : Master_Stock_Item

Fungsi : Tabel ini digunakan untuk mencatat jumlah stok yang ada.

Tabel 3.3 Struktur Tabel Master_Stock_Item

Field Name	Type	Length	Information
ID	Integer		<i>Primary key</i>
ITEM_ID	Varchar	50	Merupakan <i>Foreign Key</i> ke tabel Master_Item
QUANTITY	int		
SLOC_ID	Int		<i>Foreign Key</i> ke Master_Sloc
SUPPLIER_ID	int		<i>Foreign Key</i> ke Master_Supplier
BATCH	Varchar	50	
ARRIVAL_DATE	Date		
FINISHED_DATE	Date		

Tabel 3.3 Struktur Tabel Master_Stock_Item (Lanjutan)

Field Name	Type	Length	Information
PRODUCTION_LINE_ID	Int		<i>Foreign Key</i> ke Master_Production_Line
STATUS	Int		

Nama tabel : Master_Procurement

Fungsi : Tabel ini digunakan untuk menyimpan informasi-informasi yang terkait dengan pembelian tetapi tidak mencakup detail barang yang dibeli.

Tabel 3.4 Struktur Tabel Master_Procurement

Field Name	Type	Length	Information
ID	Int		<i>Primary key</i>
DATE	Date		
STATUS	Int		
DESCRIPTION	Varchar	500	
SUPPLIER_ID	int		<i>Foreign Key</i> ke Master_Supplier

Nama tabel : Master_Procurement_Detail

Fungsi : Tabel ini digunakan untuk menyimpan detail barang yang dibeli dari Master_Procurement

Tabel 3.5 Struktur Tabel Master_Procurement_Detail

Field Name	Type	Length	Information
ID	Int		<i>Primary Key</i>
PROCUREMENT_ID	Int		<i>Foreign Key</i> ke Master_Procurement
ITEM_CODE	Varchar	50	<i>Foreign Key</i> ke Master_Item

Tabel 3.5 Struktur Tabel Master_Procurement_Detail (Lanjutan)

Field Name	Type	Length	Information
QUANTITY	Int		
SLOC_ID	Int		<i>Foreign Key</i> ke Master_Sloc (Tempat penyimpanan barang)
DESCRIPTION	varchar	200	
STATUS	Int		
BATCH	varchar	50	
DATE_RECEIVED	Date		

Nama tabel : Master_Item

Fungsi : Tabel yang digunakan untuk menyimpan data barang.

Tabel 3.6 Struktur Tabel Master_Item

Field Name	Type	Length	Information
CODE	Varchar	50	<i>Primary Key</i>
NAME	Varchar	50	
DESCRIPTION	Varchar	50	
ITEM_TYPE	Int		Tipe item, contoh kayu. <i>Foreign Key</i> ke Master_Item_Type
BASE_MEASURE_ID	Int		Satuan dasar. <i>Foreign Key</i> ke Master_Base_Measure
STATUS	Int		
STATUS_ITEM_ID	Int		Status barang apakah barang jadi atau bahan baku. <i>Foreign Key</i> ke Master_Status_Item.

Nama tabel : Master_Transformation

Fungsi : Tabel yang menyimpan data tentang informasi perubahan barang.

Tabel 3.7 Struktur Tabel Master_Transformation

Field Name	Type	Length	Information
ID	int		<i>Primary Key</i>
PROCESS_NAME	Varchar	50	
DATE	Date		
DURATION	Int		
DESCRIPTION	Varchar	200	
NOTE	Varchar	200	
STATUS	Int		
PRODUCTION_LINE_ID	Int		<i>Foreign Key ke Master_Production_Line</i>

Nama tabel : Master_Transformation_Detail

Fungsi : Tabel yang digunakan untuk menyimpan detail transformasi yang terjadi.

Tabel 3.8 Struktur Tabel Master_Transformation_Detail

Field Name	Type	Length	Information
ID	Int		<i>Primary Key</i>
TRANSFORMATION_ID	Int		<i>Foreign Key ke Master_Transformation_detail</i>
STOCK_ID	Int		<i>Foreign Key ke Master_Stock_Item</i>
QUANTITY	Int		
STATUS	Int		

Nama tabel : Master_User

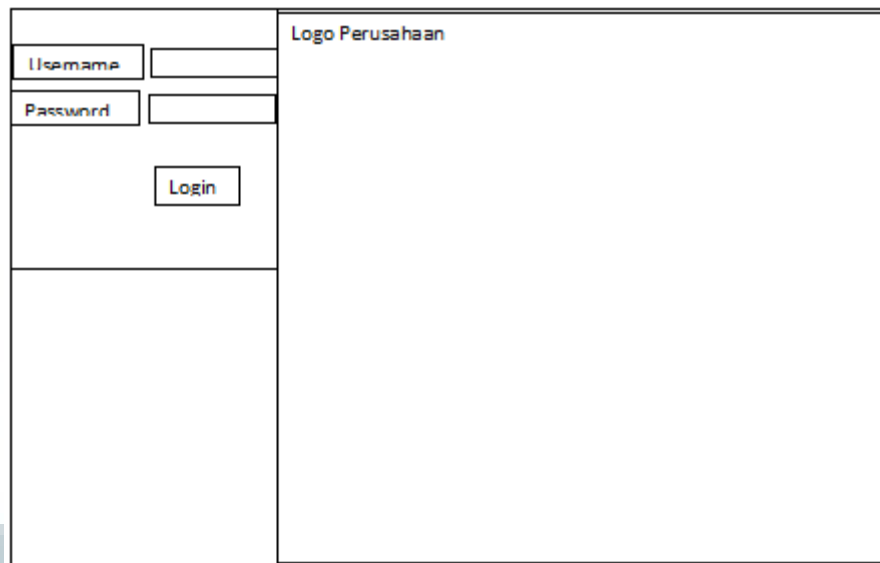
Fungsi : Tabel yang menyimpan data *username*, *password*, *database*, tanggal aktif, tipe *user*, serta apakah *user* tersebut diblok dari sistem.

Tabel 3.9 Struktur Tabel Master_User

Field Name	Type	Length	Information
USERNAME	Varchar	50	<i>Primary Key</i> , <i>username</i> untuk <i>login</i> ke sistem.
PASSWORD	Varchar	32	<i>Password</i> yang dienkripsi dalam MD5
DIRECTEDDB	Int		Penggunaan redirek <i>database</i>
ACTIVE_UNTIL	Date		Tanggal aktif, lebih dari tanggal tersebut tidak bisa login.
IS_BLOCKED	Int		
USER_TYPE	Int		Tipe user. <i>Foreign Key</i> ke Master_User_type

B. Desain Antarmuka

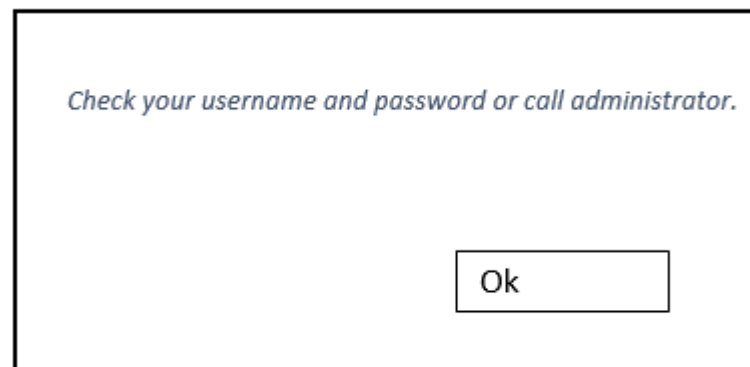
Pada sub sistem yang dibuat ini terdapat beberapa rancangan yang akan digunakan. Untuk rancangan tampilan antarmuka untuk *login* ada pada Gambar 3.8.



The image shows a login form with a header area on the right containing the text "Logo Perusahaan". On the left side, there are two input fields: the top one is labeled "Username" and the bottom one is labeled "Password". Below these fields is a "Login" button. The form is enclosed in a rectangular border.

Gambar 3.8 Desain antarmuka *login*

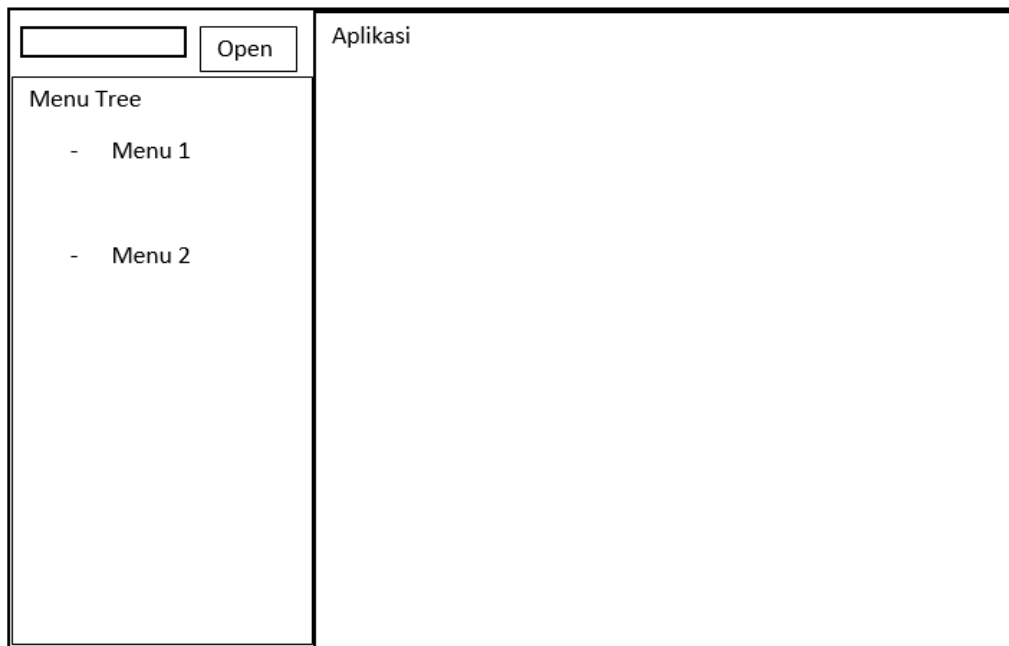
Tampilan di atas akan dilihat saat program baru dijalankan. *User* akan diminta untuk *login* ke dalam sistem. Apabila *user* salah memasukkan *username* atau *password* maka akan muncul popup dialog dengan desain berikut.



The image shows a dialog box with a black border. Inside, the text "Check your username and password or call administrator." is displayed in a blue, italicized font. At the bottom right of the dialog box, there is an "Ok" button.

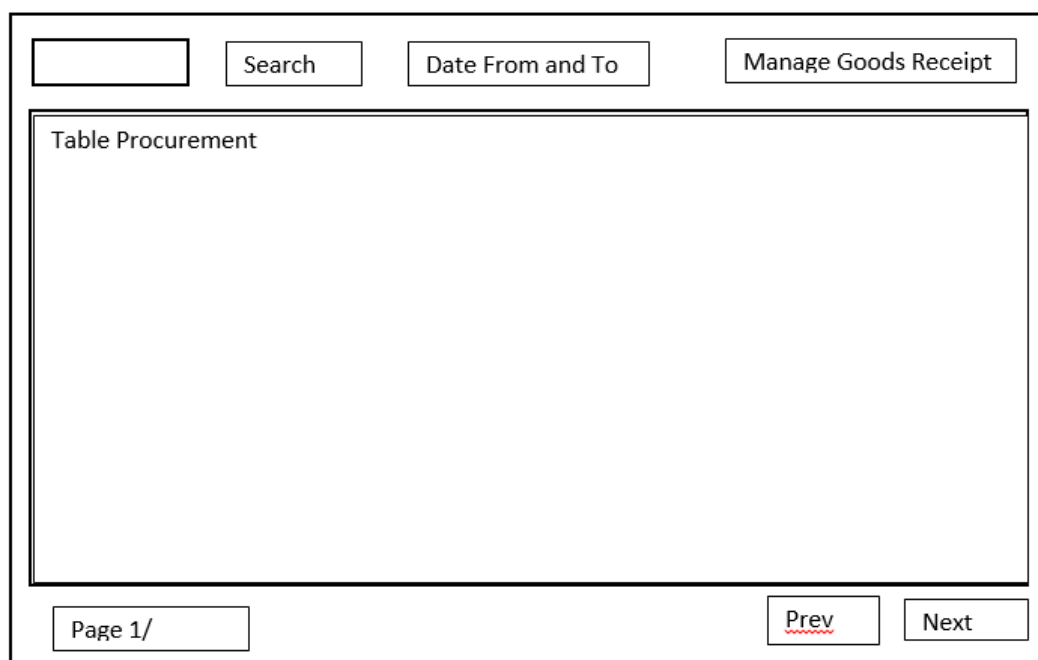
Gambar 3.9 Desain antarmuka *popup dialog gagal login*

Jika *user* berhasil masuk ke dalam sistem maka akan ditampilkan menu yang akan digunakan untuk menggunakan sistem seperti yang ditunjukkan pada Gambar 3.10.



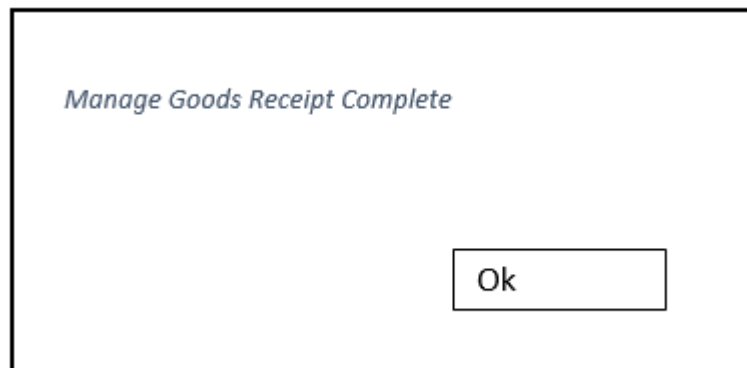
Gambar 3.10 Desain antarmuka *menu* setelah login

Pada tampilan di atas, sebelah kanan akan digunakan untuk tempat aplikasi yang akan dibuka dari menu di bagian kiri. Kemudian *user* dapat memilih bagian inventori dan produksi. Desain antarmuka *list goods receipt* sebagai berikut.



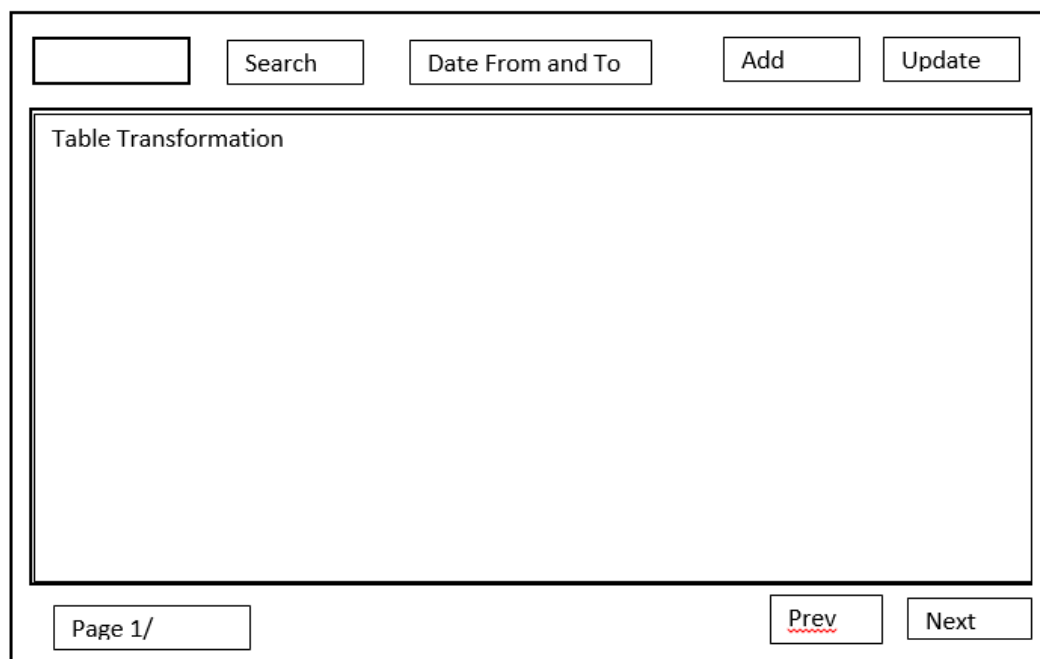
Gambar 3.11 Desain antarmuka *list goods receipt*

User dapat melakukan pencarian dari kolom yang disediakan serta memilih tanggal dari data. Setelah itu *user* bisa memilih pembelian yang barangnya akan diterima. Apabila sudah selesai maka akan muncul *popup* sesuai pada Gambar 3.12 .



Gambar 3.12 Desain antarmuka *popup manage goods receipt*

Untuk desain antarmuka transformasi juga mirip dengan *goods receipt* namun sedikit berbeda sesuai dengan gambar 3.13.



Gambar 3.13 Desain antarmuka *list transformasi*

Dari gambar tersebut dapat dilihat bahwa terdapat *button search* serta *add*, dimana *user* dapat melakukan pencarian. Jika *user* memilih *button add* maka akan melakukan akses ke form *add* atau *edit*, sedangkan jika *button search* yang dipilih maka hasil pencarian akan dimunculkan di dalam bagian konten.

Kemudian apabila memilih *form add* atau *edit* maka akan muncul desain antarmuka sebagai berikut.

Gambar 3.14 Desain antarmuka *create transformation*

Pada gambar di atas terdapat tempat untuk mengisi data-data yang diperlukan untuk proses tersebut namun pada bagian input dan output user memilih lagi dari *list item* maupun *list stock*. Untuk *modify* atau *update transformation* juga mengikuti desain antarmuka di atas. Apabila user memilih tombol *add* untuk input item atau output item maka akan muncul tampilan antar muka pada Gambar 3.15 dan 3.16 berikut.

Input Item to Transform

Item

Item Type

Base Measure

Quantity

Gambar 3.15 Desain antarmuka *input item*

Output Item to Transform

Item

Item Type

Base Measure

Quantity

Storage Location

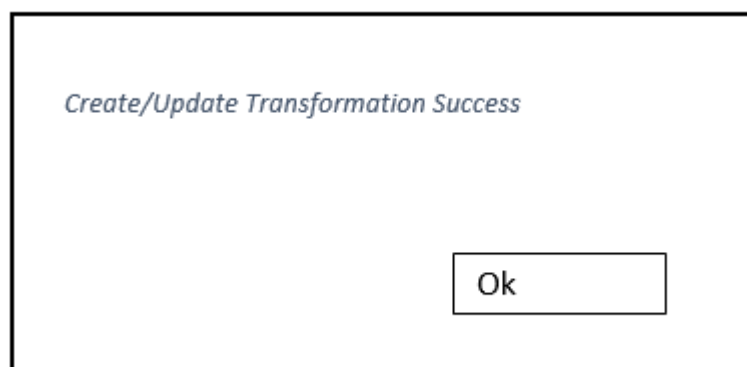
Gambar 3.16 Desain antarmuka *output item*

Setelah itu jika *user* memilih *select item* pada *input* atau *output item* maka akan muncul *form* dengan desain seperti Gambar 3.17 di bawah ini.



Gambar 3.17 Desain antarmuka *select item*

Setelah memasukkan semua data dan menyelesaikan create atau update transformation maka akan muncul popup informasi sesuai dengan desain antarmuka seperti Gambar 3.18 berikut ini.

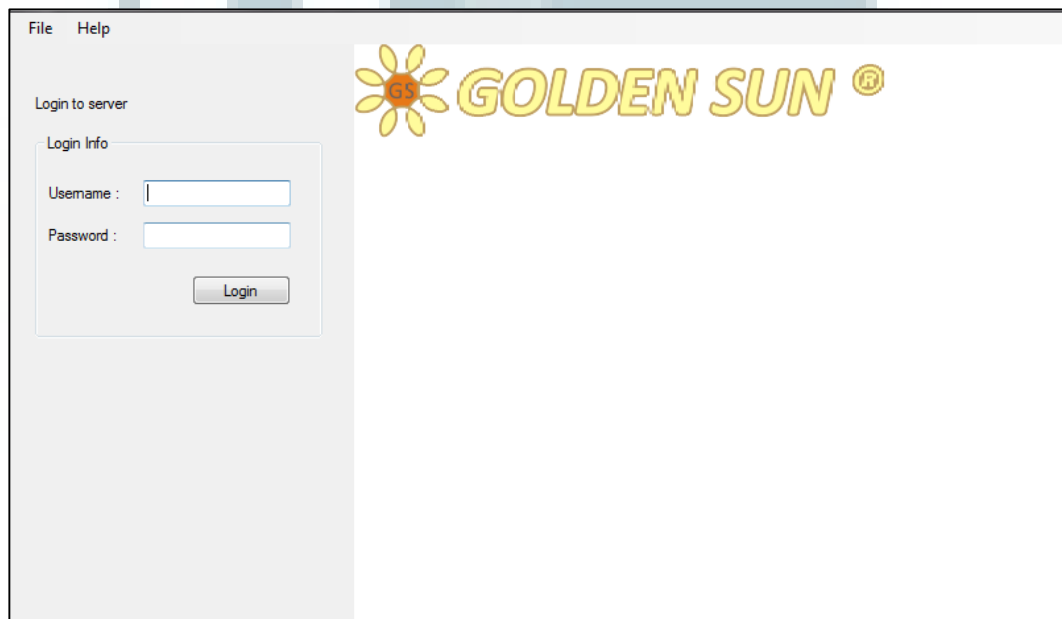


Gambar 3.18 Desain antarmuka *popup create/update transformation*

C. Implementasi

C.1 *Form Login* dan *Menu User*

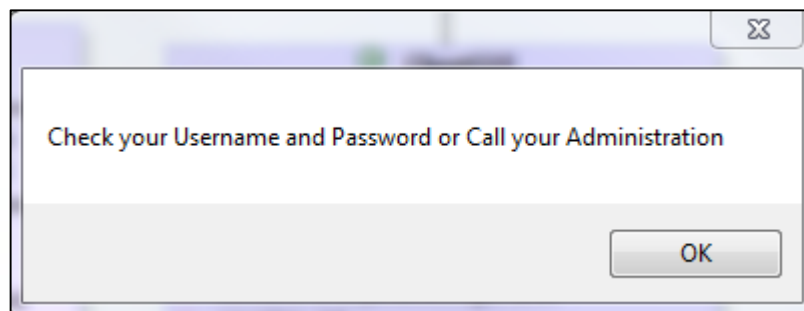
Berikut merupakan hasil *print screen* yang diambil dari hasil implementasi sub sistem yang telah dibuat. Sebagai halaman awal seperti yang telah dijelaskan pada bagian antarmuka, maka *user* akan mengakses *form login*, seperti gambar berikut dibawah ini dan kemudian memasukkan *username* serta *password*.



The screenshot shows a web application window with a title bar containing 'File' and 'Help' menus. The main content area is divided into two sections. On the right, there is a logo for 'GOLDEN SUN' featuring a yellow sun icon with 'GS' inside and the text 'GOLDEN SUN' in yellow. On the left, there is a 'Login to server' section. Inside this section, there is a 'Login Info' box containing two input fields: 'Username' and 'Password', and a 'Login' button below them.

Gambar 3.19 *Form Login*

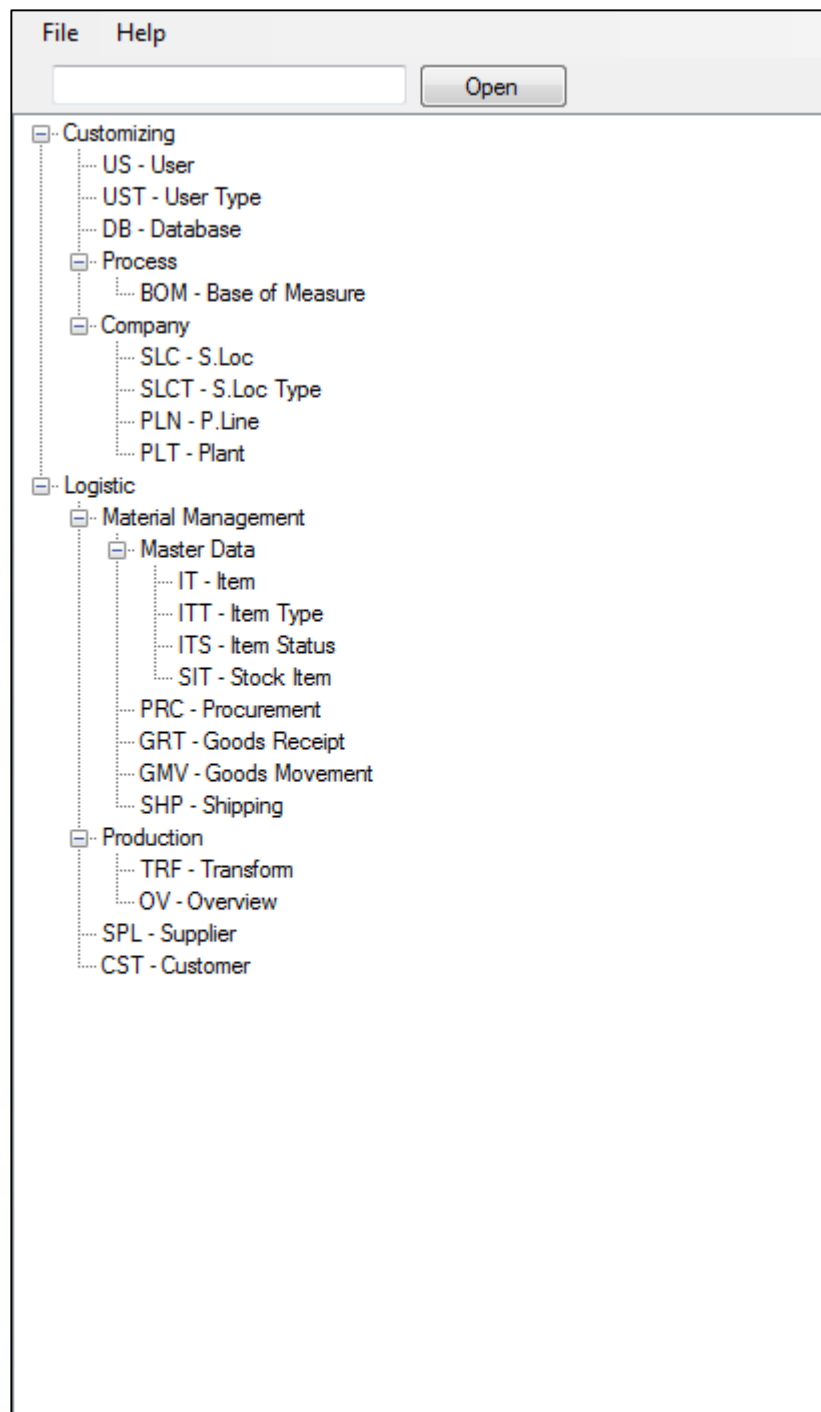
Untuk *form login* tersebut, telah dibuat *function* untuk *authentication* sehingga apabila *user* memasukkan *username* ataupun *password* yang tidak sesuai dengan data yang tersimpan di dalam *database* maka *form login* tersebut akan menunjukkan pesan kesalahan seperti berikut ini.



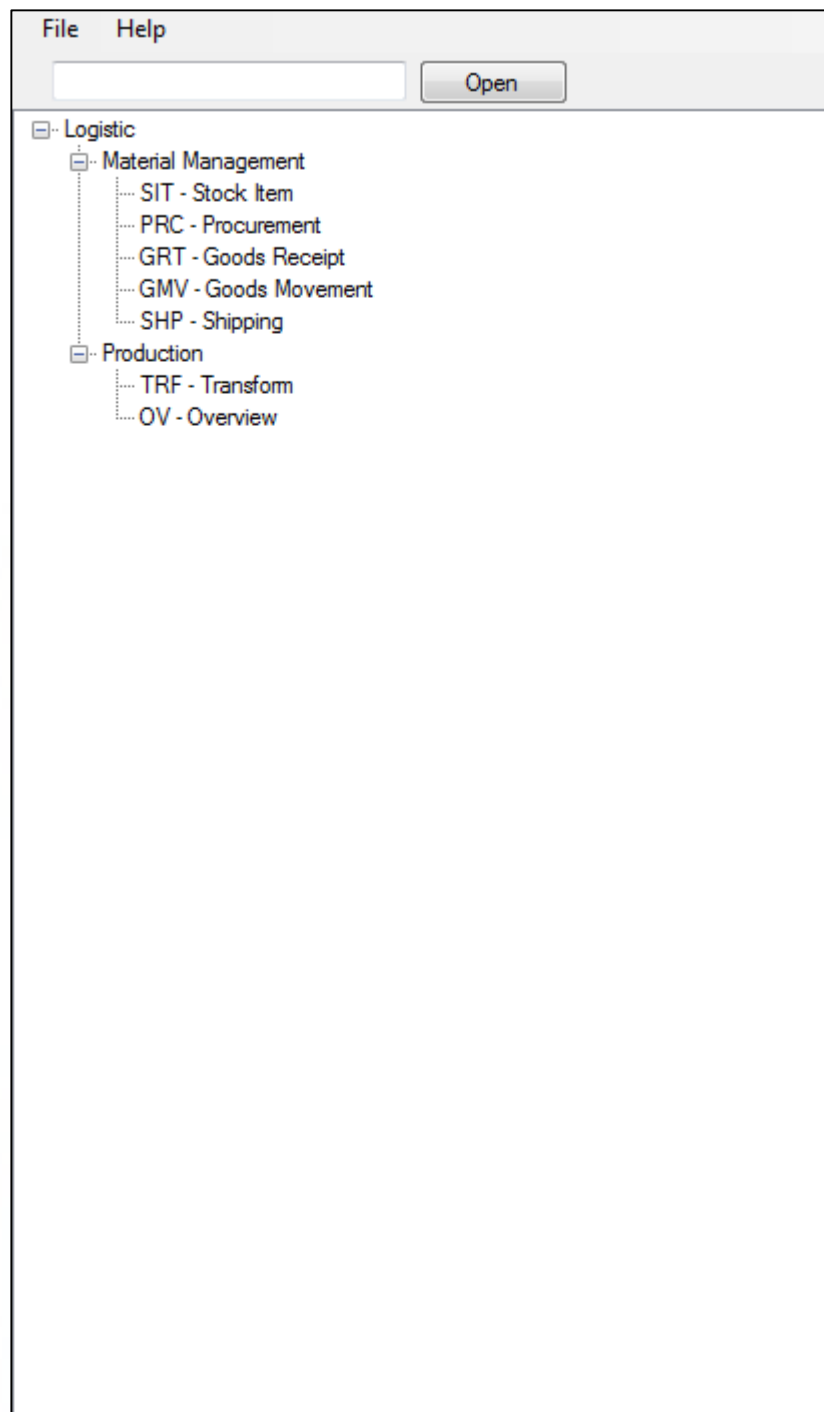
Gambar 3.20 *Popup gagal login*

Namun apabila *user* berhasil masuk ke sistem maka akan menuju menu. Untuk *user* terbagi menjadi dua tipe, yaitu *Admin* dan *User*, dengan perbedaan hak akses pada menu user dan admin seperti pada gambar 3.21 dan 3.22 berikut ini.

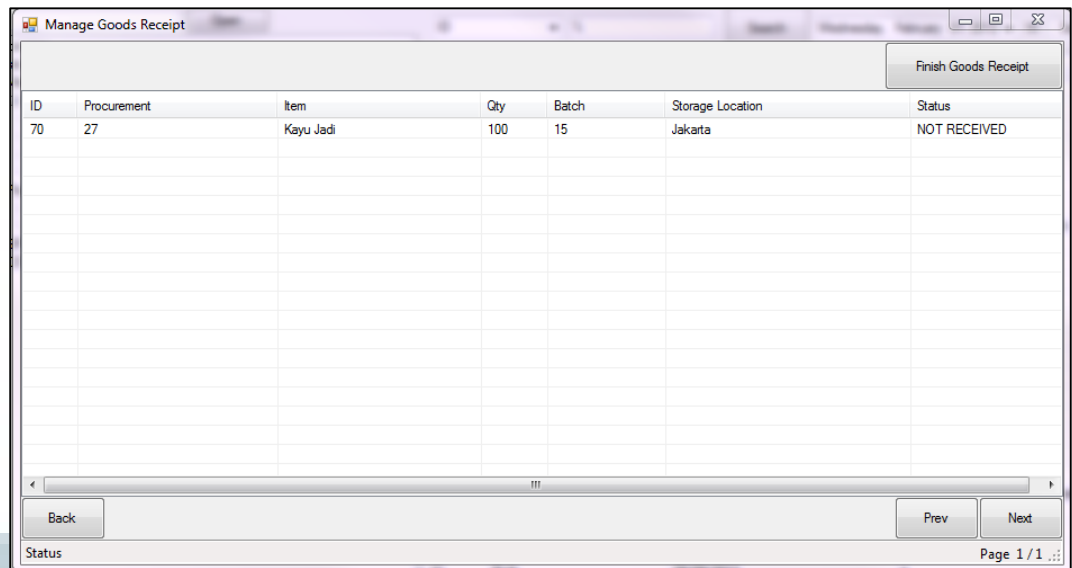
UMMN



Gambar 3.21 Menu yang bisa diakses sebagai *Admin*

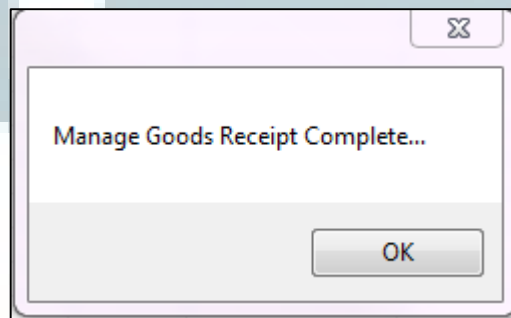


Gambar 3.22 Menu yang bisa diakses sebagai *User*



Gambar 3.24 Manage goods receipt

Pada gambar di atas terlihat data masih dalam status *NOT RECEIVED*. Apabila barang sudah diterima maka di-*double click* data tersebut untuk mengubah status menjadi *RECEIVED*. Setelah itu pilih button *Finish Goods Receipt* dan akan muncul *popup* berikut.



Gambar 3.25 Popup goods receipt success

Apabila popup tersebut muncul maka barang telah diterima dan telah masuk ke dalam gudang. Tabel-tabel yang bersangkutan juga sudah diupdate saat *popup* tersebut muncul.

C.3 Transformation

CREATE TRANSFORMATION

Process Name

Note

Date

Description

Duration day(s)

Production Line

Input

ID	Item	Qty	Base of Measure	Item Type

Output

ID	Item	Qty	Base of Measure	Item Type

Add Delete Create Cancel

Gambar 3.27 *Create Transformation*

Setelah mengisi data yang harus diisi di atas maka user juga harus memasukkan item yang akan menjadi input dan item yang akan menjadi output. Apabila pada *create transformation* kita memilih button add input maka akan muncul tampilan seperti berikut.

UMMN

Item I/O

Input Item to Transform

Item :

Item Type :

Base Measure :

Quantity : from

Gambar 3.28 Add Input

Kemudian pada form ini kita harus memilih itemnya terlebih dahulu. Dengan menekan tombol *select item* maka akan muncul tampilan seperti berikut.

Stock List

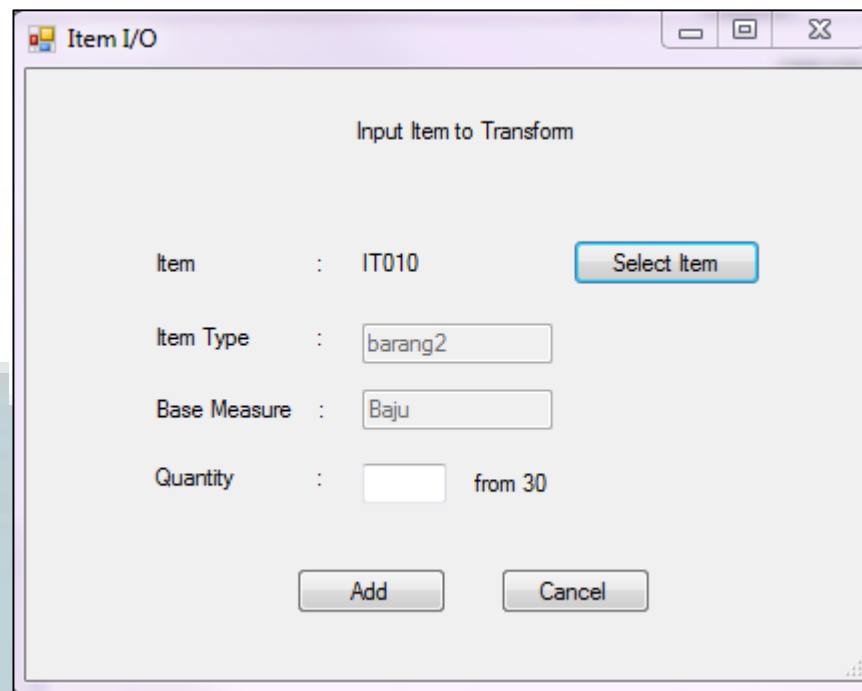
ID: 2

ID	Item	Qty	Storage Location	Supplier	Batch	Arrival Date	Finished Date	Production Line	Status
2	Benang	30	Jakarta Pusat	Rudi	Batch 2	31/07/2013	31/07/2013	0	NOT ACTIVE
12	Meja	1	Banten	Baguss		NULL	19/08/2013	2	NOT ACTIVE

Searching ID column with keyword 2 Page 1 / 1

Gambar 3.29 Select Item Input

Pada form ini setelah kita memilih item yang dipilih maka akan kembali ke form sebelumnya. Dapat dilihat pada gambar 3.30 berikut.

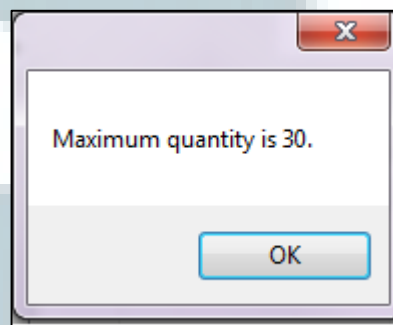


The screenshot shows a dialog box titled "Item I/O" with a subtitle "Input Item to Transform". It contains the following fields and buttons:

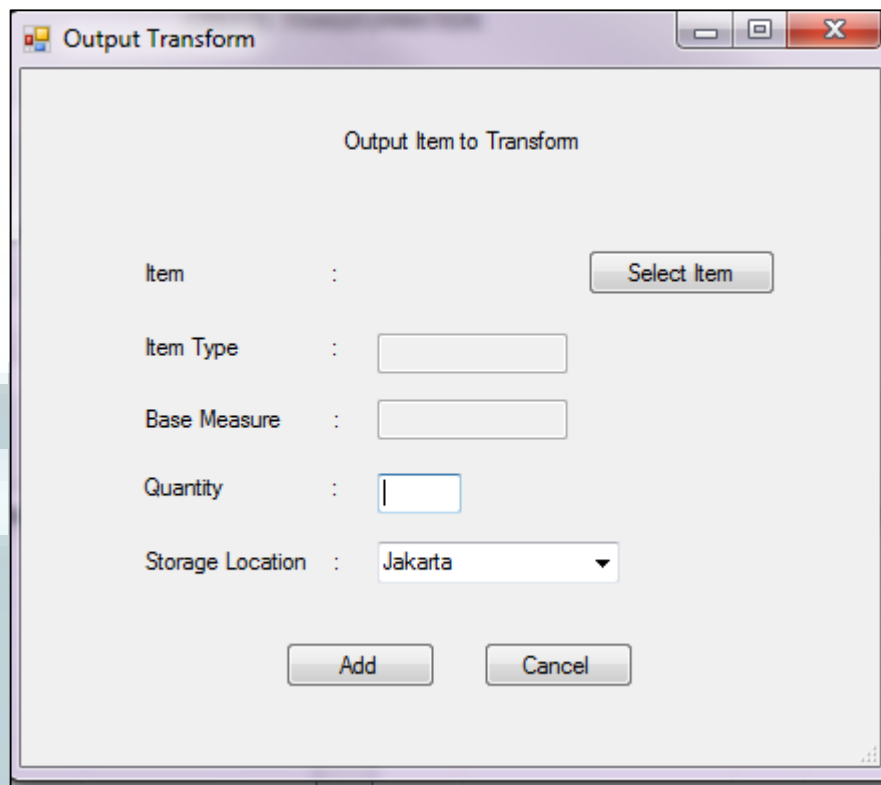
- Item : IT010 (with a "Select Item" button)
- Item Type : barang2
- Base Measure : Baju
- Quantity : [] from 30
- Buttons: Add, Cancel

Gambar 3.30 *Item Input*

User tinggal memasukkan jumlah barang yang akan menjadi input dalam proses perubahan barang ini. User tidak dapat memasukkan barang lebih dari yang ada dalam stok. Apabila user berusaha memasukkan jumlah lebih dari stok maka akan keluar peringatan sesuai gambar berikut ini.

Gambar 3.31 *Item Warning*

Setelah memasukkan *item input* user juga harus memasukkan item output, dimana nantinya akan menghasilkan *item output* tersebut. Pada *form create transformation* terdapat tombol *add* untuk *item output*. Ketika tombol tersebut dipilih maka akan muncul tampilan seperti gambar dibawah ini.



Output Transform

Output Item to Transform

Item :

Item Type :

Base Measure :

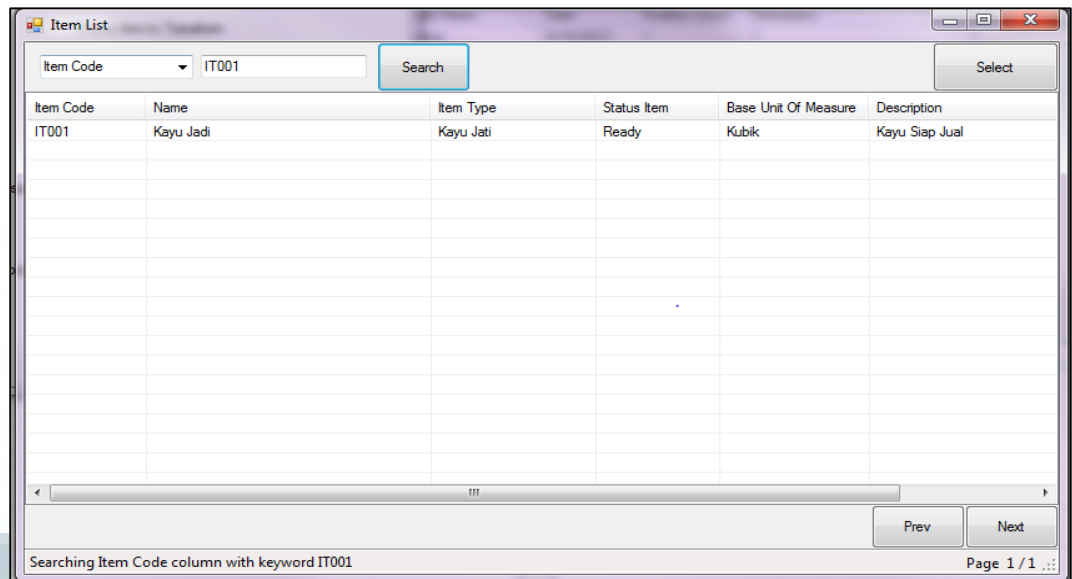
Quantity :

Storage Location :

Gambar 3.32 *Item Output*

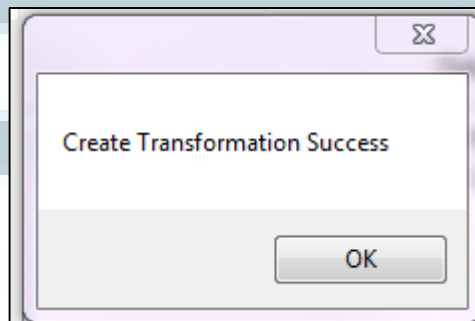
Untuk form ini user juga harus memilih select item terlebih dahulu baru memasukkan jumlah yang akan dihasilkan dari proses ini serta lokasi barang hasil akan disimpan. Untuk tampilan *select item output* akan terlihat seperti gambar berikut.

UMMN



Gambar 3.33 Select Item Output

Apabila semua data sudah dimasukkan dan diselesaikan maka proses create transformation sudah selesai dan akan muncul popup sesuai gambar berikut.



Gambar 3.34 Popup Create Transformation Success

Untuk *update transformation* prosesnya sama hanya *create* berubah menjadi *update* dan *user* harus memilih data *transformasi* yang akan diupdate terlebih dahulu.

3.3.2 Kendala yang Ditemukan

Kendala yang ditemukan dalam pembuatan sub sistem inventori dan produksi barang adalah sebagai berikut.

1. Tidak adanya *training* atau tutorial khusus mengenai C# secara mendalam, sehingga penulis kesulitan dalam memahami serta menggunakan C# secara mendalam karena kurangnya pembahasan mendalam C# dari pelajaran kampus.
2. Pada awal pengerjaan, penulis kesulitan untuk menentukan platform apa yang bisa dipakai untuk berkerja dengan tim secara efisien.

3.3.3 Solusi atas kendala yang ditemukan

Dari kendala-kendala tersebut berikut ini adalah solusi untuk menyelesaikan kendala tersebut.

1. Penulis harus menggunakan fasilitas internet untuk dapat mencari *code snippet* serta info-info untuk dapat menguasai C# secara mendalam lagi.
2. Penulis mencari informasi untuk platform yang bisa digunakan untuk bekerja di dalam tim dan menemukan Team Foundation Server yang bisa digunakan dengan baik.

UMMN