PERANCANGAN APLIKASI DASHBOARD WMS BERBASIS WEB SERVICE DENGAN MENGGUNAKAN TEKNOLOGI .NET WEB SERVICE DAN PHP

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat Mencapai Gelar Ahli Madya
Program Diploma III Ilmu Komputer
Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Sebelas Maret

Disusun Oleh :

RYAN PERMANA
NIM. M3108120

PROGRAM DIPLOMA III TEKNIK INFORMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS SEBELAS MARET
SURAKARTA
2011

HALAMAN PENGESAHAN

PERANCANGAN APLIKASI *DASHBOARD* WMS BERBASIS *WEB SERVICE* DENGAN MENGGUNAKAN TEKNOLOGI .NET *WEB SERVICE* DAN PHP

Disusun oleh :

Ryan Permana
NIM. M3108120
dengan judul :

Tugas Akhir ini telah diterima dan disahkan
oleh dewan penguji Tugas Akhir
Program Diploma III Ilmu Komputer
pada hari Kamis tanggal 27 Januari 2011

Dewan Penguji:

1. Penguji 1 : Didiek S. Wiyono, ST, MT
   NIP : 197503312005011001 (.....)

2. Penguji 2 : Sri Arum SZ, S.Kom
   NIDN : 0610038202 (.....)

3. Penguji 3 : Agus Purbayu S.Si
   NIDN : 0629088001 (.....)

Mengetahui,	Mengetahui,Ketua
Dekan FMIPA UNS,	Program Diploma III Ilmu Komputer FMIPA UNS,
Prof. Drs. Sutarno, M.Sc. Ph.D.	Drs.Y.S. Palgunadi, M.Sc.
NIP 196008091986121001	NIP 195604071983031004

ABSTRACT

Designing WMS Dashboard application based on Web Service using .NET Web Service Technology and PHP. Ryan Permana NIM M3108120. Program of Diploma III Faculty of Mathematics and Natural Science Sebelas Maret University.2011

Warehouse is a storage media in a Supply Chain Network. Warehouse Management System needs an item inventory, configuration saving method of an item, integration among warehouses, and a warehouse finances. All of the aspects must be well arranged, in order to avoid warehouses's work performance decrease that can affect to supply chain profit. To avoid the supply chain from losses, a warehouses need a system that can help to maintain and manage all of warehouse's variables. One of system that can implemented is WMS Dashbaord. WMS Dashboard is an application that can show the datas from warehouses realtime and with high accuration.

WMS Dashboard's services are made by using ASP.NET Web Service Technology which use SOAP protocol and WSDL interface so it can be used for multi platform system to system communication which using HTTP protocol. The database of WMS Dashboard is made by using MS SQL Server DBMS which has fully compatibility with other applications that made by .NET Framework. Object oriented PHP is need to made the application for client side.

WSDL of WMS dashboard application consist of 56 services. The Web Service services has been integrated with WMS Dashboard Application which based on PHP, so it would be accessed easily, and it can be used to help maintain and manage the Warehouse System variables.

Keywords : WMS Dashboard, ASP .NET Web Service, MS SQL, Object Oriented PHP

HALAMAN INTISARI

Perancangan Aplikasi *Dashboard* Wms Berbasis *Web Service* Dengan Menggunakan Teknologi .Net *Web Service* Dan PHP. Ryan Permana NIM M3108120. Program Diploma III Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Sebelas Maret.2011

Gudang adalah sarana penyimpanan stock barang dalam sebuah rantai supply (*Supply chain*). System manajemen dalam pergudangan memerlukan adanya inventaris barang masuk dan keluar, pengaturan metode penyimpanan barang, integrasi antar gudang serta pengaturan keuangan gudang. Seluruh aspek tersebut harus diatur dengan baik, sebab bila tidak kinerja gudang akan menurun dapat dapat berujung kepada meruginya rantai supply.Untuk mencegah terjadinya kerugian maka dibutuhkan sebuah sistem yang dapat membantu dalam pengawasan dan pengelolaan variable- variable yang ada dalam sistem pergudangan. Salah satu sistem yang dapat diterapkan adalah Sistem *Dashboard* Manajemen Pergudangan atau biasa disebut sebagai Warehouse Management System *Dashboard* (WMS-Dashboard). WMS *Dashboard* adalah sebuah aplikasi yang dapat menampilkan data dari variable- variable yang ada dalam gudang secara *realtime* dengan tingkat keakuratan yang tinggi.

Service untuk aplikasi WMS *Dashboard* dibuat dengan menggunakan teknologi ASP.NET *Web Service* yang menggunakan protokol SOAP dan menggunakan *interface* WSDL sehingga dapat digunakan untuk komunikasi antar system dengan platform yang berbeda- beda melalui protokol HTTP. Database untuk service aplikasi WMS *Dashboard* menggunakan DBMS MS SQL Server yang mendukung intergasi dengan aplikasi- aplikasi yang dibuat dengan .NET Framework. Aplikasi WMS *Dashboard* untuk bagian clientnya dibuat dengan menggunakan bahasa pemrograman PHP dengan menggunaan metode pemrograman berbasis objek.

Telah dibuat WSDL untuk *Web Service* aplikasii WMS *Dashboard* dengan layanan sebanyak 56 service. Layanan- layanan pada *Web Service* tersebut telah diintegrasikann dengan aplikasi WMS *Dashboard* yang berbasis PHP, sehingga mudah untuk diakses dan dapat digunakan untuk membantu dalam pengawasan dan manajemen variable- variable yang ada dalam sistem pergudangan.

Kata kunci : *WMS Dashboard, ASP .NET Web Service*, MS SQL, PHP berbasis Objek .

BAB V
PENUTUP

1.1 Kesimpulan

Berdasarkan aplikasi yang telah dirancang, dapat disimpulkan bahwa:

1. *Web Service* ASP. Net untuk aplikasi WMS *Dashboard* telah berhasil dibangun dengan menggunakan bahasa pemrograman C# dengan memanfaatkan aplikasi Visual Studio 2008.
2. *Web Service* untuk WMS *Dashboard* memiliki total 56 *Service* yang mengacu kepada 14 tabel dari database WMS.
3. Aplikasi client WMS *Dashboard* telah berhasil dibuat dengan menggunakan bahasa pemrograman PHP dengan memanfaatkan aplikasi Dreamweaver CS 4 dan Xcelsius.

1.2 Saran

Saran yang dapat disampaikan pada penulisan tugas akhir ini adalah:

1. *Web Service* WMS *Dashboard* hendaknya distandardisasikan dan didaftarkan pada UDDI, sehingga *service* tersebut dapat dimanfaatkan oleh masyarakat luas.
2. Untuk meningkatkan mobilitas dan update data secara otomatis *WMS dashboard* sebaiknya diintegrasikan dengan teknologi RFID .
3. Sekuritas *Web Service* hendaknya ditingkatkan dengan melakukan enkripsi pada XML yang ditransportasikan
4. Untuk manajemen loading data pada aplikasi WMS *dashboard* disisi client sebaiknya memanfatkan frame atau teknologi Jquery sehingga lebih meng-efektifkan *loading time*, sehingga user tidak mennunggu terlalu lama saat mengakses aplikasi WMS *Dashboard*.

DAFTAR PUSTAKA

Didiek S Wiyono.(2009),TESIS: *Analisis dan Perancangan Aplikasi Web dan Mobile Supply Chain Management pada Distribusi Komoditas Padi Pascapanen (Studi Kasus Sistem SAPA Sukabumi)*,Institut Teknologi Bandung

Andy Harris. (2002), *Microsoft C# Programming for the Absolute Beginner*. Premiere Press.

Martin Verwijmeren (2004), Software component architecture in supply chain management, *Journal Computers in Industry*, 53,pp. 165–178

Holy Icun Yunarto dan Martinus Getty Santika (2005), *Business concepts implementation series in inventory management*, Jakarta: Elex Media Komputindo

Steve Banker. (2006), *WMS Overview* , http://www.frame-wx.com/eng/imgs/02.pdf diakses pada tanggal 16 November 2010

Kadir Abdul, 2004. *Dasar Pemrograman Web Dinamis Menggunakan PHP*. Yogyakarta : Andi

Wikipedia,2010.*Dashboard(Business)*,http://en.wikipedia.org/wiki/Dashboard_(business), diakses pada 7 Januari 2011

Richard Robert. 2006. *Pro PHP XML and Web Services*. New York : Appress.

Christian Nagel, et all. 2010. *Professional C# 4 and .NET 4*. Wiley Publishing

Doug Rosenberg, Scot Kendall. 2001. *Applying Use Case Driven Object with UML : an Annotated e-Commerce Example* . Upper Sadle River : Adison-Wesley.

Kalen Delaney.2000.*Inside Microsoft SQL Server 2000*. Microsoft Press:USA

KPI Library team. 2009.*Warehouse Key Performance Indicator*. http://kpilibrary.com/categories/warehouse , diakses pada 6 Januari 2010

Hendrawan Muh Alfatih. 2009. *Supply Chain Management (SCM)*. *http://kuliahonline.hendrawan.net/download/scm.zip* . diakses pada tanggal 20 mei 2009.

PERANCANGAN APLIKASI DASHBOARD WMS BERBASIS WEB SERVICE DENGAN MENGGUNAKAN TEKNOLOGI .NET WEB SERVICE DAN PHP

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat Mencapai Gelar Ahli Madya
Program Diploma III Ilmu Komputer
Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Sebelas Maret

Disusun Oleh :

RYAN PERMANA
NIM. M3108120

PROGRAM DIPLOMA III TEKNIK INFORMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS SEBELAS MARET
SURAKARTA
2011

PERANCANGAN APLIKASI *DASHBOARD* WMS BERBASIS *WEB SERVICE* DENGAN MENGGUNAKAN TEKNOLOGI .NET *WEB SERVICE* DAN PHP

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat Mencapai Gelar Ahli Madya
Program Diploma III Ilmu Komputer
Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Sebelas Maret

Disusun Oleh :

RYAN PERMANA
NIM. M3108120

PROGRAM DIPLOMA III TEKNIK INFORMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS SEBELAS MARET
SURAKARTA
2011

HALAMAN PERSETUJUAN

PERANCANGAN APLIKASI *DASHBOARD* WMS BERBASIS *WEB SERVICE* DENGAN MENGGUNAKAN TEKNOLOGI .NET *WEB SERVICE* DAN PHP

Disusun Oleh

RYAN PERMANA
NIM. M3108120

Tugas Akhir ini disetujui untuk dipresentasikan
pada Ujian TA
pada tanggal 14 Januari 2011

Pembimbing,

Didiek.S.Wiyono,ST, MT.
NIP 197503312005011001

HALAMAN PENGESAHAN

PERANCANGAN APLIKASI *DASHBOARD* WMS BERBASIS *WEB SERVICE* DENGAN MENGGUNAKAN TEKNOLOGI .NET *WEB SERVICE* DAN PHP

Disusun oleh :

Ryan Permana
NIM. M3108120
dengan judul :

Tugas Akhir ini telah diterima dan disahkan
oleh dewan penguji Tugas Akhir
Program Diploma III Ilmu Komputer
pada hari Kamis tanggal 27 Januari 2011

Dewan Penguji:

1. Penguji 1 : Didiek S. Wiyono, ST, MT
   NIP : 197503312005011001 (.....)

2. Penguji 2 : Sri Arum SZ, S.Kom
   NIDN : 0610038202 (.....)

3. Penguji 3 : Agus Purbayu S.Si
   NIDN : 0629088001 (.....)

Mengetahui,	Mengetahui,Ketua
Dekan FMIPA UNS,	Program Diploma III Ilmu Komputer FMIPA UNS,
Prof. Drs. Sutarno, M.Sc. Ph.D.	Drs.Y.S. Palgunadi, M.Sc.
NIP 196008091986121001	NIP 195604071983031004

ABSTRACT

Designing WMS Dashboard application based on Web Service using .NET Web Service Technology and PHP. Ryan Permana NIM M3108120. Program of Diploma III Faculty of Mathematics and Natural Science Sebelas Maret University.2011

Warehouse is a storage media in a Supply Chain Network. Warehouse Management System needs an item inventory, configuration saving method of an item, integration among warehouses, and a warehouse finances. All of the aspects must be well arranged, in order to avoid warehouses's work performance decrease that can affect to supply chain profit. To avoid the supply chain from losses, a warehouses need a system that can help to maintain and manage all of warehouse's variables. One of system that can implemented is WMS Dashbaord. WMS Dashboard is an application that can show the datas from warehouses realtime and with high accuration.

WMS Dashboard's services are made by using ASP.NET Web Service Technology which use SOAP protocol and WSDL interface so it can be used for multi platform system to system communication which using HTTP protocol. The database of WMS Dashboard is made by using MS SQL Server DBMS which has fully compatibility with other applications that made by .NET Framework. Object oriented PHP is need to made the application for client side.

WSDL of WMS dashboard application consist of 56 services. The Web Service services has been integrated with WMS Dashboard Application which based on PHP, so it would be accessed easily, and it can be used to help maintain and manage the Warehouse System variables.

Keywords : WMS Dashboard, ASP .NET Web Service, MS SQL, Object Oriented PHP

HALAMAN INTISARI

Perancangan Aplikasi *Dashboard* Wms Berbasis *Web Service* Dengan Menggunakan Teknologi .Net *Web Service* Dan PHP. Ryan Permana NIM M3108120. Program Diploma III Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Sebelas Maret.2011

Gudang adalah sarana penyimpanan stock barang dalam sebuah rantai supply (*Supply chain*). System manajemen dalam pergudangan memerlukan adanya inventaris barang masuk dan keluar, pengaturan metode penyimpanan barang, integrasi antar gudang serta pengaturan keuangan gudang. Seluruh aspek tersebut harus diatur dengan baik, sebab bila tidak kinerja gudang akan menurun dapat dapat berujung kepada meruginya rantai supply.Untuk mencegah terjadinya kerugian maka dibutuhkan sebuah sistem yang dapat membantu dalam pengawasan dan pengelolaan variable- variable yang ada dalam sistem pergudangan. Salah satu sistem yang dapat diterapkan adalah Sistem *Dashboard* Manajemen Pergudangan atau biasa disebut sebagai Warehouse Management System *Dashboard* (WMS-Dashboard). WMS *Dashboard* adalah sebuah aplikasi yang dapat menampilkan data dari variable- variable yang ada dalam gudang secara *realtime* dengan tingkat keakuratan yang tinggi.

Service untuk aplikasi WMS *Dashboard* dibuat dengan menggunakan teknologi ASP.NET *Web Service* yang menggunakan protokol SOAP dan menggunakan *interface* WSDL sehingga dapat digunakan untuk komunikasi antar system dengan platform yang berbeda- beda melalui protokol HTTP. Database untuk service aplikasi WMS *Dashboard* menggunakan DBMS MS SQL Server yang mendukung intergasi dengan aplikasi- aplikasi yang dibuat dengan .NET Framework. Aplikasi WMS *Dashboard* untuk bagian clientnya dibuat dengan menggunakan bahasa pemrograman PHP dengan menggunaan metode pemrograman berbasis objek.

Telah dibuat WSDL untuk *Web Service* aplikasii WMS *Dashboard* dengan layanan sebanyak 56 service. Layanan- layanan pada *Web Service* tersebut telah diintegrasikann dengan aplikasi WMS *Dashboard* yang berbasis PHP, sehingga mudah untuk diakses dan dapat digunakan untuk membantu dalam pengawasan dan manajemen variable- variable yang ada dalam sistem pergudangan.

Kata kunci : *WMS Dashboard, ASP .NET Web Service*, MS SQL, PHP berbasis Objek .

HALAMAN MOTTO

"Keberhasilan dengan kerja keras adalah Keberhasilan dengan hasil yang paling Memuaskan"

HALAMAN PERSEMBAHAN

"Tugas Akhir ini penulis persembahkan semua orang yang telah memberi support kepada penulis untuk dapat terus berjuang menyelesaikan Laporan Tugas Akhir ini, khususnya untuk Ibu Sri Rubijantini, Bapak Bambang Supriyadi, Teteh Esti Wibawati, Teteh Lyana Sulistyanti dan Teman- teman Teknik Informatika B 2008: Sidig, Dito, Arif, Hanung, Bocun, DK, Kebo, Rena, Ucup, Ginus, Yuzril, Brian, Agil, Kholis, Atik, Trihandayani, Fibri, Debi and the gank, Angga, Risang, Rochmat, Dian, Danang, Wahyonx dan kawan-kawan lain yang belum disebut namanya ☺ "

KATA PENGANTAR

Puji dan syukur penulis panjatkan ke hadirat ALLAH SWT, karena berkat rahmat dan karunia-NYA, penulis dapat menyelesaikan laporan Tugas Akhir ini. Shalawat dan salam semoga selalu tercurah kepada manusia paling sempurna, Rasulullah Muhammad SAW beserta keluarga suci kenabiannya.

Laporan ini ditulis untuk memenehui syarat kelulusan Kurikulum Tingkat Diploma III Jurusan Teknik Informatika Unisversitas Sebelas Maret Surakarta.

Dalam pengerjaan laporan tugas akhir ini sejak awal hingga akhir, penulis telah mendapat banyak bantuan dan dukungan dari berbagai pihak yang sudah sepantasnya penulis mengucapkan rasa terima kasih yang sedalam- dalamnya kepada :

1. Allah SWT yang selalu memberikan kekuatan untuk menyelesaikan laporan Tugas Akhir ini.
2. Bapak Drs. Y. S Palgunadi, M.Sc. selaku ketua Program D3 Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Sebelas Maret.
3. Bapak Didiek Sri Wiyono, M.T. selaku dosen pembimbing TA yang telah banyak memberikan pengarahan, dukungan moril dan saran. sehingga penulis dapat menyelesaikan laporan ini dengan baik;
4. Kedua orang tuaku tercinta yang selalu memberikan doa dan dukungannya.
5. Kedua kakaku tersayang, Esti Wibawati dan Lyana Sulistyanti, terima kasih dukungan dan bantuannya.
6. My Special Friend, Faradila Azka, terima kasih atas supportnya.
7. Teman-teman D3 Teknik Informatika B angkatan 2008, atas Semangat dan dukungannya.

Penulis menyadari bahwa dengan pengalaman dan ilmu yang amat terbatas yang dimiliki oleh penulis, tentu masih banyak terdapat kekurangan pada laporan

Tugas Akhir ini, untuk itu penulis sangat menghargai dan menantikan kritik serta saran yang membangun dari pembaca.

Akhir kata, penulis berharap agar tulisan ini dapat bermanfaat bagi seluruh pembacanya, terima kasih.

Surakarta, Januari 2011

Penyusun

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

BAB I
PENDAHULUAN

1.1 Latar Belakang Masalah

Gudang sebagai sarana penyimpanan stock sumber daya merupakan bagian yang memiliki peran penting dalam efektivitas sebuah rantai pasok. Sebagai contoh dalam sebuah perusahaan produksi pasti memiliki gudang untuk menyimpan barang- barang mentah untuk bahan baku produksinya. Tanpa adanya gudang, penyimpanan barang- barang tersebut tidak akan terorganisir dengan baik. Dengan fungsinya sebagai penyimpanan stock barang mentah maupun barang jadi maka gudang memerlukan sebuah sistem manajemen untuk mengorganisir kegiatan- kegiatan yang ada dalam gudang. Sebuah gudang yang manajemennya tidak tertata dengan baik akan menyebabkan rusaknya rantai pasok dan meruginya perusahaan.

Dalam manajemen sebuah gudang terdapat banyak hal yang perlu dikelola dengan baik, seperti menentukan metode penyimpanan stock sesuai dengan kriteria yang dibutuhkan stok tersebut, pengintegrasian antar gudang dalam satu wilayah dengan wilayah lainnya, pengelolaan keuangan seperti pengelolaan uang masuk dan keluar serta pengelolaan keuntungan, dan lain sebagainya, Pengelolaan hal- hal tersebut membutuhkan ketelitian dan keakuratan, serta kestabilan dalam pemantauannya, sehingga dibutuhkan sebuah sistem yang mampu diandalkan untuk melakukannya dengan tiga spesifikasi tersebut (teliti, akurat dan stabil). Salah satu sistem yang diajukan untuk membantu masalah yang terjadi dalam manajemen pergudangan adalah Sistem *Dashboard* Manajemen Pergudangan atau biasa disebut sebagai Warehouse Management System *Dashboard* (WMS-Dashboard).

Salah satu jenis usaha yang menggunakan gudang sebagai sarana penyimpanan adalah usaha furniture. Usaha furniture memerlukan gudang untuk menyimpan dua jenis barang yang berbeda yaitu bahan baku untuk pembuatan furniture dan barang jadi yang siap dijual ke pasaran. Kedua jenis barang tersebut memiliki fungsi dan karakteristik yang berbeda, sehingga dalam metode

penyimpanannya pun akan berbeda. Metode penyimpanan yang salah dapat menyebabkan barang menjadi rusak. Selain metode penyimpanan, pengelolaan inventaris/ jumlah ketersediaan barang dalam sebuah gudang furniture juga perlu diperhatikan, karena dengan pengelolaan inventaris yang kurang akurat dapat berakibat pada kurang baiknya aliran perputaran barang, sehingga dapat menyebabkan 'menganggur'nya barang dalam waktu lama. 'Menganggur'nya barang dalam waktu lama dapat menyebabkan rusaknya barang.

Sistem pergudangan perusahaan- perusahaan furniture di Indonesia yang kebanyakan masih menggunakan metode manual dan belum saling terintegrasi memiliki tingkat keakuratan dan ketelitian yang rendah, dan memiliki peluang untuk terjadinya *human error* yang dapat berakibat terjadinya manipulasi data dan ruginya perusahaan. Selain itu akibat dari ketidakakuratan dan ketidaktelitian manajemen pergudangan dapat berakibat pada langkanya sumber daya dari alam, seperti kayu dan lain sebagainya, karena eksploitasi sumber daya alam dengan maksud penimbunan sumber daya untuk bahan produksi yang dapat menyebabkan kerusakan alam yang permanen dan lebih jauh lagi dapat menyebabkan habisnya sumber daya untuk produksi.

WMS-*Dashboard* diharapkan mampu menangani masalah dalam pengelolaan penyimpanan sumber daya dari sebuah rantai pasok yang masih belum tertata dengan baik, terutama dalam masalah pemantauan variabel- variabel yang ada pada gudang seperti jumlah stock dalam gudang dan keadaan stock dalam gudang dengan tingkat keakuratan, ketelitian, kestabilan tingkat flexibilitas yang tinggi. Salah satu teknologi yang mampu mengintegrasikan informasi melalui platform dan device yang berbeda- beda (tingkat flexibilitas tinggi) dengan tingkat kekakuratan, ketelitian dan kestabilan yang menjanjikan dan melalui perantara internet adalah *Web Service.*

1.2 Rumusan Masalah

Rumusan masalah dari penulisan tugas akhir ini adalah : " Bagaimana cara membuat sebuah aplikasi WMS-*Dashboard* yang mampu membantu proses

manajemen pergudangan kayu untuk kepentingan pasokan bahan baku usaha furniture.

1.3 Batasan Masalah

Batasan masalah dalam penulisan Tugas Akhir ini adalah :

1. Pembuatan *Web Service* yang menyediakan service- service mengenai manajemen pergudangan untuk digunakan pada server.
2. Pembuatan aplikasi WMS-*Dashboard* yang memanfaatkan *web service* WMS untuk digunakan pada sisi client.

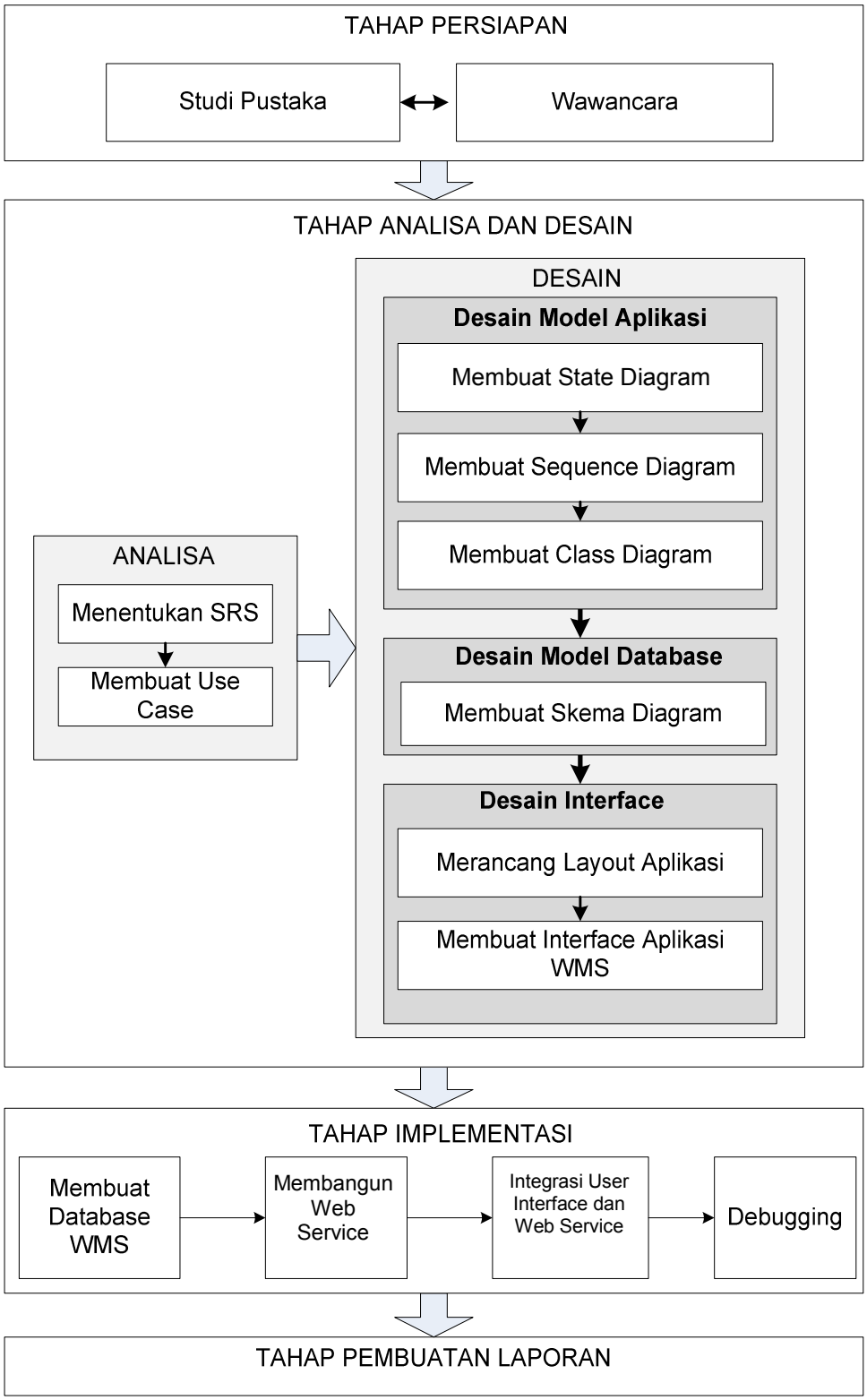
1.4 Tujuan dan Manfaat

Tujuan penyusunan Tugas Akhir ini adalah membangun sebuah aplikasi WMS-*Dashboard* untuk membantu penanganan proses manajemen pergudangan kayu. Manfaat- manfaat yang penulis harapkan dari penyusunan tugas akhir ini antara lain :

1. Untuk Penulis:
   Penyusunan tugas akhir ini merupakan wadah penulis untuk menerapkan ilmu- ilmu yang telah penulis peroleh selama proses belajar di bangku kuliah, serta sebagai portofolio penulis yang dapat dijadikan referensi ketika mencari pekerjaan di kemudian hari.
2. Untuk pengguna :
   Produk akhir dari tugas akhir ini diharapkan dapat mengoptimalkan kinerja gudang dengan melalui pemantauan variable- variable yang ada di dalam gudang secara real time

1.5 Metodologi Penelitian

Laporan tugas akhir ini disusun dengan menggunakan metode penelitian sebagai berikut :



Gambar 1. Bagan metode penelitian

1.6 Sistematika Penulisan

Sistematika penulisan laporan tugas akhir ini adalah sebagai berikut :

1. Bab I Pendahuluan

   Bab ini membahas tentang latar belakang masalah dari penulisan tugas akhir ini, rumusan masalah, batasan masalah, tujuan penulisan tugas akhir, manfaat yang didapatkan dari penulisan tugas akhir, metodologi serta sistematika penulisan yang dipake pada penulisan tugas akhir ini.

2. Bab II Landasan Teori

   Bab ini membahas tentang teori-teori yang digunakan penulis sebagai dasar untuk menyusun tugas akhir ini.

3. Bab III Analisa Dan Perancangan

   Bab ini membahas tentang bentuk desain dan perancangan aplikasi WMS-Dashboard. Pemodelan yang dipakai untuk merancang aplikasi WMS-*Dashboard* ini adalah dengan pembuatan SRS (System Requirement Spesification) dan UML (*Unified Model Language)*. Diagram UML yang digunakan untuk merancang aplikasi WMS-*Dashboard* ini adalah Use Case diagram, State diagram, Sequence diagram, Class diagram, Component diagram dan Deployment diagram.

4. Bab IV Implementasi Dan Pembahasan

   Bab ini membahas tentang ini membahas tentang implementasi aplikasi WMS-*Dashboard* pada jaringan komputer, spesifikasi *hardware* maupun *software* yang dipakai, serta analisa hasil aplikasi WMS-*Dashboard* yang kemudian dibentuk menjadi interface dari projek WMS-*Dashboard* ini

5. Bab V Penutup

   Bab ini membahas tentang kesimpulan dan saran yang penulis ambil dari penulisan tugas akhir ini.

BAB II
LANDASAN TEORI

2.1 Pengertian Gudang

Gudang adalah suatu tempat yang digunakan untuk menyimpan barang baik yang berupa *raw material*, barang *work in process*/setengah jadi atau *finished good*. Dari kata gudang maka didapatkan istilah pergudangan yang berarti merupakan suatu kegiatan yang berkaitan dengan gudang. Menurut Holy Icun Yunarto dan Martinus Getty Santika (2005) kegiatan tersebut dapat meliputi kegiatan *movement* (perpindahan), *storage* (penyimpanan) dan *information transfer* (transfer informasi).

Menurut Holy Icun Yunarto dan Martinus Getty Santika (2005) dalam bukunya menyebutkan beberapa macam tipe gudang, yaitu:

1. *Manufacturing plant warehouse*

   Manufacturing plant warehouse adalah gudang yang ada di pabrik. Transaksi di dalam gudang ini meliputi penerimaan dan penyimpanan material, pengambilan material, penyimpanan barang jadi ke gudang, transaksi internal gudang, dan pengiriman barang jadi ke *central warehouse, distribution warehouse,* atau langsung ke konsumen.

2. *Central warehouse*

   Central warehouse adalah gudang pokok. Transaksi di dalam *central warehouse* meliputi penerimaan barang jadi (dari *manufacturing warehouse*, langsung dari pabrik, atau dari *supplier*), penyimpanan barang jadi ke gudang, dan pengiriman barang jadi ke *distribution warehouse.*

3. *Distribution warehouse*

   Distribution warehouse adalah gudang distribusi. Transaksi dalam gudang ini meliputi penerimaan barang jadi (dari *central warehouse*, pabrik, atau *supplier*), penyimpanan barang yang diterima gudang, pengambilan dan persiapan barang yang akan dikirim, dan pengiriman barang ke konsumen. Terkadang *distribution warehouse* juga berfungsi sebagai *central warehouse*.

4. *Retailer warehouse*

Retailer warehouse adalah gudang pengecer, jadi dengan kata lain dapat dikatakan gudang yang dimiliki toko yang menjual barang langsung ke konsumen.

2.2 Pengertian Warehouse Management System

Warehouse Management System atau Sistem Manajemen Pergudangan adalah sebuah sistem yang bertugas untuk mengatur sumber daya gudang seperti ruang simpan dan stock yang ada di dalam gudang dan serta mengatur aktivitas-aktiitas yang berhubungan dengan sumber daya tersebut seperti *shipping* (pengiriman), *receiving* (penerimaan), *putaway* (penyimpanan), *move* (pergerakan) dan *picking* (pengambilan).

Paradigma baru yang terjadi pada sistem pergudangan sekarang ini adalah melakukan integrasi proses-proses yang ada dengan menggunakan suatu teknologi seperti WiFi LAN, Radio Frequency, Biztalk, Email dan teknologi informasi lainnya. Dengan WMS yang juga terintegrasi dengan teknologi- teknologi tersebut, kita dapat mengontrol proses pergerakan dan penyimpanan dengan lebih baik, pemakaian *space* gudang dengan lebih optimal, meningkatkan efektifitas proses penerimaan dan pengiriman serta mengetahui jumlah stok dengan lebih akurat pada setiap waktu.

Jika pada penerapan WMS telah optimal maka kelebihan- kelebihan yang disebutkan dalam paragraf sebelumnya dapat dicapai dan akhirnya dapat memberikan keuntungan pada perusahaan karena secara prinsip WMS akan mengoptimalkan tenaga kerja, mengurangi waktu proses, mengurangi proses inventory yang tidak perlu dan akhirnya akan meningkatkan pelayanan kita kepada customer selanjutnya.

Di lain pihak, penerapan WMS juga tidak mudah dan membutuhkan penggodokan yang cukup matang. Dari desain *Bussiness Process* sampai dengan teknis harus fix sehingga hasilnya sesuai dengan yang diinginkan.

Tidak setiap gudang dapat atau harus menerapkan WMS karena adakalanya suatu gudang cukup menerapkan sistem pergudangn yang sederhana. Contohnya pada gudang dengan skala kecil atau jenis unit handling yang mudah.

Selain itu, keinginan untuk berinvestasi dari perusahaan pun ikut berperan dalam penerapan WMS. Dana investasi WMS yang terbilang cukup besar, tentunya tidak ingin berakhir dengan sia-sia tanpa hasil.

2.3 Pengertian Supply Chain Management (SCM)

Supply chain management adalah metode, alat dan cara pengelolaan suatu hubungan *supply chain* ditantara perusahaan-perusahaan terkait, sehingga dapat bekerja sesuai dengan yang diharapkan. Definisi SCM menurut Council of Logistics Management, 2004 adalah :

> "Supply Chain Management is the systematic, strategic coordination of the traditional business functions within a particular company and across businesses within the supply chain for the purpose of improving the long-term performance of the individual company and the supply chain as a whole".

SCM melakukan pendekatan kolaborasi antar sistem yang bekerja didalam tiap perusahaan yang berada didalamnya, sehingga diperlukan suatu protokol yang mampu menjembatani perbedaan-perbedaan yang ada diantara platform sistem yang dipakai oleh tiap-tiap perusahaan tersebut. Pendekatan kolaborasi SCM lebih menitik beratkan pada cangkupan eksternal dengan perusahaan-perusahaan partner, bukan internal antar divisi didalam perusahaan tersebut.

Menurut Hendrawan Alfatih, 2004. Tantangan yang harus dihadapi dalam membangun sebuah SCM yang handal adalah :

1. Kompleksitas *supply chain* yang terbentuk

Adanya kompleksitas yang melibatkan internal perusahaan maupun eksternal perusahaan. Internal perusahaan contoh : antara bagian marketing dengan produksi, marketing seringkali membuat kesepakatan dengan pelanggan tanpa mengecek secara baik kemampuan produksi, perubahan jadual produksi secara tiba-tiba karena marketing menyepakati perubahan order dengan

pelanggan. Di sisi lain bagian produksi sering resistant dengan perubahan mendadak.

Dengan eksternal misalnya antara supplier yang menginginkan pemesanan produknya jauh-jauh hari sebelum waktu pengiriman dan sedapat mungkin pesanan tidak berubah. Supplier juga menginginkan pengiriman segera setelah produksinya selesai. Disisi lain perusahaan menghendaki fleksibilitas yang tinggi dengan mengubah jumlah, spesifikasi maupun jadual pengiriman bahan baku yang dipesan. Perusahaan juga menginginkan supplier menggunakan JIT yaitu mengirimkan produk dalam waktu yang tepat dan kuantitasnya kecil-kecil. Kompleksitas yang lain adalah dalam pembayaran, budaya dan bahasa.

2. Ketidakpastian

Ketidakpastian menimbulkan ketidakpercayaan diri terhadap rencana yang dibuat. Sebagai akibatnya, perusahaan sering menciptakan pengaman di sepanjang supply chain. Pengaman ini bisa berupa safety stock, safety time, atau kapasitas produksi maupun transportasi.

2.4 WMS Key Performance Indicator (WMS KPI)

Untuk mengetahui kinerja sebuah system yang berjalan perlu dilakukan perhitungan *Key Performance Indicator* (KPI). KPI adalah indikator kunci tingkat keberhasilan dari sebuah system. Dalam WMS terdapat indikator – indikator yang merupakan hasil perhitungan dari setiap operasi transaksional pada sistem pergudangan yang dilakukan oleh WMS.

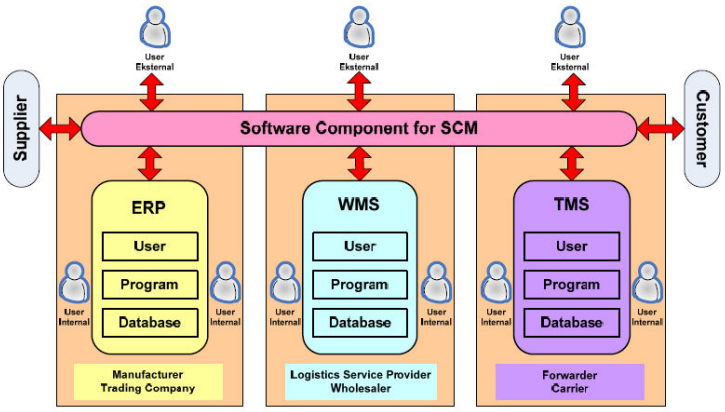
WMS Key Performance Indicator perlu dihitung, sehingga dapat diketahu kinerja dari sebuah WMS. Indikator – indikator kunci (KPI) yang perlu dihitung dapat dilihat seperti pada tabel 1:

Tabel 1. *WMS Key Performance Indicators*

KPI	Deskripsi
Percentage Lost Stock	Persentase barang yang hilang dalam gudang
Percentage Damaged stock	Persentase barang rusak dalam gudang
Percentage of Used Warehouse Space	Persentase jumlah area/ wilayah gudang yang terpakai
Percentage of Activity that Appropiate with SOP	Persentase aktivitas dalam gudang yang sesuai dengan SOP (*Standard of Procedure*)
QOS (Quality Of Service) Percentage	Persentase kuaitas pelayanan dari gudang

2.5 Hubungan WMS dengan SCM

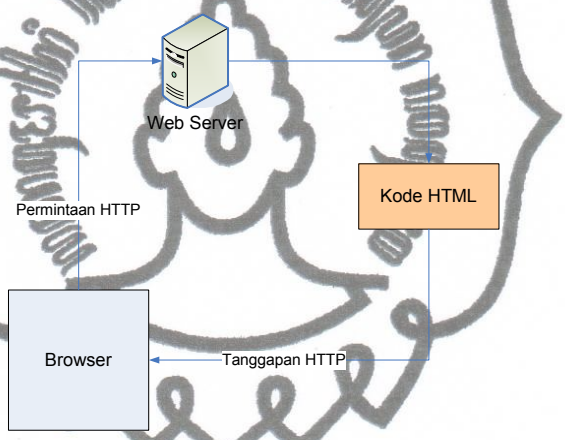
Martin Verwijmeren berpendapat dalam tulisannya bahwa manajemen local dari suatu arsitektur SCM dikerjakan oleh tiga objek utama yaitu *Enterprise Resource Plannning (ERP), Warehouse Management System (WMS) dan Transportation Management System (TMS)*. SCM sendiri berjalan di atas manajemen local dan berfungsi sebagai integrator dari manajemen local kepada *supplier* dan *customer*. Dibawa ini merupakan skema hubungan antara SCM dengan WMS dan objek- objek lainnya :



Gambar 2. Arsitektur SCM dengan objek lain.

2.6 Pengertian Website

Website adalah suatu media publikasi elektronik yang terdiri dari halaman-halaman web (*web page*) yang terhubung satu dengan yang lain menggunakan *link* yang dilekatkan pada suatu teks atau image. Website dibuat pertama kali oleh Tim Barners Lee pada tahun 1990. Website dibangun dengan menggunakan bahasa *Hypertext Markup Language* (HTML) dan memanfaatkan protokol komunikasi *Hypertext Transfer Protocol* (HTTP) yang terletak pada *application layer* pada referensi *layer* OSI. Halaman website diakses menggunakan aplikasi yang disebut internet *browser*. Gambar dibawah ini menunjukan skema kerja pemrosesan file HTML sampai ditampilkan di *browser* (Kadir Abdul. 2004) .



Gambar 3. Skema kerja protokol HTTP

Menurut Jasmadi (2008), Fungsi dari website adalah :

3. Fungsi Komunikasi

Website berfungsi sebagai media komunikasi antara pembuat/pemilik dengan pengunjung atau pengunjung dengan pengunjung lain. Komunikasi dilakukan dengan menggunakan aplikasi *web messanger*, *web forum*, *web chat*, *web mail*, dan lain sebagainya.

4. Fungsi Informasi

Website berfungsi untuk menyediakan informasi bagi pengunjung.

5. Fungsi Hiburan

Website menjadi sarana hiburan, menyediakan layanan *online game*, *video streaming*, *music streaming*, dan lain sebagainya.

6. Fungsi Transaksi

Website berfungsi sebagai sarana untuk melaksanakan transaksi bisnis seperti : *online order*, pembayaran menggunakan kartu kredit, pembayaran dengan *e-gold*, dan sebagainya.

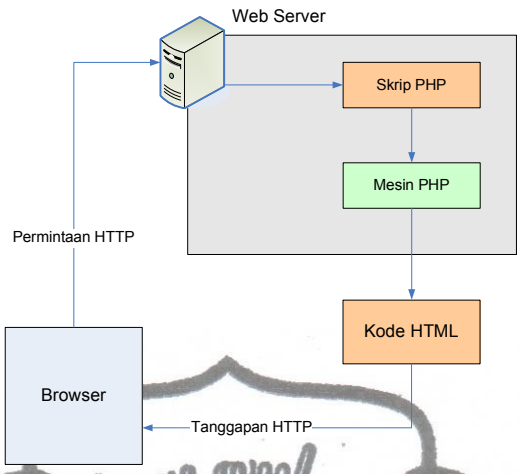
2.7 Pengertian Dashboard

Dashboard adalah sebuah istilah yang digunakan dalam teknologi SI untuk mewakili sebuah aplikasi yang dapat menampilkan informasi- informasi secara Realtime. Teknologi dashboard muncul dengan diilhami dari teknologi dashboard yang ada pada kendaraan bermotor.

Sebuah aplikasi dashboard akan terdiri dari grafik- grafik dan indikator-indikator yang menunjukkan skala keadaan dari variable- variable yang ada dalam sebuah perusahaan. Indikator- indikator tersebut akan menjadi pedoman bagi perusahaan untuk mengambil langkah yang tepat, sehingga tidak mengalami kerugian dalam usahanya.

2.8 Pengertian PHP

PHP (PHP: *Hypertext Preprocessor*) merupakan salah satu dari bahasa pemrograman berbasis website. PHP bersifat *server-side programming*, artinya kode PHP yang ditulis akan dieksekusi di sisi server sehingga pengunjung tidak dapat melihat *source code* dari skrip PHP yang dibangun. Adapun proses eksekusi kode PHP didalam sisi server ditunjukan oleh gambar dibawah ini (Kadir Abdul. 2004) :



Gambar 4. Proses Eksekusi kode PHP

Keunggulan PHP dibanding bahasa pemograman web yang lain antara lain : bersifat *multi platform*, *open source*, memiliki fasilitas untuk OOP(*Object Oriented Programming*) yang merupakan teknik pemrograman yang paling handal dan banyak digunakan saat ini, bersifat gratis, memiliki dukungan API (*Application Programming Interface*) yang sangat lengkap serta didukung oleh hampir semua web hosting yang ada didunia (Lavin Peter. 2006).

2.9 Web Service

Web service merupakan salah satu implementasi dari teknologi XML pada proses pertukaran data (data *exchange*) antar platform yang berbeda. Definisi *web service* menurut Richards Robert, 2006. adalah :

> "A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format(specifically WSDL).Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML seriali-zation in conjunction with other Web-related standards".

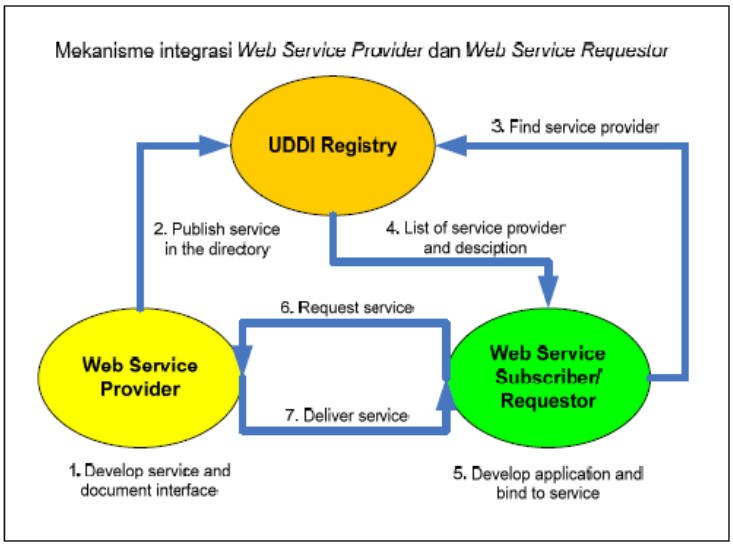
Menurut Richards, *web service* dapat digunakan untuk berkomunikasi antara mesin satu dengan mesin yang lain melalui *interface* perantara yang umumnya berupa WSDL(Web Service Definition Language), layanan ini biasa bekerja pada protokol HTTP dengan bentuk *response* dan *request* berupa SOAP *messange*.

SOAP (*Simple Object Access Protocol*) adalah standar untuk bertukar pesan-pesan berbasis XML melalui jaringan komputer atau sebuah jalan untuk program yang berjalan pada suatu sistem operasi (OS) untuk berkomunikasi dengan program pada OS yang sama maupun berbeda dengan menggunakan HTTP dan XML sebagai mekanisme untuk pertukaran data (Wikipedia, 2009).

Dalam Web Service terdapat tiga actor utama, yaitu :

1. Web Service provider → merupakan pemilik dari layanan yang akan mempublikasikannya menggunakan WSDL
2. Service Registry (UDDI Registry)→ merupakan pihak yang mempublikasikan layanan- layanan milik service provider yang telah berbentuk WSDL.
3. Web Service Requestor → merupakan web service yang akan mencari layanan yang ia butuhkan ke Service registry, kemudian Service Registry akan memberikan informasi mengenai layanan yang ia cari dalam bentuk WSDL, kemudian Web Service Requestor akan mengambil WSDL tersebut, untuk mendapatkan cara mengakses layanan dari Web Service Provider.

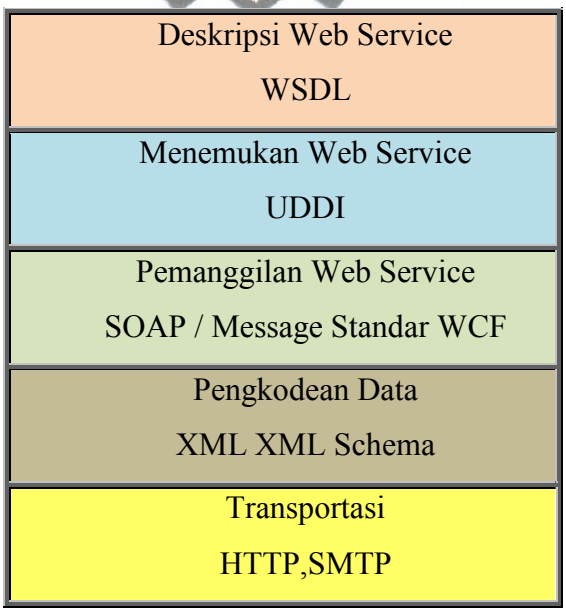
Untuk lebih jelasnya, silahkan lihat ilustrasi di bawah ini :



Gambar 5. Mekanisme Integrasi Web Service Provider dengan Web Service Requestor

Sebelum sebuah layanan dari Web Provider dimanfaatkan oleh Web Requestor, terdapat beberapa tahap yang harus dilakukan, antara lain:

1. Tahap mendeskripsikan Web Service → pada tahap ini web service provider akan mendeskripsikan layanan- layanan miliknya dalam bentuk WSDL untuk selanjutnya didaftarkan ke UDDI.
2. Tahap Menemukan Web Service → pada tahap ini web service requestor menemui UDDI untuk mencari WSDL mengenai service yang ia butuhkan.
3. Pemanggilan Web Service → setelah WSDL dari web service provider ditemukan, maka web service requestor akan membuat aplikasi yang akan mengirimkan SOAP message/ Message standar WCF kepada Web Service Provider untuk mengakses layanan yang dibutuhkan.
4. Pengkodean Data → Baik data yang dikirimkan ke web service provider maupun data yang diterima oleh web service requestor dikirimkan dalam bentuk XML.
5. Transportasi → Data yang dikirimkan ke web service provider maupun data yang diterima oleh web service requestor dikirimkan melalui HTTP atau SMTP.

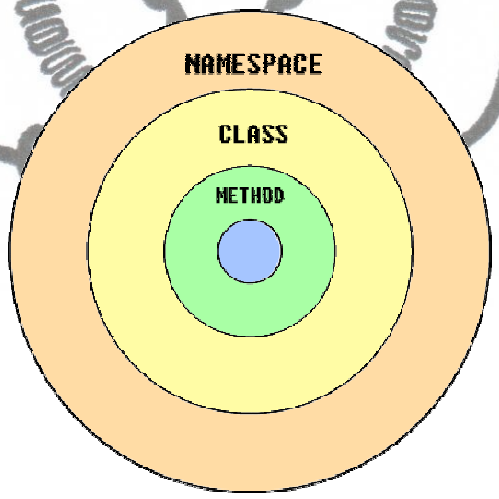


Gambar 6. Ilustrasi Pembentukan dan Pemanfaatan Web Service

2.10 Pengertian C#

Bahasa Pemrograman C# (red. C Sharp) adalah sebuah bahasa pemrograman tingkat menengah yang dapat digunakan untuk membuat sebuah program eksekusi. C# merupakan bahasa pengembangan dari bahasa C++. C# dikembangkan oleh team dari Microsoft, pengembangan bahasa C# awalnya dikhususkan untuk *Framework* .NET. Meskipun pengembangan bahasa C# dilakukan oleh tim dari Microsoft namun C# tetap menjadi bahasa yang flexible, karena compiler C # juga tersedia untuk platform- platform lain seperti FreeBSD, Linux dan Macintosh.

Bahasa pemrograman C# merupakan bahasa pemrograman berbasis Objek, sama seperti Java. Sebuah program yang dibuat dengan bahasa pemrograman C# memiliki 3 layer utama, yang dapat diilustrasikan dengan gambar dibawah ini :



Gambar 7. 3 layer utama dari sebuah program C#

Layer terluar dari sebuah program C# adalah Namespace. Namespace merupakan layer paling umum dan terbesar dalam bahasa pemrograman C#. Dalam kehidupan nyata, Namespace bisa dianalogikan sebagai Negara Bagian/ Provinsi dalam sebuah wilayah Negara.Namespace merupakan bagian dari program C# yang memilki fungsi untuk mengelompokkan bagian- bagian kecil dari sebuah program C#.

Layer ke 2 merupakan layer *class*.Sebuah Namespace biasanya terbuat dari satu atau kebih classes. Sebuah class merupakan definisi dari sebuah objek spesifik. Semua hal yang dapat dideskripsikan oleh sebuah computer (database, file, image) dapat dienskripsikan sebagai sebuah objek. Hal- hal yang dapat dilakukan objek dikenal dengan nama *Methods* dan karakteristik/ sifat- sifat dari sebuah objek dikenal dengan nama *Properties*.

Layer terakhir dikenal dengan nama *Method*. Sebuah *Class* dalam program C# selalu memiliki *method*. *Method* adalah sebuah struktur yang terdiri dari instruksi- instruksi yang dapat dilakukan oleh sebuah *Class*. Sebagian besar program memiliki *Method* khusus yang biasa disebut sebagai *Main() Method*. *Main() Method* adalah sebuah *Method* yang akan dieksekusi pertama kali saat sebuah program dijalankan.

Visual Studio.NET merupakan sebuah editor visual untuk beberapa bahasa pemrograman yang berjalan dalam platform Microsoft seperti Visual Basic dan C++. Pada Visual Studio 2005, Microsoft menambahkan sebuah bahasa pemrograman lagi yaitu bahasa C#.

Visual Studio bukan satu- satunya editor untuk bahasa C#, pembuatan sebuah program dengan bahsa C# dapat juga menggunakan editor yang lain, baik editor visual (SharpDevelop) maupun editor text (Notepad ++, TextPad maupun JEdit). Sehinggan dengan maupun tanpa Visual Studio C# tetap disebut sebagai C#.

2.11 Pengertian ASP .NET Web Service

ASP.NET web services adalah sebuah teknologi web service keluaran Microsoft yang menggunakan protokol SOAP (Simple Object Access Protocol) dan dibuat menggunakan .Net *Framework*. ASP.Net Web Service menggunakan bahasa pemrograman C# dan teknik- teknik pemrograman .Net *Framework*, namun meskipun menggunakan bahasa dan teknik yang beraroma kental dengan Microsoft, service- service yang dibuat dengan .Net *Framework* tetap bisa dikonsumsi oleh berbagai macam platform, selama platform tersebut mendukung penggunaan protokol HTTP untuk berhubungan dengan Server. Sehingga bisa

disebut juga service- service yang dibuat menggunakan .Net *Framework* merupakan service yang multi platform dan juga multi OS.

Web Service yang dibuat dengan menggunakan .Net *Framework* memiliki 2 file utama, yaitu file service yang berekstensi *.asmx dan file kode pendukung/*code behind* yang berekstensi *.cs. File dengan ekstensi asmx biasanya berisi konfiguraasi bahasa yang digunakan dalam service dan juga konfigurasi code behind yang digunakan, sedangkan untuk file dengan ekstensi *.cs akan berisi method- method dari service tersebut yang dibangun dengan bahasa pemrograman C#.

2.12 Pengertian DBMS Microsoft SQL Server

Microsoft SQL Server atau biasa disebut sebagai SQL Server adalah DBMS (Database Management System) keluaran Microsoft.. SQL Server pertama dikeluarkan untuk platform OS/2 (-+ 1988) dengan kerjasama antara Microsoft, Sybase dan Ashton-Tate. Pada tahun 1992 Microsoft baru mengeluarkan SQL Server untuk platform Windows, platform pertama yang dapat mengaplikasikan SQL Server adalah platform Windows NT.

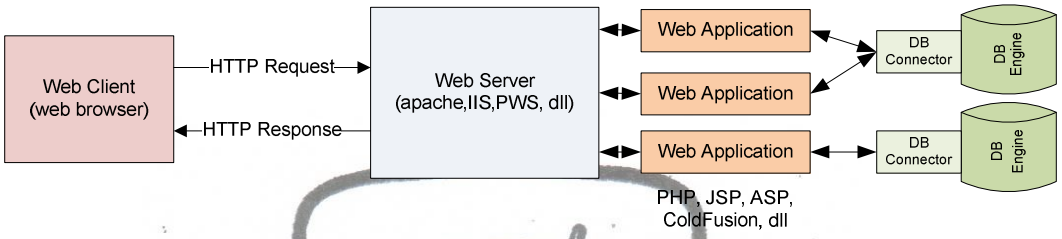
SQL Server memiliki beberapa kelebihan, antara lain :

1. Kehandalan performa dengan jumlah pengguna yang tinggi. SQL Server dapat digunakan untuk aplikasi- aplikasi dengan jumlah pengguna yang tinggi seperti web sites.
2. Keamanan data di SQL Server lebih terjaga dan terjamin dibandingkan dengan DBMS lain.
3. Pengelolaan database menjadi lebih mudah karena dengan adanya fasilitas *automated repair*, *transaction logs*, *trigger* dan *stored procedure*.

2.11 Pengertian Web Server

Web server adalah suatu perangkat lunak yang berfungsi untuk melayani aktifitas *request* and *reply file-file* web. Salah satu web server yang paling banyak digunakan untuk web- web yang dikembangkan dengan platform .NET adalah IIS Web Server. Keunggulan IIS antara lain : mendukung aplikasi yang dibuat dengan

menggunakan *Framework* .NET dan dapat meningkatkan nilai jual suatu aplikasi karena aplikasi yang dibuat dengan *Framework* .NET dikarenakan .NET adalah aplikasi yang komersil. Berikut adalah bagan arsitektur *web service* :



Gambar 8. Arsitektur Web Server

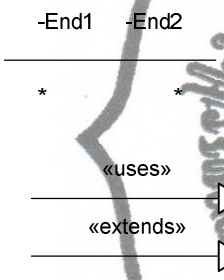
2.13 Pengertian UML (Universal Model Language)

UML adalah salah salah satu bahasa visual untuk mempresentasikan dan mengkomunikasikan sistem melalui penggunaan diagram dan teks pendukung (Doug Rosenberg, Scot Kendall. 2001). Guna fungsi pemodelan visual ini, UML menggunakan 8 jenis diagram standard, yaitu :

2.13.1 Use Case

Use Case digunakan pada saat pelaksanaan tahap *requirment* dalam pengembangan suatu sistem informasi. Use Case menggambarkan hubungan antara entitas yang biasa disebut aktor dengan suatu proses yang dapat dilakukannya. Berikut adalah simbol-simbol yang digunakan dalam Use Case beserta deskripsinya.

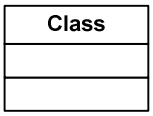
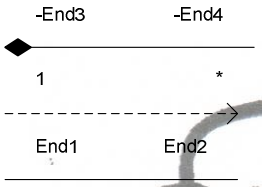
Tabel 2.Simbol Use Case

No.	Simbol	Nama	Deskripsi
1.		Case	Menggambarkan proses / kegiatan yang dapat diakukan oleh aktor
2.		Actor	Menggambarkan entitas / subyek yang dapat melakukan suatu proses
3.		Relation	Relasi antara case dengan actor ataupun case dengan case lain.

2.13.2 Class Diagram

Class Diagram digunakan untuk menggambarkan stuktur kelas dan obyek yang akan digunakan dalam sistem yang akan dibangun. Class Diagram digunakan pada tahap analisa dan desain aplikasi. Berikut adalah simbol-simbol yang digunakan dalam Class Diagram.

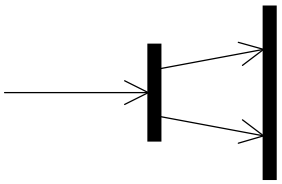
Tabel 3 Simbol Class Diagram

No.	Simbol	Nama	Deskripsi
1.		Class	Menggambarkan sebuah kelas yang terdiri dari attribut dan method
2.		Relation	Menggambarkan hubungan komponen-komponen didalam Static Diagram.

2.13.3 State Chart Diagram

State Chart Diagram digunakan untuk menjelaskan siklus hidup dari sebuah elemen. State Chart digunakan dalam tahap desain dalam pembangunan suatu aplikasi. Berikut ini adalah simbol-simbol yang digunakan dalam State Chart Diagram.

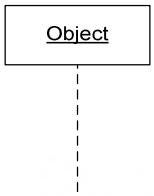
Table 4 Simbol State Chart Diagram

No.	Simbol	Nama	Deskripsi
1.		State	Menggambarkan kondisi suatu elemen
2.		Transition	Menggambarkan aliran siklus state (kondisi) suatu elemen
3.		Decision	Menggambarkan suatu percabangan logika dalam sistem

2.13.4 Sequence Diagram

Sequence Diagram digunakan untuk menjelaskan aliran pesan dari suatu Class ke Class lain secara sequensial (berurutan). Sequence Diagram digunakan pada tahap desain aplikasi. Berikut adalah simbol yang digunakan dalam Sequence Diagram

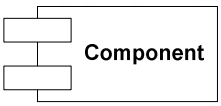
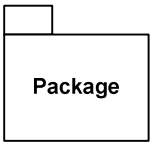
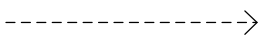
Tabel 5 Simbol Sequnce Diagram

No.	Simbol	Nama	Deskripsi
1		*Object*	Menggambarkan pos-pos obyek yang pengirim dan penerima message
2		Message	Menggambarkan aliran pesan yang dikirim oleh pos-pos obyek

2.13.5 Component Diagram

Component Diagram digunakan untuk menjelaskan hubungan komponen-komponen sistem. Komponen digunakan dalam tahap desain aplikasi. Berikut adalah simbol-simbol yang digunakan dalam Component Diagram.

Tabel 6 Simbol Component Diagram

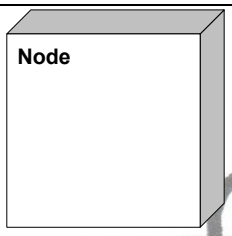
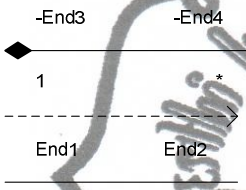
No.	Simbol	Nama	Deskripsi
1.		Component	Menggambarkan sebuah Komponen
2.		Package	Menggambarkan sebuah package dari class-class yang bekerja sama membentuk suatu fungsi tertentu.
3		Dependency	Menggambarkan hubungan antar komponen

2.13.6 Deployment Diagram

Deployment Diagram digunakan untuk menjelaskan implementasi aplikasi yang dibuat ke dalam sebuah environment. Deployment Diagram digunakan

dalam tahap desain aplikasi. Berikut ini adalah simbol-simbol yang digunakan dalam Deloyment Diagram.

Tabel 7 Simbol Deployment Diagram

No.	Simbol	Nama	Deskripsi
1.		Node	Menggambarkan Node sistem atau *environment*.
2.		Relation	Menggambarkan hubungan *node-node* didalam Static Diagram.

BAB III
DESAIN DAN PERANCANGAN

3.1 *System Requirement Spesification* (SRS)

Untuk merancang suatu aplikasi perlu diketahui dan diidentifikasi terlebih dahulu spesifikasi aplikasi yang akan dibuat yang disesuaikan dengan kebutuhan dari sisi user, fungsionalitas sistem yang akan dirancang serta dukungan lingkungan yang dibutuhkan.

Untuk menganalisa kebutuhan dari user, dapat digunakan sebuah tool bernama *System Requirement Spesification* (SRS). SRS akan menjabarkan apa yang dibutuhkan user dalam sebuah aplikasi.

SRS untuk aplikasi dashboard WMS ini dibagi menjadi 2 bagian yaitu SRS fungsional (SRS-WDF) yang menjabarkan kebutuhan user yang berkaitan langsung dengan fungsi utama dari aplikasi dan SRS non-fungsional (SRS-WDNF) yang menjabarkan kebutuhan user yang memanfaatkan fungsi tambahan dari aplikasi. Kedua SRS tersebut akan dijabarkan sebagai berikut :

Tabel 7. SRS Fungsional

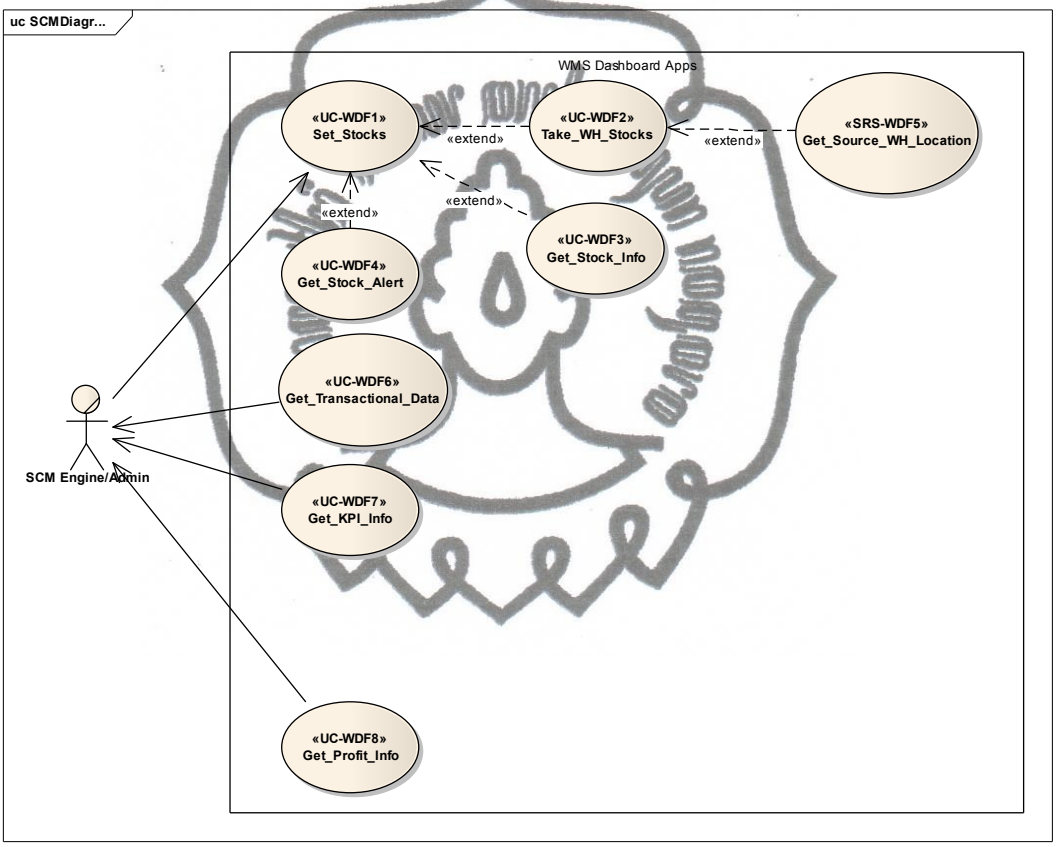
KODE SRS	DESKRIPSI KEBUTUHAN FUNGSIONAL
SCM Engine/Admin	
SRS-WDF 1	Set stocks specification (Fill the stock)
SRS-WDF 2	Take stocks from warehouse
SRS-WDF 3	Get stocks information
SRS-WDF 4	Get stock alert
SRS-WDF 5	Get the source warehouse location
SRS-WDF 6	Get All Transactional Data
SRS-WDF 7	Get KPI Information
SRS-WDF 8	Get Profit Information
WMS Dashboard Application	
SRS-WDF 9	Set effectiveness of warehouse space percentage
SRS-WDF 10	Set lost stock percentage
SRS-WDF 11	Set damaged stock percentage
SRS-WDF 12	Set Quality Of Service percentage
SRS-WDF 13	Set the warehouse activity that appropriate with SOP percentage
SRS-WDF 14	Calculate used space in warehouse
SRS-WDF 15	Calculate unused space in warehouse
SRS-WDF 16	Calculate lost stock
SRS-WDF 17	Calculate damaged stock
SRS-WDF 18	Calculate Stock Amount
SRS-WDF 19	Calculate warehouse activity that appropriate with SOP
SRS-WDF 20	Calculate warehouse activity that un-appropriate with SOP
SRS-WDF 21	Calculate total of warehouse activity
SRS-WDF 22	Calculate warehouse total income
SRS-WDF 23	Calculate Warehouse total Outcome
SRS-WDF 24	Organize stock information data
SRS-WDF 25	Set Profit Information

Tabel 8. SRS Non Fungsional

KODE SRS	DESKRIPSI KEBUTUHAN NON-FUNGSIONAL
User/Owner	
SRS-WDNF 1	Get User Friendly Interface
SRS-WDNF 2	Set Login/Logout
SRS-WDNF 3	Set Password
SRS-WDNF 4	Set Login Email
SRS-WDNF 5	See&Print Transactional Data
SRS-WDNF 6	See&Print KPI Information
SRS-WDNF 8	See&Print Profit Information
SCMEngine/Admin	
SRS-WDNF 9	Authoring Process
General User	
SRS-WDNF10	Get WMS Dashboard Info

3.2 Use Case Diagram

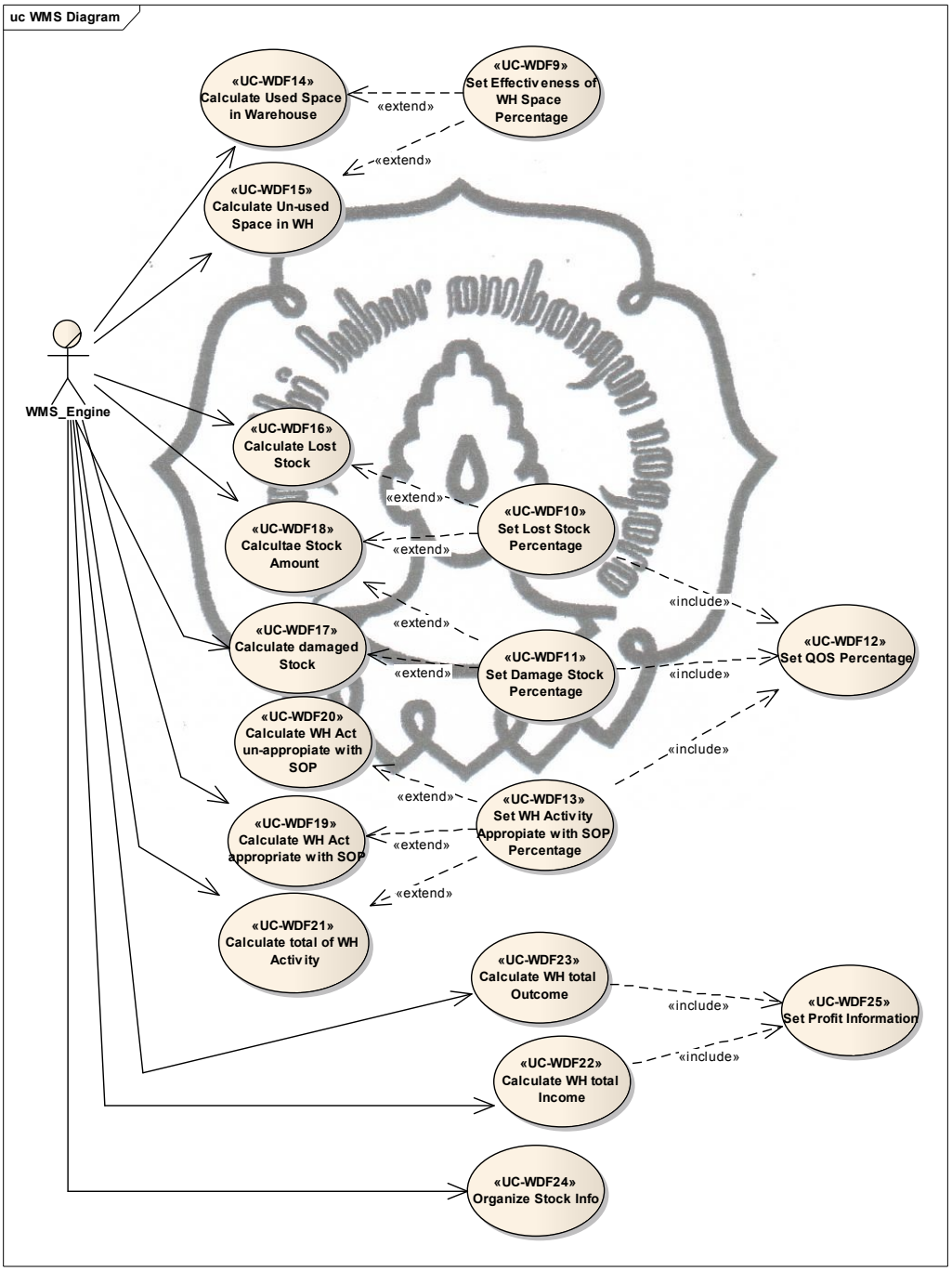
Berdasarkan SRS Fungsional dan non- Fungsional WMS Service dan berdasarkan aktor- aktor yang terlibat didalamnya, use case diagram untuk WMS Service dapat dibagi menjadi 3 bagian yaitu use case fungsional untuk aktor SCM/Admin, dan WMS *Engine* serta use case non fungsional. 3 Use Case diagram tersebut adalah sebagai berikut:



Gambar 9 . Use Case Diagram SCM /Admin

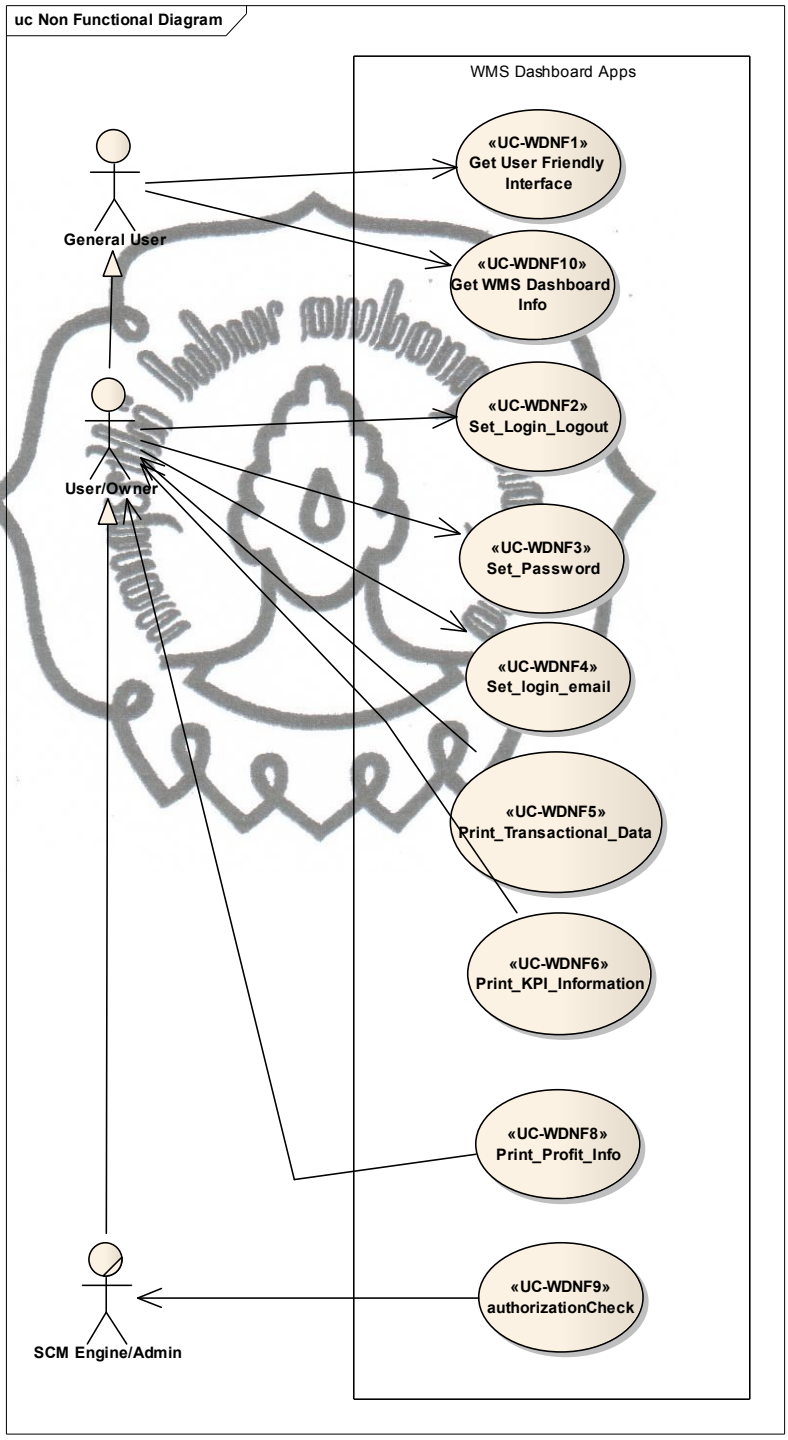
Use case diagram untuk SCM /Admin terdiri dari 8 use case, dari 8 use case terdapat 5 use case yang saling berhubungan, yaitu use case Set Stocks, Get Stock Alert, Take WH Stock, Get Stock Info dan Get Source WH Location. Kelima use case tersebut saling dihubungkan dengan relasi extend. Use case take WH Location, Get Stock info dan Get Stock Alert akan muncul/ aktif jika use case Set Stock ada. Use case Get Source WH Location juga bergantung kepada

use case Take WH Stock, karena jika use Take WH Stock, maka use case Get Source WH Location juga tidak ada.



Gambar 10 . Use Case Diagram WMS Engine

Use case diagram dari WMS *Engine* untuk WMS Service. Pada use case diagram ini terdapat aktivitas- aktivitas yang dapat dilakukan oleh WMS Service. Use Case Diagram ini terdiri dari 18 Use Case.



Gambar 11 . Use Case Diagram Non Fungsional

Use case diagram non fungsional berisi aktivitas- aktivitas yang dapat dilakukan oleh user, dimana aktivitas- aktivitas tersebut tidak berpengaruh pada fungsi utama dari WMS Service. Dari use diagram ini juga dapat dilihat user- user yang dapat menggunakan aplikasi WMS Dashboard dan juga level generalisasinya.

Berikut ini adalah tabel kesesuaian use case dengan SRS dari WMS Service :

Tabel 9. Tabel Kesesuaian SRS dan UC Fungsional

FUNCTIONAL NEEDS	**SRS**	**Use Case**
Set stocks specification (Fill the stock)	SRS-WDF 1	UC-WDF1
Take stocks from warehouse	SRS-WDF 2	UC-WDF2
Get stocks information	SRS-WDF 3	UC-WDF3
Get stock alert	SRS-WDF 4	UC-WDF4
Get the source warehouse location	SRS-WDF 5	UC-WDF5
Get All Transactional Data	SRS-WDF 6	UC-WDF6
Get KPI Information	SRS-WDF 7	UC-WDF7
Get Profit Information	SRS-WDF 8	UC-WDF8
Set effectiveness of warehouse space percentage	SRS-WDF 9	UC-WDF9
Set lost stock percentage	SRS-WDF 10	UC-WDF10
Set damaged stock percentage	SRS-WDF 11	UC-WDF11
Set Quality Of Service percentage	SRS-WDF 12	UC-WDF12
Set the warehouse activity that appropriate with SOP percentage	SRS-WDF 13	UC-WDF13
Calculate used space in warehouse	SRS-WDF 14	UC-WDF14
Calculate unused space in warehouse	SRS-WDF 15	UC-WDF15
Calculate lost stock	SRS-WDF 16	UC-WDF16
Calculate damaged stock	SRS-WDF 17	UC-WDF17
Calculate Stock Amount	SRS-WDF 18	UC-WDF18
Calculate warehouse actiity that appropiate with SOP	SRS-WDF 19	UC-WDF19
Calculate warehouse actiity that un-appropiate with SOP	SRS-WDF 20	UC-WDF20
Calculate total of warehouse activity	SRS-WDF 21	UC-WDF21
Calculate warehouse total income	SRS-WDF 22	UC-WDF22
Calculate Warehouse total Outcome	SRS-WDF 23	UC-WDF23
Organize stock information data	SRS-WDF 24	UC-WDF24
Set Profit Information	SRS-WDF 25	UC-WDF25

Tabel 10. Tabel Kesesuaian SRS dan UC Non Fungsional

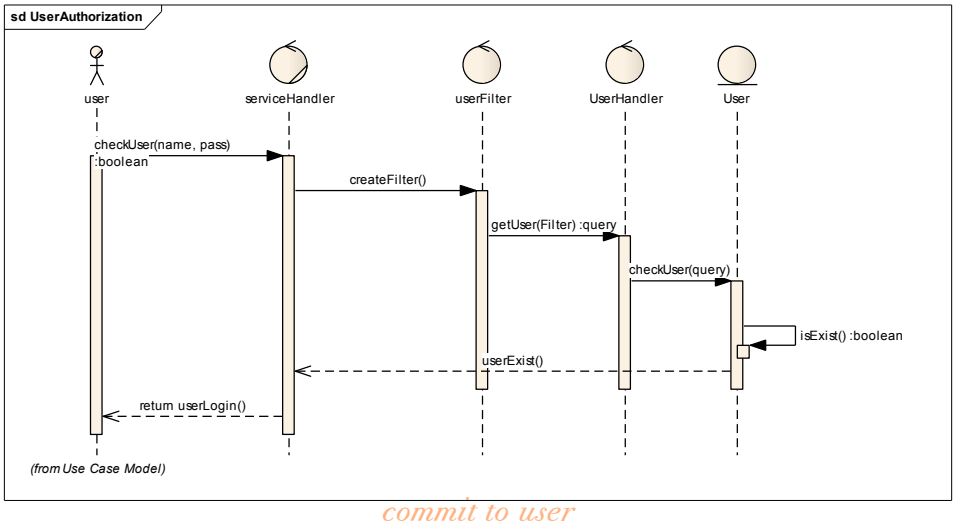
NON- FUNCTIONAL NEEDS	SRS	Use Case
Get User Friendly Interface	SRS-WDNF 1	UC-WDNF1
Set Login/Logout	SRS-WDNF 2	UC-WDNF2
Set Password	SRS-WDNF 3	UC-WDNF3
Set Login Email	SRS-WDNF 4	UC-WDNF4
Print Transactional Data	SRS-WDNF 5	UC-WDNF5
Print KPI Information	SRS-WDNF 6	UC-WDNF6
Print Profit Information	SRS-WDNF 8	UC-WDNF8
Authoring Processing	SRS-WDNF 9	UC-WDNF9
Get WMS Dashboard Info	SRS-WNDF10	UC-WDNF10

3.3 Sequence Diagram

Sequence diagram adalah, diagram yang menggambarkan hubungan class-class dan method- method yang digunakan saat sebuah fitur/ menu diaktifkan dalam sebuah aplikasi. Sequence diagram untuk aplikasi WMS Dashboard dibagi menjadi sembilan yaitu sequence diagram login, sequence diagram set stock, sequence diagram get transaction data, sequence diagram get stock info, sequence diagram take stock, sequence diagram get warehouse source, sequence diagram get KPI info, seequence diagram get profit info dan yang terakhir adalah sequence diagam get alert.

3.3.1. Sequence Login

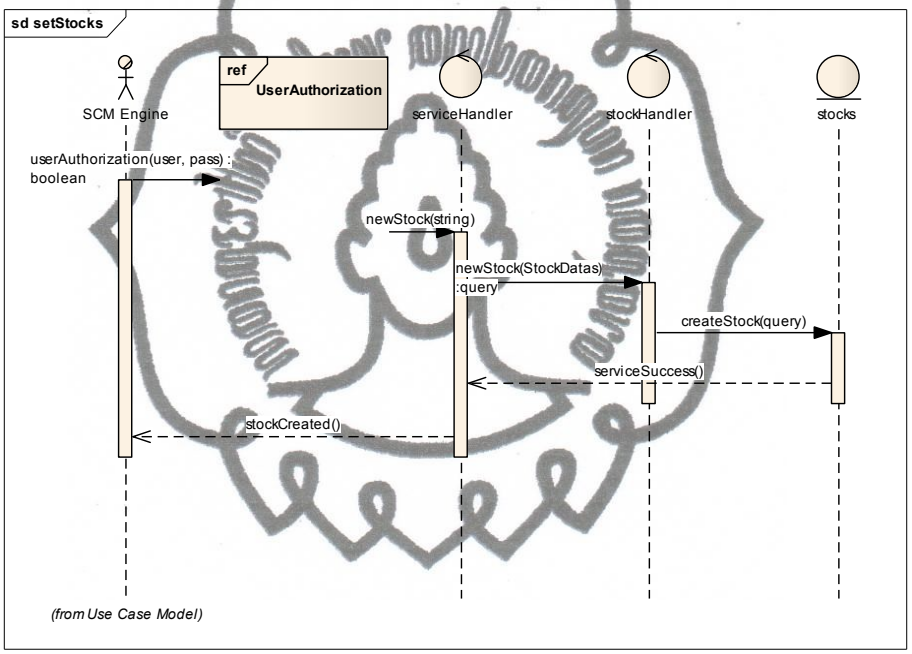
Sequence diagram ini menjelaskan class- class dan method apa saja yang digunakan saat aktivitas login dilakukan. Saat login terdapat empat class yang saling berinteraksi, untuk lebih jelasnya silahkan lihat pada gambar 12:



Gambar 12 . Sequence Diagram untuk Proses Login

3.3.2. Sequence Set Stock

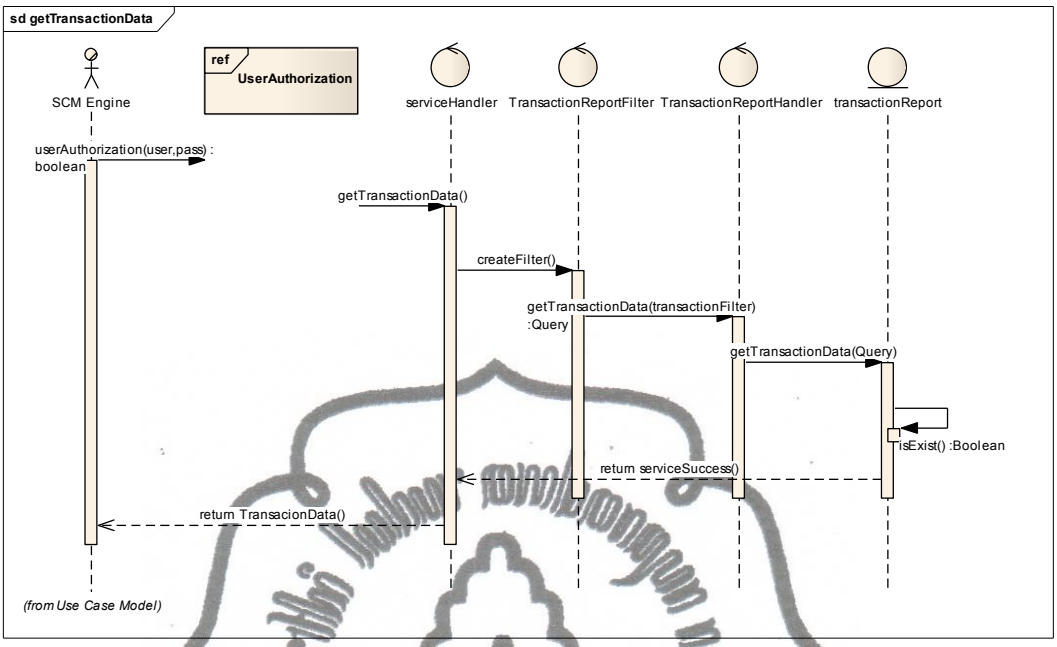
Sequence diagram untuk set stock, memanfaatkan empat class dari aktivitas login dan tiga class khusus yang saling berinteraksi. Empat class dari class login dapat dilihat pada bagian sebelumnya tepatnya pada gambar 12. Sedangkan untuk tiga class khususnya adalah class serviceHandler, class stockHandler dan class stocks.Method yang digunakan ada empat yaitu method userAuthorization(), newStock(string), newStock(StockData) dan createStock(). Untuk lebih jelasnya dapat dilihat pada gambar 13 dibawah ini:



Gambar 13 . Sequence Diagram untuk Proses Set Stock

3.3.3.Sequence GetTransactionalData

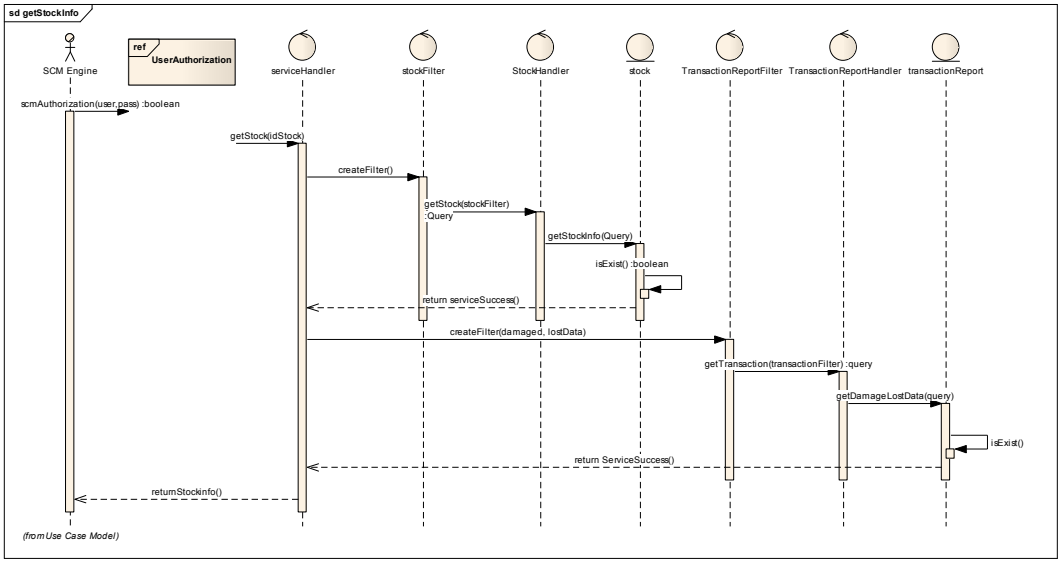
Aktivitas GetTransactionData memanfaatkan total delapan class yang saling berinteraksi. Delapan class tersebut terdiri dari empat class untuk authorisasi user dan empat class untuk mengambil data transactional. Sedangkan untuk methodnya aktivitas ini melibatkan lima method dari satu proses dan empat class. Untuk lebih jelasnya dapat dilihat pada gambar 14 :



Gambar 14 . Sequence Diagram untuk Proses Get Transactional Data

3.3.4.Sequence Get Stock Info

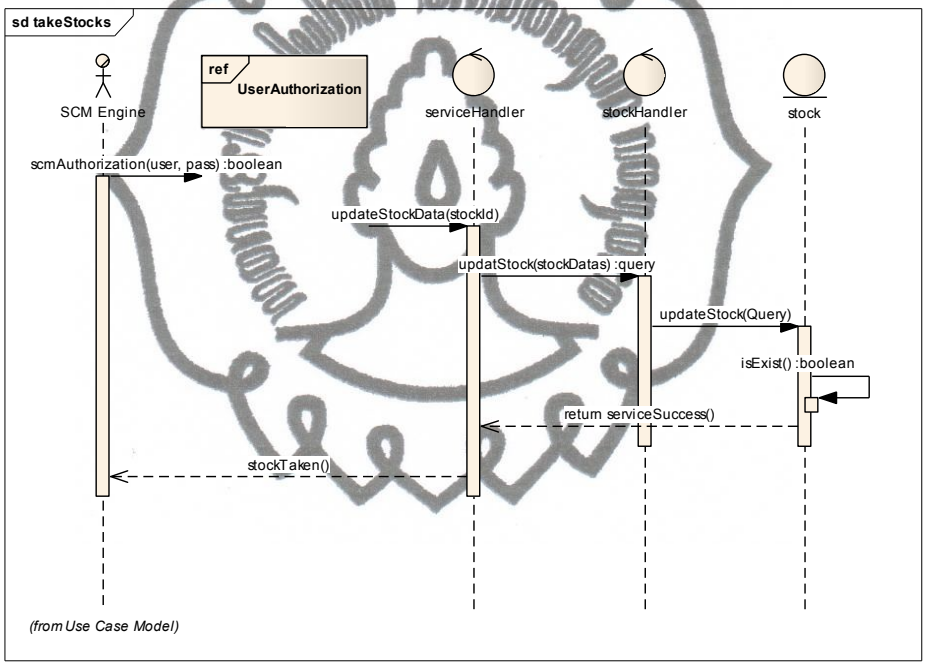
Sequence diagram ini menjelaskan class- class dan method apa saja yang digunakan saat aktivitas GetStockInfo dilakukan. Saat aktivitas GetStockInfo dilakukan terdapat sebelas class yang saling berinteraksi, yang terdiri dari empat class untuk autorisasi dan tujuh class untuk mendapat info stock. Untuk lebih jelasnya silahkan lihat pada gambar 15:



Gambar 15 . Sequence Diagram untuk Proses Get Stock Info

3.3.5.Sequence Take Stock

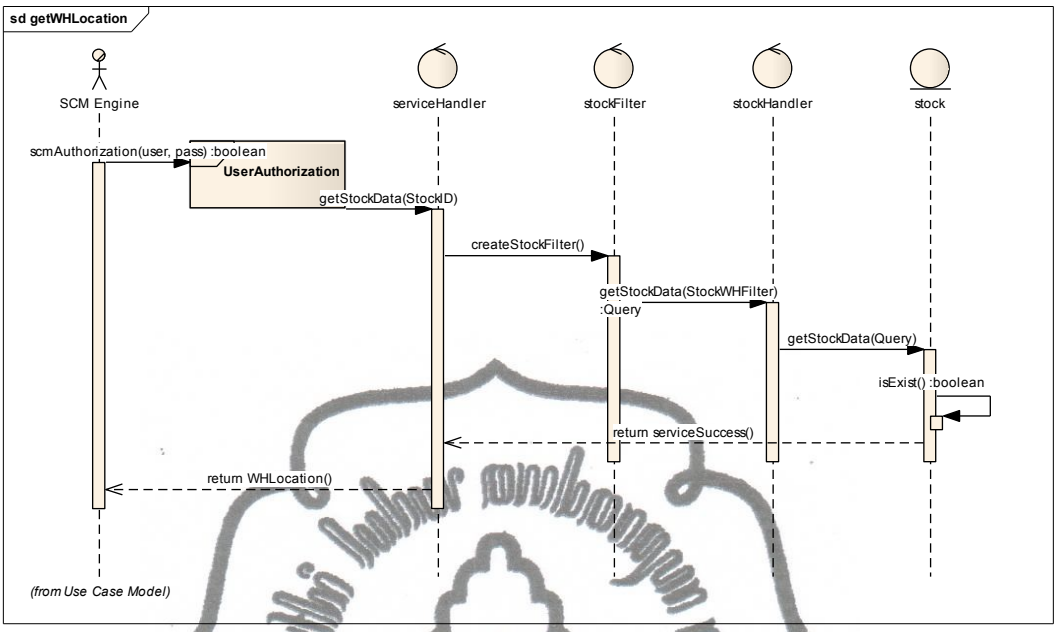
Aktivitas Take Stock, atau mengambil barang dari gudang memanfaatkan tujuh class yang saling berinteraksi. Empat class untuk authorisasi user dan tiga class untuk mengambil stock. Tiga class tersebut adalah class- class khusus yang berhubungan dengan tabel stock pada database. Sedangkan untuk method yang dipanggil, terdapat empat buah method yang digunakan salah satunya adalah method untuk authorisasi user yaitu scmAuthorization(). Adanya method scmAuthorization mengindikasikan bahwa user yang dapat mengambil stock hanya SCM. Untuk lebih jelasnya dapat dilihat pada gambar 16:



Gambar 16 . Sequence Diagram untuk Proses Mengambil/Take Stock

3.3.6.Sequence get Warehouse Source

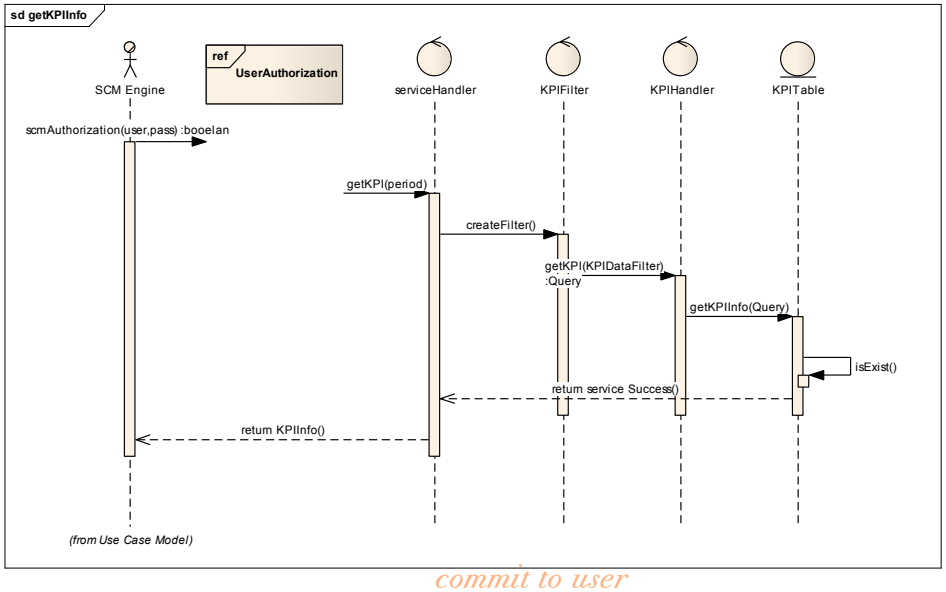
Aktivitas getWarehouseSource memanfaatkan total delapan class yang saling berinteraksi. Delapan class tersebut terdiri dari empat class untuk authorisasi user dan empat class untuk mengambil info mengenai gudang sumber untuk pengambilan stock. Sedangkan untuk methodnya aktivitas ini melibatkan lima method dari satu proses dan empat class. Untuk lebih jelasnya dapat dilihat pada gambar 17 :



Gambar 17 . Sequence Diagram untuk Proses get Warehouse Source

3.3.7. Sequence Get KPI Info

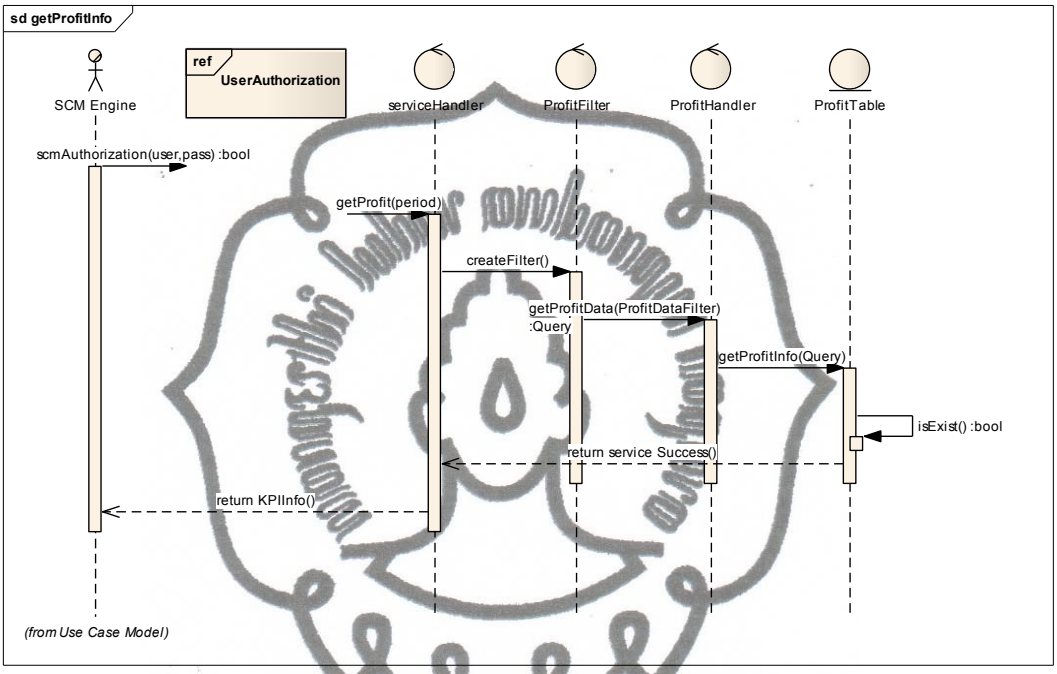
Sequence diagram ini menjelaskan class- class dan method apa saja yang digunakan saat aktivitas GetKPIInfo dilakukan. Saat aktivitas GetKPIInfo dilakukan terdapat delapan class yang saling berinteraksi, yang terdiri dari empat class untuk autorisasi dan empat class untuk mendapat info KPI. Untuk lebih jelasnya silahkan lihat pada gambar 18:



Gambar 18. Sequence Diagram untuk Proses Get KPI Info

3.3.8. Sequence Get Profit Info

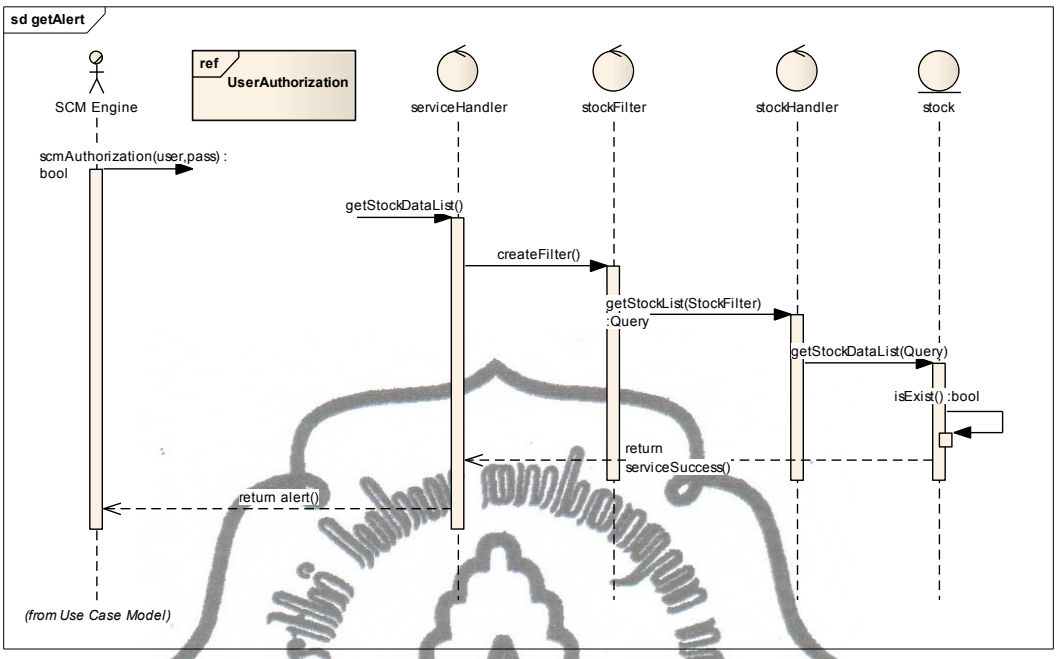
Sequence diagram untuk aktivitas GetProfitInfo memanfaatkan delapan class dan lima method yang saling berhubungan.Class- class dan method- method yang saling berubungan tersebut akan menghasilkan sebuah laporan keuntungan yang dapat dilihat oleh user, salah satunya SCM.



Gambar 19 . Sequence Diagram untuk Proses Get Profit Info

3.3.9. Sequence Get Alert

Aktivitas Get Alert memanfaatkan delapan class yang saling berinteraksi. Empat class untuk authorisasi user dan empat class untuk mengambil mendapat alert. Empat class tersebut adalah class- class khusus yang berhubungan dengan tabel stock pada database. Sedangkan untuk method yang dipanggil, terdapat empat buah method yang digunakan salah satunya adalah method untuk authorisasi user yaitu scmAuthorization().. Untuk lebih jelasnya dapat dilihat pada gambar 20:



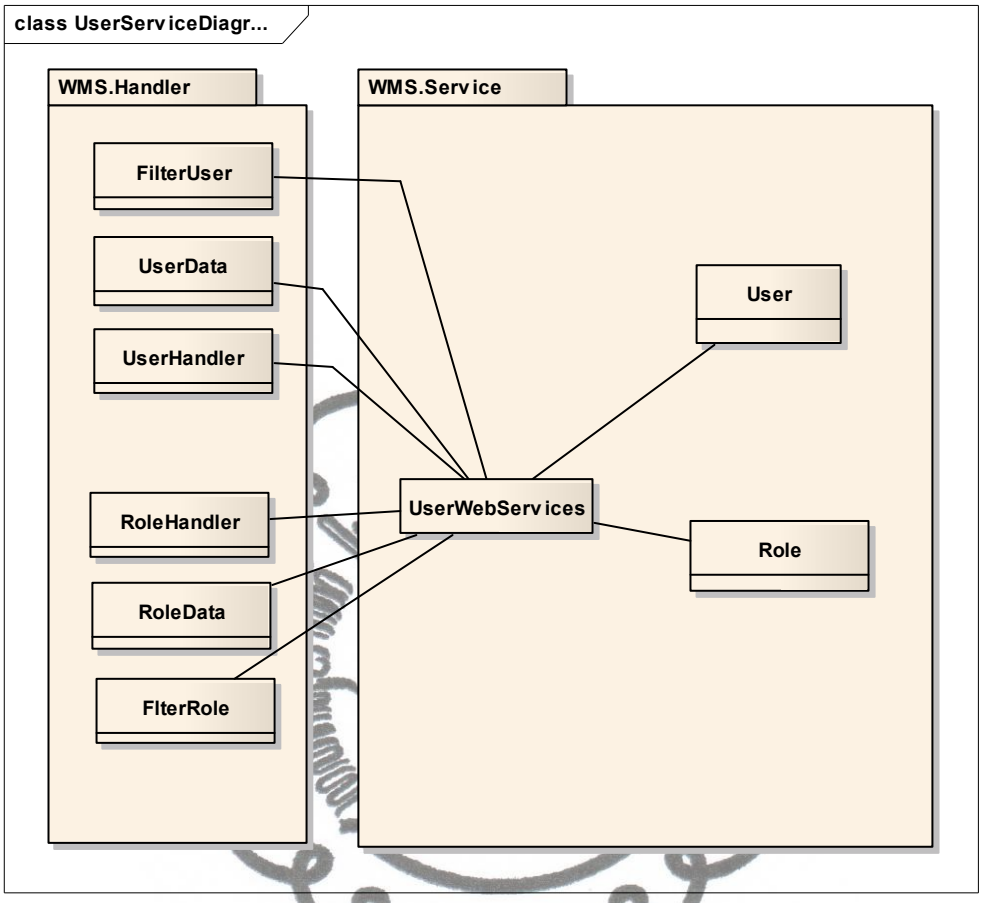
Gambar 20 . Sequence Diagram untuk Proses Get alert

3.4 Class Diagram

Class Diagram WMS Dashboard dibagi menjadi 3 bagian yaitu class diagram untuk User Service, untuk Warehouse Service dan untuk Report Service.

3.4 .1. User Service Class Diagram

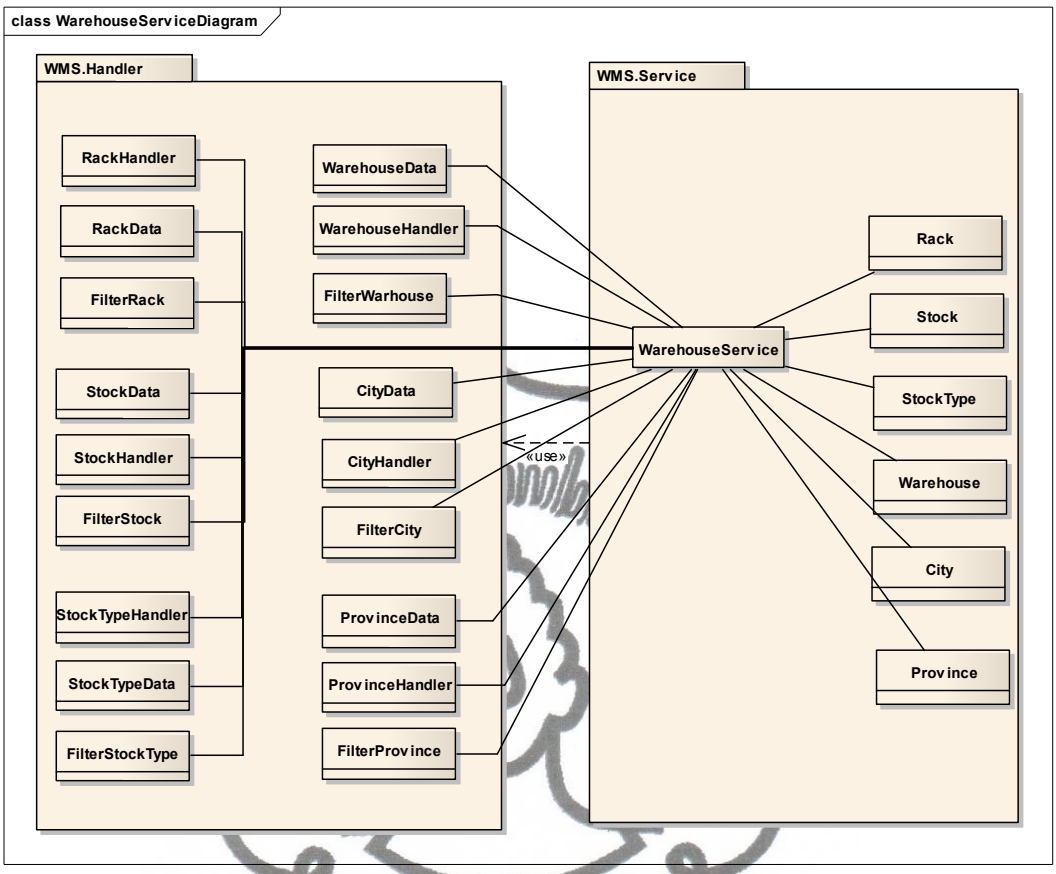
Class diagram untuk user service, menjelaskan mengenai class- class apa saja yang digunakan untuk membuat userService. Pada diagram ini tampak hubungan antar class yang membangun UserService. Class yang menjadi pusat adalah class UserWebService, sedangkan 8 class lainnya adalah class pendukung dari class web service. Class- class tersebut terbagi menjadi 4 bagian, yaitu class handler, class data, class filter dan class data untuk service (class dengan nama sesuai dengan nama tabel pada database ex: class User). Untuk lebih jelasnya dapat dilihat pada gambar 21 berikut ini:



Gambar 21 . Class Diagram untuk User Service

3.4.2. Warehouse Service Class Diagram

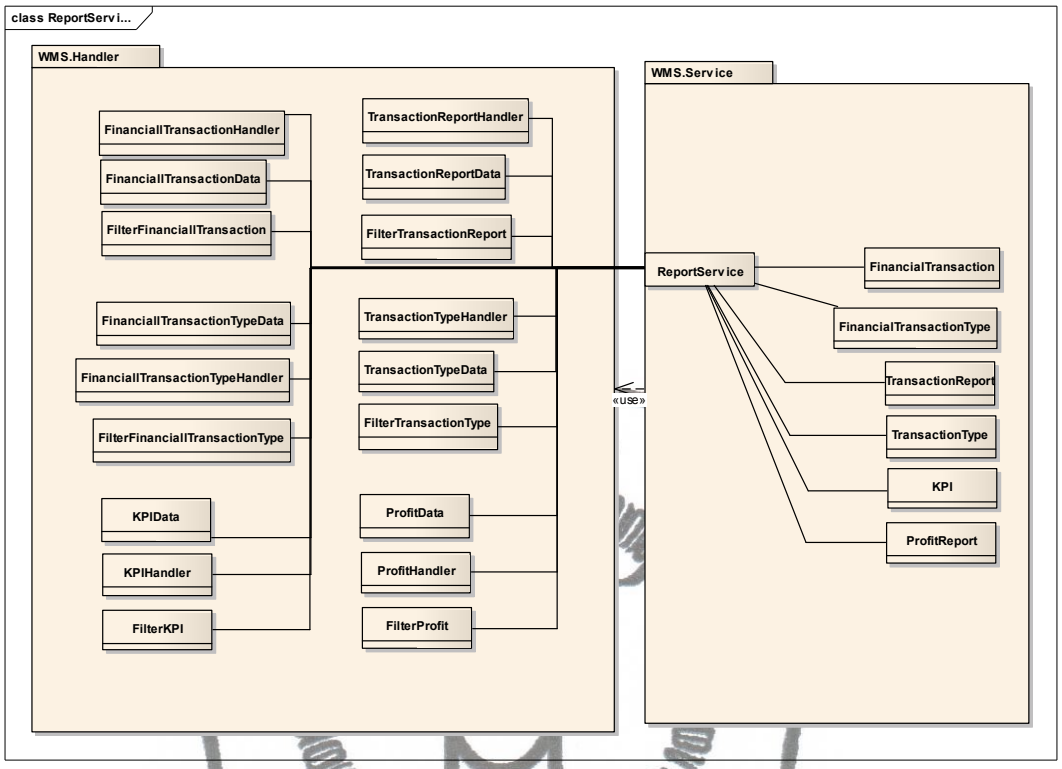
Warehouse service class diagram, adalah class diagram yang menjelaskan class- class yang saling berhubungan untuk membangun Warehouse Service. Warehouse Service adalah service yang mengurusi hal- hal mengenai masalah gudang, antara lain stock dan space. Dalam diagram class Warehouse Service tampak 25 class yang saling berhubungan, dengan class Warehouse Service sebagai *core*/ pusat dari diagram ini. Untuk lebih jelasnya dapat dilihat pada gambar 22 berikut ini:



Gambar 22 . Class Diagram untuk Warehouse Service

3.4 .3.ReportServiceClassDiagram

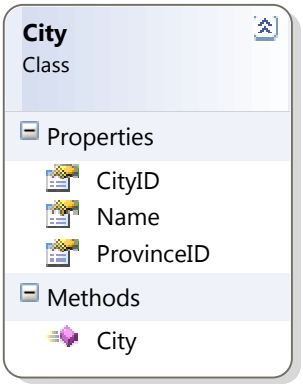
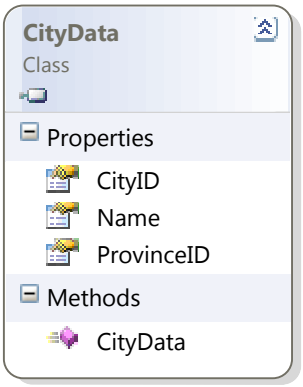
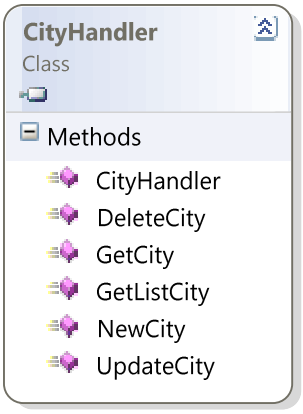
Class diagram untuk report Service menggambarkan 25 class yang saling berhubunan untuk membentuk service untuk membuat report. Class yang saling berhubungan terbagi menjadi enam kategori, yaitu class untuk Financial Transaction, class untuk Financial Transaction Type, class untuk Transaction, class untuk Transaction Type, class untuk Profit Report dan class untuk KPI. Setiap class tersebuut dibagi menjadi 4 bagian class pembangun yaitu class handler, class data, class filter dan class untuk kontainer data dari tabel database. Untuk lebih jelasnya dapat dilihat pada gambar 23 berikut ini:

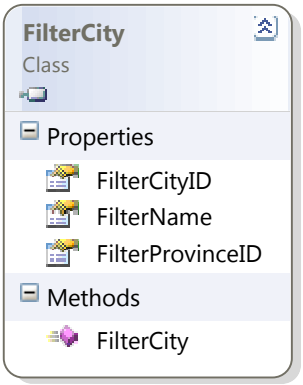
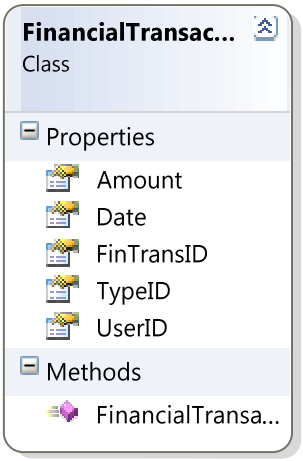
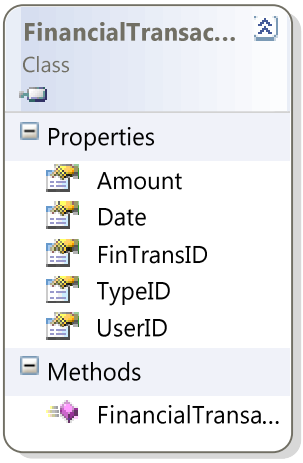


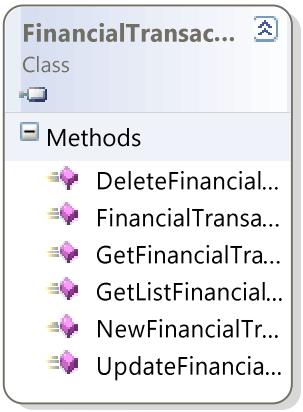
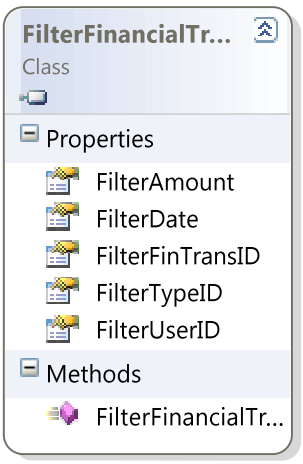
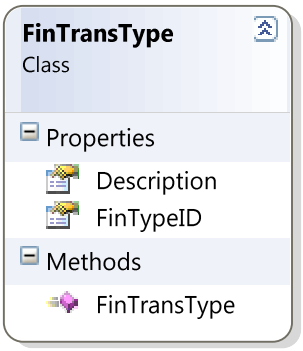
Gambar 23 . Class Diagram untuk Report Service

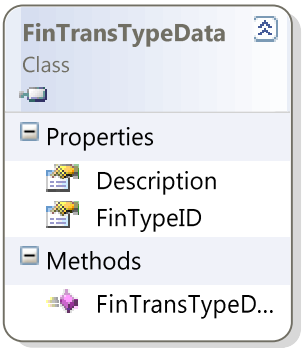
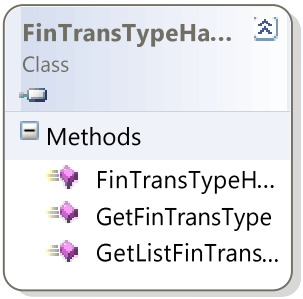
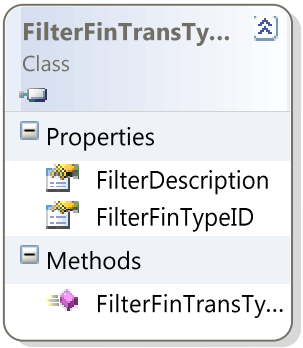
Berikut ini adalah tabel yang berisi deskripsi dari class- class yang ada pada aplikasi WMS Dashboard dan telah digambarkan pada class diagram sebelumnya.

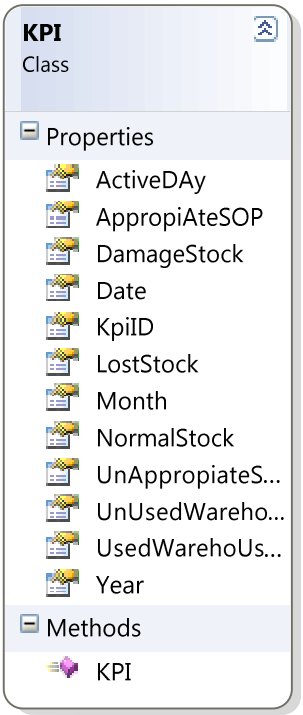
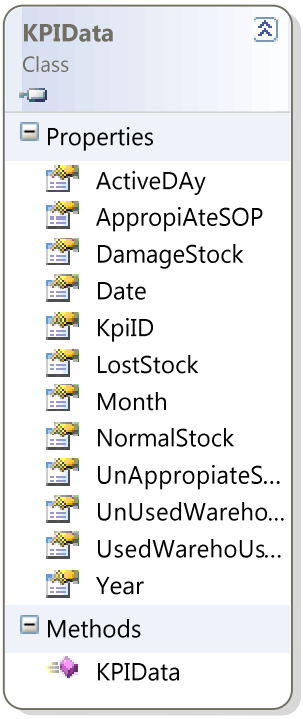
Tabel 11. Tabel Deskripsi Class

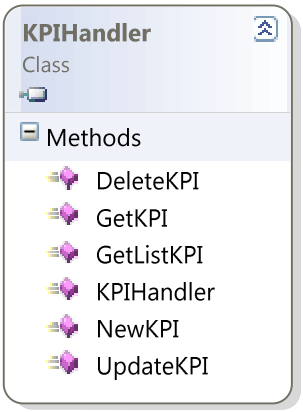
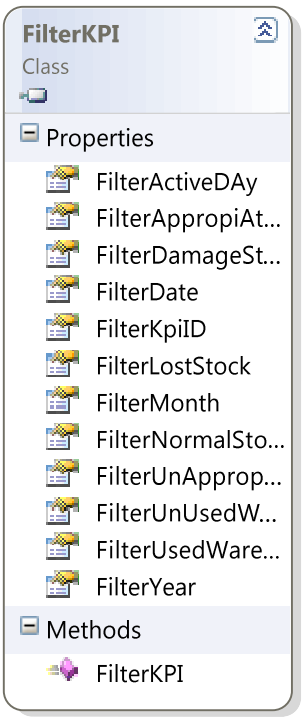
KODE CLASS	NAMA CLASS	METHOD & PROPERTIES	KETERANGAN
CD-WD1	City		Class City, digunakan sebagai class untuk membuat objek city, yang digunakan sebagai container data yang diambil dari class data, dan digunakan dalam Service
CD-WD2	CityData		Class CityData, digunakan sebagai class untuk membuat objek city, yang akan digunakan oleh class handler sebagai container dari data yang diambil dari database
CD-WD3	City Handler		Class CityHandler, berisi method-method yang dapat digunakan untuk manipulasi data pada tabel City

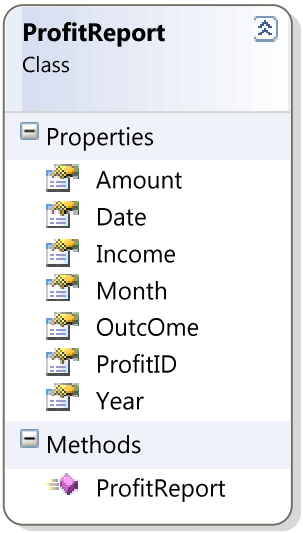
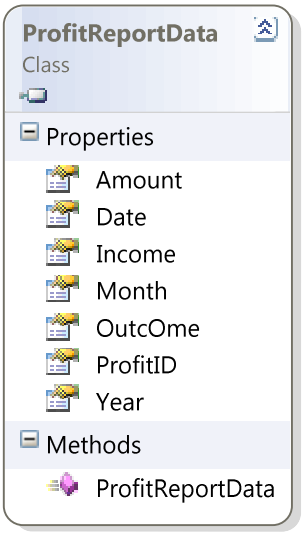
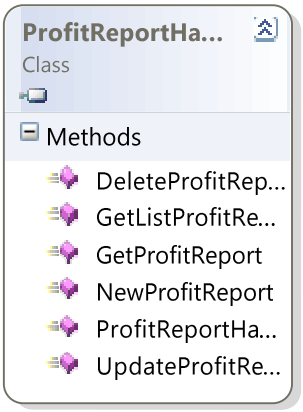
CD-WD4	FilterCity		Class FilterCity, digunakan oleh class handler untuk memfilter data- data yang akan diambil dari tabel City
CD-WD5	Financial Transaction		Class FinancialTransaction, digunakan sebagai class untuk membuat objek FinancialTransaction, yang digunakan sebagai container data yang diambil dari class data, dan digunakan dalam Service
CD-WD6	Financial Transaction Data		Class FinancialTransactionData, digunakan sebagai class untuk membuat objek FinancialTransactionData, yang akan digunakan oleh class handler sebagai container dari data yang diambil dari database

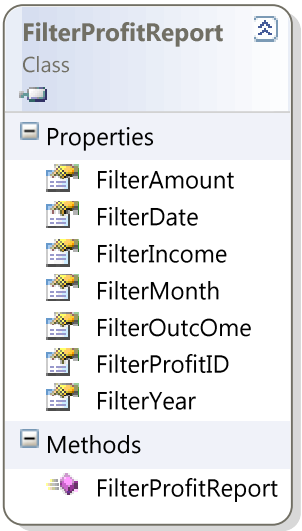
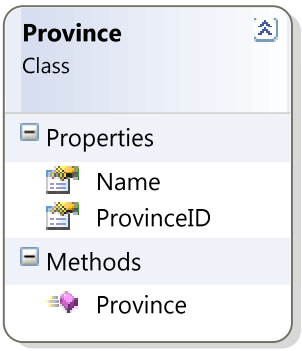
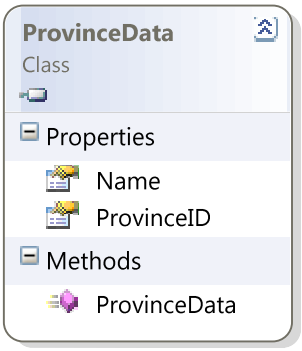
CD-WD7	Financial Transaction Handler		Class FinancialTransactionHandler, berisi method-method yang dapat digunakan untuk manipulasi data pada tabel FinancialTransaction
CD-WD8	Filter Financial Transaction		Class FilterFinancialTransaction, digunakan oleh class handler untuk memfilter data- data yang akan diambil dari tabel FinancialTransaction
CD-WD9	FinTrans Type		Class FinTransType, digunakan sebagai class untuk membuat objek FinTransType, yang digunakan sebagai container data yang diambil dari class data, dan digunakan dalam Service

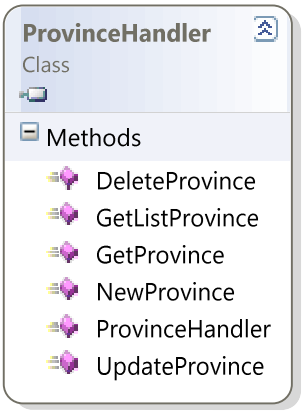
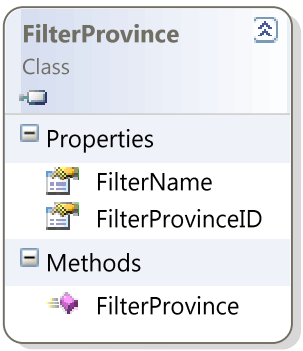
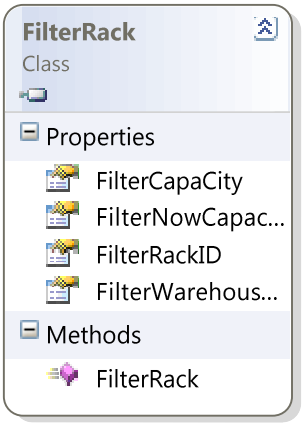
CD-WD10	FinTrans TypeData		Class FinTransTypeData, digunakan sebagai class untuk membuat objek FinTransTypeData, yang akan digunakan oleh class handler sebagai container dari data yang diambil dari database
CD-WD11	FinTrans Type Handler		Class FinTransTypeHandler, berisi method-method yang dapat digunakan untuk manipulasi data pada tabel FinTransType
CD-WD12	Filter FinTrans Type		Class FilterFinTransType, digunakan oleh class handler untuk memfilter data- data yang akan diambil dari tabel FinTransType

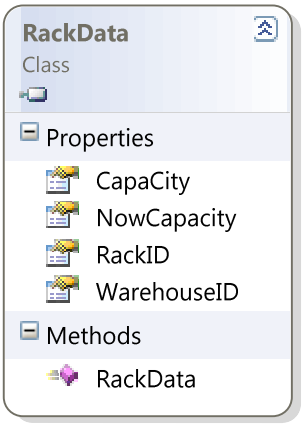
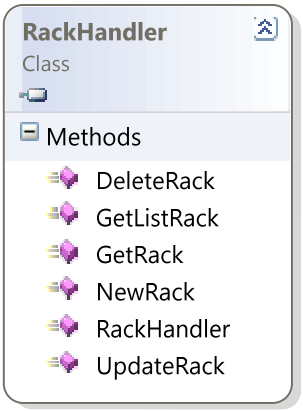
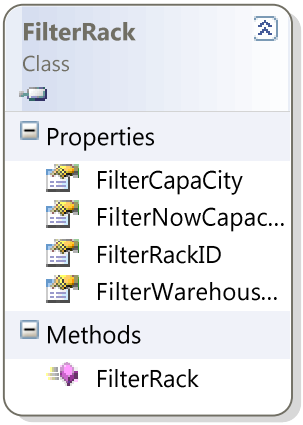
CD-WD13	KPI		Class KPI, digunakan sebagai class untuk membuat objek KPI, yang digunakan sebagai container data yang diambil dari class data, dan digunakan dalam Service
CD-WD14	KPIData		Class KPIData, digunakan sebagai class untuk membuat objek KPIData, yang akan digunakan oleh class handler sebagai container dari data yang diambil dari database

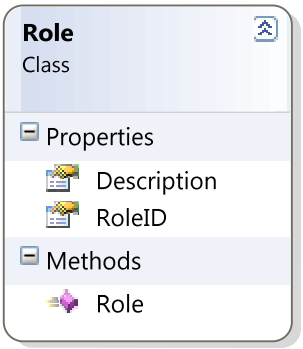
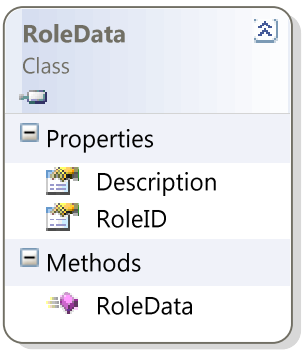
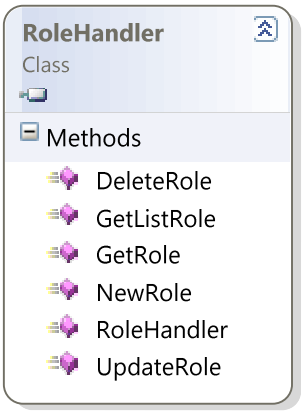
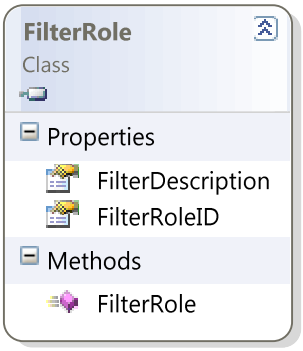
CD-WD15	KPIHandler		Class KPIHandler, berisi method-method yang dapat digunakan untuk manipulasi data pada tabel KPI
CD-WD16	FilterKPI		Class FilterKPI, digunakan oleh class handler untuk memfilter data- data yang akan diambil dari tabel FilterKPI

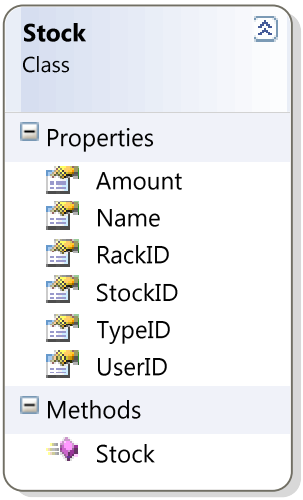
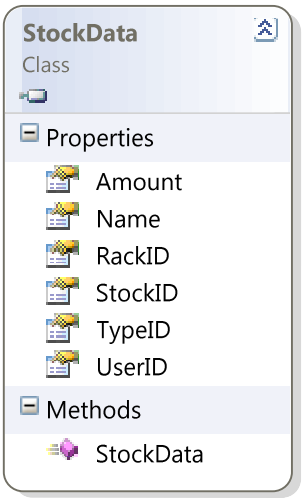
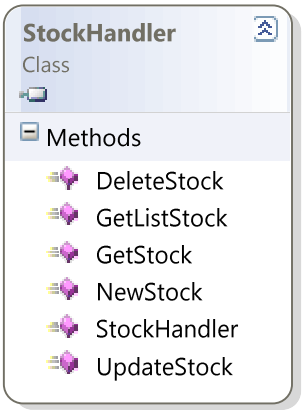
CD-WD17	ProfitReport		Class ProfitReport, digunakan sebagai class untuk membuat objek ProfitReport, yang digunakan sebagai container data yang diambil dari class data, dan digunakan dalam Service
CD-WD18	ProfitReportData		Class ProfitReportData, digunakan sebagai class untuk membuat objek ProfitReportData, yang akan digunakan oleh class handler sebagai container dari data yang diambil dari database
CD-WD19	ProfitReportHandler		Class ProfitReportHandler, berisi method-method yang dapat digunakan untuk manipulasi data pada tabel ProfitReport

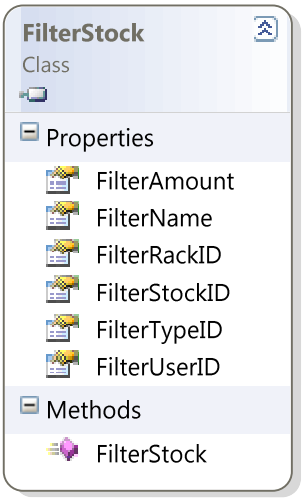
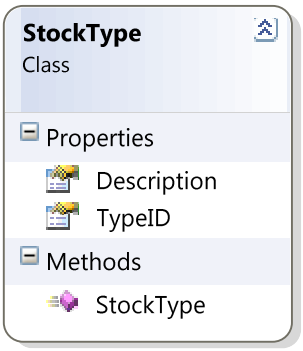
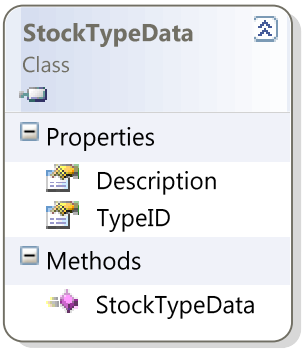
CD-WD20	Filter ProfitReport		Class Filter ProfitReport, digunakan oleh class handler untuk memfilter data- data yang akan diambil dari tabel ProfitReport
CD-WD21	Province		Class Province, digunakan sebagai class untuk membuat objek Province, yang digunakan sebagai container data yang diambil dari class data, dan digunakan dalam Service
CD-WD22	Province Data		Class ProvinceData, digunakan sebagai class untuk membuat objek ProvinceData, yang akan digunakan oleh class handler sebagai container dari data yang diambil dari database

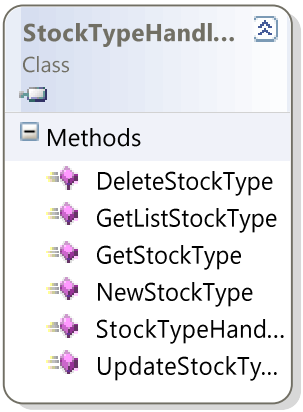
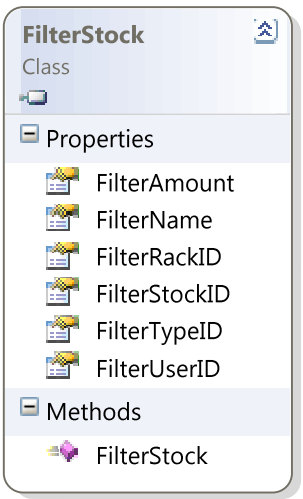
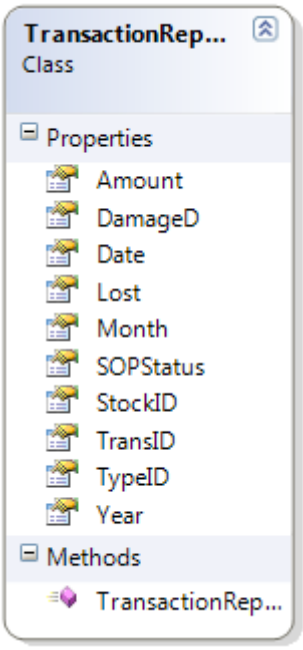
CD-WD23	Province Handler		Class ProvinceHandler, berisi method-method yang dapat digunakan untuk manipulasi data pada tabel Province
CD-WD24	Filter Province		Class FilterProvince, digunakan oleh class handler untuk memfilter data- data yang akan diambil dari tabel Province
CD-WD25	Rack		Class Rack, digunakan sebagai class untuk membuat objek Rack, yang digunakan sebagai container data yang diambil dari class data, dan digunakan dalam Service

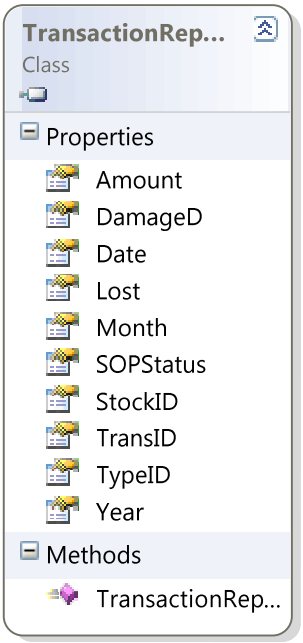
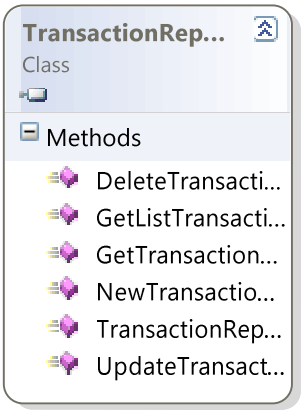
CD-WD26	RackData		Class RackData, digunakan sebagai class untuk membuat objek RackData, yang akan digunakan oleh class handler sebagai container dari data yang diambil dari database
CD-WD27	RackHandler		Class RackHandler, berisi method-method yang dapat digunakan untuk manipulasi data pada tabel Rack
CD-WD28	FilterRack		Class FilterRack, digunakan oleh class handler untuk memfilter data- data yang akan diambil dari tabel Rack

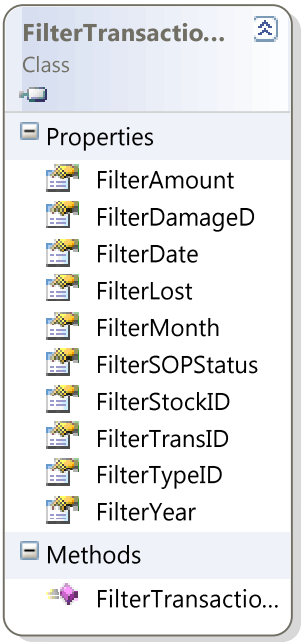
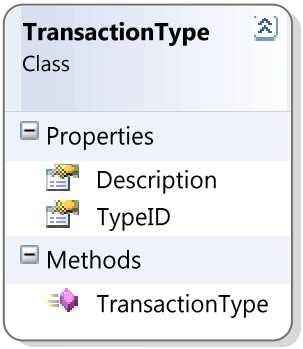
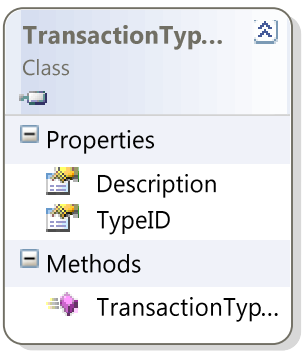
CD-WD29	Role		Class Role, digunakan sebagai class untuk membuat objek Role, yang digunakan sebagai container data yang diambil dari class data, dan digunakan dalam Service
CD-WD30	RoleData		Class RoleData, digunakan sebagai class untuk membuat objek RoleData, yang akan digunakan oleh class handler sebagai container dari data yang diambil dari database
CD-WD31	RoleHandler		Class RoleHandler, berisi method-method yang dapat digunakan untuk manipulasi data pada tabel Role
CD-WD32	FilterRole		Class FilterRole, digunakan oleh class handler untuk memfilter data- data yang akan diambil dari tabel Role

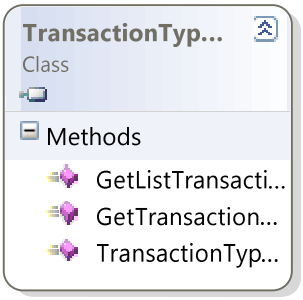
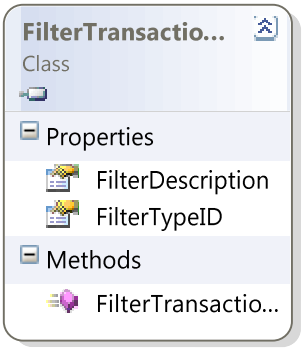
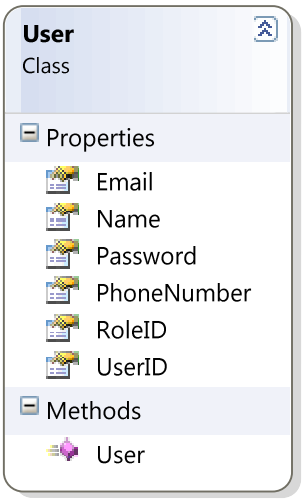
CD-WD33	Stock		Class Stock, digunakan sebagai class untuk membuat objek Stock, yang digunakan sebagai container data yang diambil dari class data, dan digunakan dalam Service
CD-WD34	StockData		Class StockData, digunakan sebagai class untuk membuat objek StockData, yang akan digunakan oleh class handler sebagai container dari data yang diambil dari database
CD-WD35	StockHandler		Class StockHandler, berisi method-method yang dapat digunakan untuk manipulasi data pada tabel Stock

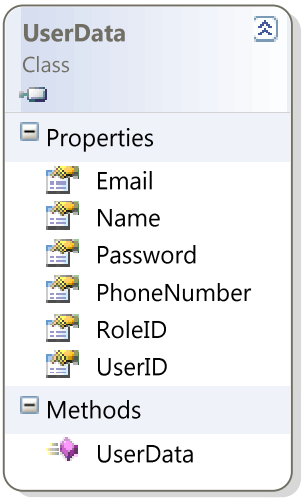
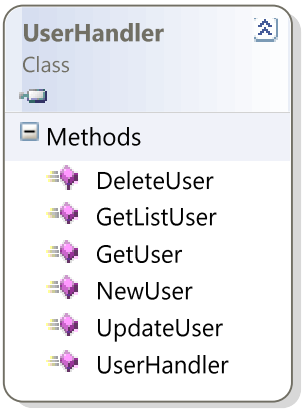
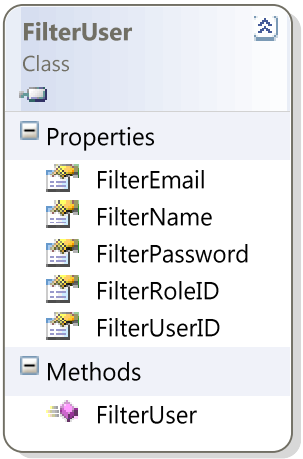
CD-WD36	FilterStock		Class FilterStock, digunakan oleh class handler untuk memfilter data- data yang akan diambil dari tabel Stock
CD-WD37	StockType		Class StockType, digunakan sebagai class untuk membuat objek StockType, yang digunakan sebagai container data yang diambil dari class data, dan digunakan dalam Service
CD-WD38	StockTypeData		Class StockTypeData, digunakan sebagai class untuk membuat objek StockTypeData, yang akan digunakan oleh class handler sebagai container dari data yang diambil dari database

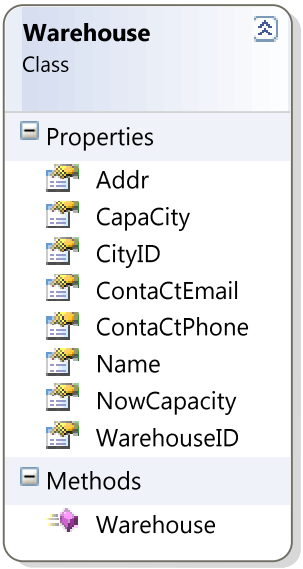
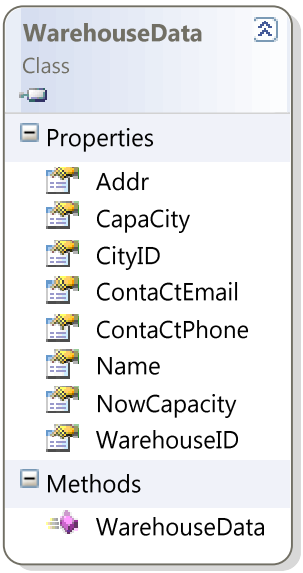
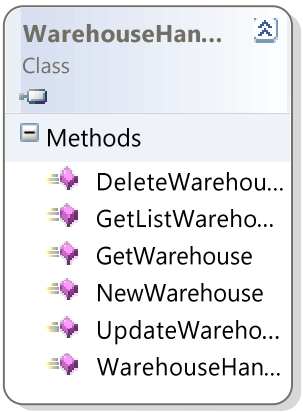
CD-WD39	StockTypeHandler		Class StockTypeHandler, berisi method-method yang dapat digunakan untuk manipulasi data pada tabel StockType
CD-WD40	FilterStockType		Class FilterStockType, digunakan oleh class handler untuk memfilter data- data yang akan diambil dari tabel StockType
CD-WD41	TransactionReport		Class TransactionReport, digunakan sebagai class untuk membuat objek TransactionReport, yang digunakan sebagai container data yang diambil dari class data, dan digunakan dalam Service

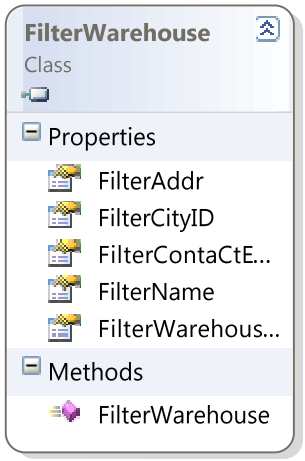
CD-WD42	TransactionReportData		Class TransactionReportData, digunakan sebagai class untuk membuat objek TransactionReportData, yang akan digunakan oleh class handler sebagai container dari data yang diambil dari database
CD-WD43	TransactionReportHandler		Class TransactionReportHandler, berisi method-method yang dapat digunakan untuk manipulasi data pada tabel TransactionReport

CD-WD44	Filter TransactionReport		Class FilterTransactionReport, digunakan oleh class handler untuk memfilter data- data yang akan diambil dari tabel TransactionReport
CD-WD45	TransactionType		Class TransactionType, digunakan sebagai class untuk membuat objek TransactionType, yang digunakan sebagai container data yang diambil dari class data, dan digunakan dalam Service
CD-WD46	TransactionTypeData		Class TransactionTypeData, digunakan sebagai class untuk membuat objek TransactionTypeData, yang akan digunakan oleh class handler sebagai container dari data yang diambil dari database

CD-WD47	TransactionTypeHandler		Class TransactionTypeHandler, berisi method-method yang dapat digunakan untuk manipulasi data pada tabel TransactionType
CD-WD48	FilterTransactionType		Class FilterTransactionType, digunakan oleh class handler untuk memfilter data- data yang akan diambil dari tabel TransactionType
CD-WD49	User		Class User, digunakan sebagai class untuk membuat objek User, yang digunakan sebagai container data yang diambil dari class data, dan digunakan dalam Service

CD-WD50	UserData		Class UserData, digunakan sebagai class untuk membuat objek UserData, yang akan digunakan oleh class handler sebagai container dari data yang diambil dari database
CD-WD51	UserHandler		Class UserHandler, berisi method-method yang dapat digunakan untuk manipulasi data pada tabel User
CD-WD52	FilterUser		Class FilterUser, digunakan oleh class handler untuk memfilter data- data yang akan diambil dari tabel User

CD-WD53	Warehouse		Class Warehouse, digunakan sebagai class untuk membuat objek Warehouse, yang digunakan sebagai container data yang diambil dari class data, dan digunakan dalam Service
CD-WD54	WarehouseData		Class WarehouseData, digunakan sebagai class untuk membuat objek WarehouseData, yang akan digunakan oleh class handler sebagai container dari data yang diambil dari database
CD-WD55	WarehouseHandler		Class WarehouseHandler, berisi method-method yang dapat digunakan untuk manipulasi data pada tabel Warehouse

CD-WD56	FilterWarehouse		Class FilterWarehouse, digunakan oleh class handler untuk memfilter data- data yang akan diambil dari tabel Warehouse

Class- class yang telah dijelaskan pada tabel diatas digunakan pada aktivitas-aktivitas yang ada pada use case. Dibawa ini adalah tabel kesesuaian antara use case dan class- class yang digunakannya:

Tabel 12. Tabel Kesesuaian Class Diagram dan UC Fungsional

Class Diagram	Use Case	Functional Needs
CD-WD34, CD-WD35,	UC-WDF1	Set stocks specification (Fill the stock)
CD-WD34, CD-WD35,	UC-WDF2	Take stocks from warehouse
CD-WD33, CD-WD34, CD-WD35, CD-WD36	UC-WDF3	Get stocks information
CD-WD33, CD-WD34, CD-WD35, CD-WD36	UC-WDF4	Get stock alert
CD-WD33, CD-WD34, CD-WD35, CD-WD36, CD-WD25, CD-WD26, CD-WD27, CD-WD28	UC-WDF5	Get the source warehouse location
CD-WD41, CD-WD42, CD-WD43, CD-WD44	UC-WDF6	Get All Transactional Data
CD-WD13, CD-WD14, CD-WD15, CD-WD16	UC-WDF7	Get KPI Information
CD-WD17, CD-WD18, CD-WD19, CD-WD20	UC-WDF8	Get Profit Information
CD-WD13, CD-WD14, CD-WD15, CD-WD16	UC-WDF9	Set effectiveness of warehouse space percentage
CD-WD13, CD-WD14, CD-WD15, CD-WD16	UC-WDF10	Set lost stock percentage
CD-WD13, CD-WD14, CD-WD15, CD-WD16	UC-WDF11	Set damaged stock percentage
CD-WD13, CD-WD14, CD-WD15, CD-WD16	UC-WDF12	Set Quality Of Service percentage
CD-WD13, CD-WD14, CD-WD15, CD-WD16	UC-WDF13	Set the warehouse activity that appropriate with SOP percentage
CD-WD14, CD-WD15,	UC-WDF14	Calculate used space in warehouse
CD-WD14, CD-WD15,	UC-WDF15	Calculate unused space in warehouse
CD-WD14, CD-WD15,	UC-WDF16	Calculate lost stock
CD-WD14, CD-WD15,	UC-WDF17	Calculate damaged stock
CD-WD14, CD-WD15,	UC-WDF18	Calculate Stock Amount
CD-WD14, CD-WD15,	UC-WDF19	Calculate warehouse activity that appropriate with SOP
CD-WD14, CD-WD15,	UC-WDF20	Calculate warehouse activity that un-appropriate with SOP
CD-WD13, CD-WD14, CD-WD15, CD-WD16	UC-WDF21	Calculate total of warehouse activity
CD-WD17, CD-WD18, CD-WD19, CD-WD20	UC-WDF22	Calculate warehouse total income
CD-WD17, CD-WD18, CD-WD19, CD-WD20	UC-WDF23	Calculate Warehouse total Outcome
CD-WD34,CD-WD35	UC-WDF24	Organize stock information data
CD-WD18, CD-WD19,	UC-WDF25	Set Profit Information

Tabel 13. Tabel Kesesuaian Class Diagram dan UC Non-Fungsional

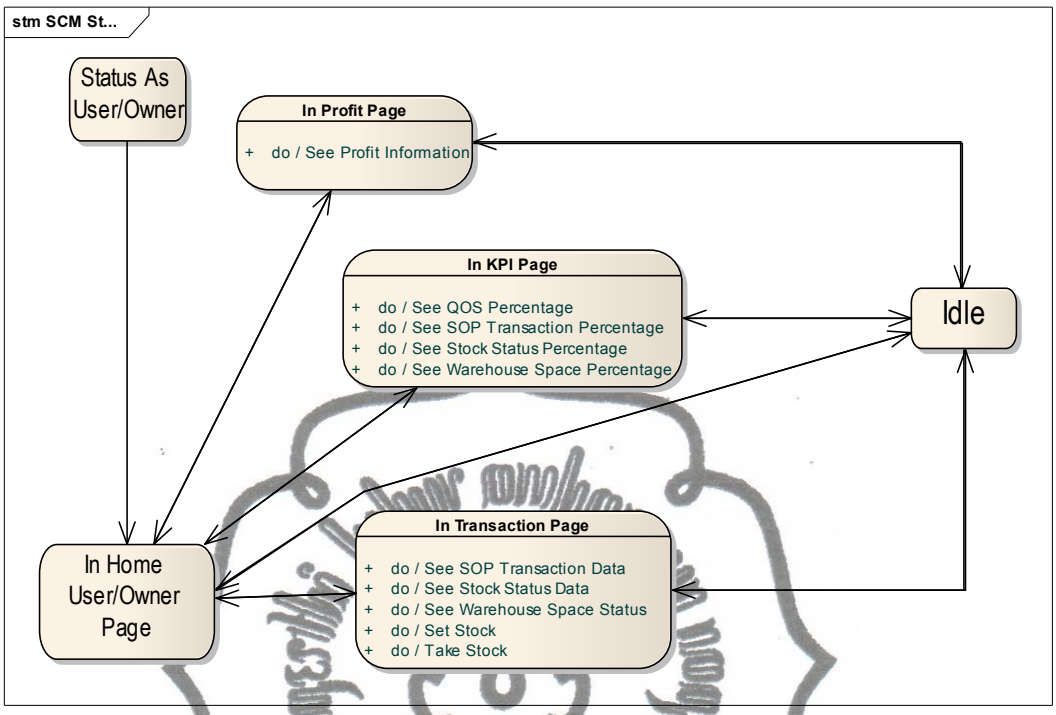
Class Diagram	Use Case	Non- Functional Needs
-	UC-WDNF1	Get User Friendly Interface
CD-WD49, CD-WD50, CD-WD51, CD-WD52	UC-WDNF2	Set Login/Logout
CD-WD50, CD-WD51	UC-WDNF3	Set Password
CD-WD50, CD-WD51	UC-WDNF4	Set Login Email
CD-WD41, CD-WD42, CD-WD43, CD-WD44	UC-WDNF5	Print Transactional Data
CD-WD13, CD-WD14, CD-WD15, CD-WD16	UC-WDNF6	Print KPI Information
CD-WD17, CD-WD18, CD-WD19, CD-WD20	UC-WDNF8	Print Profit Information
CD-WD49, CD-WD50, CD-WD51, CD-WD52	UC-WDNF9	Authoring Processing
-	UC-WDNF10	Get WMS Dashbaord Info

3.5 State Diagram

State diagram untuk WMS Dashboard dibagi menjadi 3 bagian menurut usernya, yaitu untuk Owner, untuk General User dan untuk SCM.

3.5.1.Owner/User State Diagram

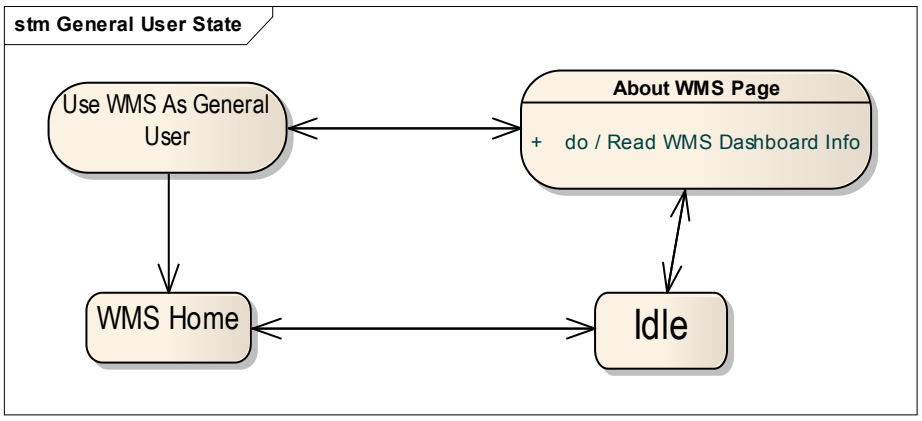
Pada state diagram untuk user/owner dijelaskan mengenani kondisi- kondisi yang dapat dialami oleh user/owner selama menggunakan aplikasi WMS Dashboard. Pada gambar 24 terdapat enam buah kondisi yang dapat dialami oleh user/ owner. Lima kondisi akan akan dapat dialami oleh user/ owner jika user/ owner telah mengalami kondisi pertama yaitu kondisi ‘Login as User/ Owner’. Jika user/ owner telah melakukan login tersebut maka User/ Owner akan dapat mengalami lima kondisi lainnya, yaitu kondisi saat berada pada halaman home, profit, KPI, transaction dan daam kondisi Idle(diam). Gambar 24 akan menjelaskan state diagram untuk user/owner:



Gambar 24 . State Diagram Untuk Owner/User

3.5.2. General User State Diagram

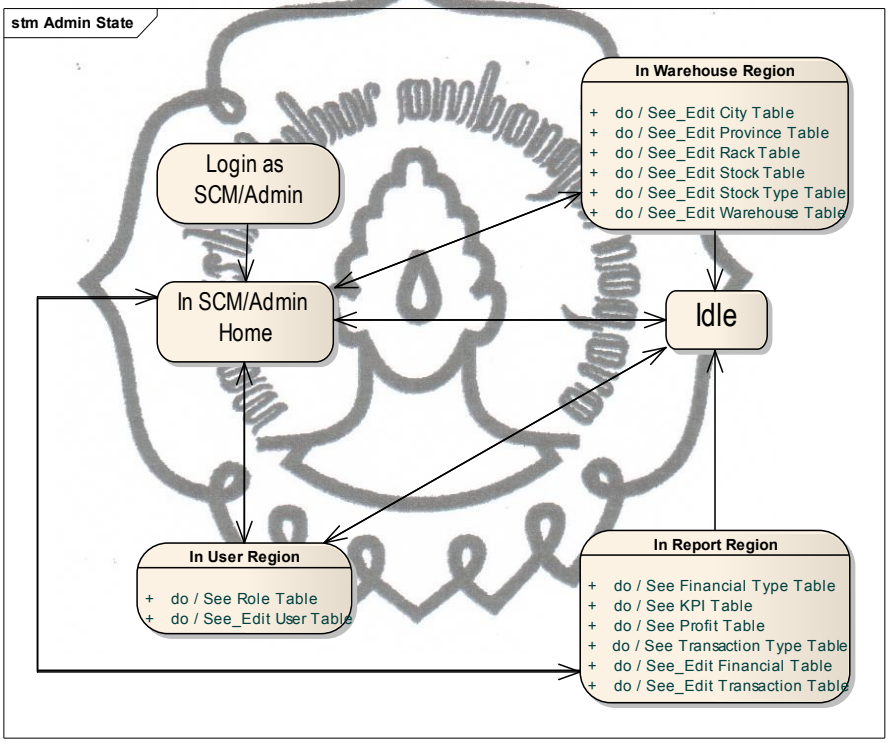
General user sebagai tingkatan user paling rendah, hanya memiliki empat kondisi saat ia menggunakan aplikasi WMS Dashboard. General user hanya dapat memasuki WMS Home untuk selanjutnya melakukan login, membaca info WMS atau diam/ idle. Untuk lebih jelasnya tampak pada gambar 25 dibawah ini :



Gambar 25 . State Diagram Untuk General User

3.5.3. SCM/Admin State Diagram

SCM/ admin memiliki state diagram, dengan kondisi yang memiliki banyak fiturnya. Jumlah kondisi untuk SCM/ admin saat menggunakan aplikasi WMS Dashboard ada enam buah kondisi, dengan satu kondisi awal yang harus dialami sebelum mengalami lima kondisi lainnya, yaitu kondisi 'login as SCM/ Admin'. Lima kondisi lainnya adalah kondisi dimana SCM/ admin dapat melakukan editing data terhadap tabel- tabel yang ada pada database WMS Dashboard.

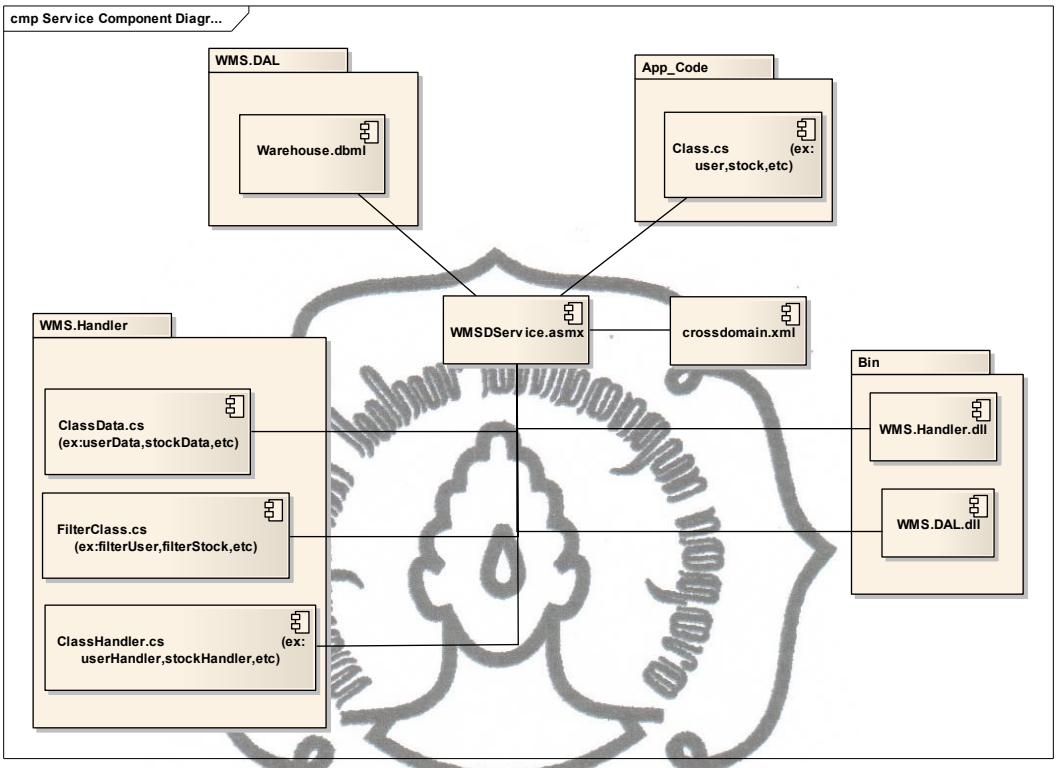


Gambar 26 . State Diagram Untuk SCM/Admin

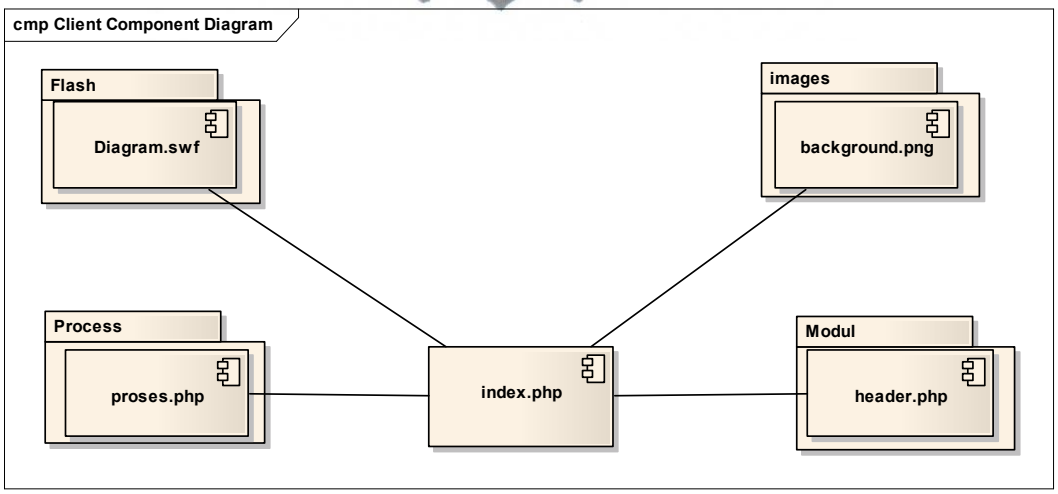
3.6 Component Diagram

Component diagram dari aplikasi WMS Dashboard dibagi menjadi 2, yaitu component diagram untuk Service dan component diagram untuk bagian client (aplikasi WMS Dashboard sendiri). Pada component diagram untuk bagian Service terdapat empat buah folder pendukung, yang akan mendukung core dari aplikasi WMS Dashboard, yaitu file Service.asmx. Sedangkan untuk component diagram pada bagian client terdapat 4 buah folder pendukung dengan sebuah file core yaitu index.php. Untuk component diagram pada bagian Service, didominasi dengan file- file dari . Net Framework, sedangakan untuk component diagram

untuk bagian client didominasi dengan file- file PHP. Berikut ini adalah kedua komponen tersebut:



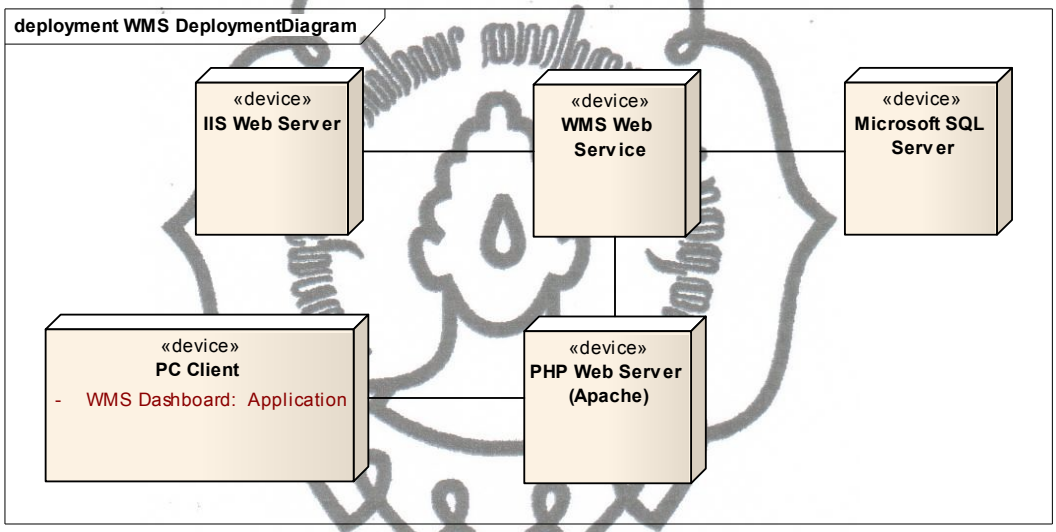
Gambar 27 . Component Diagram Untuk Bagian Service



Gambar 28. Component Diagram Untuk Bagian Client

3.7 Deployment Diagram

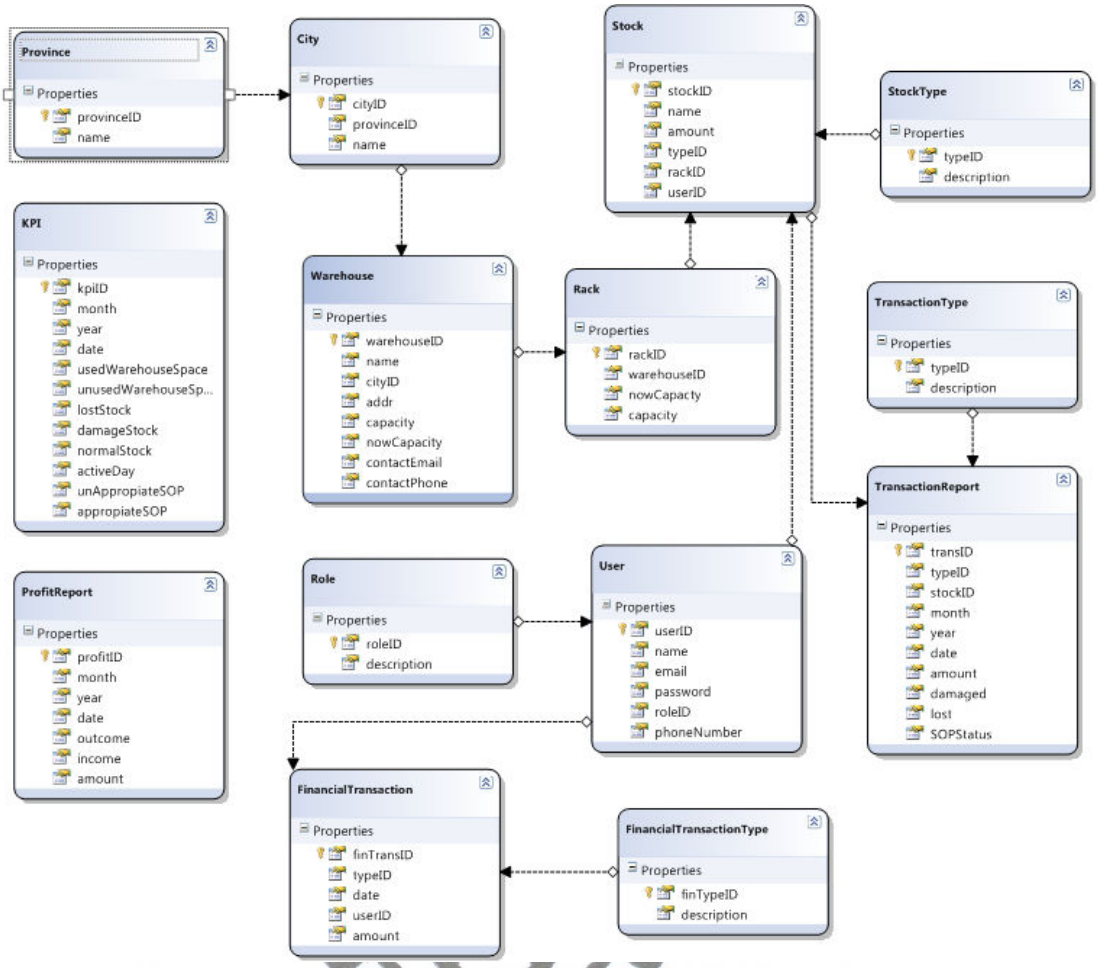
Pada deployment diagram dibawah ini, dijelaskan mengenai hubungan antar hardware- hardware yang digunakan dalam aplikasi WMS Dashboard. Terdapat lima group hardware yang saling terkoneksi, yang harus tersedia untuk berjalannya aplikasi WMS Dashboard lengkap dengan servicenya. Lima hardware tersebut adalah server dengan IIS, server dengan PHP web server, server database dengan MS.SQL, server untuk service dan pc untuk client. Gambar 29 berkut ini, akan menjelaskan lebih detail:



Gambar 29. Deployment Diagram Untuk WMS Dashboard

3.8 Skema Diagram

Skema diagram adalah diagram dari database milik WMS Service yang didalamnya dijelaskan mengenai hubungan antar tabel- tabel yang ada dalam database WMS Service. Dalam service WMS Dashboard terdapat 14 tabel, dimana 12 tabel saling terkoneksi satu sama lain, dan dua tabel lainnya merupakan tabel trigger yang akan menampung data- data dari 12 tabel lainnya. Dibawah ini adalah gambar 30 yang menunjukka skema diagram dari WMS Service :



Gambar 30. Skema Diagram Untuk WMS Dashboard

BAB IV
IMPLEMENTASI DAN ANALISA

4.1 Implementasi Aplikasi WMS *Dashboard* & Service

Implementasi aplikasi WMS *Dashboard* dan Service menyangkut kebutuhan hardware dan software untuk menggunakan aplikasi WMS *Dashboard* dan Service, Skema layer aplikasi, gambar implementasi pada jaringan, list WSDL yang tersdia pada WMS Service serta daftar menu utama yang ada di aplikasi WMS Dashboard.

4.1.1 Kebutuhan hardware

Aplikasi WMS *Dashboard* dan Service menggunakan reomendaasi harware sebagai berikut :

1. Server Mandiri
    a. Web Server
        i. Processor Intel Dual Core 2,2 GHz
        ii. RAM 2 GHz
        iii. Harddisk 20 GB SCSI
    b. Database Server
        i. Microsoft Server
        ii. RAM 8 GHz
        iii. Harddisk 240 Gb SATA
2. Server Hosting
3. Client
    a. PC
        i. Processor Pentium Dual Core
        ii. RAM 1 GB
        iii. Harddisk 120 GB
        iv. Koneksi internet min 56kbps

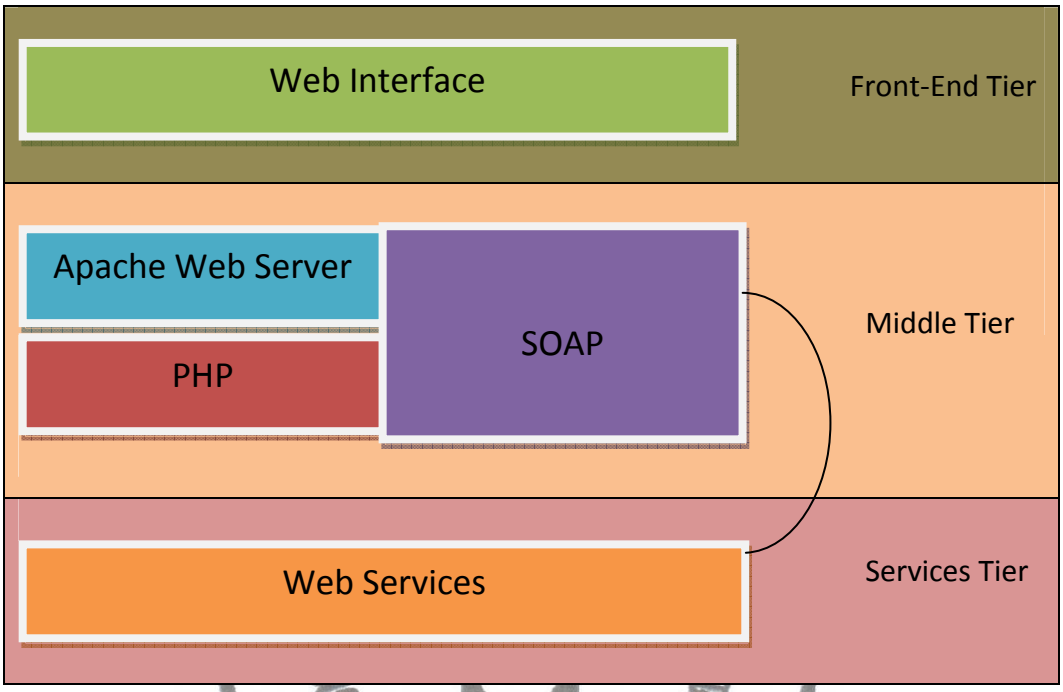
4.1.2 Kebutuhan Software

Aplikasi WMS Dashbaord dan Server menggunakan spesifikasi rekomendasi software sebagai berikut :

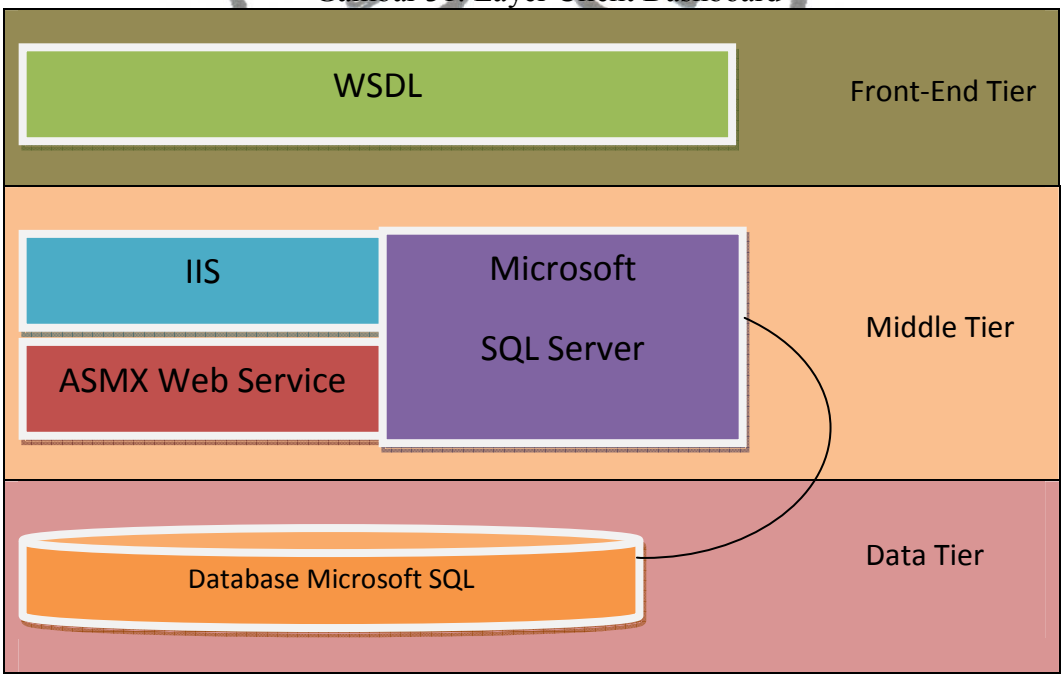
1. Server
    a. Web Server
        i. OS Windows Server 2008
        ii. IIS Web Server
    b. Database Server
        i. Windows OS
        ii. Microsoft SQL Server
2. Client
    a. PC
        i. OS Windows XP SP 2 Professional
        ii. Web Browser IE versi 7.0 atau Firefox versi 3
        iii. Adobe Flash Player 10

4.1.3 Layer Aplikasi SCM

Layer aplikasi adalah stuserktur lapisan teknologi yang digunakan untuk membentuk aplikasi. Layer aplikasi *Dashboard* WMS terdiri dari 2 diagram layer yaitu diagram untuk server dan diagam untuk client, dimana setiap diagram terdiri dari 3 buah lapisan. Arsitektur layer aplikasi *Dashboard* WMS digambarkan sebagai berikut :



Gambar 31. Layer Client Dashboard

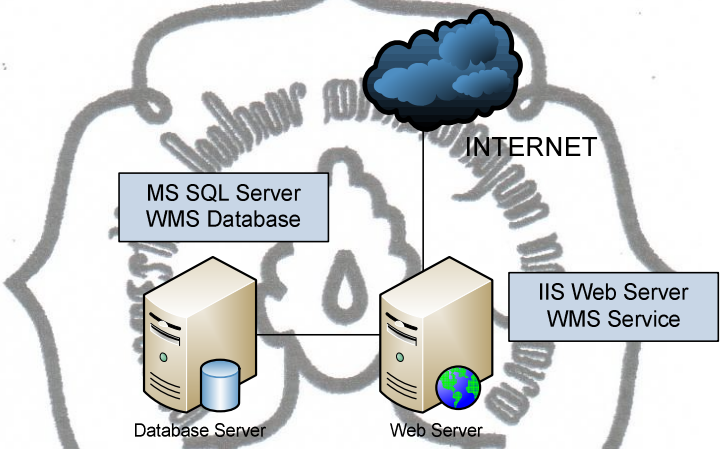


Gambar 32. Layer Server Dashboard

4.1.4 Arsitektur Aplikasi WMS *Dashboard* pada Jaringan Komputer

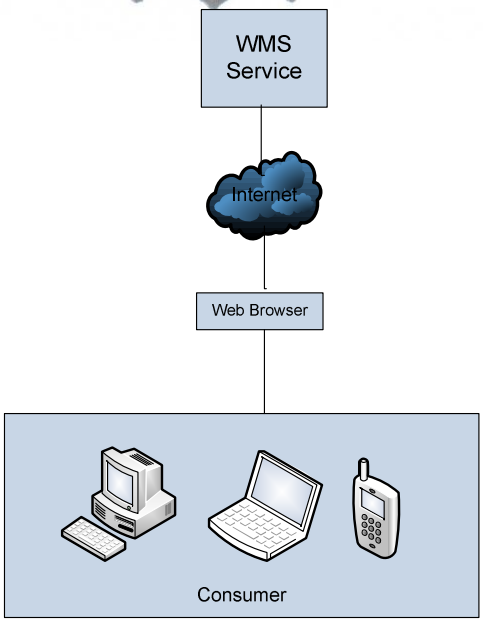
Implementasi aplikasi WMS *Dashboard* pada jaringan komputer dapat dilihat dari 2 sisi, yaitu sisi client dan sisi server(bagian service). Implementasi aplikasi WMS *Dashboard* pada jaringan komputer digambarkan sebagai berikut :

1. Sisi Server (Bagian Service)



Gambar 33. Arsitektur Jaringan di sisi server

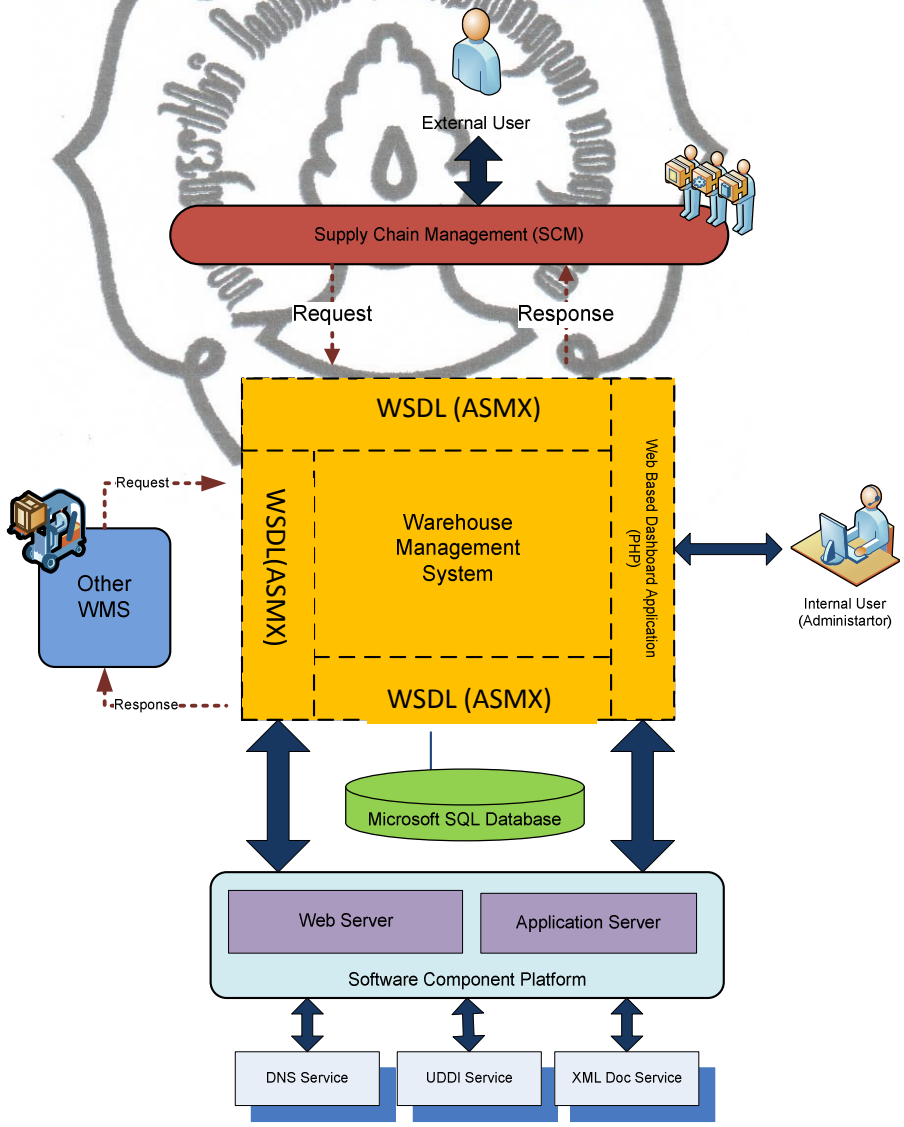
2. Sisi Client (Bagian UI untuk User)



Gambar 34. Arsitektur Jaringan di sisi client

4.1.5 Arsitektur Aplikasi WMS *Dashboard* dengan User dan System Lain

Aplikasi WMS *Dashboard* selain dapat langsung digunakan oleh User, juga dapat digunakan oleh System lain. System lain memanfaatkan service yang dimiliki oleh WMS *Dashboard* dengan cara mengambil WSDL dari service yang dibutuhkannya. Pengambilan WSDL tersebut dapat menggunakan protokol SOAP atau HTTP POST. Seangkan untuk user dapat mengakses aplikasi WMS *Dashboard* dengan memanfaatkan protokol HTTP dengan menggunakan PHP. Berikut ini adalah arsitektur WMS *Dashboard* dengan user dan system lainnya:



Gambar 35. Arsitektur WMS dengan User dan System Lain

4.2 Hasil dan Pembahasan

4.2.1 WMS Service (System Interface)

Service untuk aplikasi WMS *Dashboard* dibagi menjadi 2 Service yaitu :

1. WMSDService

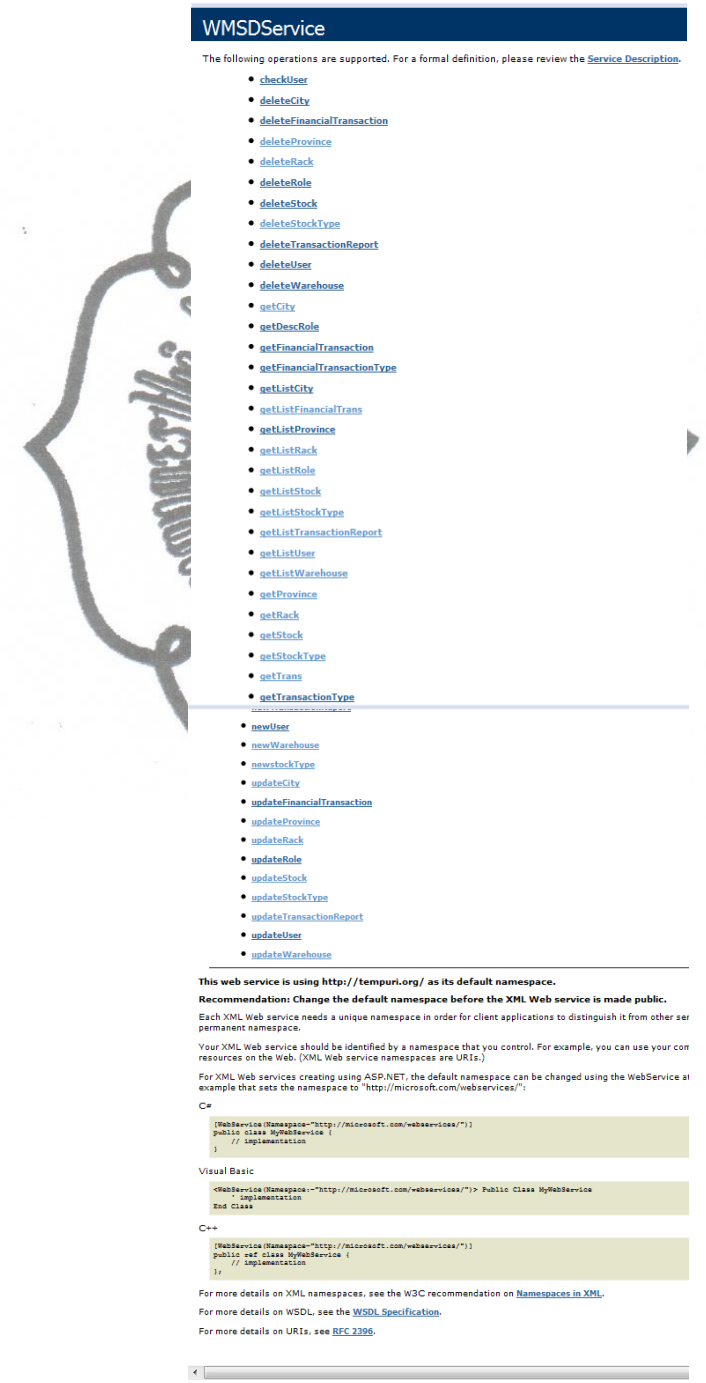
WMSD Service berisi service- service yang berhubungan dengan tabel-tabel data mentahan dari WMS. Service- service yang ada pada WMSD service antara lain :

Tabel 15 . Tabel Service- service pada WMSDService

NAMA SERVICE	KEGUNAAN SERVICE
checkUser	Memeriksa user yang menggunakan Service dan menggunakan database
deleteCity	Menghapus data pada tabel City
deleteFinancialTransaction	Menghapus data pada tabel FinancialTransaction
deleteProvince	Menghapus data pada tabel province
deleteRack	Menghapus data pada tabel rack
deleteRole	Menghapus data pada tabel role
delete*Stock*	Menghapus data pada tabel *stock*
delete*Stock*Type	Menghapus data pada tabel *stock*type
deleteTransactionReport	Menghapus data pada tabel transactionreport
deleteUser	Menghapus data pada tabel user
deleteWarehouse	Menghapus data pada tabel warehouse
getCity	Mengambil data dari tabel city
getDescRole	Mengambil deskripsi dari tiap role pada tabel role
getFinancialTransaction	Mengambil data dari tabel financial transaction
getFinancialTransactionType	Mengambil data dari tabel financialtransactiontype
getListCity	Mengambil data dari tabel city beuserpa list
getListFinancialTrans	Mengambil data dari tabel financial transaction beuserpa list
getListProvince	Mengambil data dari tabel province beuserpa list
getListRack	Mengambil data dari tabel rack beuserpa list
getListRole	Mengambil data dari tabel role beuserpa list
getList*Stock*	Mengambil data dari tabel *stock* beuserpa

	list
getList*Stock*Type	Mengambil data dari tabel *stock*type beuserpa list
getListTransactionReport	Mengambil data dari tabel transactionreport beuserpa list
getListUser	Mengambil data dari tabel user beuserpa list
getListWarehouse	Mengambil data dari tabel warehouse beuserpa list
getProvince	Mengambil data dari tabel province
getRack	Mengambil data dari tabel rack
get*Stock*	Mengambil data dari tabel *stock*
get*Stock*Type	Mengambil data dari tabel *stock*type
getTrans	Mengambil data dari tabel transaction
getTransactionType	Mengambil data dari tabel transactiontype
getUser	Mengambil data dari tabel user
getWarehouse	Mengambil data dari tabel warehouse
getusername	Mengambil data username dari tabel user
newCity	Menambah data pada tabel City
newFinancialTransaction	Menambah data pada tabel FinancialTransaction
newProvince	Menambah data pada tabel province
newRack	Menambah data pada tabel rack
newRole	Menambah data pada tabel role
new*Stock*	Menambah data pada tabel *stock*
newTransactionReport	Menambah data pada tabel TransactionReport
newUser	Menambah data pada tabel User
newWarehouse	Menambah data pada tabel Warehouse
new*stock*Type	Menambah data pada tabel StoockType
updateCity	Mengubah data pada tabel City
updateFinancialTransaction	Mengubah data pada tabel Financial Transaction
updateProvince	Mengubah data pada tabel province
updateRack	Mengubah data pada tabel rack
updateRole	Mengubah data pada tabel role
update*Stock*	Mengubah data pada tabel *stock*
update*Stock*Type	Mengubah data pada tabel *stock*type
updateTransactionReport	Mengubah data pada tabel transactionreport
updateUser	Mengubah data pada tabel user
updateWarehouse	Mengubah data pada tabel warehouse

Berikut ini adalah *Screenshot* dari service yang ada saat diusernnning di browser :



Gambar 36. *Screenshot* dari WMSD Service

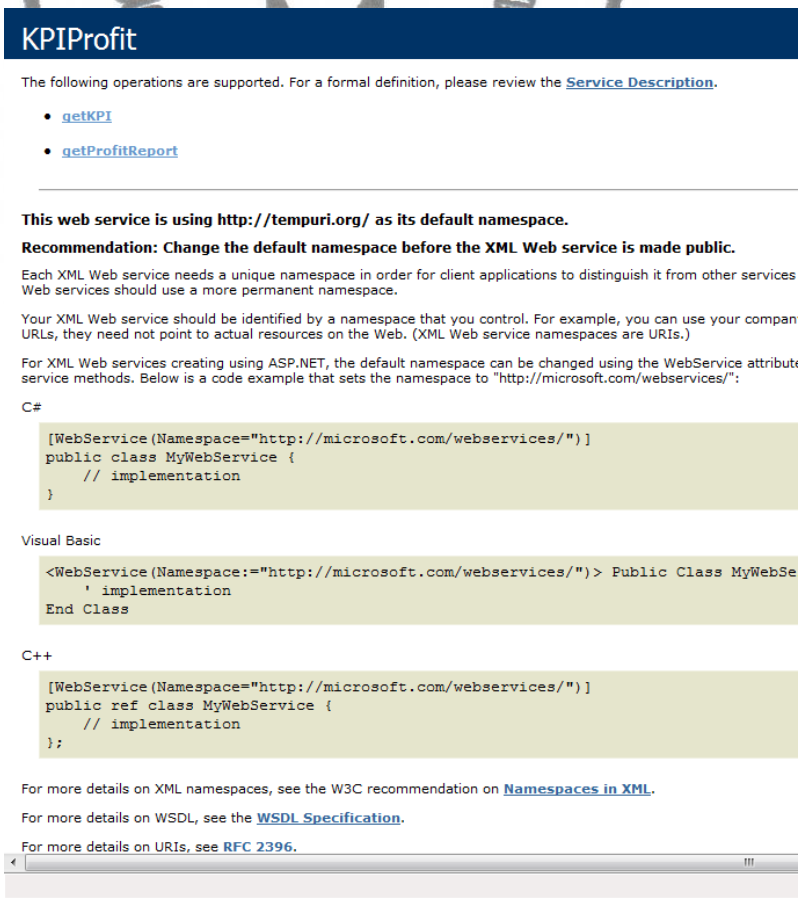
2. KPIProfit Service

KPIProfit Service berisi service- service yang berhubungan langsung dengan tabel KPI dan tabel Profit, dimana kedua tabel tersebut akan menjadi sumber dari pembuatan report. Servicenya antara lain:

Tabel 16. Tabel Service- service pada KPIProfit Service

NAMA SERVICE	KEGUNAAN SERVICE
getKPI	Mendapatkan data dari table KPI beuserpa list
getProfitReport	Mendapatkan data dari table profitReport beuserpa list

Berikut ini adalah *Screenshot* dari service yang ada saat diusernnning di browser :



Gambar 37. *Screenshot* dari KPIProfit Service

4.2.2 WMS *Dashboard* Application (User Interface)

User Interface dari aplikasi WMS *Dashboard* dibagi menjadi 3 bagian, yaitu User Interface untuk General User, Internal User/Admin dan Regular User/Owner dari gudang/*Stock*.

4.2.2.1 User Interface untuk General User/ Visitor

User Interface untuk Internal User/Admin terdiri dari 2 Halaman utama, yaitu :

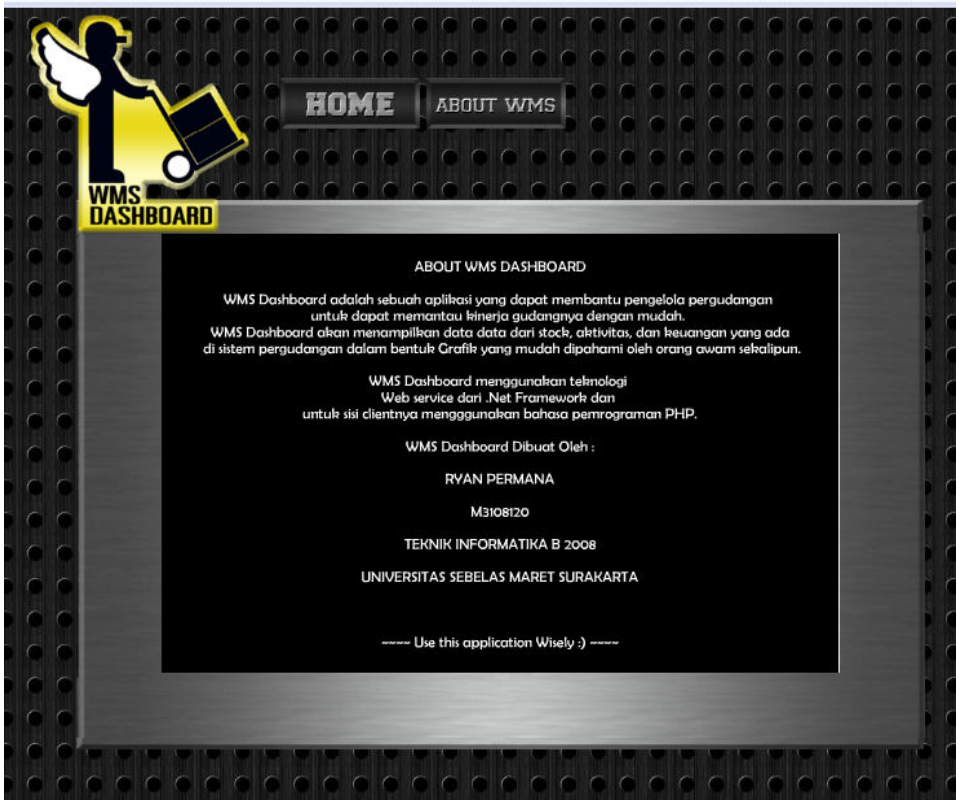
1. Halaman Login

   Halaman Login adalah halaman pertama yang akan muncul saat visitor mengakses aplikasi WMS *Dashboard* ini. Pada halaman ini visitor diminta untuk mengisi username/email dan password yang dimilikinya. Apabila username/email yang dimasukkan benar maka visitor akan dibawa menuju interface sesuai dengan level accountnya. Berikut ini adalah *Screenshot* dari halaman login :

Gambar 38. Halaman Login

2. Halaman About WMS *Dashboard*

Pada halaman ini Visitor yang notabene tidak bisa melakukan apa- apa, karena bukan merupakan user yang berhak mengakses fungsi dari aplikasi WMS *Dashboard*, masih dapat melihat informasi mengenai WMS *Dashboard*. Berikut ini adalah *screenshot* dari halaman About WMS yang membahas mengenai WMS *Dashboard*.



Gambar 39. Halaman Abour WMS Dashboard

4.2.2.2 User Interface untuk Admin/SCM

User Interface untuk Admin/ SCM terdiri dari 15 Halaman utama, yaitu :

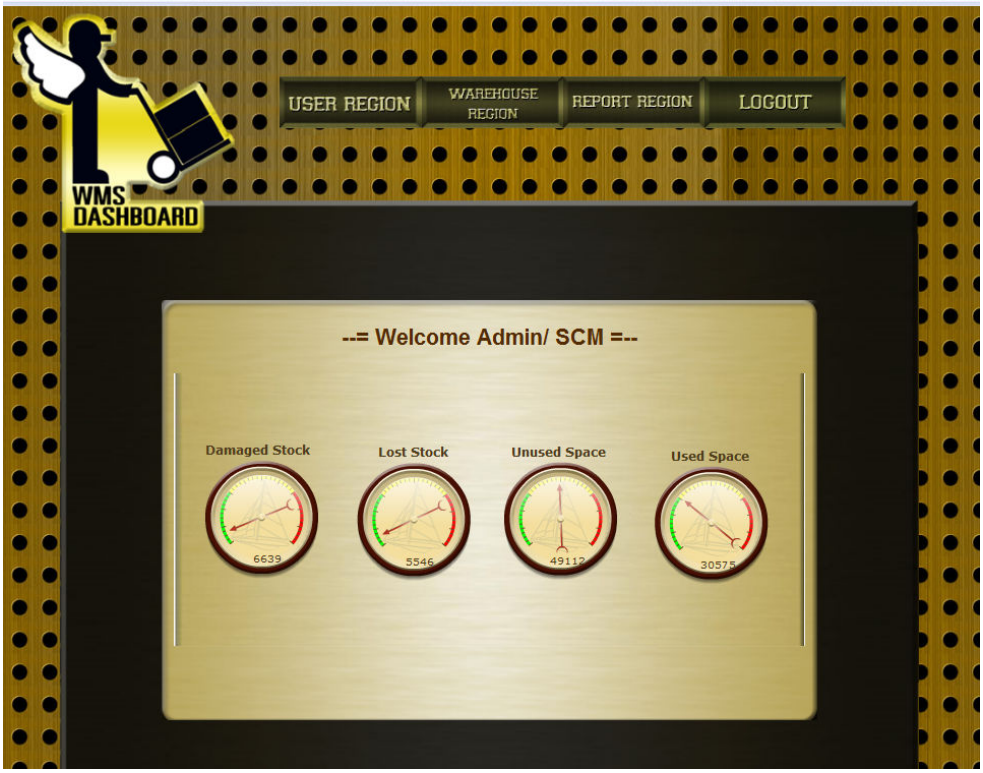
1. Halaman Home

Halaman Home untuk admin adalah halaman pertama yang akan muncul setelah visitor melakukan login sebagai admin. Dari halaman ini user

dapat mengakses halaman- halaman lain melalui menu di bagian atas. Selain itu pada halaman home dari admin terdapat 4 buah indikator alert yang menunjukkan jumlah stock yang rusak dan hilang serta menunjukkan wilayah gudang yang terpakai dan tidak terpakai. Indikator alert tersebut didapat dari perhitungan :

1 bagian a1ert = 1/3 x jumlah total variable x 100%

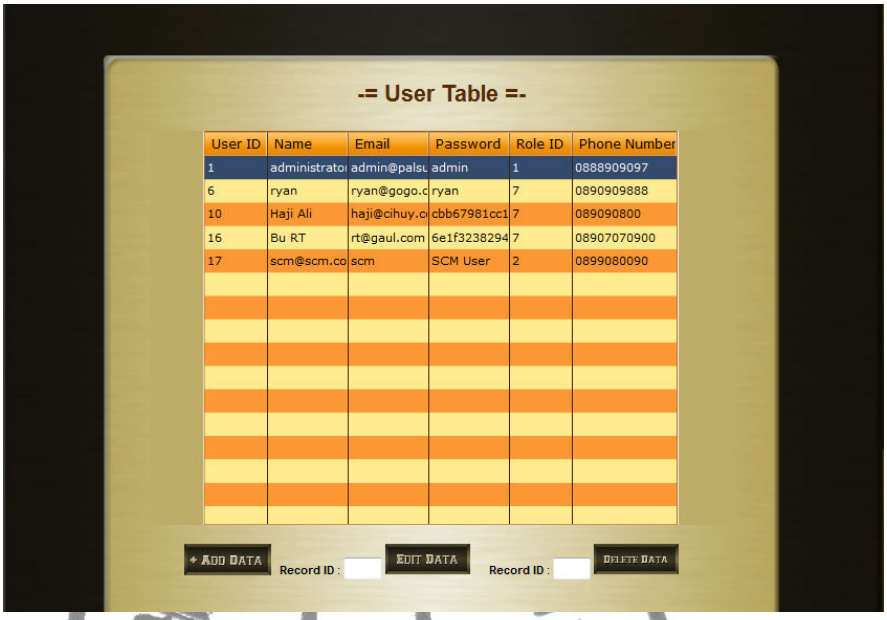
Berikut ini adalah *Screenshot* dari halaman home untuk admin:



Gambar 40. Halaman Home Admin/ SCM

2. Halaman Tabel User

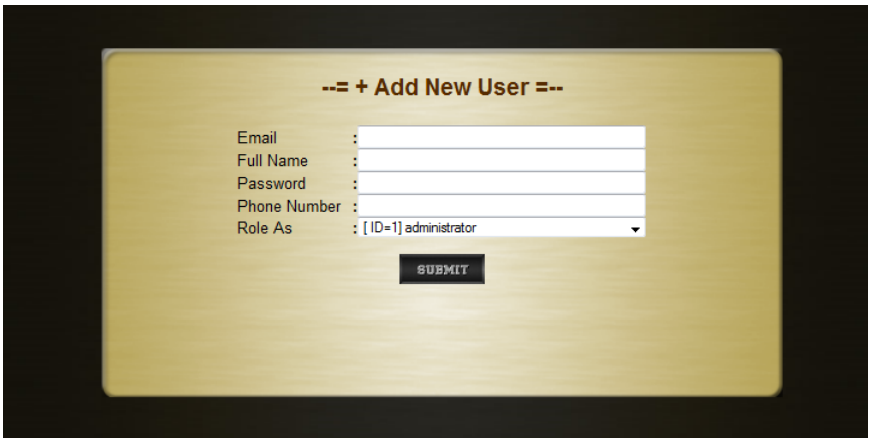
Halaman tabel user, adalah halaman yang akan memunculkan data- data dari tabel user. Pada halaman ini admin dapat memantau user yang terdaftar pada aplikasi ini, selain itu admin juga dapat menambah, mengubah atau menghapus data user yang ada. Berikut adalah *Screenshot* dari halaman tabel user :



Gambar 41. Halaman Table User

Dibagian bawah halaman ini terdapat tiga *button* untuk memodifikasi data pada tabel user, yaitu *button* add data untuk menambah data, *button* edit data untuk mengubah data dan *button* delete data untuk menghapus data. Untuk *button* edit dan delete terdapat sebuah text box untuk menentukan *record* nomer berapa yang akan dihapus.

Bila admin memilih *button* add data maka akan muncul form untuk menambah data dari tabel user, berikut ini adalah form tambah data dari tabel user :



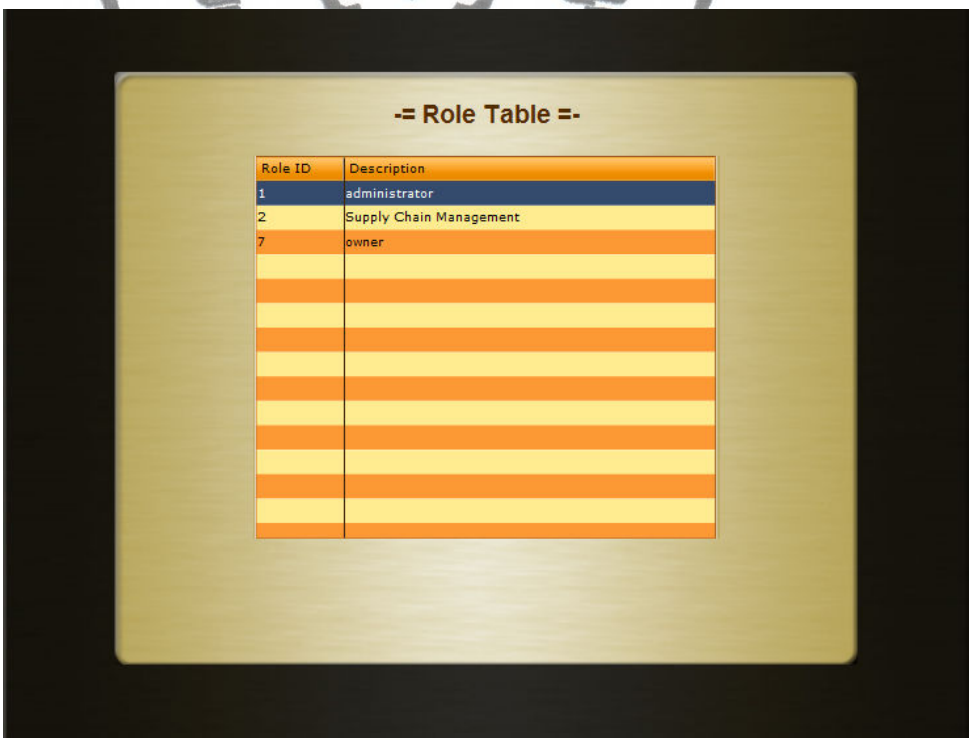
Gambar 42. Halaman Tambah User

Pada form tersebut terdapat *field* Role As yang beuserpa combo box, data dari combo box tersebut diambil dari tabel role. Maksud dari *field* tersebut adalah menentukan jenis account dari user yang akan ditambahkan.

Sedangkan untuk form updatenya akan memunculkan jenis form yang sama, hanya saja dengan *field- field* yang telah terisi data dari *record* yang akan diedit.

3. Halaman Tabel Role

Halaman tabel role, adalah halaman yang akan memunculkan data- data dari tabel role. Tabel role adalah tabel yang berisi jenis- jenis user yang dapat mengakses aplikasi WMS Dashboard. Berikut adalah *Screenshot* dari halaman tabel role :

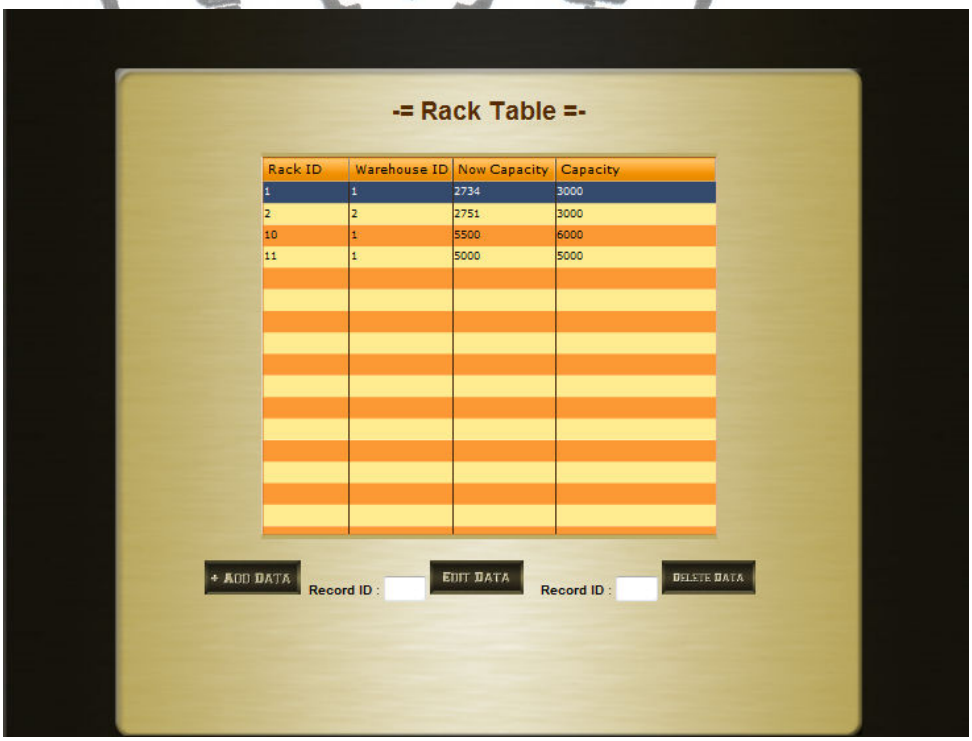


Gambar 43. Halaman Table Role

Pada tabel ini tidak disediakan tombol untuk menambah, mengedit atau menghapus *record* yang ada. Karena tabel role bukanlah tabel yang dapat diubah oleh admin.

4. Halaman Tabel Rack

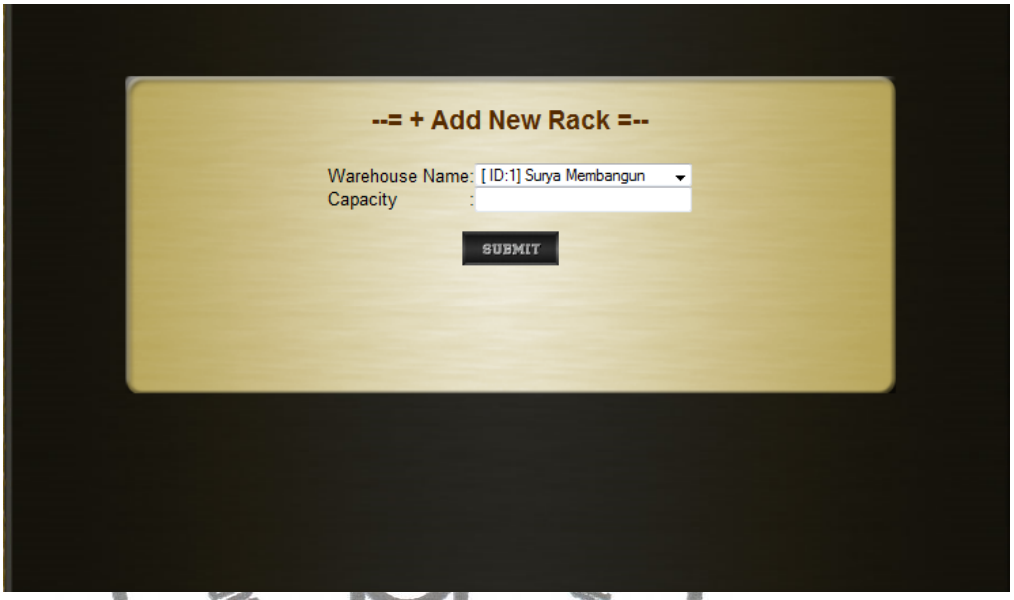
Halaman tabel rack, adalah halaman yang akan memunculkan data- data dari tabel rack. Tabel rack adalah tabel yang berisi rack- rack yang menjadi tempat untuk menyimpan *stock- stock* milik owner. Tabel ini juga memunculkan asal warehouse dari rack yang terdaftar. Selain itu tabel ini juga memberikan informasi kapasitas terkini dari tiap rack yang ada. Informasi terkini kapasitas rack akan beuserbah saat ada *stock* yang disimpan di dalam rack tersebut. Berikut ini adalah *Screenshot* dari halaman tabel rack:



Gambar 44. Halaman Table Rack

Pada halaman ini tersedia tombol untuk memanipulasi data pada tabel, sama seperti pada tabel user. Tombol ini disediakan karena rack adalah

variable yang dimungkinkan untuk teusers bertambah atau diubah kapasitasnya. Berikut ini adalah *Screenshot* dari form tambah data dari tabel rack :



Gambar 45. Halaman Tambah Rack

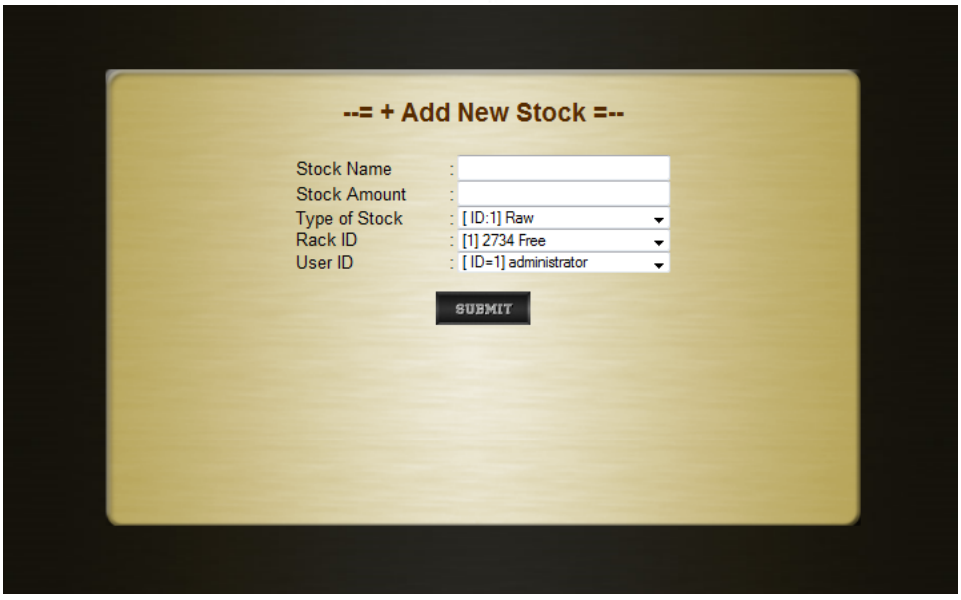
Pada form ini hanya ada 2 *field* yang dapat diisi oleh admin, karena unuk *field* nowCapacity dan RackID akan terisi secara otomatis. Rack ID akan terisi otomatis karena fitur auto_increment dari database, dan untuk *field* nowCapacity akan terisi otomatis dengan mengacu pada *field* capacity dari tabel rack dan *field* amount dari table *stock*.

5. Halaman Tabel *Stock*

   Halaman tabel *stock*, adalah halaman yang akan memunculkan data- data dari tabel *stock*. Tabel *stock* adalah tabel yang berisi data- data dari *stock* yang disimpan dalam rack pada sbuah gudang. Pada tabel ini juga terdapat data user yang memiliki *stock* tersebut, selain itu juga terdapat jenis *stock* yang disimpan pada rack. Berikut ini adalah *Screenshot* dari halaman tabel *stock*:

Gambar 46. Halaman Table *Stock*s

Data pada tabel *stock* adalah data yang dapat ditambah, diubah dan dihapus, karena *stock* meuserpakan variable yang dinamis pada sistem pergudangan. Berikut ini adalah form untuk menambah data *stock*:

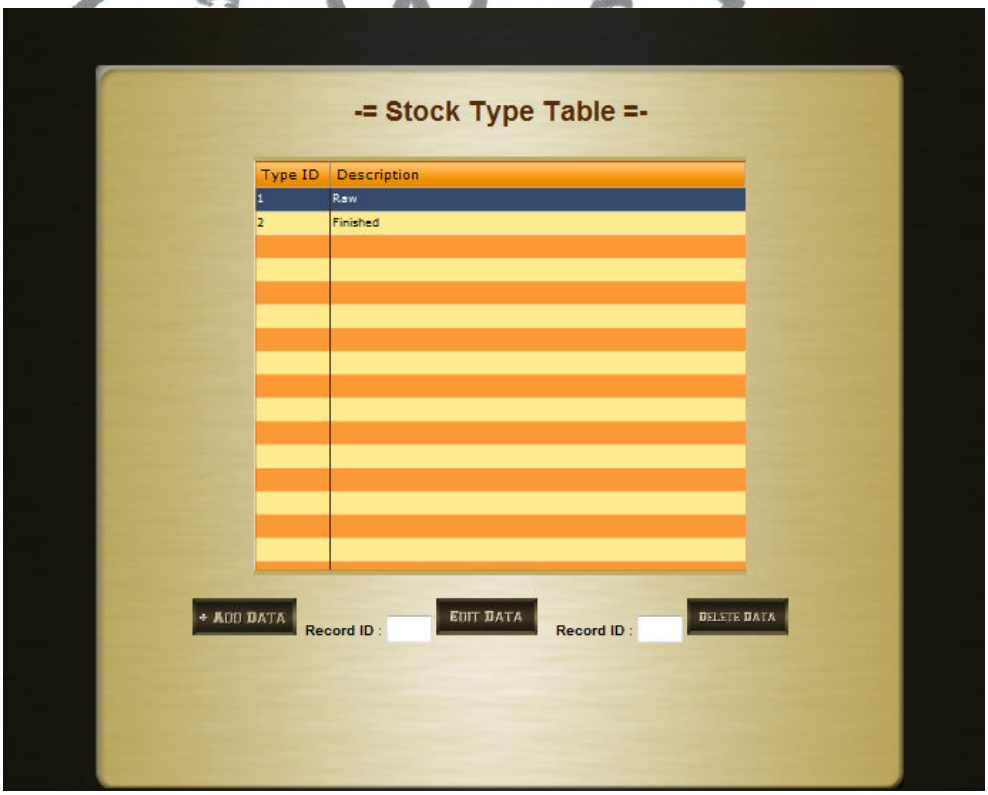


Gambar 47. Halaman Tambah *Stock*

3 combo box pada form ini mendapat data dari table lain. Untuk combo box Type of *Stock* mendapat data dari tabel *stock* type, untuk combo box Rack ID mendapat data dari tabel rack dan untuk *field* UserID mendapat data dari tabel User.

6. Halaman Tabel *Stock* Type

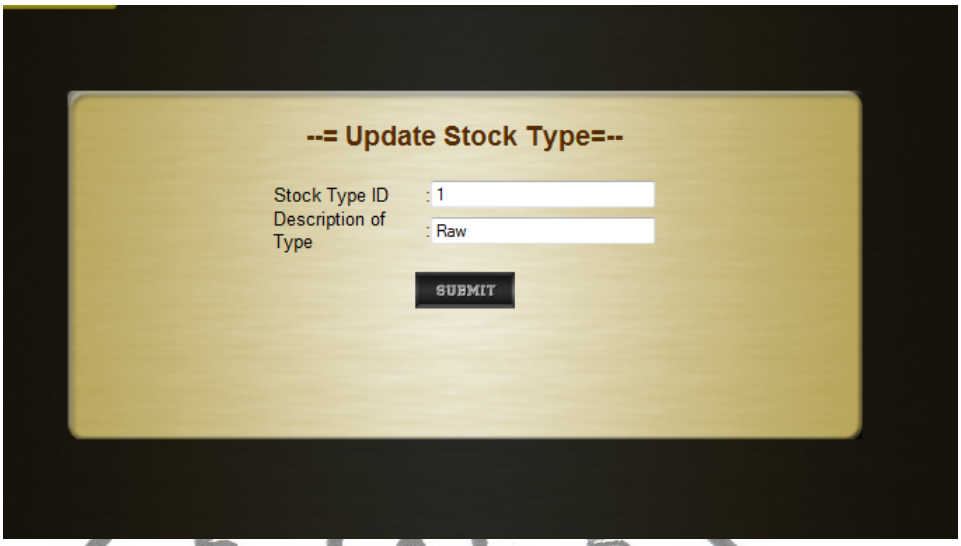
   Halaman tabel *stock* type, adalah halaman yang akan memunculkan data-data dari tabel *stock* type. Tabel *stock* type adalah tabel yang berisi data-data jenis *stock* yang dapat disimpan ke dalam rack. Jenis *stock* yang dapat disimpan ke dalam rack ada beberapa macam antara lain Raw untuk barang mentah dan Finished untuk barang yang telah melalui proses finishing. Berikut ini adalah *Screenshot* dari halaman tabel *stock*:



Gambar 48. Halaman Table *Stock* Type

Jenis *Stock* adalah variable yang dapat ditambah atau diubah datanya karena itu disediakan pula form untuk memanipulasi data pada tabel

*stock*type. Berikut ini adalah form untuk memanipulasi data dari tabel *stock*type:



Gambar 49. Halaman Update *Stock* Type

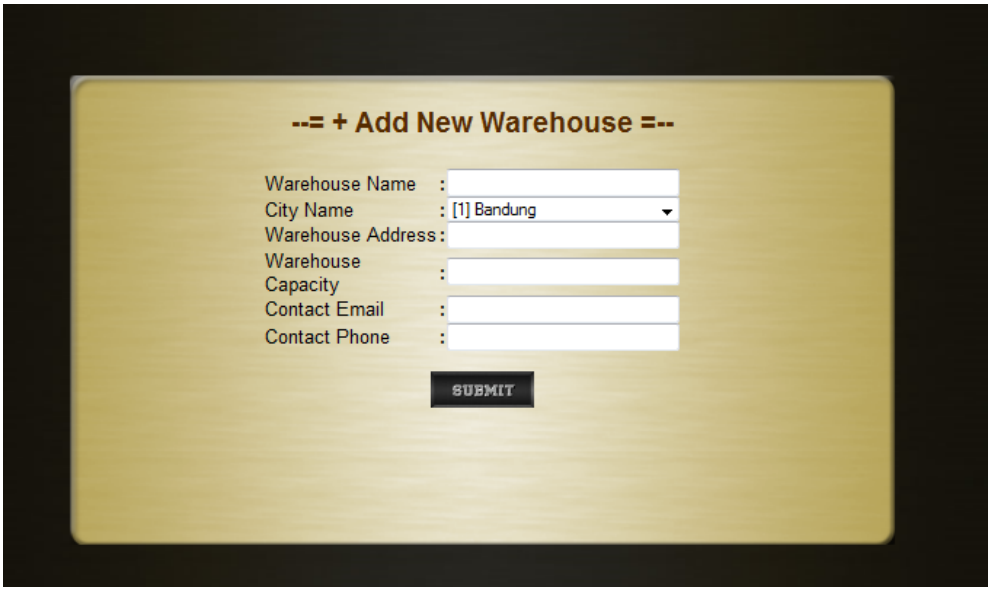
Tampilan diatas adalah tampilan saat admin akan mengubah data dalam tabel *stock*type dengan nomor *record* 1.

7. Halaman Tabel Warehouse

   Halaman tabel warehouse, adalah halaman yang akan memunculkan data-data dari tabel warehouse. Tabel warehouse adalah tabel yang berisi data-data mengenai warehouse yang diiatur oleh WMS ini. Pada tabel warehouse terdapat nama, alamat dan kapasitas dari warehouse yang ada. Sama seperti tabel rack, tabel warehouse juga memiliki sebuah *field* yang dinamis, yatu *field* now capacity. *Field* ini akan beubah saat ada *stock* yang disimpan kedalam gudang. Semakin banyak *stock* yang disimpan . maka semakin berkurang pula kapasitas dari sebuah gudang.Berikut ini adalah *Screenshot* dari halaman tabel warehouse:

Gambar 50. Halaman Table Warehouse

Warehouse adalah sebuah variable dinamis, yang dapat bertambah atau berkurang, oleh karena itu pada halaman ini diberi fasilitas untuk menambah, mengedit atau menghapus *record* yang ada pada tabel. Berikut ini adalah form untuk menambah sebuah warehouse kedalam tabel:



Gambar 51. Halaman Tambah Warehouse

8. Halaman Tabel City

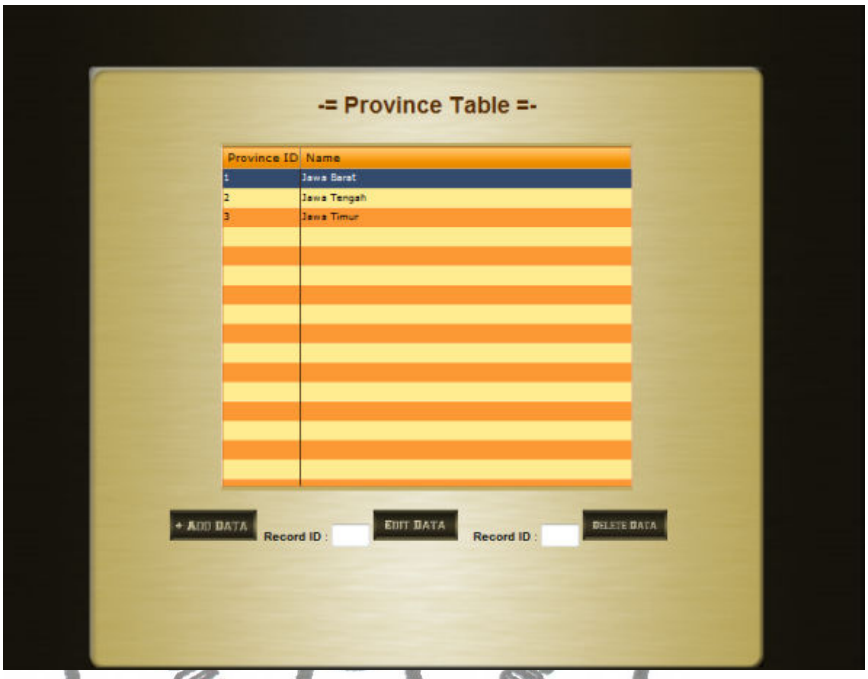
Halaman tabel city, adalah halaman yang akan memunculkan data- data dari tabel city. Tabel city adalah tabel yang berisi data- data mengenai kota- kota yang dimana sebuah gudang berada. Tabel ini berisi mengenai nama dari kota dan juga dimana letak provinsi dari suatu kota. Tabel kota berhubungan dengan tabel warehouse, setiap warehouse pasti memiliki sebuah kota. Oleh karena itu pada tabel warehouse juga terdapat *field* cityID yang menunjukkan kota dari suatu warehouse.Berikut ini adalah *Screenshot* dari halaman tabel city:

Gambar 52. Halaman Table City

9. Halaman Tabel Province

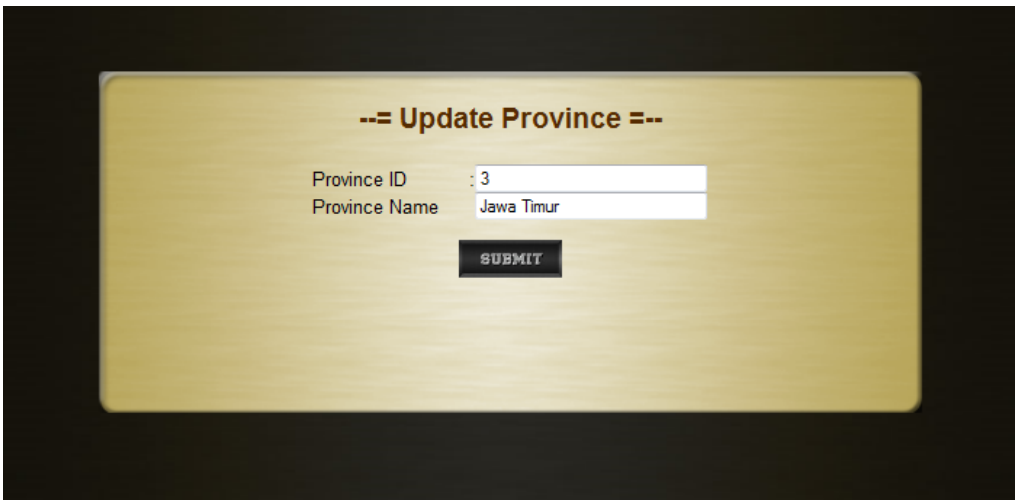
Halaman tabel province, adalah halaman yang akan memunculkan data-data dari tabel province. Tabel province adalah tabel yang berisi data- data mengenai provinsi- provinsi yang menjadi induk dari sebuah kota. Dalam tabel kota juga terdapat province ID yang menunjukkan provinsi dari

sebuah kota, karena setiap kota pasti memiliki provinsi. Berikut ini adalah *Screenshot* dari halaman table province:



Gambar 53. Halaman Table Province

Province yang terdaftar pada WMS ini dapat bertambah, karena itu pada halaman ini diberi tombol add data, edit data dan delete data untuk memanipulasi data pada tabel. Berikut ini adalah form edit province saat melakukan editing pada *record* no 3 :

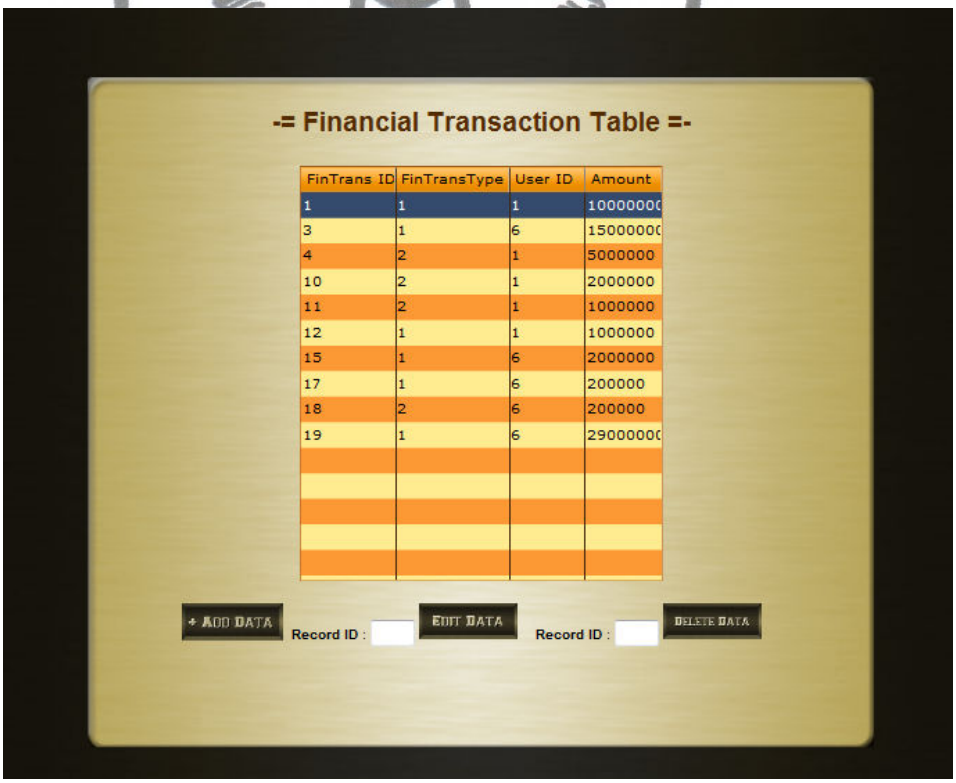


Gambar 54. Halaman Update Province

Saat user memilih *record* id no 3 dan menekan tombol edit data maka akan muncul form update province dengan textbox yang telah terisi data- ata dari *record* no 3. Sehingga user dapat memiliki panduan untuk mengubah data yang diinginkan.

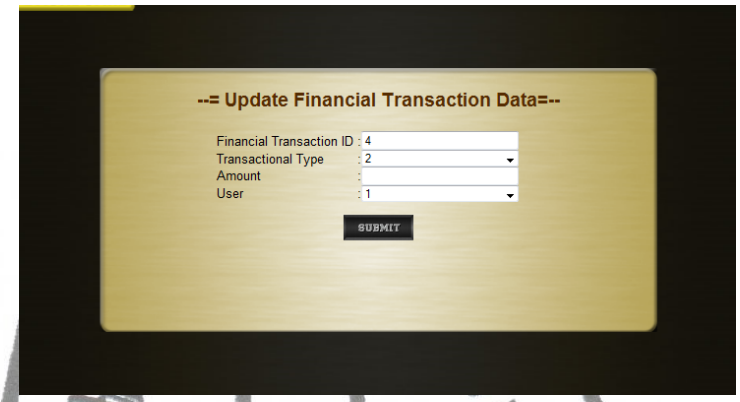
10. Halaman Tabel Financial Transaction

Halaman tabel Financial Transaction, adalah halaman yang akan memunculkan data- data dari tabel Financial Transaction. Tabel Financial Transaction adalah tabel yang berisi data- data mengenai Transaksi-transaksi Financial/ keuangan yang tejadi pada sistem pergudangan. Transaksi keuangan pada sistem pergudangan ada 2 jenis, yaitu income/uang masuk dan outcome/ uang keluar. Jenis transaksi kuangan itu mengacu kepada tabel FinancialTransactionType. Berikut ini adalah *Screenshot* dari halaman table province:



Gambar 55. Halaman Tabel Financial Transaction

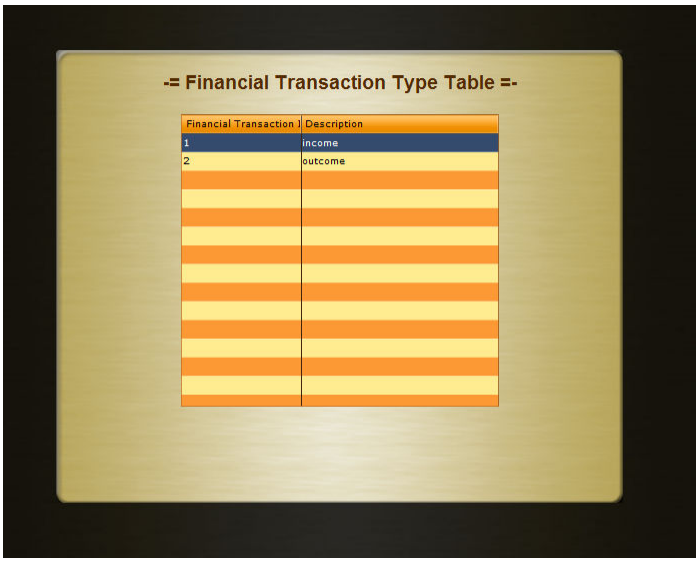
Apabila ada kesalahan dalam penginputan data transaksi keuangan, admin masih dapat mengedit data dari *record* yang salah ia masukkan, caranya adalah dengan menuliskan no *record* yang akan diedit dan menekan tombol edit, maka akan muncul form edit seperti dibawah ini :



Gambar 56. Halaman Update Financial Transaction

11. Halaman Tabel Financial Transaction Type

Tabel Financial Transaction Type bukanlah tabel yang bisa dimanipulasi datanya oleh admin, karena sudah dapat dpastikan bahwa jenis tranaksi financial pada sistem pergudangn hanya ada 2 yaitu income/uang masuk/pendapatan dan outcome/uang keluar/pengeluaran. Berikut ini adalah *Screenshot* dari halaman Tabel Financial Transaction Type:



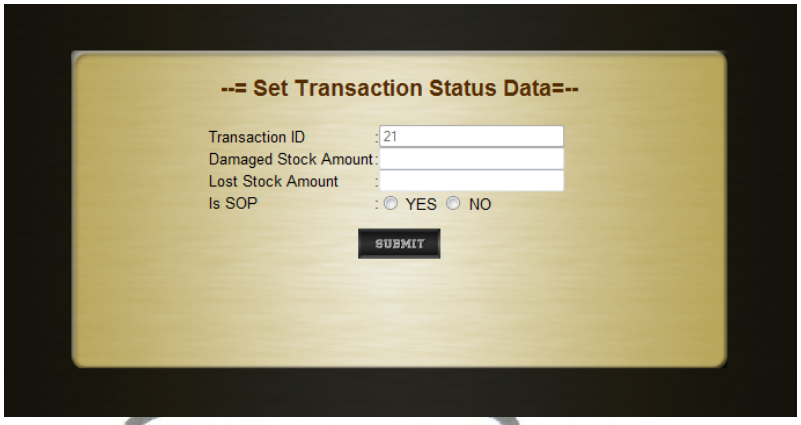
Gambar 57. Halaman Table Financial Transaction Type

12. Halaman Tabel Transaction

Halaman ini adalah halaman yang akan menampilkan data data dari tabel transaction. Tabel transaction adalah tabel yang mencatat transaksi-transaksi yang terjadi pada sistem pergudangan. Tabel ini akan melakukan penambahan data secara otomatis saat ada barang yang masuk dan disimpan ke dalam gudang. Berikut ini adalah *Screenshot* dari halaman tabel transaction :

Gambar 58. Halaman Table Transaction

Pada halaman ini admin dapat melakukan update data untuk mengeset banyak barang yang usersak, yang hilang serta apakah aktivitas tersebut memenuhi SOP atau tidak. Berikut ini adalah form update data untuk *record* ke 21 dari tabel transaksi :

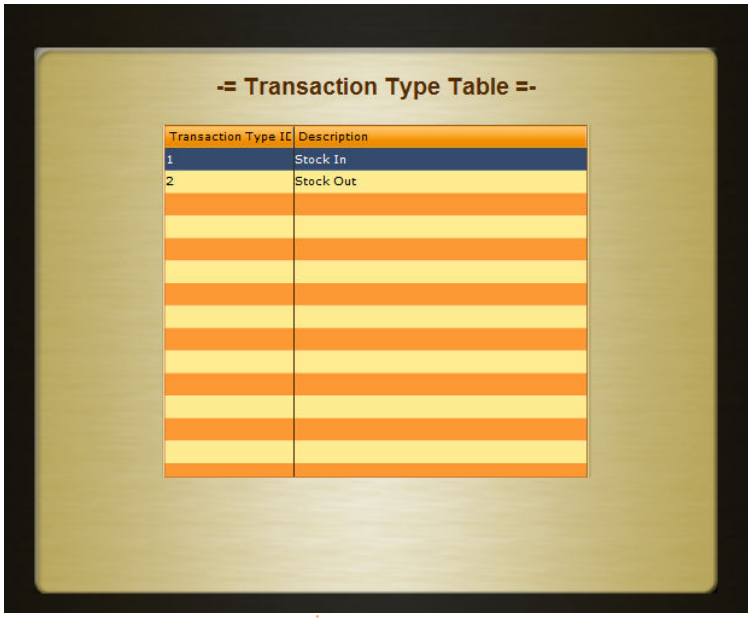


Gambar 59. Halaman Set Transaction Data

Data yang telah diupdate akan tersimpan pada tabel, dan data tersebut akan ter-rekap pada tabel KPI.

13. Halaman Tabel Transaction Type

Halaman tabel transaction type, adalah halaman yang akan memunculkan data- data dari tabel transaction type. Tabel *stock* type adalah tabel yang berisi data- data jenis transaksi yang dapat dilakukan pada sistem pergudangan. Transaksi disini adlah transaksi keluar masuknya barang. Tabel ini tidak memiliki fasilitas add, edit dan delete karena data pada tabel jenis transaksi adalah jenis data yang tidak dapat beuserbah. Berikut ini adalah *Screenshot* dari halaman tabel *stock*:



Gambar 60. Halaman table Transaction Type

14. Halaman Tabel KPI

Tabel KPI atau Tabel Key Performance Indicator adalah tabel yang berisi data data yang dapat menjadi indikator perfoma dari sistem pergudangan. Pada tabel ini dapat dlihat report perbulan dari sistem pergudangan. Report tersebut memuat beberapa data, yaitu status *stock*, status space dan status SOP sebuah aktivitas. Tabel ini akan teuserpdate secara otomatis setelah admin menambah/ mengubah data *stock* dan menge-set status *stock*, serta terpenuhi tidaknya SOP pada tabel Transaksi. Berikut ini adalah *Screenshot* dari halaman tabel KPI :

Gambar 61. Halaman Table KPI

15. Halaman Tabel Profit

Halaman tabel profit adalah halaman yang manampilkan data data dari table profit. Tabel profit adalah table yang berisi laporan keuntungan dari system pergudangan. Table ini akan teuserpdate secara otomatis, saat

admin menambah data pada table financial transaction. Apabila admin menambah data income maka *field* data income akan bertambah jumlah datanya dan apabila admin menambah data outcome maka *field* data outcome akan bertambah datanya. Berikut ini adalah *Screenshot* dari halaman table profit:

Gambar 62. Halaman Table Profit

4.2.2.3 User Interface untuk User/Owner

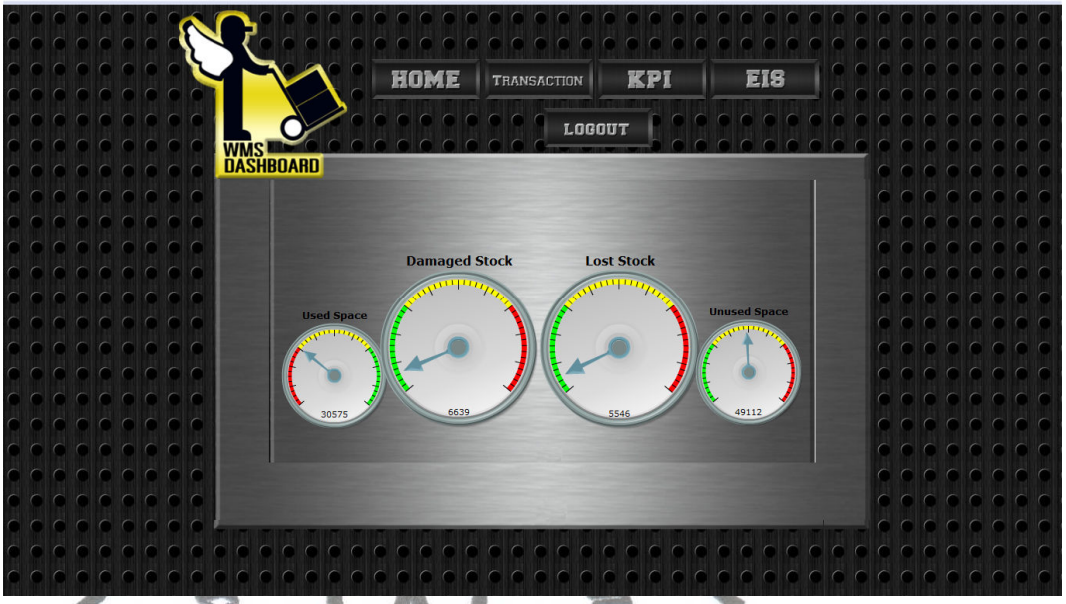
User Interface untuk User/Owner terdiri dari 10 Halaman utama, yaitu :

1. Halaman Home

Halaman Home untuk User/Owner adalah halaman pertama yang akan muncul setelah visitor melakukan login sebagai USER. Dari halaman ini USER dapat mengakses halaman- halaman lain melalui menu di bagian atas. Selain itu pada halaman home dari admin terdapat 4 buah indikator alert yang menunjukkan jumlah stock yang rusak dan hilang serta menunjukkan wilayah gudang yang terpakai dan tidak terpakai. Indikator alert tersebut didapat dari perhitungan :

1 bagian a1ert = 1/3 x jumlah total variable x 100%

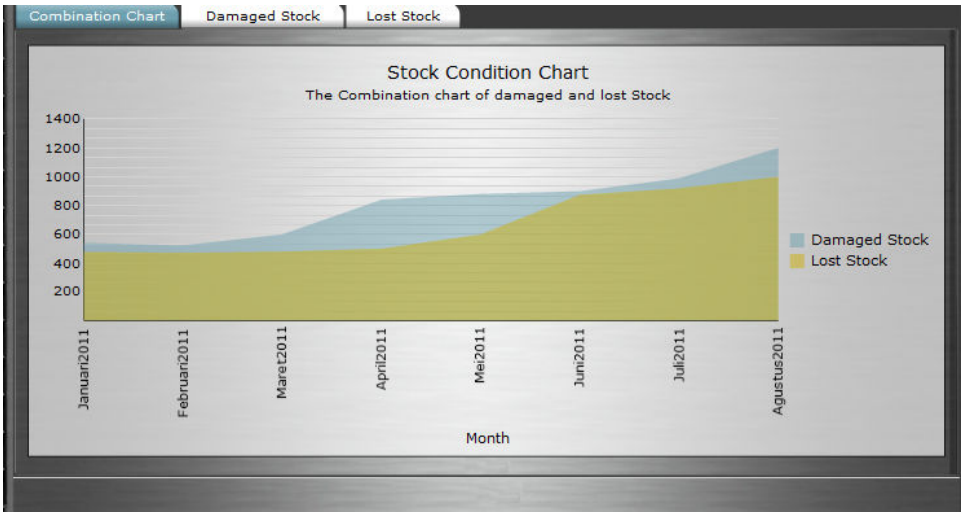
Berikut ini adalah *Screenshot* dari halaman home untuk USER:



Gambar 63. Halaman Home User/Owner

2. Halaman *Dashboard Stock* Status

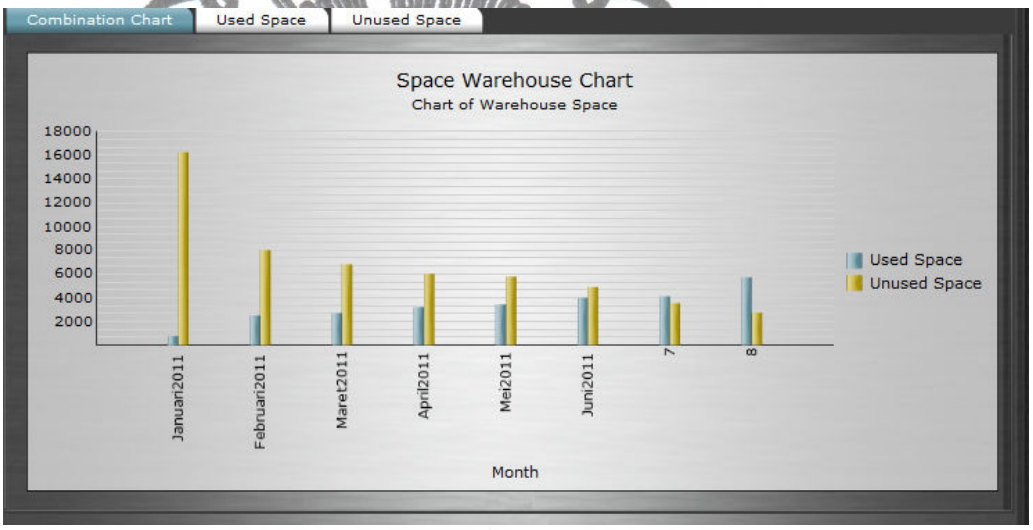
Pada halaman *Dashboard Stock* Status, USER dapat memantau keadaan *stock* pada gudangnya melalui grafik damaged *stock*, lost *stock* gabungan keduanya. Berikut ini adalah *Screenshot* dari halaman Dashbaord *Stock* Status:



Gambar 64. Halaman *Dashboard Stock* Status

3. Halaman *Dashboard* Warehouse Space Status

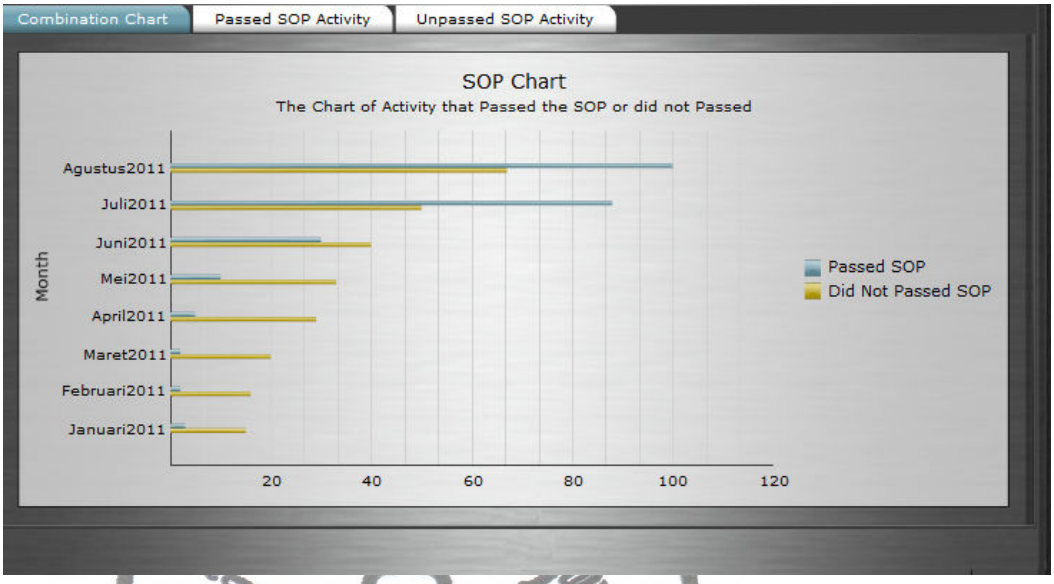
   Halaman *Dashboard* Warehouse Space Status akan menampilkan re-presentasi data- data dari tabel KPI teusertama pada bagian penggunaan space pada gudang, yaitu pada *field* used dan un-used space dalam bentuk grafik. Pada halaman ini USER dapat melihat data – data tersebut pada grafik used space, un-used space dan gabungan dari keduanya. Berikut ini adalah *Screenshot* dari halaman *Dashboard* Warehouse Space Status:



Gambar 65. Halaman *Dashboard* space status

4. Halaman *Dashboard* SOP Activity Status

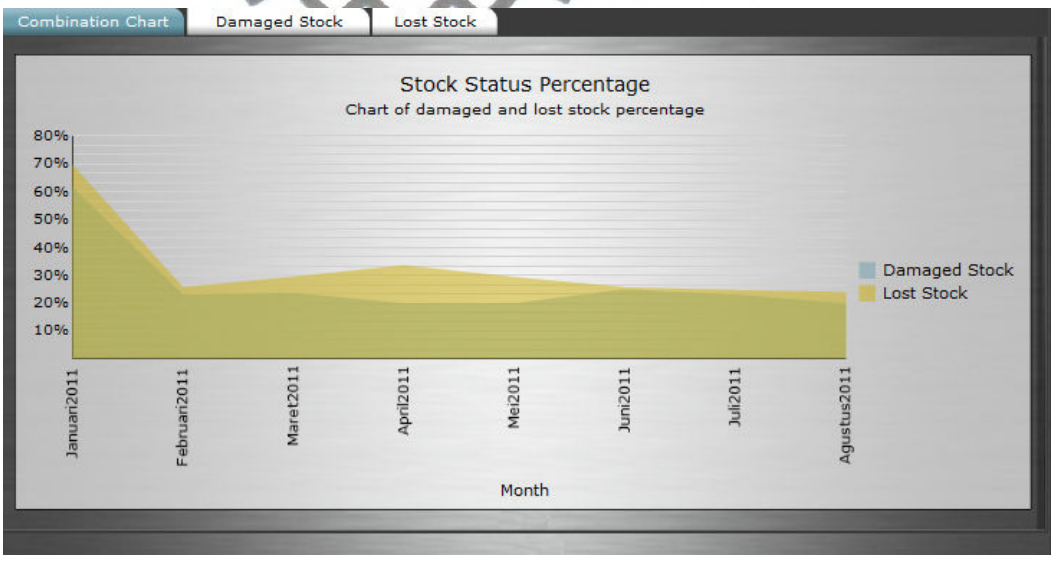
   Saat membuka halaman *Dashboard* SOP Activity Status, USER akan disugukan re-presentasi beuserpa grafik dari data- data tabel KPI teusertama dari *field* appropiate SOP dan un-appropiate SOP. Berikut adalah *Screenshot* dari halaman *Dashboard* SOP Activity Status:



Gambar 66. Halaman *Dashboard* SOP activity status

5. Halaman *Dashboard Stock* Percentage

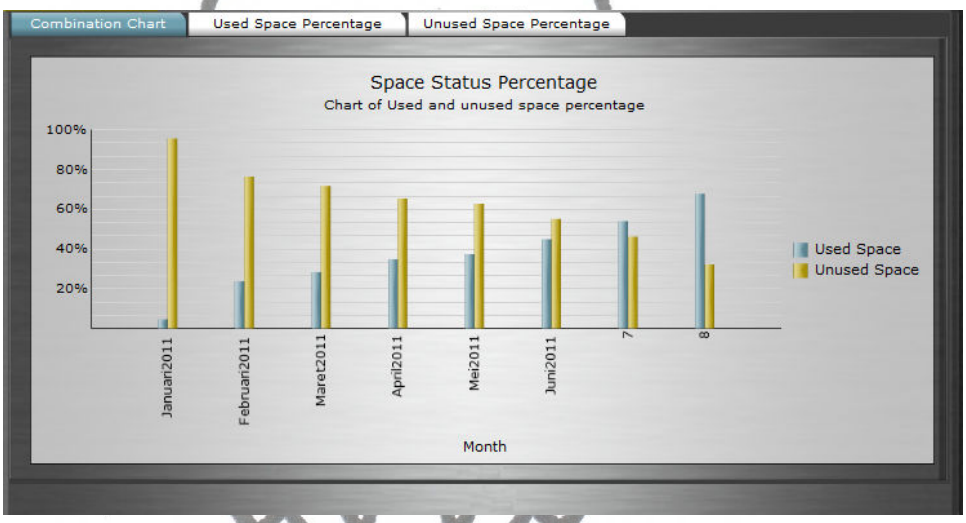
   Pada halaman *Dashboard Stock* Status Percentage, USER dapat memantau persentase barang yang rusak (damaged *stock*), barang yang hilang (lost *stock*) dan gabungan dari kedua melalui grafik seperti diperlihatkan dibawah ini :



Gambar 67. Halaman *Dashboard stock* percentage

6. Halaman *Dashboard* Warehouse Space Percentage

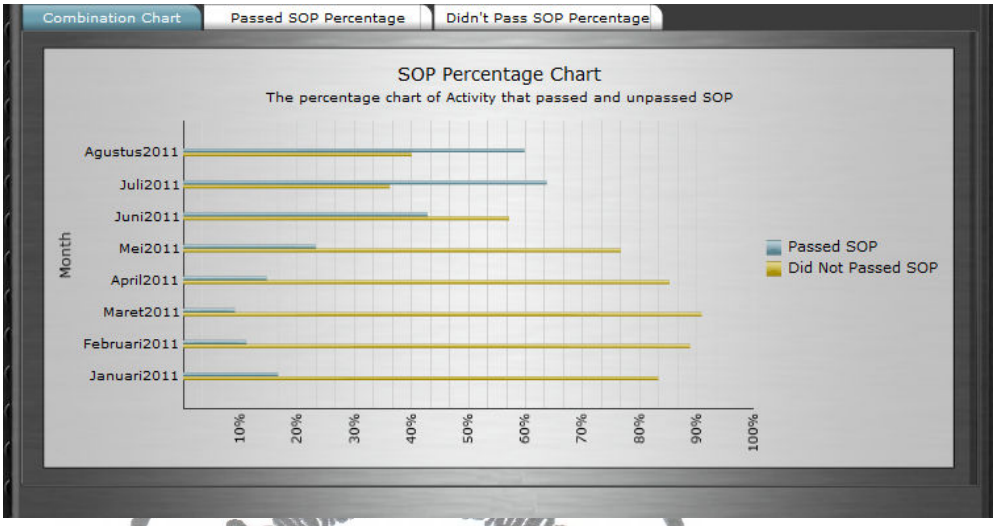
   Halaman *Dashboard* Warehouse Space Percentage akan menampilkan pengolahan data- data dari tabel KPI teusertama pada bagian penggunaan space pada gudang, yaitu pada *field* used dan un-used space menjadi nilai persentase. Pada halaman ini USER dapat melihat persentase data – data tersebut pada grafik used space percentage, un-used space percentage dan gabungan dari keduanya. Beriut ini adalah *Screenshot* dari halaman *Dashboard* Warehouse Space Percentage:



Gambar 68. Halaman *Dashboard* space status percentage

7. Halaman *Dashboard* SOP Activity Percentage

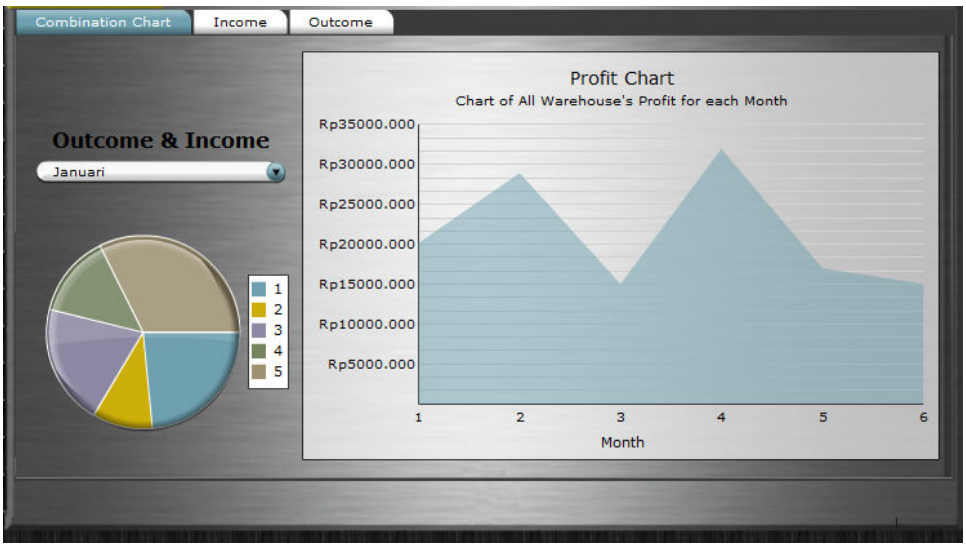
   Saat mengakses halaman *Dashboard* SOP Activity Percentage, USER akan ditampilkan data- data persentase dari aktivitas- aktivitas yang sesuai dan tidak sesuai dengan SOP dengan periode laporan per 1 bulan seperti halaman yang ditampilkan screenshotnya dibawah ini :



Gambar 69. Halaman *Dashboard* SOP percentage

8. Halaman *Dashboard* Profit

Halaman *Dashboard* Profit adalah halaman yang menampilkan keuntungan total yang didapat gudang- gudang yang ikut dalam sistem pergudangan. Pada halaman ditampilkan grafik keuntungan sistem pergudangan setiap periode per bulannya, selain itu juga ditampilkan grafik pemasukan dan pengeluaran yang terjadi setiap bulannya. Berikut ini adalah *Screenshot* dari halaman *Dashboard* Profit:



Gambar 70. Halaman *Dashboard* Profit

BAB V
PENUTUP

1.1 Kesimpulan

Berdasarkan aplikasi yang telah dirancang, dapat disimpulkan bahwa:

1. *Web Service* ASP. Net untuk aplikasi WMS *Dashboard* telah berhasil dibangun dengan menggunakan bahasa pemrograman C# dengan memanfaatkan aplikasi Visual Studio 2008.
2. *Web Service* untuk WMS *Dashboard* memiliki total 56 *Service* yang mengacu kepada 14 tabel dari database WMS.
3. Aplikasi client WMS *Dashboard* telah berhasil dibuat dengan menggunakan bahasa pemrograman PHP dengan memanfaatkan aplikasi Dreamweaver CS 4 dan Xcelsius.

1.2 Saran

Saran yang dapat disampaikan pada penulisan tugas akhir ini adalah:

1. *Web Service* WMS *Dashboard* hendaknya distandardisasikan dan didaftarkan pada UDDI, sehingga *service* tersebut dapat dimanfaatkan oleh masyarakat luas.
2. Untuk meningkatkan mobilitas dan update data secara otomatis *WMS dashboard* sebaiknya diintegrasikan dengan teknologi RFID .
3. Sekuritas *Web Service* hendaknya ditingkatkan dengan melakukan enkripsi pada XML yang ditransportasikan
4. Untuk manajemen loading data pada aplikasi WMS *dashboard* disisi client sebaiknya memanfatkan frame atau teknologi Jquery sehingga lebih meng-efektifkan *loading time*, sehingga user tidak mennunggu terlalu lama saat mengakses aplikasi WMS *Dashboard*.