

# Pengembangan Sistem Administrasi Pergudangan PT. PLN (Persero) Unit Keramasan

**Guntoro Barovich**

Teknik Informatika, Jl. Basuki Rahmat No.5, Palembang  
STMIK PalComTech, Palembang  
guntoro@palcomtech.ac.id

## **Abstrak**

*Sistem administrasi pergudangan merupakan salah satu bentuk manajemen tata kelola gudang agar barang-barang yang tersimpan didalamnya terdokumentasi dengan baik. Sistem administrasi pergudangan juga diterapkan di perusahaan negara seperti PT. PLN yang digunakan untuk mencatat keluar masuknya suku cadang yang digunakan dan ketersediaan suku cadang yang dibutuhkan. Perangkat lunak sistem administrasi pergudangan bisa diterapkan diberbagai platform baik berbasis desktop atau web. Perangkat lunak yang dikembangkan ini berbasiskan web dengan menggunakan siklus pengembangan menggunakan metode RUP sebagai metode pengembangan sistem. Perangkat lunak yang dikembangkan menggunakan model MVC dan pengujian sistem yang dilakukan menggunakan pengujian blackbox testing. Perangkat lunak ini dibangun bertujuan untuk memudahkan dalam pemesanan suku cadang dan memonitor transaksi dan ketersediaan suku cadang. Hasil pengujian yang didapatkan bahwa intergasi antar modul : model database, controller dan user interface berjalan sesuai yang diharapkan hanya saja end session login mengalami kegagalan kembali ke halaman awal jika session login melewati batas waktu lebih dari 24 jam*

**Kata kunci**— MVC, Pergudangan, RUP, Web

## **Abstract**

*The administrative system of warehousing is one form of management tatakelola warehouse in order for the goods stored therein well-documented. Warehousing systems administration is also applied in the company of countries like pt. PLN that is used to record the out the inclusion of parts used and the availability of spare parts required. Warehousing systems administration software can be applied in various platforms of either desktop or web-based. This software developed using web-based development cycle using the RUP method as a method of system development. Software developed using the MVC model and testing system testing done using blackbox testing. The software is built for easy aiming in ordering parts and monitor transactions and availability of spare parts. The test results suggested that the intergasi between modules: database model, controller and user interface runs as expected just end the session login failure back to the home page if the login session past the deadline for more than 24 hours.*

**Keywords**— MVC, RUP, Warehousing, Web

## **1. PENDAHULUAN**

PT. PLN (Persero) merupakan salah satu perusahaan energi non pertambangan yang bergerak di bidang energi listrik yang bertanggung jawab terhadap ketersediaan listrik nasional. Untuk menciptakan ketersediaan listrik secara luas, perusahaan harus menyediakan banyak pembangkit listrik dari berbagai jenis pembangkit listrik. Mulai dari pembangkit listrik bertenaga uap, pembangkit listrik bertenaga gas, pembangkit listrik bertenaga air dan masih banyak lagi. Semakin banyaknya pembangkit yang dibangun, maka akan semakin sulit perusahaan untuk mengelola kendali persediaan suku cadang (*part*) setiap pembangkit tersebut. Solusi yang digunakan adalah mengupayakan suatu sektor kerja agar bisa memajemen setiap pembangkit yang dikelola. Setiap sektor mengelola beberapa unit pembangkit. Pembangkit listrik tenaga disel unit Sungai Jauro dan pembangkit listrik tenaga gas Merah Mata merupakan unit pembangkit listrik dibawah pengawasan sektor Kramasan, Kertapati Palembang. Setiap

tugas dan tanggung jawab sektor ini diawasi oleh kantor cabang yaitu kantor cabang Palembang dan Sumatera Selatan, Jambi dan Bengkulu (S2JB).

Semakin banyaknya unit pembangkit mengakibatkan kebutuhan *part* pembangkit juga harus selalu tersedia untuk menjaga kestabilan produksi listrik, khususnya di area Palembang. Hal ini diperlukan agar tidak sering terjadi pemadaman dikarenakan adanya kerusakan pembangkit yang membutuhkan waktu lama karena harus melakukan perbaikan atau penggantian *part*. Semakin banyaknya *part* yang dibutuhkan, maka diperlukan suatu administrasi yang baik dalam melakukan pengelolaan *part* tersebut yang tersimpan diberbagai lokasi gudang yang dikelola oleh perusahaan. Sistem administrasi tata kelola *part* yang tersimpan di berbagai lokasi gudang PT. PLN (Persero) pada dasarnya sudah cukup baik. Dimana setiap *part* didata dalam bentuk dokumen digital. Hanya saja mekanisme kerja yang dilakukan saat ini tidak terintegrasi satu sama lain, seperti data masuk dibuat terpisah dengan data stok gudang dan sistem pengajuan permohonan penggunaan *part* diajukan secara tertulis yang membutuhkan waktu dalam pengajuan hingga *part* diambil

Berdasarkan hal tersebut perlu adanya perubahan sistem yang sebelumnya berjalan semi digital diubah menjadi *full* digital. Mulai data masuk barang hingga pengajuan permohonan barang dan pelaporan disajikan dalam satu sistem yang terintegrasi dan pengajuan permohonan *part* bisa dilakukan dari *site project* pembangkit ke gudang dimana *part* tersimpan. Sistem kerja seperti ini bisa terwujud dengan menerapkan sistem informasi yang tersebar di jaringan internal perusahaan yang memungkinkan setiap unit pembangkit atau sektor pembangkit mampu mengakses dan melakukan pengecekan ketersediaan *part* serta permohonan penggunaan *part* yang tersedia, sehingga admin gudang hanya memeriksa dan memvalidasi permohonan penggunaan *part* serta penyetujuan permohonan bisa di lihat oleh petinggi dari masing-masing divisi yang melakukan transaksi.

Sistem informasi sudah banyak digunakan dalam berbagai kegiatan yang difungsikan sebagai penunjang kerja dari setiap perusahaan yang menggunakannya, dikarenakan kemudahan dalam pengendalian, pengawasan dan pelaporan yang diberikan oleh sistem tersebut. Sistem informasi (sisfo) bisa berisikan informasi umum baik berupa informasi pendistribusian bantuan logistik seperti yang digunakan oleh Badan Penanggulangan Bencana Daerah Yogyakarta yang menyajikan rute dan jarak tempuh dan kebutuhan logistik sehingga informasi yang diberikan lebih detail, efektif dan efisien [1]. Sistem informasi tidak hanya berisikan informasi umum seperti informasi kelulusan ataupun informasi penjualan. Sisfo juga bisa digunakan sebagai sistem informasi tata kelola pergudangan yang tidak hanya digunakan oleh perusahaan besar saja tetapi sistem informasi juga digunakan pada perusahaan retail seperti alfamart dimana sistem informasi digunakan sebagai acuan dalam ketersediaan barang yang selalu berubah-ubah mengikuti kebutuhan konsumen sehingga meminimalisis ketersediaan barang yang tidak terpakai yang mengakibatkan melebihi kapasitas gudang dan kerugian perusahaan [2].

Sistem informasi juga digunakan pada perusahaan penyedia jasa makanan seperti pada perusahaan jasa boga yang bergerak dibidang jasa penyediaan makanan bagi perusahaan pesawat udara. Sistem informasi *inventory* digunakan untuk mengontrol setiap ketersediaan makanan yang masuk dan keluar hingga sistem transaksi yang melibatkan banyak perusahaan udara yang memesan makanan. Sehingga memberikan kemudahan dalam melakukan transaksi dan pengontrolan ketersediaan barang [3]. Sistem informasi khususnya sistem inventori juga berguna bagi perusahaan otomotif, dimana sistem ini bisa memberikan kemudahan dalam melakukan tata kelola ketersediaan barang dan memudahkan bagi pihak marketing dalam menganalisis produk mana saja yang memiliki peningkatan nilai jual berdasarkan perilaku konsumen dalam hal memilih produk [4].

Sistem informasi pergudangan juga digunakan oleh perusahaan pertamina dalam mengelola produk bukan bahan bakar minyak yang meliputi ketersediaan pelumas dan gas yang dikelola oleh PT. Pertamina VII depot Bitung. Sistem informasi sangat membantu perusahaan ini dalam mengendalikan persediaan barang dan persiapan produk bebas bahan bakar untuk meningkatkan kinerja perusahaan [5].

Untuk membangun suatu sistem informasi, ketepatan suatu rancangan sisfo dapat menjadi acuan atau dasar untuk membuat aplikasi tentu saja dengan data *warehouse* yang jelas dan detail berdasarkan analisa yang telah dilakukan dan didapatkan sebelumnya pada fase awal sebelum perancangan [6]. Sistem informasi kebanyakan dikembangkan atau dibangun menggunakan bahasa pemrograman web yang bisa diakses oleh banyak orang melalui halaman web. Sistem informasi web digunakan karena

kemudahan akses dan tampilan yang jelas untuk bisa dilihat dengan menyediakan sajian data yang lengkap dan mudah untuk dioperasikan. Sistem informasi berbasis web memudahkan pengembang untuk memutakhirkan sajian data jika terjadi penambahan-penambahan data yang diperlukan [7].

Sistem informasi tidak hanya berfokus pada sistem komputer saja seiring perkembangan zaman, teknologi pun terus berkembang dimana sistem informasi digital harus sudah ada dalam genggaman. Sistem informasi pergudangan sekarang sudah bisa terintegrasi ke dalam media *smartphone* dengan menggunakan basis sistem operasi tertentu yang mendukung bahasa pemrograman. Penerapan sistem informasi pada suatu *smartphone* bertujuan agar data yang keluar dan masuk tetap bisa dimonitor walaupun admin tidak berada di tempat kerja, tetapi admin bisa mengetahui barang apa saja yang ditambahkan di gudang dan yang dikeluarkan dari gudang [8]. Hal tersebut memudahkan atasan untuk memonitor transaksi yang terjadi di dalam gudang.

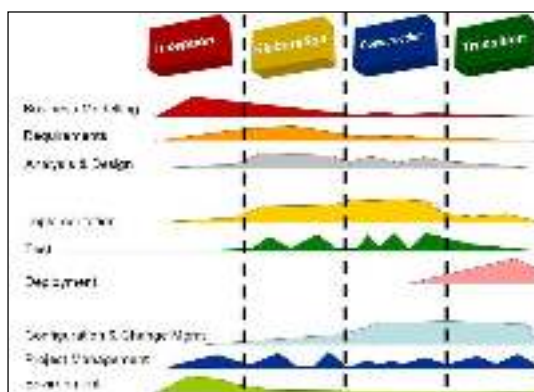
Dalam mengembangkan sistem informasi bisa diterapkan dengan menggunakan *framework* ataupun tanpa *framework* serta menggunakan dukungan arsitektur *Model View Controller* (MVC). *Framework* MVC digunakan dalam mengelola dan mengalokasi serta memantau suatu ruangan secara *realtime* pada perangkat lunak *monitoring* ruangan [9]. *Framework* MVC juga digunakan dalam mengembangkan pengolahan data dari sistem pengolahan yang sudah ada ke sistem yang lebih baik dengan memanfaatkan basis pemrograman web dengan menggunakan konsep UML pada studi kasus data *customer base* dan market share PT. Telkomsel [10].

Berdasarkan sumber diatas bisa disimpulkan bahwa sistem informasi tidak hanya sebatas pada informasi umum yang ditujukan pada sistem administrasi pada umumnya tetapi sistem informasi juga bisa diterapkan pada administrasi pergudangan yang berfungsi untuk mengawasi transaksi data gudang. Sistem informasi tidak hanya diterapkan pada basis web tetapi juga bisa diintegrasikan dengan sistem *mobile* dimana setiap orang yang terhubung di dalam sistem tersebut bisa mengetahui kegiatan apa saja yang dilakukan dalam pengolahan data gudang.

Penelitian ini bertujuan mengembangkan sistem administrasi pergudangan yang saat ini masih bersifat semi digital dan tidak efisien menjadi *full* digital dalam melakukan manajemen pergudangan. Hasil penelitian ini diharapkan mampu memberikan kemudahan dalam melakukan manajemen, pengontrolan dan pengawasan data gudang. Pengembangan perangkat lunak dalam penelitian ini menggunakan pendekatan RUP dan *framework* MVC digunakan sebagai acuan dalam pembangunan atau *developing* perangkat lunak yang digunakan. Menilai sistem administrasi ini layak digunakan atau tidak dalam manajemen gudang, maka dilakukan serangkaian uji antar *module* menggunakan pengujian *blackbox*.

## 2. METODE PENELITIAN

Metode yang digunakan dalam penelitian ini menggunakan metode *Rational Unified Process* (RUP). Penulis menggunakan RUP karena metode ini menggunakan kumpulan informasi best practises dalam pengembangan perangkat lunak, berorientasi pada objek dan menggunakan use-case driven serta fokus aktifitas pengembangan model perangkat lunak menggunakan *Unified Model Language* (UML) [11]



Gambar 1. Phase Rational Unified Process

Metode RUP memiliki 4 fase dalam pelaksanaannya, empat fase tersebut terdiri dari fase *inception*, *elaboration*, *contruction* dan *transition*. Rincian yang dilakukan dalam penelitian ini berdasarkan fase yang teradapat pada metode RUP antara lain :

1. *Inception* (fase permulaan)

Dalam fase ini adalah pendefinisian batasan-batasan kegiatan yang meliputi analisis kebutuhan user dimana dalam siklus yang berjalan saat ini, para petugas mekanik memohon pergantian *part* ke supervisor, dari supervisor form pengajuan *part* diserahkan ke petugas gudang, petugas gudang akan memeriksa apakah *part* yang dibutuhkan tersedia atau tidak, jika tidak ada petugas gudang memberikan informasi ke supervisor bahwa *part* tidak tersedia. Ketidak efisiensinya waktu dalam siklus yang berjalan saat ini adalah mekanik tidak bisa memeriksa secara *realtime part* tersedia atau tidak sebelum dilakukan pengajuan pergantian *part*. Melakukan perancangan kebutuhan *user* ditahap awal perangkat lunak (perancangan arsitektur dan *use case* diagram sistem)

2. *Elaboration* (fase perencanaan)

Dalam *fase elaboration* melakukan perancangan perangkat lunak yang meliputi *user interface* (desain *user interface* dibuat dalam bentuk sketsa dan ditampilkan dalam bentuk sebenarnya pada fase berikutnya), desain database yang akan digunakan sebagai basis data sistem informasi pergudangan. Fitur-fitur dalam sistem ini antara lain : pemohon hanya bisa melihat ketersediaan *part* dan melakukan permohonan *part*, sedangkan petugas gudang dan kepala gudang bisa menggunakan seluruh fitur yang disediakan pada sistem mulai dari penambahan data *part* hingga pelaporan data transaksi.

3. *Construction* (fase konstruksi)

Dalam fase *construction* dilakukan pengimplementasian atau pembangunan perangkat lunak yang sudah direncanakan dari awal secara detail. Pada fase ini sistem informasi dibangun menggunakan *Model View Controller* (MVC), dimana *controller* digunakan sebagai kendali untuk menciptakan aktifitas *Create, Read, Update dan Delete* (CRUD). Sedangkan *Model* digunakan sebagai koneksi antara database dan perangkat lunak. *View* digunakan untuk membuat tampilan *user interface* yang berhadapan langsung dengan pengguna, *view* dibangun menggunakan pendekatan blade pada pola pemrograman php.

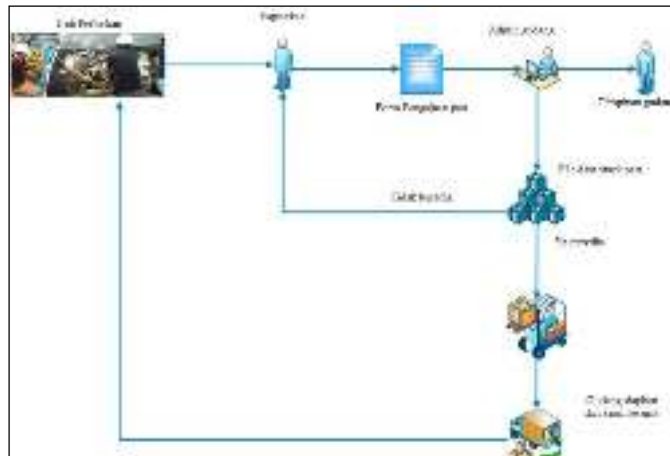
4. *Transition* (fase transisi)

Dalam tahap ini perangkat lunak yang dibangun kemudian dilakukan pengujian antar modul, konsistensi *controller* terhadap *model* yang dipanggil dan *view* atau *user interface* yang digunakan. Pengujian sistem ini menggunakan pengujian *blackbox testing*. Dari hasil pengujian yang didapatkan kemudian diterbitkan rekomendasi untuk bisa disosialisasikan ke pengguna untuk melihat respon pengguna apakah sistem ini membantu dalam melakukan manajemen atau administrasi pergudangan atau tidak membantu sama sekali

### 3. HASIL DAN PEMBAHASAN

#### 3.1 Alur Pengajuan Suku Cadang

Pada Gambar 1 memperlihatkan alur kerja dalam melakukan permohonan *part* yang berjalan pada saat ini yang merupakan tahapan awal dari RUP yaitu fase *inception*



**Gambar 2. Alur Pengajuan Suku Cadang**

Penjelasan dari Gambar 1 tersebut, yaitu : mekanik melaporkan *part* yang harus di ganti ke *supervisor*. *Supervisor* membuat formulir pengajuan pergantian *part* ke admin gudang. Admin gudang memeriksa ketersediaan *part*, jika tidak tersedia maka perbaikan dihentikan. Jika *part* tersedia akan diarahkan ke gudang untuk mengeluarkan *part* yang diajukan dan dikirim ke unit yang diperbaiki.

### 3.2 Alur Pengajuan Suku Cadang

Perancangan *blue print* sistem administrasi pergudangan mengacu pada kebutuhan pengguna yang terlibat di dalam sistem yang didefinisikan pada Tabel 1. Pengguna sistem di definisikan terdiri dari 4 aktor yaitu admin, petugas gudang, *member* (pemohon) dan pimpinan gudang yang memiliki fungsi yang sama dengan petugas gudang. Tabel 1 ini merujuk pada *fase inception* dalam RUP

**Tabel 1. Definisi Aktor Sistem Administrasi Pergudangan**

No	Aktor	Deskripsi
1	Admin	Melakukan manipulasi <i>Create, Read, Update, Delete</i> (CRUD) terhadap semua konten yang tersedia didalam perangkat lunak termasuk pemberian hak akses atau <i>privilage user</i> yang masuk dalam kategori member, petugas gudang dan pimpinan gudang.
2	Petugas Gudang	Melakukan CRUD terhadap fasilitas yang sudah ditentukan sebelumnya oleh admin berdasarkan kategorinya yaitu melakukan pendataan posisi gudang, penambahan stok <i>part</i> , menampilkan data-data pemohon <i>part</i> dan pembuatan laporan-laporan.
3	Member	melakukan pengajuan <i>part</i> , sebelum melakukan pengajuan maka member harus melihat ketersediaan stok yang tersedia di gudang.
4	Pimpinan Gudang	Melakukan pemantauan kegiatan pengeluaran dan penambahan stok data gudang.

### 3.3 Use Case Diagram

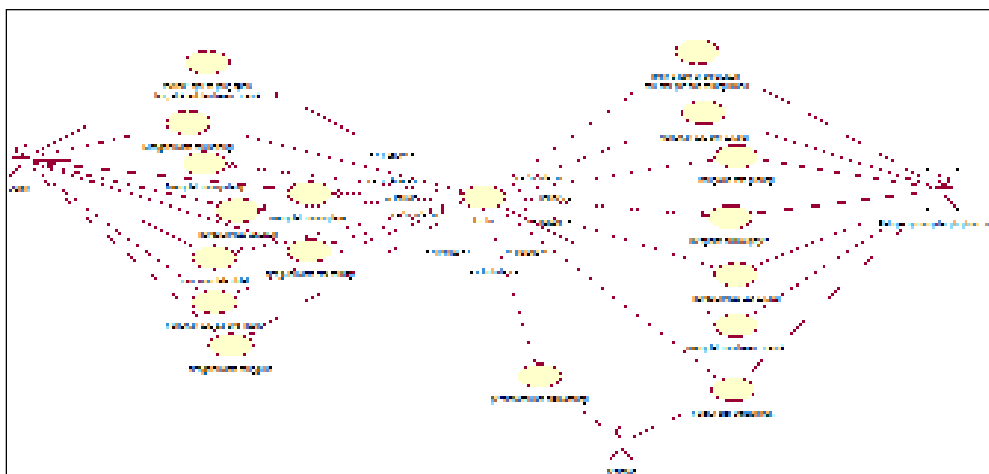
Tabel 2 merupakan penjelasan dari setiap aktor yang terlibat di dalam sistem dalam penelitian ini yang dikembangkan dari fase *inseption* yang didapatkan sebelumnya menjadi dasar dari *fase elaboration* dalam tahapan RUP yang disajikan pada Gambar 2

**Tabel 2. Definisi Use Case Sistem Informasi Administrasi Gudang**

No	Aktor	Use Case	Deskripsi
1	Admin	Pengolahan data bon barang	<i>Use case</i> ini terdiri dari menambah, mengubah dan menghapus data bon barang
2		Pengolahan data pengguna	<i>Use case</i> ini terdiri dari penambahan, mengubah dan menghapus kategori pengguna sistem yang terdiri dari <i>warehouse officer</i> , pimpinan divisi, member serta <i>admin system</i>

3		Mengolah data gudang	<i>Use case</i> ini terdiri dari penambahan, perubahan dan penghapusan data lokasi gudang.
4		Mengolah data suplayer	<i>Use case</i> ini terdiri dari penambahan, perubahan dan penghapusan data suplayer
5		Melihat data laporan-laporan	<i>Use case</i> ini hanya berupa menampilkan data laporan dari tiap-tiap <i>use case</i> yang ada pada sistem.
6		Mengolah data karyawan	<i>Use case</i> ini terdiri dari penambahan, penghapusan dan perubahan data karyawan
7		Mengolah data divisi	<i>Use case</i> ini terdiri dari perubahan, penambahan dan penghapusan data divisi.
8		Mengolah data barang masuk	<i>Use case</i> ini terdiri dari penambahan, perubahan dan penghapusan barang masuk
9		Mengolah data stok barang	<i>Use case</i> ini hanya menampilkan data stok <i>part</i> terakhir .
10	Admin, petugas gudang, <i>member</i> dan pimpinan	Login	<i>Use case</i> ini merupakan proses pemverifikasian <i>username</i> dan <i>password</i> dari masing-masing <i>user</i> .
11	Member, petugas gudang dan pimpinan	Melihat ketersediaan <i>part</i>	<i>Use case</i> ini merupakan proses untuk melihat ketersediaan <i>part</i> yang tersedia di gudang.
12	Pemohon atau <i>member</i>	Permohonan bon <i>part</i>	<i>Use case</i> ini berisikan penambahan permohonan pengajuan bon <i>part</i> kepada petugas gudang atau pimpinan divisi gudang.
13	Petugas gudang dan pimpinan gudang	Mengolah data barang masuk	<i>Use case</i> ini terdiri dari penambahan, perubahan dan penghapusan barang masuk.
14		Mengolah data stok barang	<i>Use case</i> ini hanya menampilkan stok barang terakhir.
15		Mengolah data suplayer	<i>Use case</i> ini terdiri dari penambahan, perubahan dan penghapusan data <i>suplayer</i> .
16		Mengolah data gudang	<i>Use case</i> ini terdiri dari penambahan, perubahan dan penghapusan data lokasi gudang.
17		Mengolah data bon barang	<i>Use case</i> ini terdiri dari perubahan dan penghapusan data bon barang yang diajukan oleh <i>user</i> .
18		Melihat laporan	<i>Use case</i> ini hanya menampilkan data laporan-laporan meliputi laporan pengajuan bon hingga stok terakhir dari <i>part</i> yang tersedia di gudang.

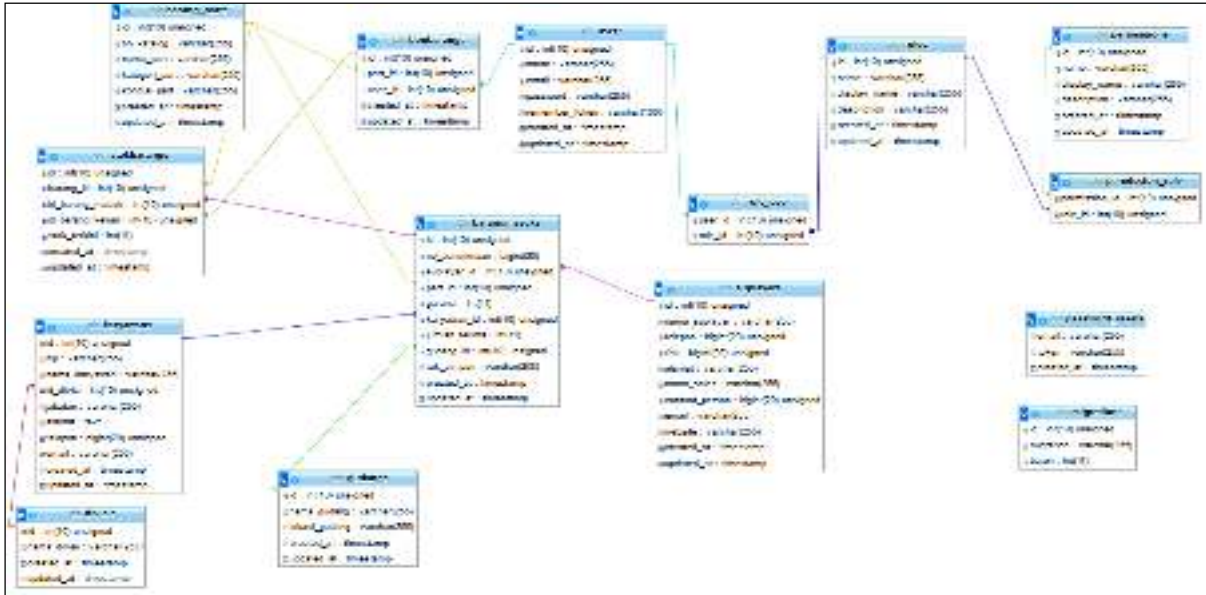
Pada Gambar 3 menunjukkan *use case diagram* yang merupakan hasil representasi dari definisi aktor yang telah disebutkan pada table 2 yang juga merujuk pada *fase elaboration* dalam tahapan RUP.



Gambar 3. *Use Case Diagram* Sistem

### 3.4 Desain Database

Adapun desain database yang digunakan dalam penerapan perangkat lunak sistem informasi pergudangan tampak pada Gambar 4. Gambar 4 merujuk pada fase perencanaan (*elaboration*) dalam tahapan RUP.



Gambar 4. Desain Database Sistem Informasi Pergudangan

### 3.5 Desain Database

Dalam perangkat lunak yang dikembangkan menggunakan pendekatan *Model View* dan *Controller* (MVC) yang masuk dalam *fase construction*. *User interface* yang disajikan pada Gambar 5 sampai dengan Gambar 8 merupakan hasil *view* yang bisa dilihat oleh *user*. Hasil *view* telah terintegrasi dengan model yang memegang kendali koneksi masing-masing tabel dari database yang telah ditampilkan pada Gambar 4, serta *view* juga telah terintegrasi dengan *controller* dalam melakukan kegiatan atau fungsi *create*, *read*, *update* dan *delete* (CRUD). *View* atau *user interface* yang tersedia pada sistem terdiri dari 4 kategori pengguna sesuai dengan perencanaan awal seperti tampak pada Gambar 3, antara lain :

#### 1. *User guest*

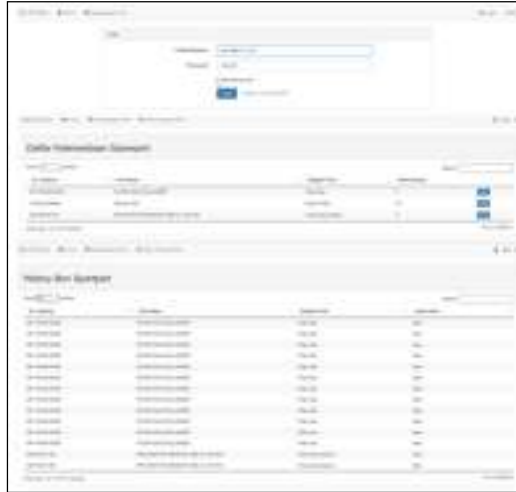
Fasilitas yang diberikan pada *user guest* adalah *home* dan menu ketersediaan *part* yang dimiliki dan tersimpan dalam gudang. Pada fasilitas ini *user guest* bisa melakukan *order part*. *order part* bisa dilakukan jika *user guest* tersebut sudah memiliki data *user* tetap yang masuk kategori *member*. Jika *user guest* tersebut tidak memiliki data keanggotaan sebelumnya di dalam sistem, maka *user guest* tersebut harus mendaftarkan diri ke dalam sistem sebelum melakukan *order part*. *View* atau *user interface order part* pada *user guest* disajikan pada Gambar 5.



Gambar 5. User interface user Guest

## 2. *Member* atau user pengguna

*User* yang terdaftar merupakan reposisi dari *user guest* menjadi *member*. Fasilitas yang disediakan untuk *member* tidak jauh berbeda dengan *user guest* yaitu menu *home*, ketersediaan *part* dan *history* dari *part* yang sudah di *order* oleh *member* tersebut. Yang masuk dalam kategori *member* adalah karyawan pelaksana (mekanik atau kepala mekanik). *User interface* yang diberikan kepada *user member* tampak pada Gambar 6.



**Gambar 6.** *User interface user member*

## 3. *Warehouse officer* dan pimpinan gudang

*User* yang masuk dalam kategori *warehouse officer* memiliki fitur yang lebih lengkap di bandingkan *user member* dimana pada kategori *user* ini memiliki fasilitas untuk memajemen lokasi gudang, supplay, manajemen data *sparepart*, *request* bon, manajemen data barang masuk, laporan. *View* atau *user interface* dari *warehouse officer* disajikan pada Gambar 7.



**Gambar 7.** *User interface Warehouse Officer*

## 4. *Admin System*

*Admin system* dalam perangkat lunak ini memiliki fasilitas menu yang lebih lengkap jika dibandingkan dengan kategori *user* yang lain. *Admin system* memiliki fasilitas menu yang dimiliki oleh *user warehouse officer* dan dilengkapi dengan manajemen data pengguna, mulai dari penambahan karyawan yang menjadi *resource* awal pengguna sistem. Tampak pada Gambar 8.





**Gambar 8. User Interface Admin System Warehouse**

*User interface* yang disajikan pada Gambar 5 sampai dengan Gambar 8 merupakan penerapan dari *user interface* yang terintegrasi dengan database (model) dan *controller* yang telah dijelaskan sebelumnya pada *fase elaboration* yang dikembangkan dengan menggunakan model MVC pada *fase construction* dalam metode RUP yang digunakan.

### 3.6 Pengujian Sistem

Tabel 3 menjelaskan pengujian antar modul dari perangkat lunak yang digunakan. Pengujian ini untuk melihat konsisten dan integritas antar modul dari perangkat lunak yang digunakan. Pengujian antar modul ini merujuk pada framework MVC yang digunakan dimana menguji konsistensi dan koneksi *model* terhadap *view* dan *controller*. Tahapan pengujian sistem ini merujuk pada *fase transition* dalam metode RUP yang digunakan sebelum diterapkan ke dalam sistem jaringan internal perusahaan.

**Tabel 3. Hasil pengujian perangkat lunak sistem informasi administrasi pergudangan**

No	Schema	Model	Controller	View	hasil
1	Login	AutenticatesUsers.php	LoginController.php	login.blade.php	True
2	Register	RegistersUsers.php	RegisterController.php	Register.blade.php	True
3	Divisi	Divisi.php	DivisiController.php	divisis.index.blade.php divisis.create.blade.php divisis.edit.blade.php	True
4	Karyawan	Karyawan.php	KaryawanController.php	karyawans.index.blade.php karyawan.create.blade.php karyawan.edit.blade.php	True
5	Pengguna sistem	User.php	UserController.php	penggunasistems.index.blade.php penggunasistems.create.blade.php penggunasistems.edit.blade.php	True
6	Status keanggotaan	Role.php	RolesController.php	roleusers.index.blade.php roleusers.create.blade.php roleusers.edit.blade.php	True
7	Gudang	Gudang.php	GudangsController.php	gudangs.index.blade.php gudangs.create.blade.php gudangs.edit.blade.php	True
8	Supplayer	Supplayer.php	SupplayerController	supplayers.index.blade.php supplayers.create.blade.php supplayers.edit.blade.php	True
9	Sparepart	Barang_parts.php	Barang_partsController.php	spareparts.index.blade.php spareparts.create.blade.php spareparts.edit.blade.php	True

10	Barangmasuk	Barangmasuk.php	BarangmasuksController.php	barangmasuks.index.blade.php barangmasuks.create.blade.php barangmasuks.edit.blade.php	True
11	Bonbarang	Bonbarang.php	BonbarangsController.php	bonbarangs.index.blade.php bonbarangs.create.blade.php bonbarangs.edit.blade.php	True
12	Stokbarang	Stokbarang.php	StokbarangController.php	Stocks.index.blade.php	true
13	Laporan	Laporan.php	ReportsController.php	Laporans.stokbarang.blade.php Laporans.barangmasuk.blade.php Laporans.barangkeluar.blade.php Laporans.bonbarang.blade.php Laporans.suplayer.blade.php	True
14	Historybon	Bonbarang.php	HistoryController.php	members.index.blade.php	True
15	Ketersediaan part	Stock.php	GuestController.php	guests.index.blade.php	True

Hasil dari serangkaian pengujian yang telah dilakukan dinyatakan bahwa integrasi antar *model*, *view* dan *controller* atau modul serta fungsi yang telah di buat berdasarkan keterangan file-file yang tertera pada Tabel 3 dinyatakan berfungsi dengan baik sesuai yang diharapkan di awal perancangan. *Session login* mengalami kegagalan kembali ke halaman awal pada saat *logout* jika *user* melakukan kondisi *idle* tanpa aktifitas selama kurun waktu 24 jam atau dalam kurun waktu tertentu. Token yang digunakan sebagai enkripsi *key* memuat *time server* yang digunakan dalam proses enkripsi. Dan *token key* akan berubah-ubah setiap harinya pada saat *user* melakukan aktifitas login.

Adapun tahapan yang dilakukan dalam pengembangan sistem dalam penelitian ini mengikuti fase-fase dalam metode RUP yang dijelaskan pada Tabel 4.

**Tabel 4. Penjelasan Tahapan Pengembangan Sistem Terhadap Fase RUP**

	Tahapan-Tahapan dalam RUP			
	<i>fase inception</i>	<i>fase elaboration</i>	<i>fase construction</i>	<i>fase transition</i>
tahapan pengembangan perangkat lunak	alur pengajuan <i>part</i> dan desain skenario aktor	use case diagram dan desain database	user interface	pengujian sistem

#### 4. KESIMPULAN

Berdasarkan hasil pengujian yang dilakukan pada perangkat lunak yang dikembangkan dihasilkan antara lain :

1. Hasil pengujian yang dilakukan sebelum sistem diterapkan di dalam sistem perusahaan didapatkan bahwa sistem yang dibuat sesuai dengan yang diharapkan sebelumnya. Perangkat lunak yang dikembangkan bisa memberikan informasi secara *real time* tentang stok *part* yang tersedia ataupun tidak tersedia, sehingga pengguna (mekanik) bisa melihat ketersediaan *part* dan admin gudang bisa memonitor *part* mana yang harus di *restock* maupun dilakukan pengajuan untuk pembelian *part* ke pihak terkait.
2. Pimpinan gudang dan *supervisor* bisa memonitor pergerakan dari *part* yang digunakan sehingga memudahkan pihak terkait dalam melakukan pengawasan.
3. Setiap tahapan-tahapan dalam pengembangan perangkat lunak ini sesuai dengan tahapan pada metode pengembangan sistem yaitu RUP dan *framework MVC* yang digunakan. Dimana *fase inception* memuat data alur pengajuan *part* dan desain skenario aktor, *fase elaboration* memuat data *use case* diagram dan desain database, *fase construction* memuat *user interface* pengguna sistem serta *fase transition* memuat data pengujian sistem.
4. Integrasi antar *model view controller* yang bangun menggunakan *framework MVC* dinyatakan berjalan dengan sangat baik sesuai dengan yang diharapkan.

5. Seluruh modul berjalan dengan baik tetapi hanya ada 1 permasalahan yang dinyatakan gagal yaitu kondisi *logout* pada saat masa *user* melakukan *idle time* lebih dari 1 hari atau lebih dari batas waktu yang sudah ditetapkan. Dimana seharusnya pada saat *user* melakukan *logout* ataupun *refresh link* akan kembali ke *login*. Hal ini kemungkinan disebabkan oleh *regenerate session token* yang digunakan pada *framework* yang digunakan.
6. Perangkat lunak yang dibangun di uji coba pada sistem yang berdiri sendiri atau *stand alone* bukan yang terintegrasi dengan sistem jaringan internal perusahaan.
7. Perangkat lunak hanya menggunakan satu sistem autentikasi yaitu autentikasi login.

## 5. SARAN

Perangkat lunak yang dikembangkan ini masih memiliki kelemahan dan kekurangan yang diharapkan bisa disempurnakan pada pengembangan berikutnya. Perangkat lunak yang dibangun menggunakan basis web. Diharapkan adanya penerapan infrastruktur jaringan yang bisa mendistribusikan perangkat lunak ini ke seluruh sektor atau unit produksi ataupun unit pembangkit di internal perusahaan. Hal ini terjadi dikarenakan setiap sektor mengelola banyak unit pembangkit diharapkan media server yang akan digunakan dalam penerapan perangkat lunak ini menggunakan pendekatan multi server. Dimana *user interface* terpisah dengan *database* menghindari kerusakan pada sistem yang digunakan yang dapat berpotensi terhadap kehilangan data.

## DAFTAR PUSTAKA

- [1] Sanjani, Arif Lukman, dkk. 2014. Rancang Bangun Sistem Informasi Penggajian Pegawai dan Remunerasi Jasa Medis pada Rumah Sakit Bedah Surabaya. JSIKA Vol. 3, No. 1 (2014), hal 87-93. ISSN: 2338-137x.
- [2] Puryanto, Tono dan Sutikno. 2016. Sistem Perencanaan Penambahan Stok Barang Menggunakan Metode Fuzzy C-Means dan Fuzzy Sukamoto (Studi Kasus di Distributor Alfamart Semarang). TEKNOLOGI, Vol. 02, No. 02, Hal. 43-52, ISSN : 2476-8812
- [3] Sudana, Oka. A.A.K. 2007. Sistem informasi Manajemen Inventori pada Perusahaan Layanan Jasaboga Pesawat Udara. Majalah ilmiah Teknik Elektro. Vol. 6, No. 1. Hal.13-19. ISSN: 2503-2372
- [4] Fanani, Zainul dan Faisal Muhammad. 2014. Rancang Bangun Sistem informasi Inventory Menggunakan Metode Association Rules di CV Damar Langit. Jurnal Matics, Vol.4. No. 2. Hal. 85-99. ISSN: 2477-2550
- [5] Rorimpandey, Gladly. G., dan Kembuan, Olivia. 2016. Analisa dan Perancangan Sistem Informasi Inventori Produk non-BBM PT. Pertamina (Persero). Informal, Vol. 1, No. 1, Hal. 14-24. ISSN : 2503-250X.
- [6] Astuti, Dwi Puspita. 2011. Sistem Informasi Penjualan Obat pada Apotek Jati Farma Arjosari. Jurnal Sentra Penelitian Engineering dan Edukasi (Jurnal Speed) vol. 3 No. 4. Hal 34-39. ISSN : 1979-9330, ISSN Online : 2088-0154.
- [7] Shah, Umm-e-Mariya., Shamim, Azra., dan Kazmi Madiha. 2011. Enhanced Architecture of a Web Warehouse based on Quality Evaluation Framework to Incorporate Quality Aspects in Web Warehouse Creation. International Journal of Advanced Computer Science and Applications (IJACSA), Vol. 2, No. 8. Hal. 55-59. ISSN : 2156-5570
- [8] Triwibowo, Dodi., Kridalukmana, Rinta., dan Martono, Kurniawan Teguh. 2015. Pembuatan Aplikasi Terintegrasi, Pendataan Barang di Gudang Berbasis Android. JTSiskom Vol. 3, No. 2, Hal. 320-334, ISSN : 2338-0403
- [9] Safriadi, Novi. 2012. Academic Resources Management Information System (ARMIS) dengan Arsitektur Model View Controller (MVC). Jurnal ELKHA Vol. 4 No. 1. Hal. 1-6. ISSN : 1858-1463.
- [10] Asri, Yessi. 2012. Analisis dan Perancangan Aplikasi Web Menggunakan Framework Yii sebagai Media Analisis Data Customer Base dan Market Share (Studi Kasus: PT. Telkomsel). Gerbang

JIST edisi Februari, Hal. 89-110. ISSN :0853-6376.

- [11] Triwahyuni, Atin dan Saputra, Novian. 2015. Architecture E-Mall Using RUP (Rational Unified Process) Methods. Cogito Smart Journal, Vol. 1, No. 1, Hal. 1-12, ISSN : 2541-2221.