

## Pembangunan Aplikasi Sistem Informasi Pergudangan pada Rumah Sakit Umum Daerah Dr. Murjani Sampit Kabupaten Kotawaringin Timur

Hidayatullah Agung Prasetyo<sup>1</sup>, Bayu Priyambadha<sup>2</sup>, Achmad Arwan<sup>3</sup>

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya  
Email: <sup>1</sup>hidayatullahap@gmail.com, <sup>2</sup>bayu\_priyambadha@ub.ac.id, <sup>3</sup>arwan@ub.ac.id

### Abstrak

Perkembangan teknologi informasi berkembang dengan sangat pesat dan permasalahan utamanya adalah bagaimana cara untuk mengolah data sehingga menghasilkan informasi yang berguna dan mudah digunakan oleh pengguna. Sistem manajemen pergudangan yang saat ini digunakan di Rumah Sakit Dr. Murjani Sampit masih manual. Pencatatan dilakukan dengan penulisan di buku dan file excel, dengan banyaknya barang keluar dan masuk maka pencatatan dan pencarian data menjadi tidak efisien dalam hal waktu dan tenaga. Pembebanan tugas terhadap seseorang yang melakukan sistem manual ini menyebabkan kemungkinan kesalahan penulisan dan pengolahan data. Dari permasalahan tersebut maka perlu dikembangkan sebuah aplikasi sistem informasi yang dapat menangani pencatatan, penyimpanan, pengolahan dan laporan barang secara otomatis sehingga memudahkan kerja serta mengurangi kesalahan pengolahan data. Penelitian ini dilakukan dengan metode *Iterative Life Cycle* dan sistem dibangun dengan menggunakan teknologi PHP, MySQL, dan Javascript. Pengujian *compatibility* menunjukkan hasil sistem berjalan baik di Google Chrome dan Mozilla Firefox kemudian pada pengujian *white-box* didapatkan hasil 100% valid dari 20 kasus uji dan dari pengujian *black-box testing* didapatkan hasil 100% valid dari 65 kasus uji. Hasil pengujian tersebut menunjukkan bahwa sistem berhasil dianalisis kebutuhannya dan diimplementasi sesuai dengan perancangan. Dari implementasi tersebut maka dapat mengatasi permasalahan sistem yang masih manual pada rumah sakit.

**Kata kunci:** *pergudangan, iterative life cycle, php, mysql, javascript.*

### Abstract

*The information technology development is growing rapidly and the main problem is how to process data to give useful information and easy to use by user. Warehouse management system that currently used on Dr. Murjani Sampit hospital is manual. Data recording done by write on books and excel files, with the number of in and out of goods makes record and search for data become inefficient in term of time and effort. Assigning tasks to someone who did this manual system led to possible errors in writing and data processing. From these problems it is necessary to develop an information system application that can handle recording, storing, processing and reporting automatically to make work easier and reduce data processing error. Research was done by Iterative Life Cycle and system built using PHP, MySQL, and Javascript technology. Compatibility testing showed system running well on Google Chrome and Firefox Mozilla later on white-box testing showed from 20 test cases 100% result is valid and from black-box testing obtained from 65 test cases 100% result is valid. The results of these tests indicate that the system successfully analyzed its needs and implemented according with the design. System implementation may resolve the manual system problem which occur at the hospital.*

**Keywords:** *warehousing, iterative life cycle, php, mysql, javascript.*

### 1. PENDAHULUAN

Perkembangan teknologi informasi berkembang dengan sangat pesat dan permasalahan utamanya adalah bagaimana cara untuk mengolah data sehingga menghasilkan

informasi yang berguna dan mudah digunakan oleh pengguna. Sistem manajemen pergudangan yang saat ini digunakan di Rumah Sakit Dr. Murjani Sampit masih manual dan dengan banyaknya barang keluar dan masuk maka pencatatan dan pencarian data menjadi tidak efisien dalam hal waktu dan tenaga. Pembebanan

tugas terhadap seseorang yang melakukan sistem manual ini menyebabkan kemungkinan kesalahan penulisan dan pengolahan data. Sehingga pada saat ini diperlukan suatu sistem manajemen pergudangan yang lebih terstruktur agar dapat mempermudah dan mempercepat pembuatan laporan barang masuk, barang keluar sesuai klasifikasi dan pencarian data yang ada.

Rumah Sakit Umum Daerah Dr. Murjani Sampit Kabupaten Kotawaringin Timur merupakan sebuah lembaga yang tugas utama dan fungsinya adalah melayani kesehatan masyarakat. Untuk menjalankan tugas ini sangatlah perlu pelayanan yang cepat dan efisien agar semua masyarakat dapat terlayani dengan baik. Semua unit rumah sakit harus terintegrasi dan proses antar unit dapat berjalan dengan baik. Pada umumnya unit unit di RSUD Dr. Murjani Pada unit pergudangan sistem yang digunakan masih manual dan hanya unit - unit tertentu di RSUD Dr. Murjani Sampit yang memiliki sistem otomatis. Unit pergudangan masih mengalami kendala pada proses keluar dan masuk barang. Proses pencatatan data dan pengarsipan masih menggunakan buku dan Microsoft Excel. Dengan pencatatan manual ini dan tidak adanya integrasi sistem maka pencarian arsip data barang yang diinginkan sangatlah susah. Proses manual ini juga menyebabkan proses pembuatan laporan memakan waktu yang lama. Tidak adanya sistem informasi pergudangan menyebabkan barang rusak dalam penyimpanan, rusak dalam handling, salah menyimpan barang, selisih stok dan lain lain. Hal-hal tersebut dapat merugikan rumah sakit secara fisik dan finansial.

Athoilah (2014) menjelaskan dengan adanya sistem informasi pergudangan memberikan keuntungan antara lain mempercepat proses, mengetahui semua transaksi dan jumlah stok dengan lebih cepat dan akurat, mempermudah mengatur lokasi penyimpanan barang, dan lain lain. Kemudian pada penelitian Gaspar et al (2011) menjelaskan dengan adanya sistem informasi pergudangan akan mempermudah dan mempercepat pekerjaan pegawai. Melihat penjelasan permasalahan di atas maka diperlukan suatu sistem yang dapat mengintegrasikan proses yang berada pada unit pergudangan RSUD Dr. Murjani Sampit. Dari permasalahan tersebut dapat disimpulkan pada unit pergudangan masih mengalami banyak kendala sehingga kurang optimal dalam menjalankan tugas.

## 2. KAJIAN PUSTAKA

Dalam penelitian yang dilakukan oleh Athoilah (2014) dengan judul “Android-Based Mobile Information System Design for Inventory Control of Warehouse” menjelaskan adanya sistem informasi pergudangan pegawai yang bekerja dengan mudah dapat memantau seluruh transaksi yang berlangsung pada gudang. Para pegawai juga dengan mudah melakukan pencatatan transaksi barang. Adanya sistem ini dapat membantu pekerjaan pegawai yang secara langsung dapat meningkatkan tingkat produktifitas para pegawai.

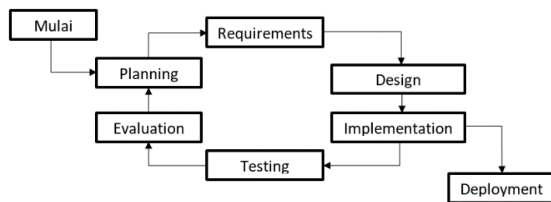
Pada penelitian Gaspar et al (2011) yang berjudul “Design and implementation of a client warehouse application over an enterprise resource planning system for mobile devices” menjelaskan dengan adanya sistem informasi pergudangan akan mempermudah dan mempercepat pekerjaan pegawai.

Arwan et al (2011) menjelaskan dengan menggunakan teknologi MySQL, aplikasi desktop perwalian STIKI berbasis client server menggunakan Borland Delphi 7 ini maka prosesan entri data perwalian mahasiswa STIKI menjadi lebih cepat.

Hasil penelitian Priyambadha (2010) et al menunjukan teknologi AJAX dalam membuat sebuah aplikasi web mail dapat menyelesaikan masalah reload halaman. Sehingga dengan teknologi tersebut sebuah aplikasi berbasis web akan lebih responsive.

Ada berbagai macam model SDLC yang mana didesain dan didefinisi untuk diikuti ketika proses pembuatan perangkat lunak. Setiap tahapan pada setiap model adalah unik dan berbebeda yang dibuat untuk memastikan kesuksesan dalam proses pengembangan perangkat lunak (Pressman, 2001).

Iterative model adalah pendekatan yang dimulai dengan implementasi sederhana dari sekumpulan kecil persyaratan perangkat lunak. Iterative model tidak berusaha untuk memulai sebuah pembuatan sistem dengan spesifikasi kebutuhan penuh namun perkembangan dirancang dan diimplementasi hanya sebagian dari sistem. Hasil dari tiap akhir iterasi kemudian ditinjau untuk mengidentifikasi kebutuhan sistem lebih lanjut yang terus meningkat hingga mendapatkan sistem yang lengkap (Miller et al, 1991).



**Gambar 1.** Iterative Model

Fase model Iterative adalah 1. *Planning & Requirements* 2. *Analysis and Design* 3. *Implementation* 4. *Testing* 5. *Evaluation*. Pada tulisannya Powel (2016) menjelaskan kelebihan iterative model adalah:

- *Inherit versioning*: Perangkat lunak akan memiliki beberapa versi yang menunjukkan tahap peluncuran aplikasi. *Versioning* memudahkan tiap iterasi yang lebih baru terus diperbaharui. Jika iterasi baru menghancurkan iterasi sebelumnya maka dapat dengan cepat di kembalikan dan di implementasi kembali untuk meminimalkan kerugian.
- Cocok untuk *Agile Organization*: Metode ini cocok untuk organisasi yang memerlukan kebutuhan yang berubah-ubah.
- *Easy adaptability*: Mudah beradaptasi antara kebutuhan dan sistem yang dibangun.

Dengan adanya kelebihan iterative model juga memiliki kekurangan antara lain:

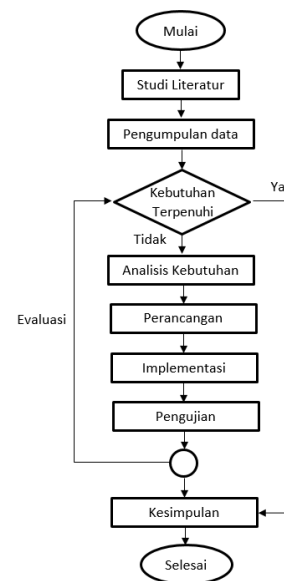
- *Costly Late-Stage Issue*: Jika pada tahap akhir terdapat permasalahan maka perbaikan untuk masalah tersebut akan mahal
- *Increased Pressure on User Engagement*: Metode ini perlu *feedback* dari *stakeholder* untuk melakukan iterasi berikutnya sehingga sangat perlu seringnya tatap muka dengan *stakeholder*.
- *Feature Creep*: Mungkin saja fitur yang di inginkan pengguna adalah bukan yang benar-benar diperlukan (Powel, 2016).

Dari penelitiannya Malhotra (2016) menyimpulkan bahwa kelemahan dari Iterative Model adalah perlunya keterlibatan stakeholder yang lebih dalam proses evaluasi. Dengan kurangnya keterlibatan stakeholder akan memperlambat dan kurang efisiennya pengerjaan suatu proyek.

### 3. METODOLOGI

Tahap ini akan menjelaskan langkah-langkah yang dilakukan dalam penelitian. Gambar 2 menunjukan alir metodologi penelitian detail alir tersebut dapat ada pada sub bab 3.1

hingga 3.6.



**Gambar 2.** Metodologi Penelitian

Karena penelitian ini menggunakan *Iterative SDLC* maka setelah pengujian siklus dapat kembali ke analisis kebutuhan menyesuaikan iterasi siklus perkembangan bila kebutuhan belum terpenuhi. Evaluasi dilakukan sebelum siklus berikutnya untuk mendapatkan kebutuhan pada iterasi selanjutnya. Setelah siklus berakhir maka kesimpulan dan saran disertakan sebagai catatan atas aplikasi dan kemungkinan arah pengembangan aplikasi selanjutnya.

#### 3.1 Pengumpulan Data

Studi literatur ini menjelaskan tinjauan teori yang digunakan sebagai penunjang penelitian ini. Teori-teori tersebut adalah sebagai berikut :

- Penelitian terdahulu
- Pengudangan
- Rekayasa Perangkat Lunak
- Unified Modeling Language
- Entity Relationship Diagram
- Apache
- PHP
- SQL
- XAMPP
- JavaScript
- Pengujian Perangkat Lunak

#### 3.2 Pengumpulan Data

Hal pertama yang dilakukan dalam analisis sistem adalah pengumpulan data. Pengumpulan data dilakukan dengan cara sebagai berikut:

1. Wawancara dilakukan melalui studi lapangan dan tanya jawab secara langsung ke Rumah Sakit Dr. Murjani Sampit. Wawancara

dengan pihak yang terkait yaitu bagian Pergudangan di Rumah Sakit tersebut.

2. Pada tahap observasi didapatkan hasil mengenai proses bisnis dalam unit Pergudangan dan permasalahan yang dihadapi pada unit tersebut.
3. Pengambilan dokumen yang diperlukan dalam pelaporan barang masuk dan barang keluar. Dokumen tambahan seperti form yang digunakan dalam pencatatan barang. Dari dokumen-dokumen tersebut didapatkanlah menakisme untuk melakukan manajemen pergudangan.

### 3.3 Analisis Kebutuhan

Analisis kebutuhan bertujuan untuk mendapatkan semua kebutuhan yang diperlukan dari sistem yang akan dibangun. Analisis kebutuhan dilakukan dengan mengidentifikasikan semua kebutuhan sistem dan siapa saja yang terlibat didalamnya. Analisis juga dilakukan untuk mengetahui kondisi lapangan yang ada.

Dengan adanya analisis maka dapat dilakukan perancangan sistem. Perancangan dibutuhkan untuk menjadi acuan implementasi sistem. Perancangan sistem ini berupa rancangan langkah – langkah kerja dalam pembuatan sistem.

### 3.4 Perancangan

Perancangan adalah tahap merancang sistem yang akan dibangun, tahap ini berupa rancangan langkah-langkah kerja dalam pembuatan sistem. Perancangan sistem dilakukan berdasarkan analisis kebutuhan. Perancangan antara lain adalah membuat use case diagram, skenario use case diagram, sequence diagram, package diagram, sequence diagram, database, dan perancangan antarmuka. Perancangan dibutuhkan untuk acuan pada implementasi sistem.

### 3.5 Implementasi

Pada Tahap implementasi, peneliti melakukan implementasi terhadap sistem yang akan dibangun. Implementasi ini didasarkan pada rancangan sistem yang telah dibuat pada tahap perancangan yaitu meliputi implementasi antar muka sistem dan implementasi database.

### 3.6 Pengujian

Pengujian perangkat lunak dilakukan untuk

mengetahui sejauh mana kinerja dan performa sistem yang telah dibuat. Apakah aplikasi sistem informasi pergudangan rumah sakit ini telah sesuai dengan spesifikasi kebutuhan yang direncanakan. Proses pengujian dilakukan dengan tiga metode yaitu *blackbox testing*, *whitebox testing*, dan *compatibility testing*.

### 3.7 Kesimpulan

Pengambilan kesimpulan diberikan jika semua tahapan telah selesai dilaksanakan. Mulai dari observasi awal, pengumpulan data, analisis kebutuhan, perancangan sistem, implementasi, pengujian dan analisis penelitian. Kesimpulan penelitian akan menjawab permasalahan penelitian. Hasil kesimpulan diberikan berdasarkan keseluruhan proses untuk menunjukkan apakah masalah penelitian telah terjawab dan tujuan tercapai. Kesimpulan juga memberikan saran untuk kepentingan pengembangan sistem.

## 4. PERANCANGAN

### 4.1 Identifikasi Ruang Lingkup

Deskripsi ruang lingkup merupakan tahapan awal yang ada pada RSUD Dr. Murjani Sampit. Tahap ini juga menjadi landasan untuk tahapan selanjutnya

#### 4.1.1 Studi Literatur

Ruang lingkup didasarkan pada proses keluar masuk barang yang ada di pergudangan RSUD Dr. Murjani Sampit. Gudang meliputi gudang inventaris dan gudang farmasi kemudian pelayanan yang ada di depo Rajal (Rawat Jalan).

#### 4.1.2 Sumber Daya Yang Terlibat

Berdasarkan permasalahan yang telah dirumuskan maka sumber daya yang terlibat di Rumah Sakit antara lain:

1. Petugas Loker adalah karyawan yang memasukan informasi pasien yang berobat.
2. Petugas Gudang Farmasi adalah karyawan pergudangan di gudang farmasi. Petugas ini bertanggung jawab untuk mendata barang masuk dan keluar yang ada di gudang farmasi serta mengirim barang farmasi ke unit yang meminta.
3. Petugas Gudang Inventaris adalah karyawan pergudangan di gudang inventaris. Petugas ini bertanggung jawab untuk mendata barang masuk dan keluar yang ada di gudang inventaris serta mengirim barang inventaris

ke unit yang meminta.

4. Petugas Depo Rajal adalah petugas yang bertugas untuk mengeluarkan atau meracik obat ke pasien sesuai dengan resep yang telah dibuat oleh dokter.

## 4.2 Analisis Kebutuhan

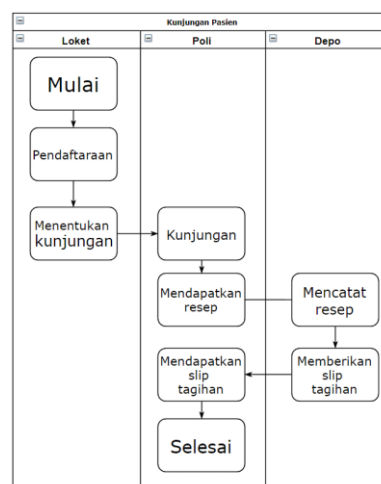
Analisis kebutuhan akan dibagi menjadi tiga yang pertama adalah identifikasi aktor kemudian identifikasi kebutuhan dan kebutuhan tersebut dimodelkan kedalam diagram. Analisis kebutuhan ini sangat penting untuk menggambarkan secara detil kebutuhan aktor akan sistem, sehingga sistem dapat berjalan sesuai dengan harapan.

### 4.2.1 Gambar Fitur Pilihan

Dalam penelitian ini aplikasi rumah sakit untuk pergudangan yang dibuat mampu memberikan solusi yaitu sistem yang terkomputasi, dapat mempersingkat waktu pelayanan, meningkatkan efisiensi waktu, mengurangi selisih jumlah barang, memudahkan petugas untuk menginput data dan mendapatkan laporan stok yang ada di gudang.

#### 4.2.2 Diagram Alir Proses

Terdapat beberapa proses yang terjadi pada beberapa unit yang terlibat yang akan dijelaskan pada diagram alir pada gambar 3. Pada saat pasien berkunjung ke rumah sakit pertama

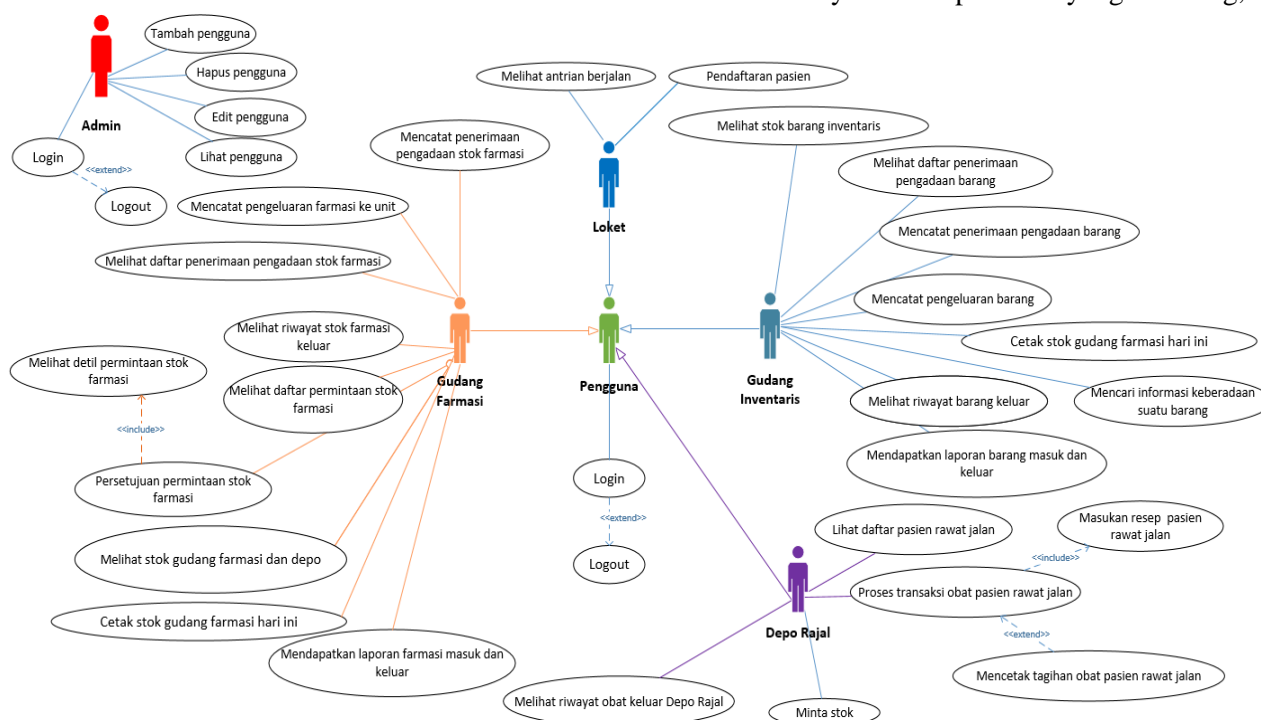


**Gambar 3.** Diagram alir

### 4.2.3 Identifikasi Aktor

Pada sistem ini terdapat 5 pengguna, masing-masing pengguna mempunyai hak akses yang berbeda. Hak akses tersebut ditentukan berdasarkan tugas pengguna tersebut, tugas pengguna adalah sebagai berikut:

1. Petugas loket: Petugas yang bertugas untuk menyambut pasien yang datang,



**Gambar 4.** *Use Case* SIMRS



melakukan pencatatan data pasien dan meneruskan ke layanan yang dibutuhkan.

2. Petugas Gudang Farmasi: Petugas yang bertugas untuk mencatat proses barang masuk dan keluar pada gudang farmasi. Petugas juga bertugas untuk mengeluarkan dan mengantarkan obat yang di pesan oleh depo rajal.

3. Petugas Gudang Inventaris: Petugas yang bertugas untuk mencatat proses barang masuk dan keluar pada gudang inventaris. Petugas bertugas untuk mendistribusikan barang pada unit – unit yang membutuhkan.

4. Petugas Depo Rajal: Petugas Depo Rajal melakukan pengeluaran obat atau peracikan sesuai dengan resep yang telah dibuat oleh dokter. Petugas depo juga melakukan pencatatan resep.

5. Admin: Admin adalah orang yang berhak untuk mengelola pengguna

#### 4.2.4 Use Case

Perbedaan use case iterasi pertama dan use case iterasi kedua adalah penambahan dari fungsionalitas mencetak stok hari ini. Gambar 4 menunjukkan *use case* aplikasi SIMRS keseluruhan. Kebutuhan gudang inventaris dan gudang farmasi adalah hampir sama namun yang membedakan adalah gudang farmasi dapat menerima permintaan barang dari depo. Sedangkan gudang inventaris dapat mencari informasi keberadaan barang. Depo rajal bertugas untuk mencatat transaksi obat yang dilakukan oleh pasien yang berkunjung. Depo juga dapat mencetak tagihan obat.

#### 4.2.5 Kebutuhan Fungsional

Penelitian ini menggunakan *iterative model* dan pada siklus pengembangan sistem ini dilakukan sebanyak dua kali iterasi sehingga kebutuhan akan dimodelkan kedalam dua buah tabel. Terdapat dua buah iterasi yang dilakukan, kebutuhan fungsional pada iterasi pertama dijelaskan pada tabel 1 dan kebutuhan fungsional iterasi kedua yang adalah penambahan fungsional dari iterasi pertama dijelaskan pada tabel 2.

**Tabel 1.** Kebutuhan fungsional iterasi 1

Kode	Spesifikasi	Deskripsi
F04	Daftar antrian pasien pada loket	Aplikasi dapat menampilkan antrian pasien yang berobat hari ini pada pengguna dengan peran petugas loket

F06	Masukan informasi pasien	Aplikasi dapat mencatat informasi pasien yang berobat serta mengarahkan pasien ke unit yang hendak dituju di loket
F07.2	Mencatat penerimaan pengadaan pada untuk gudang inventaris	Aplikasi dapat mencatat informasi barang yang masuk di gudang inventaris
F08.1	Mencatat pengeluaran barang dari gudang farmasi ke unit	Aplikasi dapat mencatat informasi barang yang keluar dari gudang farmasi ke unit
F13.1	Mencari riwayat barang keluar dari gudang inventaris	Aplikasi dapat mencari riwayat barang yang keluar dari gudang inventaris
F14.1	Laporan barang masuk dan keluar gudang inventaris	Aplikasi dapat memberikan laporan stok barang masuk dan keluar pada jangka waktu tertentu misalnya per bulan atau per tahun pada gudang inventaris. Keluaran dapat berupa cetak atau format CSV

Tabel 1 merupakan sebagian kebutuhan fungsional pada pengembangan iterasi pertama dan dari hasil ulasan stakeholder pada iterasi pertama menunjukkan sistem masih terdapat kekurangan, sehingga pada iterasi kedua terdapat kebutuhan fungsional baru yang dapat dilihat pada tabel 2.

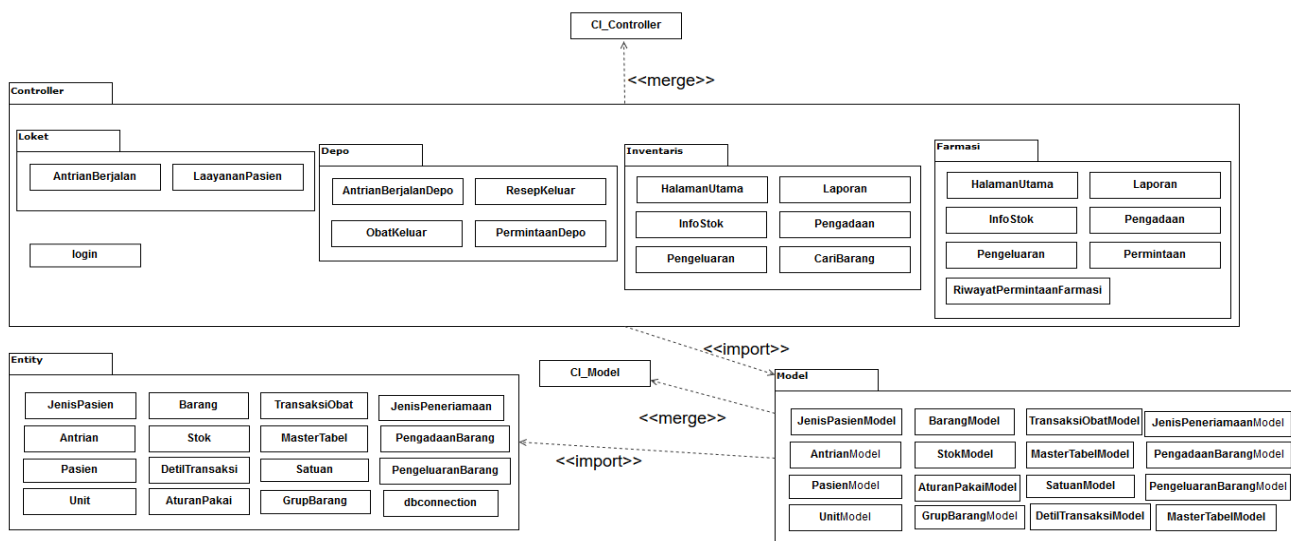
**Tabel 2.** Kebutuhan fungsional iterasi 2

Kode	Spesifikasi	Deskripsi
F25.1	Cetak stok yang tersedia hari ini pada gudang farmasi	Aplikasi dapat mencetak stok yang tersedia pada gudang farmasi pada hari ini

Kebutuhan fungsional pada tabel 2 merupakan salah satu penambahan dari kebutuhan fungsional iterasi pertama, sehingga kebutuhan sistem keseluruhan adalah gabungan antara tabel 1 dan 2.

#### 4.2.6 Kebutuhan Non Fungsional

Kebutuhan non-fungsional akan di jelaskan di tabel 3. Tidak ada perubahan kebutuhan non fungsional pada kedua iterasi sehingga hanya terdapat satu tabel yang di modelkan.



Gambar 5. Package Diagram SIMRS

Tabel 3. Kebutuhan non fungsional

Kode	Spesifikasi	Deskripsi
NF01	Compatibility Browser Google Chrome	Sistem harus dapat berjalan dengan baik pada browser dengan versi minimal 58.0.3029.96
NF02	Compatibility Browser Mozilla Firefox	Sistem harus dapat berjalan dengan baik pada browser dengan versi minimal 53.0.2

### 4.3 Perancangan Sistem

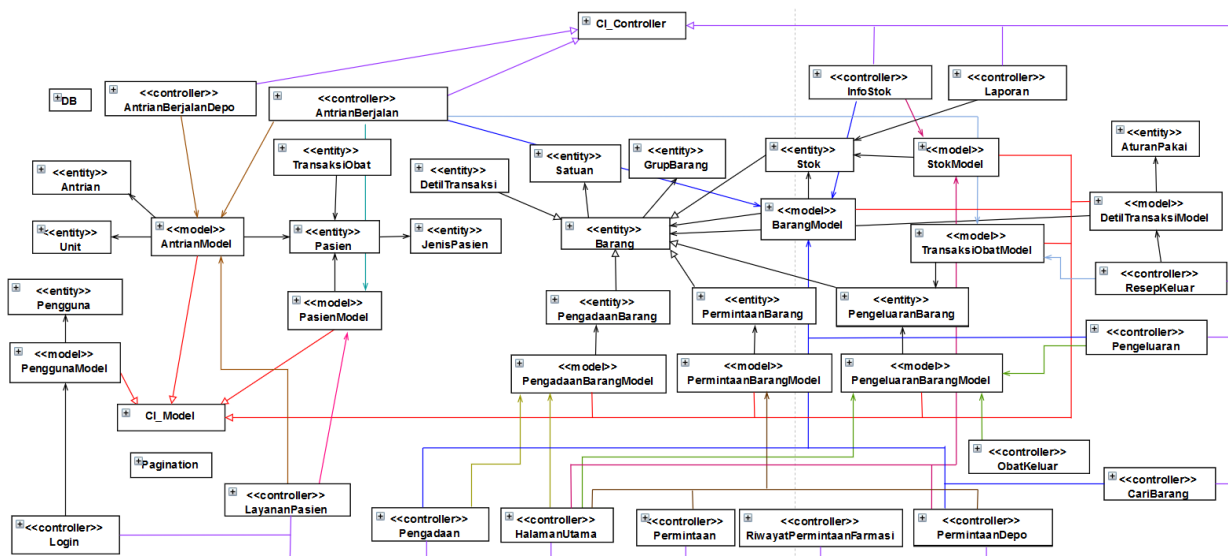
Sub bab 4.3.1 hingga 4.3.4 akan membahas tentang bagaimana merancang pemodelan sistem yang nantinya akan di buat mulai dari use case hingga perancangan antar muka.

#### 4.3.1 Package Diagram

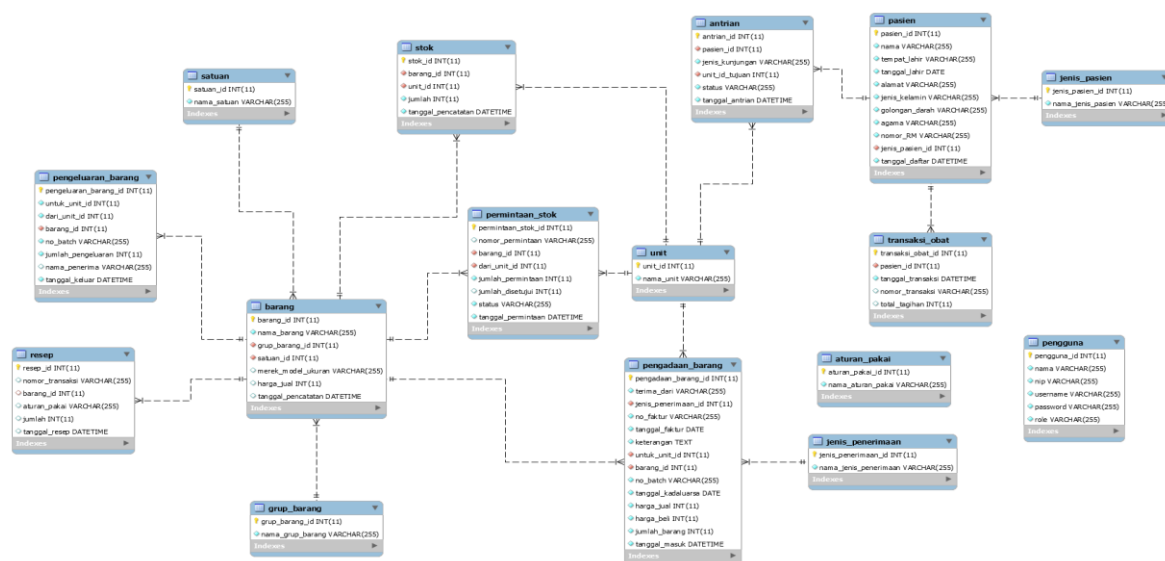
Tidak ada perubahan pemodelan package diagram pada kedua siklus perkembangan, sehingga hanya ada satu pemodelan *package diagram*. Gambar 5 menunjukkan *package diagram* sistem. Package *controller* adalah sekumpul package yang memiliki kelas *controller* yang berfungsi untuk mengatur aliran data dan tampilan. Didalam *package controller* dibagi lagi beberapa *package* berdasarkan lingkupnya. *Package controller* adalah extends dari kelas *CI\_Controller* dan mengimport *package model*. *Package model* merupakan sekumpulan kelas yang mengatur data pada sistem.

#### 4.3.2 Class Diagram

Gambar 6 adalah rancangan kelas yang akan



Gambar 6. Class diagram SIMRS



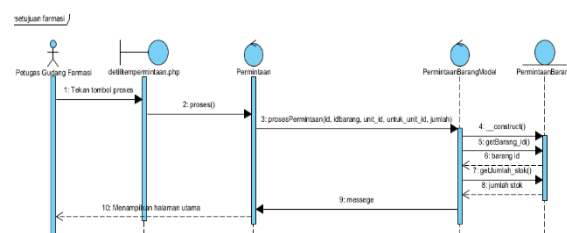
Gambar 7. Entity Relationship Diagram database SIMRS

diimplementasi kedalam sistem aplikasi SIMRS. Terdapat 3 jenis *class* yaitu *controller*, *model* dan *entity*. *Controller* berfungsi untuk mengatur aliran data dari model ke tampilan sedangkan *model* berfungsi untuk mengatur proses data dari database dan *entity class*. Pada pengembangan iterasi kedua tidak ada penambahan kelas karena kelas yang dimodelkan pada iterasi pertama sudah mencukupi untuk pengembangan pada tahap kedua, namun hanya ada satu penambahan metode didalam kelas *InfoStokModel* pada gudang farmasi dan gudang inventaris dengan nama metode *printStok()* yang digunakan untuk mencetak stok hari ini.

#### 4.3.3 Entity Relationship Diagram

Gambar 7 merupakan rancangan arsitektur database yang akan dibuat. Tidak ada perubahan dalam pemodelan ERD ini. Walaupun ada perubahan pada iterasi kedua namun karena pemodelan ERD ini sudah mencukupi persyaratan pengembangan pada iterasi kedua maka hanya digunakan satu buah pemodelan. Terdapat 15 tabel yang akan di implementasi kedalam sistem, tabel tersebut antara lain: tabel *antrian*, *aturan\_pakai\_obat*, *barang*, *grup\_barang*, *jenis\_pasien*, *jenis\_pemeriksaan*, *pasien*, *pengadaan\_barang*, *pengeluaran\_barang*, *pengguna*, *permintaan\_stok*, *satuan*, *stok*, *transaksi\_obat*, *resep*, dan *unit*.

#### 4.3.4 Sequence Diagram

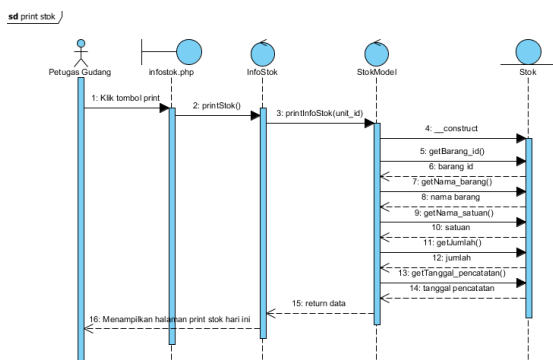


Gambar 8. Sequence Diagram Persetujuan Permintaan Stok

Gambar 8 menggambarkan interaksi pengguna yang melakukan aktifitas melihat detail permintaan stok. Setelah pengguna menekan tombol layanan pada tampilan halaman utama maka controller permintaan akan mengambil data dari kelas gudang. Setelah mendapatkan kembalian data maka data di berikan ke tampilan detail permintaan. Lalu setelah pengguna mengisi jumlah yang disetujui.

Dari hasil ulasan stakeholder pada iterasi pertama menunjukkan sistem masih terdapat kekurangan sehingga ada penambahan kebutuhan fungsional pada iterasi kedua. Dari penambahan kebutuhan fungsional tersebut maka pemodelan *sequence diagram* juga ada yang ditambahkan, penambahan *sequence diagram* iterasi kedua dijelaskan gambar 9.





Gambar 9. Sequence Diagram cetak stok hari ini

Gambar 9 menggambarkan interaksi pengguna yang melakukan aktifitas mencetak informasi stok hari ini. Setelah pengguna menekan tombol print pada halaman informasi stok maka controller memanggil method printInfoStok() pada kelas gudang. Setelah mendapatkan kembalian data maka data di berikan ke halaman printstokhariini dan mencetak halaman tersebut.

## 5. IMPLEMENTASI

### 5.1 Implementasi Database

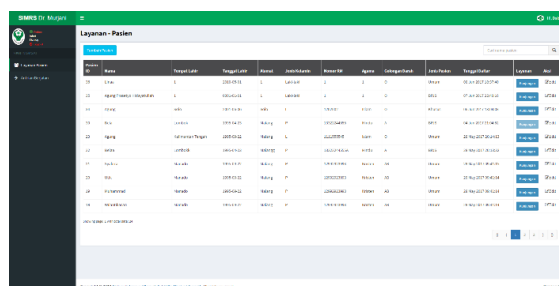
Dari perancangan *database* pada bab sebelumnya maka diimplementasikan lah kedalam *database* sesungguhnya yang akan dipakai oleh sistem aplikasi SIMRS. Gambar 10 merupakan implementasi pada database. Gambar 10 adalah hasil tangkapan layar dari phpmyadmin. Kolom tabel merupakan tabel yang telah di implementasi serta rows adalah jumlah kolom yang telah terisi pada tabel.

Table	Action
antrian	Browse Structure
aturan_pakai	Browse Structure
barang	Browse Structure
grup_barang	Browse Structure
jenis_pasien	Browse Structure
jenis_penerimaan	Browse Structure
pasien	Browse Structure
pengadaan_barang	Browse Structure
pengeluaran_barang	Browse Structure
pengguna	Browse Structure
permintaan_stok	Browse Structure
resep	Browse Structure
satuan	Browse Structure
stok	Browse Structure
transaksi_obat	Browse Structure
unit	Browse Structure
16 tables	Sum

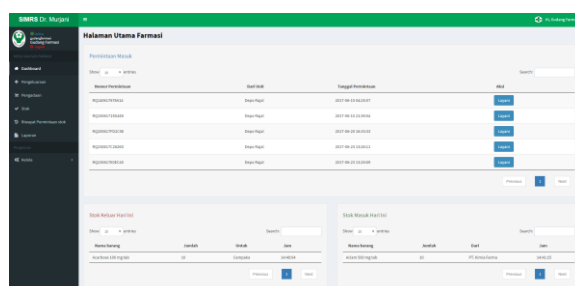
Gambar 10. Implementasi database sistem

### 5.2 Implementasi Antarmuka

Berikut ini adalah contoh implementasi dari aplikasi SIMRS



Gambar 11. Antarmuka layanan pasien



Gambar 12. Antarmuka Halaman Utama

Gambar 11 dan 12 merupakan beberapa implementasi antarmuka SIMRS. Gambar 11 adalah ketika berada di menu layanan pasien. Terdapat info pasien dan tombol layani untuk melayani kunjungan. Gambar 12 adalah antarmuka halaman utama farmasi, terdapat informasi stok keluar dan masuk serta permintaan masuk dari depo dan tombol layani permintaan tersebut.

## 6. PENGUJIAN

Terdapat 3 buah pengujian yang akan dilakukan, yang dilakukan antara lain *compatibility testing*, *blackbox testing*, *whitebox testing*,. Pada *whitebox testing* akan menggunakan *basis path testing*.

### 6.1 Compatibility Testing (Non-Fungsional)

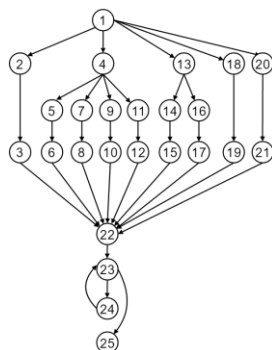
Browser	Internet Explorer	Edge	Firefox	Safari	Opera	Chrome	iOS	Android	BlackBerry
Version	8 9 10 11	15	53	≤9 10	44	58	≤8 9 10	≤3 4*	≤7.1 10.0
Critical Issues	✓	✓	✓	✓	✓	✓	✓	✓	✓
Major Issues	✓	✓	✓	✓	✓	✓	✓	✓	✓
Minor Issues	✓	✓	✓	✓	✓	✓	✓	✓	✓
1 issues on 1 pages	✓	✓	✓	✓	✓	✓	✓	✓	✓

Gambar 13. Hasil Pengujian Compatibility

Gambar 13 memperlihatkan bahwa sistem dapat berjalan dapat berjalan di browser Google Chrome versi 58 atau di bawahnya serta Mozilla Firefox 53 atau di bawahnya sesuai dengan

kebutuhan non-fungsional kecuali internet explorer versi 9 kebawah.

## 6.2 Basis Path testing



Gambar 15. FlowGraph getLaporan

### Jalur Independen getLaporan

- Jalur 1: 1-2-3-22-24-25
- Jalur 2: 1-4-5-6-22-23-24-25
- Jalur 3: 1-4-7-8-22-23-24-25
- Jalur 4: 1-4-9-10-22-23-24-25
- Jalur 5: 1-4-11-12-22-23-24-25
- Jalur 6: 1-13-14-15-22-23-24-25
- Jalur 7: 1-13-16-17-22-23-24-25
- Jalur 8: 1-18-19-22-23-24-25
- Jalur 9: 1-20-21-22-23-24-25

### Cyclomatic Complexity getLaporan

$V(G) = 25$  Regions

$V(G) = 33$  edges - 25 node + 2

$V(G) = 10$

Dengan nilai  $V(G) = 10$  memperlihatkan bahwa aplikasi SIMRS masih tergolong tidak terlalu rumit dalam pemeliharaan kode program.

Tabel 4 Kasus uji jalur independen fungsi getLaporan

No jalur	Jalur	Kasus Uji	Hasil yang diharapkan	Hasil yang didapat	Status
1	1-2-3-22-24-25	rang e="bulan" mont h=7 year =2017	return array('jumlah_barang_keluar' => '1', 'jumlah_barang_masuk' => '1', 'stok_sekangan g' => '171')	return array('jumlah_barang_keluar' => '1', 'jumlah_barang_masuk' => '1', 'stok_sekangan g' => '171')	valid
2	1-4-5-6-22-23-24-25	rang e="triwulan" mont h=3 year =2017	return array('jumlah_barang_keluar' => '0', 'jumlah_barang_masuk' => '0', 'stok_sekangan g' => '171')	return array('jumlah_barang_keluar' => '0', 'jumlah_barang_masuk' => '0', 'stok_sekangan g' => '171')	Valid
3	1-4-7-8-22-23-	rang e="triwulan"	return array('jumlah_barang_keluar' => '14',	return array('jumlah_barang_keluar' => '14',	Valid

	24-25	mont h=6 year =2017	'jumlah_barang_masuk' => '12', 'stok_sekangan g' => '171')	'jumlah_barang_masuk' => '12', 'stok_sekangan g' => '171')	
4	1-4-9-10-22-23-24-25	rang e="triwulan" mont h=9 year =2017	return array('jumlah_barang_keluar' => '1', 'jumlah_barang_masuk' => '1', 'stok_sekangan g' => '171')	return array('jumlah_barang_keluar' => '1', 'jumlah_barang_masuk' => '1', 'stok_sekangan g' => '171')	valid
5	1-4-11-12-22-23-24-25	rang e="triwulan" mont h=12 year =2017	return array('jumlah_barang_keluar' => '0', 'jumlah_barang_masuk' => '0', 'stok_sekangan g' => '171')	return array('jumlah_barang_keluar' => '0', 'jumlah_barang_masuk' => '0', 'stok_sekangan g' => '171')	valid
6	1-13-14-15-22-23-24-25	rang e="" mont h=7 year =2017	null	null	valid
7	1-13-16-17-22-23-24-25	rang e="semester" mont h=3 year =2017	return array('jumlah_barang_keluar' => '42', 'jumlah_barang_masuk' => '1', 'stok_sekangan g' => '171')	return array('jumlah_barang_keluar' => '42', 'jumlah_barang_masuk' => '1', 'stok_sekangan g' => '171')	valid
8	1-18-19-22-23-24-25	rang e="semester" mont h=7 year =2017	return array('jumlah_barang_keluar' => '37', 'jumlah_barang_masuk' => '1', 'stok_sekangan g' => '171')	return array('jumlah_barang_keluar' => '37', 'jumlah_barang_masuk' => '1', 'stok_sekangan g' => '171')	valid
9	1-20-21-22-23-24-25	rang e="tahun" mont h=7 year =2017	return array('jumlah_barang_keluar' => '15', 'jumlah_barang_masuk' => '38', 'stok_sekangan g' => '171')	return array('jumlah_barang_keluar' => '15', 'jumlah_barang_masuk' => '38', 'stok_sekangan g' => '171')	valid
10	1-20-21-22-23-24-25	rang e="" mont h=7 year =2017	return array('jumlah_barang_keluar' => '1', 'jumlah_barang_masuk' => '1', 'stok_sekangan g' => '171')	return array('jumlah_barang_keluar' => '1', 'jumlah_barang_masuk' => '1', 'stok_sekangan g' => '171')	valid

Terdapat 3 pengujian *basis path*, getLaporan adalah salah satu pengujiannya. Dari 10 kasus uji getLaporan didapatkan 10 hasil yang didapat

sesuai dengan hasil yang diharapkan.

### 6.3 Black-box testing

Tabel 4 Pengujian *blackbox testing*

Kode Fungsional	Kasus Uji	Yang diharapkan	Hasil yang di dapat	Status
F04	Petugas loket menekan menu antrian berjalan	Sistem menampilkan antrian yang sedang berjalan hari ini dengan yang masuk pertama paling atas	Sistem berhasil menampilkan antrian yang sedang berjalan hari ini dengan yang masuk pertama paling atas	Valid
F07.2	(1)Petugas gudang inventaris masuk ke menu pengadaan kemudian menekan tombol tambah barang masuk (2)Petugas gudang inventaris mengisi semua form yang disediakan kemudian menekan tombol tambah item	<ul style="list-style-type: none"> <li>Sistem mencatat semua informasi yang dimasukkan petugas</li> <li>Sistem memberi peringatan bila tidak ada informasi yang dimasukkan</li> </ul>	<ul style="list-style-type: none"> <li>Sistem mencatat semua informasi yang dimasukkan petugas</li> <li>Sistem memberi peringatan bila tidak ada informasi yang dimasukkan</li> </ul>	<ul style="list-style-type: none"> <li>Valid</li> <li>Valid</li> </ul>

Tabel 4 adalah beberapa pengujian *blackbox* dari 65 pengujian yang dilakukan. Pada 65 kasus uji *black-box testing* didapatkan 65 hasil yang didapat sesuai dengan hasil yang diharapkan maka kebutuhan fungsional sistem valid 100%.

## 7. PENUTUP

Dari hasil pengamatan mulai dari proses analisis hingga tahap pengujian maka dapat diambil kesimpulan:

1. Sistem dapat dianalisis kebutuhannya dengan melakukan wawancara petugas gudang inventaris dan petugas gudang di rumah sakit. Hasil dari wawancara tersebut kebutuhan sistem yang didapatkan kemudian dirancang kedalam *use case*, *use case scenario*, *sequence diagram*, *package diagram*, *class diagram*, *entity diagram*, dan perancangan antarmuka.
2. Aplikasi dapat diimplementasi berdasarkan perancangan yang telah dilakukan dengan menggunakan pendekatan *Object Oriented* dan penggunaan teknologi PHP, MySQL, dan JavaScript. Dengan implementasi ini permasalahan pencatatan barang masuk dan keluar yang masih manual dan sederhana pada pergudangan RSUD Dr. Murjani tersebut diharapkan akan dapat teratasi.
3. Sistem dapat diuji dengan menggunakan *compatibility testing* serta metode *white-box testing* dan *black-box testing*. Pada *compatibility testing* sistem dapat berjalan dengan baik di Google Chrome dan Mozilla Firefox browser. Pada metode *white-box* menggunakan *basis path testing* didapatkan hasil 100% valid dari 20 kasus uji. Dari pengujian *black-box testing* didapatkan hasil 100% valid dari 65 kasus uji.

## DAFTAR PUSTAKA

- Arnold, J. R. Tony. 2008. Introduction to Material Management, 6th ed., Pearson Prentice Hall.
- Arwan, Achmad. 2011. Pengembangan Aplikasi Desktop Perwalian Stiki Berbasis Client Server Untuk Mempercepat Proses Entry Data Perwalian Menggunakan Borland Delphi 7.
- Athoillah, Muhammad. 2014. Android-Based Mobile Information System Design for Inventory Control of Warehouse.
- Bunyakiati, P., Finkelstein, A. 2010. Standard Compliance testing for unified modelling language tools.
- Briand, Lionel C., Labiche, Yvan., Leduc, Johanne. 2006. Toward the Reverse Engineering of UML Sequence Diagrams for Distributed Java Software.
- Broy, Manfred., Denert, Ernst. 2002. Entity-Relationship Modeling: Historical Events, Future Trends, and Lessons Learned.

- Chaffer, Jonathan., Swedberg, Karl. 2013. Learning jQuery: Fourth Edition. Packt Publishing. [e-book]. Tersedia melalui: <<https://www.packtpub.com/web-development/learning-jquery-fourth-edition>>
- Codeigniter. 2013. About Codeigniter. [online]. Tersedia di: < <https://codeigniter.com>> [Diakses 7 Juli 2017]
- Dedeke, Adenekan., Lieberman, Benjamin. 2006. Qualifying Use Case Diagram Associations.
- Daqiqil, Ibnu. 2011. Framework Codeigniter Sebuah Panduan dan Best Practice. [e-book]. Tersedia melalui: <<https://www.amarbank.co.id/upload/ed69f40abc4d5bb8a9b4857cf461f142.pdf>> [Diakses 30 Juni 2017]
- Gaşpar, V., Madarász, L., Paralip, J., Ténaiová, K. 2011. Design and implementation of a client warehouse application over an enterprise resource planning system for mobile devices.
- Inguanez, Frankie. 2012. Entity Relationship Modelling: A short guide to designing Entity Relationship Models using Barker's notation. [e-book]. Tersedia melalui:<<https://frankieingunez.files.wordpress.com/2012/01/barkers-erd-notation.pdf>> [Diakses 30 Juni 2017]
- Jayarathne, Gayanath. 2013. PHP Fundamentals. [e-book]. Leanpub Publishing. Tersedia melalui: <<https://leanpub.com/phpfundamentals>> [Diakses 30 Juni 2017]
- jQuery. 2006. What is jQuery. [online]. Tersedia di: <<https://jquery.com/>> [Diakses 7 Juli 2017]
- Ljubuncic, Igor. 2011. Apache Web Server Complete Guide.
- Miller, Fred., Paradis, Rose., Whalen, Kevin. 1991. Iterative Development Life Cycle (IDLC): A Management Process for Large Scale Intelligent System Development
- Malhotra, Ruchika., Chug, Anuradha. 2016. Comparative Analysis of Agile Methods and Iterative Enhancement Model In Assessment of Software Maintenance.
- Pressman, Roger S. 2001 .Software Engineering: A Practitioner's Approach, Fifth Edition.
- Priyambadha, Bayu., Sarwosri, Dwi Sunaryono. 2010. Perancangan dan pembuatan antar muka surat elektronik berbasis teknologi ajax.
- Powel, Andrew . 2016. Iterative Model: What Is It And When Should You Use It. [online] Tersedia di:< <https://airbrake.io/blog/sdlc/iterative-model>> [Diakses 29 Juni 2017]
- Rumbaugh, James., Jacobson, Ivar., Booch, Grady. 2004. The Unified Modeling Language Reference Manual Second Edition. [e-book]. Addison – Wesley. Tersedia melalui: Perpustakaan Universitas of Texas Dallas< [https://www.utdallas.edu/~chung/Fujitsu/UML\\_2.0/Rumbaugh--UML\\_2.0\\_Reference\\_CD.pdf](https://www.utdallas.edu/~chung/Fujitsu/UML_2.0/Rumbaugh--UML_2.0_Reference_CD.pdf)> [Diakses 30 Juni 2017]
- Owens, Thomas. 2011. What does the cyclomatic complexity of my code mean. [online]. Tersedia di: <<https://softwareengineering.stackexchange.com/questions/101830/what-does-the-cyclomatic-complexity-of-my-code-mean>> [Diakses 6 Juli 2017]
- Sommerville, Ian. 2003. Software Engineering 6th Edition.
- Solichin, Achmad. 2009. Pemrograman Web dengan PHP dan MySQL.
- Schmuller, Joseph. 2004. Sams Teach Yourself UML in 24 Hours, Third edition. [e-book]. Sams Publishing. Tersedia melalui: <<https://groupalpha.files.wordpress.com/2007/12/uml-tutorial.pdf>> [Diakses 30 Juni 2017]
- Suehring, Steve. 2013. Javascript Step by Step: Third Edition.
- Tjahyadi, Firdaus., Saptono, Henry., Gustiarahman, Irfan. 2007. IGOS Training Tool Kit “RDBMS (Relational DataBase Management System): MySQL”.
- Vidgen, Richard. 2003. Requirements analysis and UML use cases and class diagrams.
- Warman, John. 1995. Manajemen Pergudangan.