

# SIRIUS ERP V-1.5.0

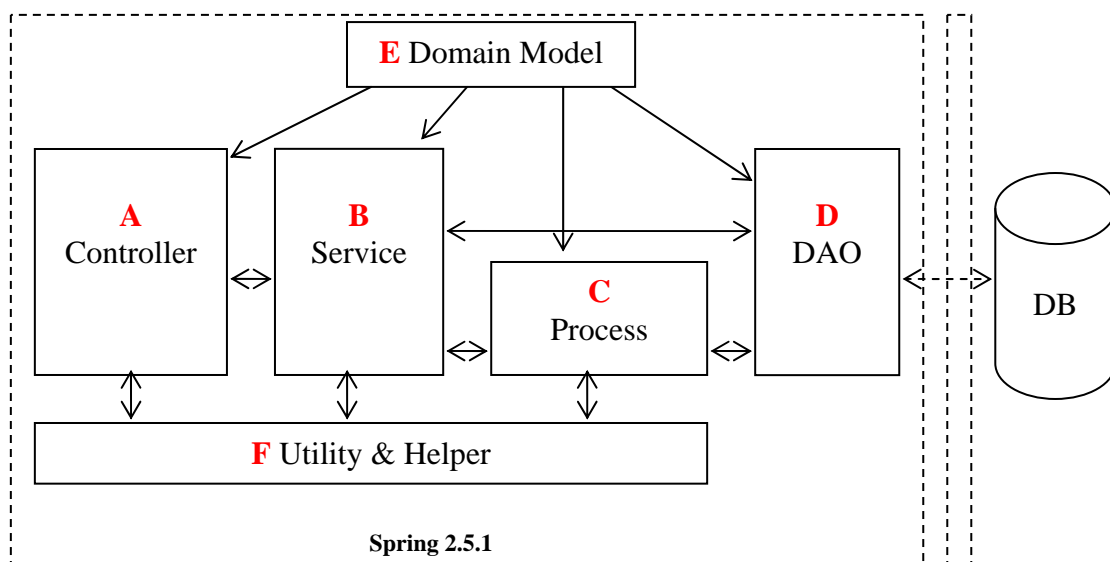
## DEVELOPER GUIDE

### 1. Architecture dan Standar Coding

#### 1.1. Technology

- JSE SDK : JSE 6
- JEE SDK : JEE 5
- MVC : Spring MVC
- JEE Framework : Spring 2.5.1
- Persistence : Hibernate 3.x
- Database : MySQL 5.x

#### 1.2. Architecture



##### A. Controller

Controller adalah component yang berada paling depan dalam lifecycle penanganan request dari user. Implementasi class controller memanfaatkan framework Spring MVC. Class controller adalah class yang di anotasi dengan @Controller.

@Controller adalah fitur spesifik yang digunakan oleh Spring MVC untuk menandakan bahwa class tersebut merupakan bagian dari controller yang kemudian dapat digunakan untuk meng-handle satu atau beberapa request dari user. Contoh implementasi dari controller sbb.

```
@Controller
public class MyController
{
    //controller code here
}
```

**Referensi : [spring-framework-reference Part V chapter 15.3 Implementing Controller](#).**

## B. Service

Service adalah class yang berisikan business logic dari lifecycle penanganan request dari user. Setiap metode pada class service harus transactional dan beberapa metode yang berhubungan dengan manipulasi data pada database me-rethrows `com.siriuserp.sdk.exception.ServiceException` serta harus mencantumkan anotasi untuk audit trails.

Contoh dari class service sbb:

```
@Transactional(rollbackFor=Exception.class)
public class MyService
{
    @Transactional(readOnly=false,.....)
    @AuditTrails(className=MyModel.class,actionType=AuditTrailsActionType.CREATE)
    public void add(MyModel model)
    {
        //write code here
    }
}
```

**Referensi : [spring-framework-reference Part IV chapter 10 Transaction Management](#).**

## C. Process

Process adalah java Class yang mengimplementasikan interface `com.siriuserp.sdk.base.Process`. Class ini digunakan sebagai extended business logic dari class service yang dimaksudkan untuk mengurangi tingkat kerumitan business logic yang terdapat pada class service serta meningkatkan readability dari class service tersebut.

Jika didalam suatu class service terdapat task yang membutuhkan beberapa pemrosesan yang melibatkan lebih dari satu table di dalam database, maka code pada class service tersebut diharuskan untuk di pecah – pecah menjadi beberapa process.

Contoh pada kasus closing Accounting Period. Untuk menyelesaikan tugas ini diperlukan beberapa langkah sbb:

- Process Income Summary
- Process Capital
- Process Currency gain and loss

- Closing Period
- Open next month period.

Untuk menyelesaikan task tersebut kita dapat membagi tugas menjadi 5 process yaitu Process untuk Income summary, capital, currency gain and loss, closing period dan opening next period.

Contoh Class Process sbb:

```
public class MyProcess implements Process<MyModel>
{
    public void execute(MyModel model)throws ServiceException
    {
        //write code here....
    }
}
```

#### D. DAO

DAO atau Data Access Object adalah sebuah class yang meng-implement interface com.siriuserp.sdk.base.BaseDao<M,Serializeable>,yang berfungsi sebagai code untuk mengakses database. Setiap class Dao diharuskan mengikuti paradigma pemrograman to interface,artinya setiap dao harus memiliki 2 class,I interface yang mengimplementasikan BaseDao,dan 1 lagi adalah class implementasi dari interface tersebut.

Contoh class DAO sbb:

```
public interface MyDao implements Process<MyModel,Long>
{
}
public class MyDaoImpl extends DaoHelper implements MyDao
{
    public void add(MyModel model)throws DataAdditionException
    {
        //write code here....
    }
    public MyModel load(MyModel model)
    {
        //write code here....
    }
}
```

## E. Model

Model adalah class yang merepresentasikan data pada database. Class model merupakan turunan dari class `com.siriuserp.sdk.base` dan harus mengikuti standard `JavaBean`.

Contoh dari domain model sbb:

```
public class MyModle extends Model implements Serializeable
{
    public String auditCode()
    {
    }
}
```

## F. Utility & Helper

Utility & Helper merupakan class – class yang berfungsi untuk membantu memudahkan dalam pemrograman.

Contoh dari Utility & Helper sbb:

`com.siriuserp.sdk.utility.StringHelper`, Class yang berfungsi untuk membantu dalam manipulasi String, antara lain untuk mengkonfersi date String menjadi `java.util.Date`, untuk mengkonfersi string bilangan menjadi `java.math.BigDecimal`, dll.

## 1.3. Standar Kode

Pada umumnya code standard yang di terapkan pada development siriuserp mengikuti aturan umum dari Sun seperti konfensi penamaan class, methode, variable, javabean.

### A. Aturan Umum.

- Semua Developer Sirius Indonesia diharuskan untuk melihat – lihat terlebih dahulu code yang sudah ada pada repository, developer hanya diperbolehkan membuat class baru hanya jika memang belum terdapat pada repository. Hal tersebut dimaksudkan agar tidak terjadinya banyak class yang memiliki fungsi yang hampir sama. Contoh:

Code yang telah ada :

```
public class BillingDaoImpl extends BillingDao
{
    // Billing loadByQueryA(query).....
}
```

Code tambahan yang dilarang :

```
public class BillingReportDaoImpl extends BillingReportDao
{
    //Billing loadByDate .....
}
```

maka semua code yang berhubungan dengan akses database untuk domain model Billing diharuskan untuk menggunakan BillingDaoImpl yang telah ada dengan cara menambahkan metode yang diinginkan di dalam class BillingDao dan BillingDaoImpl. Developer dilarang untuk membuat lagi DAO baru yang berfungsi untuk mengakses domain model yang sama walaupun untuk tujuan yang berbeda seperti contoh diatas.

- Domain Model ID harus bertipe data Long
- Property untuk data angka harus bertipe java.math.BigDecimal
- Property TRUE/ FALSE harus bertipe Boolean ( Primitive)
- Property Date Time harus bertipe java.sql.Timestamp
- Property Date harus bertipe java.util.Date
- Enum

Terkadang dalam sebuah domain model terdapat type untuk domain tersebut. Untuk tipe – tipe yang bersifat tetap, developer Sirius diwajibkan untuk menggunakan tipe data Enum. Kecuali untuk kasus – kasus yang domain type nya tidak tetap atau dapat bertambah maka developer di haruskan untuk membuat domain model baru untuk merepresentasikan type tersebut. Contoh:

```
class Mobil extends Model implements Serializeable
{
    // karna kemungkinannya sudah pasti: premium,pertamax,solar,etc.
    private EnumType bahanBakar
}
```

- Generic

Karena SiriusERP mengharuskan penggunaan JSE 5 keatas maka setiap developer diharuskan menggunakan fitur generic yang terdapat pada JSE 5 keatas.

- Collection Framework – Map, List dan Set (Implementasi: FastMap, FastList, FastSet )

Untuk Data Transfer Object ( DTO ) di prioritaskan untuk memanfaatkan class – class yang terdapat pada Collection Framework seperti Map, List, Set.

Untuk alasan performance, developer diwajibkan untuk mengutamakan penggunaan library Javalution untuk semua implementasi dari Map, List dan Set yaitu FastMap, FastList dan FastSet.

- Penciptaan Object Wrapper ( Integer, Long, Double, etc)

Untuk penggunaan atau penciptaan object wrapper dari tipe data primitive, developer diharuskan untuk menggunakan static method ( `valueOf` ) yang telah disediakan didalam object tersebut. Contoh:

```
Integer : Integer.valueOf(1);  
Long    : Long.valueOf(1);  
Double  : Double.valueOf(1);
```

## B. Buka dan Tutup Kurung { }

Buka dan tutup kurung harus diletakkan satu baris setelah deklarasi class, method, try-catch. Contoh:

```
class A  
{  
    void b()  
    {  
        try  
        {  
        }  
        catch(exception e)  
        {  
        }  
    }  
}
```

## C. Variable / Property

Nama variable / Property minimal terdiri dari tiga character, kecuali untuk property

ID. Contoh:

```
private Long id;  
private BigDecimal amount;
```

**private int i; ← dilarang menggunakan nama variable kurang dari tiga karakter**

#### D. IF ELSTE

Blok kode if else harus satu baris setelah expresi if else tersebut. Buka dan tutup kurung digunakan hanya jika code melebihi dari satu baris. Contoh

```
if (true)
    System.out.println();
else
    System.out.println();
```

Atau

```
if (true)
{
    System.out.println();
    System.out.println();
}
```

## E. DATABASE

- Table : Nama table disesuaikan dengan nama domain model, dipisahkan oleh underscore ( \_ ) jika terdapat lebih dari satu kata. Contoh: jika terdapat domain model [StockAdjustment](#), maka nama tablenya adalah [stock\\_adjustment](#).
- Field : Nama field disesuaikan dengan nama property yang ada pada domain model, dipisahkan oleh underscore ( \_ ) jika terdapat lebih dari satu kata. Contoh jika terdapat property [activationDate](#) maka field di table pada database adalah [activation\\_date](#).
- Foreign Key : Nama foreign key pada table disesuaikan dengan nama property, dipisahkan oleh underscore ( \_ ) jika terdapat lebih dari satu kata, serta diberi penambahan kata fk di depan nama field tersebut. Contoh: jika terdapat domain model yang memiliki property [Person createdBy](#) maka field tablenya adalah [fk\\_person\\_created\\_by](#).

Khusus untuk foreign key, tidak perlu membuat relasi pada sisi database antara satu table dengan table yang di referensikannya. Relasi akan dilakukan pada level domain model saja. Hal ini disebabkan untuk menjaga portabilitas dari aplikasi agar tidak terikat dengan Merek database tertentu.

- Tipe Data Field:
  - Integer : integer
  - Long : double
  - Date : date
  - boolean : char (1) (Y/N)
  - Timestamp : timestamp
  - String : varchar
  - BigDecimal : decimal (20,10)
  - Time : time



## 2. Application Module

SIRIUS ERP V-1.5 dilengkapi beberapa module sbb:

Module Name	Functionality
Administration	Party, Company Structure, Facility, Supplier, Customer, Geographic Type, Geographic, Currency, Industry Classification, Unit of Measure, Tax.
Sales	Todo
Sales Trading	Direct Sales Order, Standard Sales Order, Delivery Planning, Delivery Order, Delivery Order Realization, Consignment Order, Consignment Sales Order, Sales Return
Procurement	Purchase Requisition, Purchase Order, Direct Purchase Order, Pertamina Purchase Order, Loading Order, Collectiong Order
Transport Outsource	Transport Outsource Master Cost, Transport Outsource Contract, Vehicle Management, Shipment, Reports
Financial Accounting	Todo
Account Receivables	Todo
Account Payables	Todo
Cash and Bank Management	Todo
Inventory	Todo
Production	Todo
Human Resources	Todo
Tools	Todo
Business Inteligent	Todo

### 3. Business Model

#### 3.1 Abstract class com.siriuserp.dm.Model

Abstract class Model adalah parent dari semua class model.

abstract class com.siriuserp.dm.Model implements java.lang.Serializable			
Nama	Tipe	Akses	Keterangan
Id	Long	protected	Unique id
createdDate	Timestamp	protected	Tanggal pembuatan dokumen
updatedAt	Timestamp	protected	Tanggal pengubahan dokumen
createdBy	Person	protected	User pembuat dokumen
updatedAtBy	Person	protected	User pengubah dokumen
Version	Long	protected	Dokumen version number (digunakan oleh hibernate untuk proses Lock-ing)

#### 3.2 Class com.siriuserp.dm.Party

Class Party merupakan representasi dari pihak – pihak yang terlibat didalam aplikasi. Pihak yang terlibat dapat merupakan sebuah group/badan/instansi/perusahaan maupun perorangan/individu. Party merupakan parent class dari Organisasi dan Person.

class com.siriuserp.dm.Party extends class com.siriuserp.dm.Model			
Nama	Tipe	Akses	Keterangan
code	String	protected	Kode Party.
firstName	String	protected	Nama depan.
middleName	String	protected	Nama tengah.
lastName	String	protected	Nama belakang.
birthDate	Date	protected	Tanggal lahir.
taxCode	String	protected	NPWP.

## 4. Framework

### 4.1 Approvable Framework

Approvable framework di gunakan untuk seluruh bisnis proses yang memerlukan persetujuan pihak – pihak tertentu. Semua proses yang memerlukan persetujuan, domain model dari proses tersebut harus *extends/inheritance* dari model *com.siriuserp.sdk.dm.Approvable*.

Domain proses yang extends dari approvable akan secara otomatis keluar pada halaman list approvable pada menu **Tools > My Approval**. Akan tetapi proses pengesahan dokumen akan tetap di laksanakan pada halaman Edit dari tiap – tiap bisnis proses asalnya, framework akan secara otomatis mengarahkan pengguna ke halaman edit dari document tujuan dengan memanfaatkan property uri yang terdapat didalam model Approvable. Untuk keperluan ini telah disediakan template yang dapat langsung digunakan di dalam file jsp dari dokumen yang memerlukan persetujuan. File tempale ini dapat di temukan di **common/approval.jsp**. Agar dapat berfungsi dengan benar, Form harus memiliki id="editForm" dan name="editForm"

**<form id='editForm' name='editForm'></form>**.

class Approvable extends Model			
Field/Properties	Tipe	Akses	Keterangan
code	String	protected	Kode unique untuk setiap document.
Date	Date	protected	Tanggal pembuatan document
uri	String	protected	Uri adalah http url kemana approvable framework harus mengarahkan pengguna dalam proses persetujuan dokumen.
approvableType	ApprovableType	protected	Tipe document yang membutuhkan persetujuan,antara lain: SALES_ORDER, PURCHASE_REQUISITION, DIRECT_PURCHASE_ORDER
approver	Person	protected	User Pengesah dokumen.
organization	Organization	protected	Prusahaan/Cabang tempat dokumen dibuat.

approvalDecision	ApprovalDecision	protected	Keputusan mengenai dokumen yang terakhir kali di buat. Di dalam ApprovalDecision terdapat informasi history dari seluruh proses pengesahan dokumen.
------------------	------------------	-----------	---

class ApprovalDecision extends Model			
Field/Properties	Type	Akses	Keterangan
remark	String	private	Catatan yang di berikan oleh pengesah dokumen.
forwardTo	Person	private	Jika di putuskan bahwa dokumen ini perlu di sah kan oleh orang lain maka field forward to akan berisikan siapa pengesah berikutnya yg akan menerima dokumen tsb.
approvalDecisionStatus	ApprovalDecisionStatus	private	Status terakhir yang di berikan oleh pengesah terakhir,antara lain: APPROVE_AND_FORWARD, FORWARD, REJECTED, APPROVE_AND_FINISH, REQUESTED
historys	ApprovalHistory	private	History dari keseluruhan proses pengesahan yang pernah di lakukan.

## 4.2 Billing Framework

Billing framework digunakan untuk seluruh bisnis proses yang memerlukan penagihan. Setiap proses yang memerlukan penagihan, bisnis model detail dari proses yang bersangkutan harus extends/inheritance dari class com.siriuserp.dm.Line.

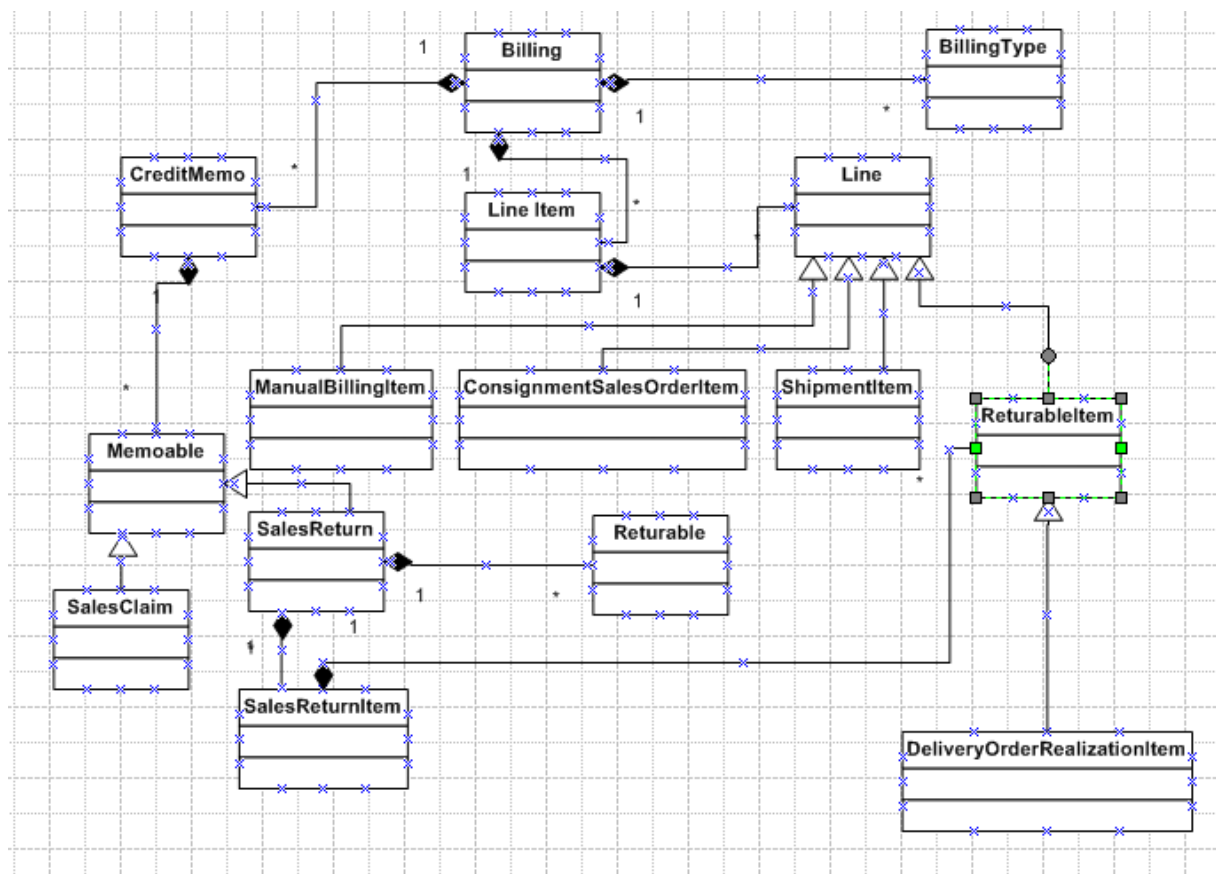
Tiap – tiap bisnis proses yang ingin memanfaatkan framework ini harus terlebih dahulu menambahkan data pada enum Billing Type, setiap entry dari billing type berisikan informasi asal usul document yang hendak di tagihkan, Transformer dan Posting Role.

**Transformer** : Transformer merupakan class implementasi dari **interface com.siriuserp.sdk.transformer**. Class ini di tujukan untuk mengubah dari Struktur class dokumen asal menjadi structure class yang dikenali oleh framework.

**PostingRole** : Posting Role merupakan class implementasi dari abstract class com.siriuserp.sdk.base.AbstractPostingRole. Class ini bertujuan untuk Automatic Posting.

Class diagram dan Screen shoot dari proses penerimaan atau pengeluaran barang dapat dilihat seperti di bawah ini:

### 4.2.1 Class Diagram billing framework



class <b>Billing</b> extends <b>Model</b> implement <b>Postingable</b>			
Field/Properties	Tipe	Akses	Keterangan
code	String	protected	Kode unique untuk setiap document.
Date	Date	protected	Tanggal pembuatan document
doNo	String	protected	Nomor DO
poNo	String	protected	Nomor PO
unpaid	BigDecimal	protected	Jumlah tagihan yang belum di bayar.
clearing	BigDecimal	protected	Jumlah Tagihan clearing.
rounding	BigDecimal	protected	Pembulatan tagihan.
status	FinancialStatus	protected	CANCELED,PAID,UNPAID
billingType	BillingType	protected	Tipe asal penagihan.
lineItems	Set<LineItem>	protected	Detil tagihan.
payments	Set<ReceiptApplication>	protected	Dokumen pembayaran tagihan.

class <b>BillingType</b> extends <b>Model</b>			
Field/Properties	Tipe	Akses	Keterangan
Name	String	protected	Nama tipe (Shipment,DOR, Consignment SO, Manual Billing, etc)
transformer	String	protected	Nama class lengkap dengan package yg merupakan implementasi dari interface <b><u>Transformer</u></b>
postingRole	String	protected	Nama class lengkap dengan package yg merupakan turunan dari class <b><u>AbstractPostingRole</u></b>

class LineItem extends Model			
Field/Properties	Type	Akses	Keterangan
line	Line	protected	Aktual dokumen yg hendak di tagihkan.
exchange	Exchange	protected	Nilai tukar mata uang pada saat transaksi dilakukan.

class Line extends Model			
Field/Properties	Type	Akses	Keterangan
paid	boolean	protected	Lunas/Belum lunas.
billed	boolean	protected	Sudah dibuat tagihan/belum.
lineItems	Set<LineItem>	protected	Referensi penagihan.

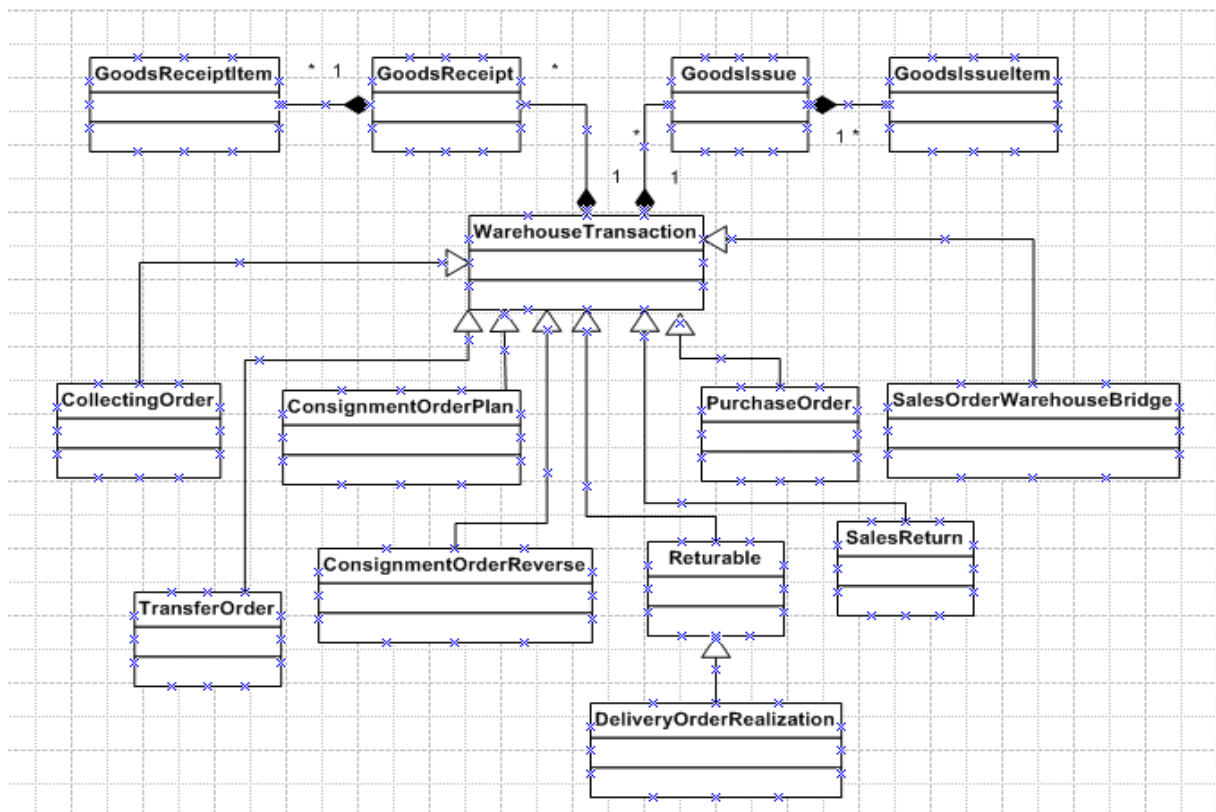
### 4.3 Warehouse Transaction Framework

Warehouse Transaction framework digunakan untuk seluruh bisnis proses yang memerlukan transaksi yang melibatkan keluar masuknya barang dari gudang seperti ( PO, TO, DO, Produksi, Retur, etc ).

Semua domain model yang menginginkan keterlibatan dalam proses pergudangan, diharuskan untuk extends/inheritance dari class com.siriuserp.dm.WarehouseTransaction. Sedangkan rincian mengenai barang yang akan dikeluarkan/dimasukan ke gudang diharuskan untuk extends/inheritance dari class com.siriuserp.dm.WarehouseTransactionItem dan mengimplementasikan beberapa method dari parent class.

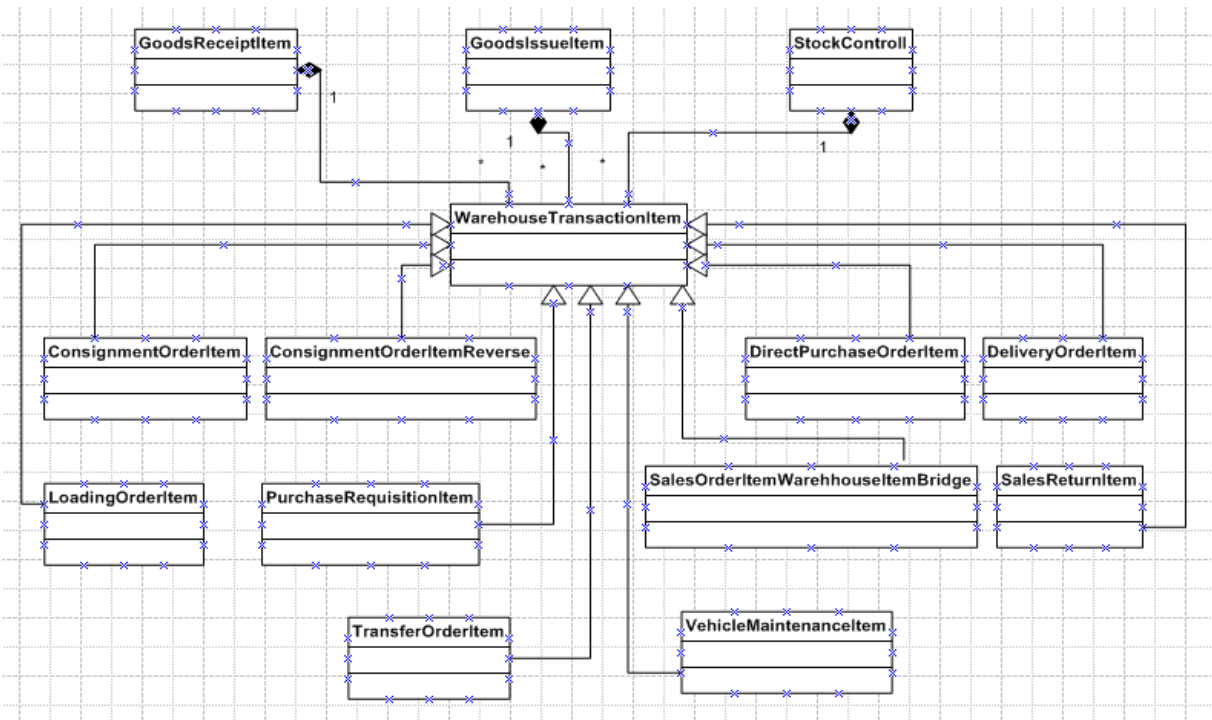
Class diagram dan Screen shoot dari proses penerimaan atau pengeluaran barang dapat dilihat seperti di bawah ini:

#### 4.3.1 Class Diagram WarehouseTransaction dan semua turunannya.

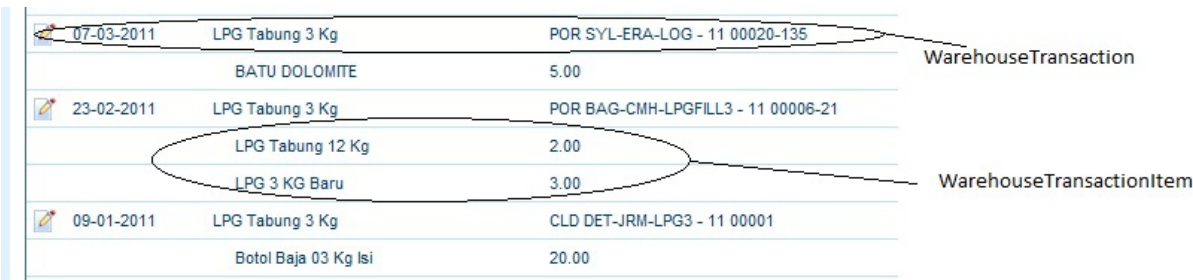




4.3.2 Class Diagram WarehouseTransactionItem dan semua turunannya.



4.3.3 Screenshoot halaman penerimaan/pengeluaran barang.



Model **com.siriuserp.sdk.dm.WarehouseTransaction** di rancang untung meng-handle transaksi barang keluar dan barang masuk ke/dari gudang. Terdapat 3 jenis transaksi keluar masuknya barang dari gudang :

Nama	Keterangan
IN	Tipe ini digunakan untuk penerimaan barang dari pihak ketiga (diluar dari internal organisasi). Transaksi yang menggunakan tipe ini akan berakibat terjadinya pencatatan pada data Stock Controll dari masing – masing product.
OUT	Tipe ini digunakan untuk pengeluaran barang ditujukan untuk pihak ketiga (diluar dari internal organisasi). Transaksi yang

	menggunakan tipe ini akan berakibat terjadinya pencatatan pada data Stock Controll dari masing – masing product.
INTERNAL	Tipe ini digunakan untuk transaksi perpindahan barang di INTERNAL organisasi. Transaksi yang menggunakan tipe ini tidak akan mengakibatkan perubahan pada data Stock Controll dari tiap product.
EXT_INTERNAL	Tipe ini digunakan untuk transaksi perpindahan barang di INTERNAL GROUP organisasi (antar organisasi yg masih di bawah induk perusahaan yang sama). Transaksi yang menggunakan tipe ini akan berakibat terjadinya pencatatan pada data Stock Controll dari masing – masing product.

Tata cara yang harus di lakukan untuk implementasi dari tipe – tipe transaksi di atas:

**IN** : Model yang ingin ikut serta pada proses barang masuk (GoodsReceipt) diharuskan untuk mengimplementasi beberapa abstract method dari WarehouseTransaction model:

Nama	Keterangan
getReceiveFrom()	Method ini digunakan sebaga informasi nama Party/Organization dari mana product yang ingin di terima berasal.
getReceiptFacility()	Mehode ini digunakan sebagai informasi Facility / Gudang tempat barang yang akan di terima di letakkan.

**OUT** : Model yang ingin ikut serta pada proses barang masuk (GoodsIssue) diharuskan untuk mengimplementasi beberapa abstract method dari WarehouseTransaction model:

Nama	Keterangan
getIssuedFrom ()	Method ini digunakan sebaga informasi nama Party/Organization dari mana product yang ingin di dikeluarkan berasal.
getIssuedFacility ()	Mehode ini digunakan sebagai informasi

	Facility / Gudang asal dari barang yang akan di keluarkan.
--	--

**INTERNAL dan EXT\_INTERNAL** : Untuk tipe ini, karena sebenarnya ini merupakan proses barang keluar dan barang masuk sekaligus, maka model diharuskan meng-implementasikan semua method yang ada baik pada tipe IN ataupun OUT.

Model **com.siriuserp.sdk.dm.WarehouseTransaction** berisikan sbb:

<b>class com.siriuserp.sdk.dm.WarehouseTransaction extends Model</b>			
<b>Field/Properties</b>	<b>Tipe</b>	<b>Akses</b>	<b>Keterangan</b>
code	String	protected	
date	Date	protected	
note	String	protected	
receiptable	boolean	protected	
issuable	boolean	protected	
verified	boolean	protected	
goodsType	goodsType	protected	
type	WarehouseTransactionType	protected	
transactionSource	WarehouseTransactionSource	protected	
organization	Organization	protected	
receipts	FastSet<GoodsReceipt>	protected	
issueds	FastSet<GoodsIssue>	protected	

Model **com.siriuserp.sdk.dm.WarehouseTransactionItem** berisikan attribute sbb:

<b>class com.siriuserp.sdk.dm.WarehouseTransactionItem extends Model</b>			
<b>Field/Properties</b>	<b>Tipe</b>	<b>Akses</b>	<b>Keterangan</b>
quantity	BigDecimal	protected	Informasi mengenai jumlah maksimum barang yang dapat diterima/dikeluarkan oleh gudang .
receipted	BigDecimal	protected	Jumlah barang yang telah di terima oleh gudang.

unreceipted	BigDecimal	protected	Jumlah barang yang belum diterima oleh gudang.
issued	BigDecimal	protected	Jumlah barang yang telah dikeluarkan oleh gudang.
unissued	BigDecimal	protected	Jumlah barang yang belum dikeluarkan oleh gudang.
issuedItems	Set<GoodsIssueItem>	protected	Informasi mengenai dokumen pengeluaran barang.
receiptedItems	Set<GoodsReceiptItem>	protected	Informasi mengenai dokumen penerimaan barang.
stockControls	Set<StockControl>	protected	Informasi mengenai stock control.