

## PERANCANGAN SISTEM INFORMASI INVENTORI BARANG MENGUNAKAN RUBY ON RAILS DI PT.XXX

RR. Isni Anisah P.H<sup>1)</sup>, Candra Mecca Sufyana<sup>2)</sup>, Bentang Saepul Rohmat<sup>3)</sup>

<sup>1), 2), 3)</sup> Manajemen Informatika Politeknik Piksi Ganesha Bandung

Jl. Gatot Subroto No. 301 Bandung

Email : [isnikng81@gmail.com](mailto:isnikng81@gmail.com)<sup>1)</sup>, [mctuta\\_echan@yahoo.com](mailto:mctuta_echan@yahoo.com)<sup>2)</sup>, [bentang\\_saepul@yahoo.co.id](mailto:bentang_saepul@yahoo.co.id)<sup>3)</sup>

### Abstrak

Penelitian ini berisi tentang program aplikasi yang dibuat untuk proses pengolahan data inventori barang di PT. XXX, yang bertujuan untuk mengetahui sistem informasi inventori barang yang sedang berjalan, faktor-faktor penghambat serta pemecahan masalah pada sistem tersebut. Dari penelitian yang telah dilakukan, terdapat faktor yang menghambat sistem inventori barang tersebut karena masih menggunakan cara manual yang dirasakan kurang efektif dan efisien. Untuk membuat perancangan sistem informasi inventori barang ini maka penulis menggunakan pendekatan berorientasi objek dengan pemodelan UML (Unified Modelling Language), ORM (Object Relational Mapping) dan diimplementasikan dengan bahasa pemrograman Ruby On Rails serta MySQL sebagai DBMS (Database Management System), serta menggunakan model pengembangan perangkat lunak prototyping. Dengan adanya aplikasi ini sebagai sarana pengolahan inventori barang maka diharapkan proses pengolahan data inventori dapat berjalan cepat dan optimal..

**Kata kunci:** Barang, UML (Unified Modelling Language), Bahasa Pemrograman Ruby On Rails.

### 1. Pendahuluan

Sistem informasi inventori barang sangat penting untuk sebuah perusahaan apalagi perusahaan tersebut bergerak di bidang produksi, yang menuntut adanya kejelasan dalam persediaan barang perusahaannya. Sistem informasi inventori barang membantu para pergerak bisnis untuk melihat kemajuan dalam perusahaannya karena dapat diambil sebuah data yang solid dari efek penjualan barang karena terdapat informasi – informasi mengenai indeks barang yang masuk dan barang yang keluar, sehingga pola keuangan perusahaan dapat diatur sesuai indeks presentase barang yang masuk maupun keluar. Ketersediaan barang dalam memenuhi pesanan pelanggan menjadi sangat penting. Namun, stok barang yang terlalu menumpuk akan mengakibatkan kerugian yang tidak sedikit karena biaya yang harus ditanggung dalam pengadaan barang tersebut<sup>[4]</sup>.

PT.XXX adalah perusahaan dagang yang memiliki produk utamanya berupa keripik dan emping di daerah

Banten. Selain keripik emping PT. XXX juga menjual produk sejenis lainnya seperti kerupuk kecimpring dan kerupuk opak. Sistem informasi inventori barang di PT.XXX hanya menggunakan sebuah buku sebagai tempat untuk mencatat tentang informasi produk yang dijual, siapa yang terlibat dan barang yang dibelinya. Hal ini mengakibatkan proses inventori barang sering tidak valid, laporan tidak lengkap dan pernah hilang tetapi tidak memiliki rekapannya, dan pemilik masih harus mencari satu persatu produknya disetiap gudang sehingga memakan waktu yang lama dan tenaga yang cukup besar.

Berdasarkan masalah yang telah diungkapkan diatas tadi, maka dibuatlah “PERANCANGAN SISTEM INFORMASI INVENTORI BARANG MENGGUNAKAN RUBY ON RAILS DI PT. XXX menggunakan metode berorientasi objek yang mencakup semua data produk (barang) masuk dan keluar, dan stok. Penulis akan mengimplementasikannya menggunakan bahasa pemrograman Ruby On Rails dengan database Mysql.

### 2. Pembahasan

#### Sistem Informasi Inventori Barang

Informasi merupakan hal yang sangat penting bagi manajemen dalam pengambilan keputusan. Informasi diperoleh dari sistem informasi (*information systems*) atau disebut juga dengan *processing system*<sup>[2]</sup>. Sebuah sistem informasi merupakan kumpulan dari perangkat keras dan perangkat lunak komputer, serta perangkat manusia yang akan mengolah data menggunakan perangkat keras dan perangkat lunak tersebut<sup>[3]</sup>. Inventory meliputi semua barang yang dimiliki perusahaan pada saat tertentu, dengan tujuan untuk dijual kembali atau dikomsumsi dalam siklus operasi normal perusahaan sebagai barang yang dimiliki untuk dijual atau diasumsikan untuk dimasa yang akan datang, semua barang yang berwujud dapat disebut sebagai inventory, tergantung dari sifat dan jenis usaha perusahaan<sup>[4]</sup>.

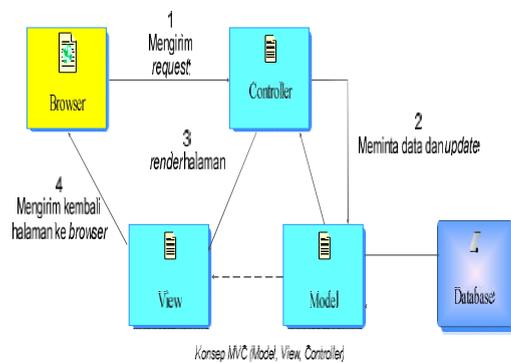
#### Bahasa Pemrograman Ruby On Rails

*Ruby on Rails* merupakan *Ruby-based framework* yang menggunakan pendekatan *Model View Controller* (MVC)<sup>[5]</sup>. Jika ingin membangun suatu *website* dengan menggunakan *Ruby*, maka dibutuhkan *Rails*. *Rails* merupakan salah satu *framework Ruby* yang paling

populer sekarang ini. Oleh karena itu dinamakan *Ruby on Rails*. Dilihat dari namanya sering ada salah pengertian bahwa bukan *Ruby* yang digunakan pada *Rails* tetapi *Rails* yang digunakan pada *Ruby*. *Rails* menggunakan *Model View Controller* (MVC) sebagai *design pattern*-nya. Struktur penyusun MVC ada 3 jenis yang saling bekerja sama, yaitu *Model*, *Controller*, dan *View*<sup>[1]</sup>. Proses yang terjadi dalam suatu *user request* adalah sebagai berikut:

1. *Client* melalui *browser* mengirimkan suatu *request* untuk suatu halaman kepada *controller* yang berada di *server*.
2. *Controller* mengambil data yang diperlukan dari *model* dalam hal melakukan *respond* terhadap *request*.
3. *Controller* me-render halaman dan mengirimkannya kepada *view*.
4. *View* mengirim halaman kembali melalui *browser* agar dapat dilihat oleh *client*

Proses di atas dapat dilihat pada Gambar 1:



Gambar 1 Konsep MVC [6]

*Ruby on Rails* (RoR) mengimplementasikan MVC dengan menyediakan 3 komponen sebagai bagian dari *framework*, yaitu<sup>[6]</sup>:

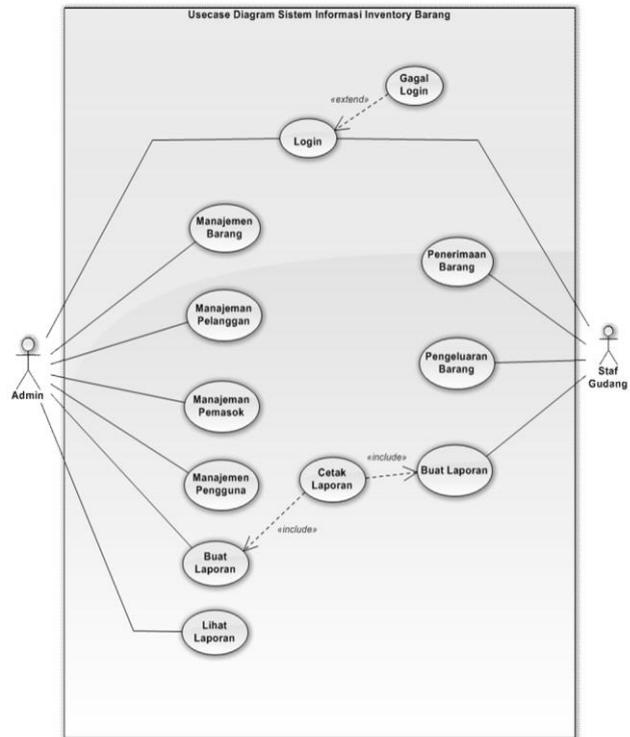
1. *Active Record*: merupakan 'Model' dalam RoR. Komponen model menyimpan data dan menyediakan fungsionalitas untuk bekerja dengan data.
2. *Action View*: komponen *View* meliputi presentasi logik dari data yang ada pada komponen *Model*. *Action View* merupakan komponen *View* pada *RoR*.
3. *Action Controller*: *controller* menyusun aliran logika. Dalam suatu aplikasi WEB, *controller* bertugas mengatur dan menyusun *flow* dari suatu logika aplikasi. *Controller* duduk sebagai batas dari suatu aplikasi dan menangkap semua *request*.

**Perancangan Unified Modelling Language**

Perancangan Sistem Informasi Inventori Barang di PT. XXX digambarkan menggunakan metode berorientasi objek, *tools tools* yang digunakan adalah *Unified Modelling Language* (UML), dimana perancangan sistem ini merupakan kebutuhan fungsional dan persiapan untuk diimplementasikan.

**Use Case Diagram**

*Usecase diagram* untuk merekam persyaratan fungsi sebuah sistem dan mendeskripsikan interaksi tipikal antara pengguna sistem dengan sistem itu sendiri<sup>[7]</sup>



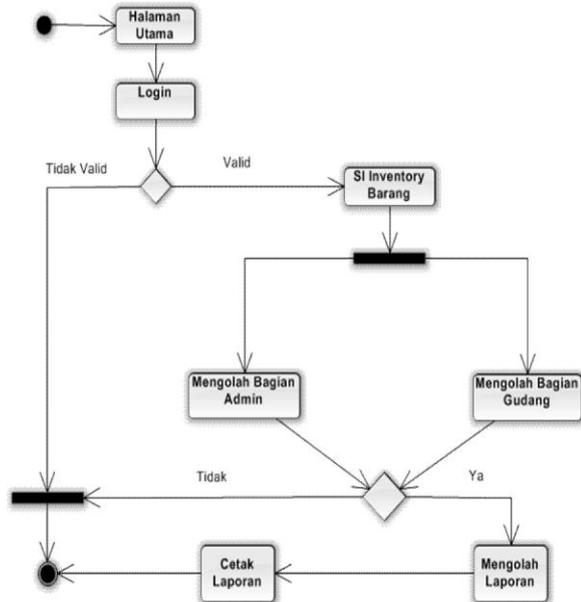
Gambar 2 Use Case Sistem Informasi Inventori Barang

Dalam diagram Use Case yang ditunjukkan pada Gambar 2 terdapat dua aktor yaitu admin dan staf gudang. Admin melakukan tujuh proses atau skenario yaitu melakukan login terlebih dahulu agar dapat mengakses proses pengolahan data lainnya yaitu manajemen barang, manajemen pelanggan, manajemen pemasok, manajemen pengguna, buat dan lihat laporan. Sedangkan aktor staf gudang melakukan empat proses yaitu login, penerimaan barang, pengeluaran barang, dan membuat laporan.

**Activity Diagram**

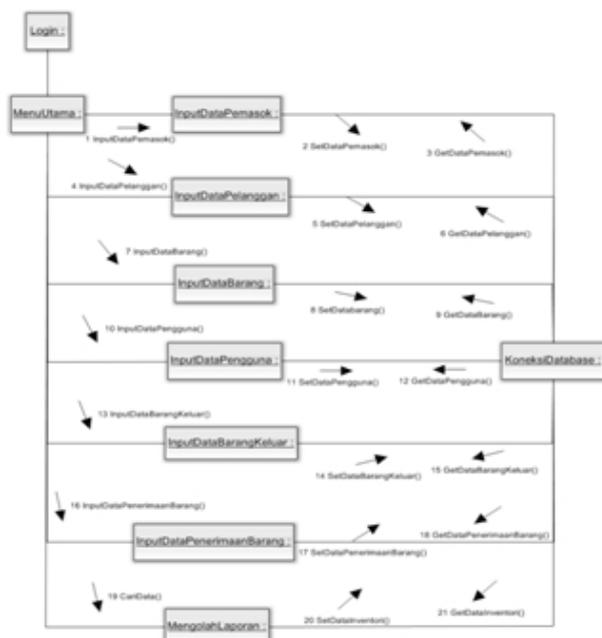
Dimodelkan dalam *activity diagram* yang digunakan untuk memodelkan *workflow*, proses bisnis dan logika procedural<sup>[7]</sup>. Gambar 3 memperlihatkan *activity diagram* sistem informasi inventori barang yang diusulkan. Pertama kali muncul dalam aplikasi ini adalah sebuah halaman utama dari aplikasi Persediaan Barang yang dimana setiap aktor hanya dapat melihat halaman utamanya saja, dan tidak dapat melakukan kegiatan karena harus melalui aktifitas login yang disediakan oleh sistem. Setiap aktor diminta untuk melakukan proses login dulu untuk memulai semua kegiatan yang dilakukan dan didalam proses login sistem akan membaca *account* yang dimasukan dan menerapkan validasi yang nantinya akan mengeluarkan pesan *error* secara langsung apabila sistem tidak sesuai dan secara otomatis sistem akan mengarahkan kemana *account*

dapat beroperasi karena sistem akan membaca *Role* dari *account* tersebut. Setelah melakukan login sistem akan langsung merespon dengan mengeluarkan ucapan salam kepada aktor, dan menjalankan *Role* dari sistem. Aktivitas aktor yang akan dilakukan dan direspon oleh sistem yang diantaranya adalah memajemen barang, memajemen pemasok, memajemen pelanggan, dan memajemen pengguna yang diantaranya adalah dapat melakukan penambahan data, perubahan data, dan penghapusan data, yang terdapat di menu bagian gudang. Kemudian terdapat aktivitas pengolahan laporan yaitu untuk melakukan pengolahan laporan data persediaan barang lalu mencetak laporan inventori tersebut.



Gambar 3. Activity Diagram

Communication Diagram

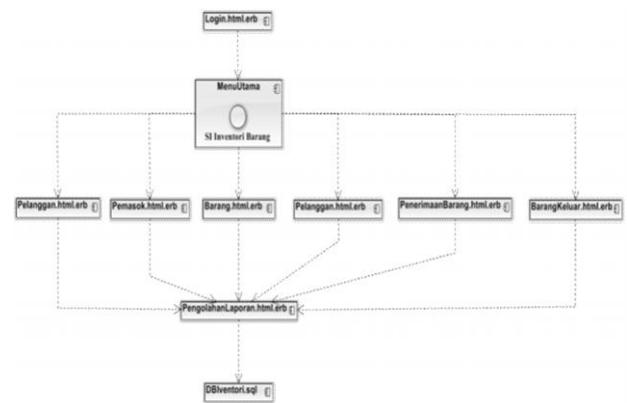


Gambar 4. Communication Diagram

*Communication Diagram* digunakan untuk memodelkan interaksi antar *object* di dalam sistem. Pada Gambar 4, setelah objek login dapat akses ke menu utama sehingga dapat melakukan input data pemasok, data pelanggan, data barang, data pengguna, data barang keluar, dan data penerimaan barang dimana output objek-objek tersebut terkoneksi pada database.

Component Diagram

*Component Diagram* merupakan pemecahan suatu sistem menjadi komponen-komponen serta menampilkan hubungan dengan antarmuka atau pemecahan komponen menjadi struktur yang lebih renah [7]. Component diagram untuk perancangan sistem yang diusulkan dapat dilihat pada Gambar 5.



Gambar 5. Component Diagram

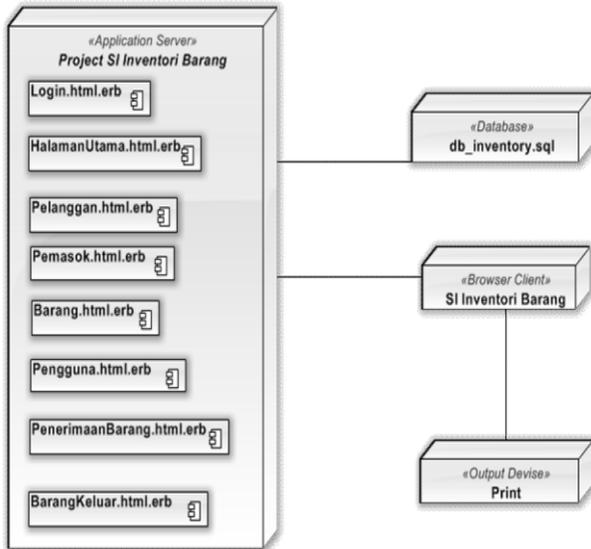
Deskripsi untuk component diagram ditunjukkan pada Tabel 2.1.

Tabel 2.1 Deskripsi Component Diagram

Nama Component	Deskripsi
Login	Komponen untuk antarmuka form login.
Main	Komponen untuk antarmuka form menu utama.
Pelanggan	Komponen untuk antarmuka form data pelanggan.
Pemasok	Komponen untuk antarmuka form data pemasok.
Barang	Komponen untuk antarmuka form data barang.
Pengguna	Komponen untuk antarmuka form data pengguna.
Penerimaan Barang	Komponen untuk antarmuka form data penerimaan barang.
Barang Keluar	Komponen untuk antarmuka form data barang keluar.
Pengolahan Laporan	Komponen untuk antarmuka form data pengolahan laporan.
Database	Komponen untuk penyimpanan dan koneksi data ke database.

**Deployment Diagram**

Deployment Diagram menunjukkan susunan fisik sebuah sistem, bagian perangkat lunak mana yang berjalan pada perangkat keras mana<sup>[7]</sup>. Gambar 6 menunjukkan deployment diagram yang diusulkan dan Tabel 2.2. menjelaskan deskripsi dari setiap node.



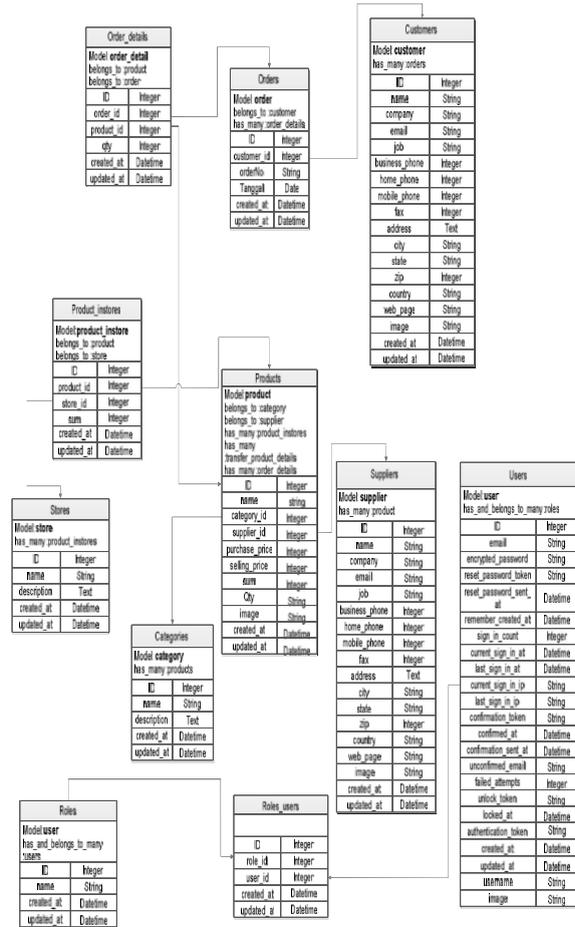
Gambar 6 Deployment Diagram

Tabel 2.2 Deskripsi Deployment Diagram

Node	Deskripsi
<<Application Server>> Project SI Inventori Barang	Node yang berisi komponen-komponen sistem yang akan dirancang.
<<Database>> Inventori	Node yang berisi database yang digunakan pada sistem Inventori.
<<Browser Client>> SI Inventori Barang	Node yang berisi aplikasi sistem inventori barang yang dikonfigurasi pada komputer client melalui mesin pencari browser engine.
<<Output Device>> Printer	Node yang berisi perangkat keras yang digunakan untuk mencetak laporan data inventori barang.

**ORM Diagram**

Objek Relational Mapping (ORM) Adalah salah satu tehnik untuk memetakan basisdata relasional ke model objek. Sedangkan menurut ORM merupakan sebuah mekanisme yang memungkinkan mengatasi, mengakses dan memanipulasi objek tanpa harus mempertimbangkan bagaimana objek berhubungan dengan Sumber data yang lain<sup>[7]</sup>.



Gambar 7 ORM Diagram

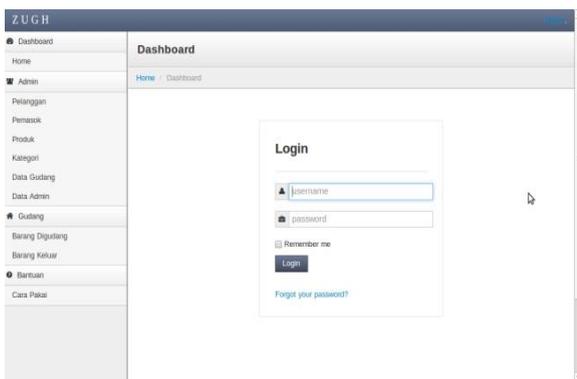
Tabel 2.3 Deskripsi ORM Diagram

Model	Keterangan
User	Model proses yang diambil dari pendefinisian Usecase Login, proses ini untuk validasi users yang akan mengakses aplikasi yang melibatkan model role.
Role	Model data yang digunakan untuk mengatur hak akses dari user dimana data dari model role ini akan di akses oleh model user.
Role_users	Hanya akan menyimpan data dari model role dan model users.
Customer	Model yang diambil dari usecase manajemen pelanggan dimana model ini berperan dalam menyimpan data dari pelanggan dan akan diambil oleh model order
Supplier	Model yang diambil dari usecase manajemen pemasok dimana model ini berperan dalam menyimpan data dari pemasok dan data dari data tersebut akan diberikan kepada model product

<b>Product</b>	Model yang diambil dari usecase manajemen barang dimana model ini berperan dalam menyimpan atribut dari barang dan akan mengambil data dari model supplier untuk melengkapi data dari siapa barang tersebut ada,
<b>Category</b>	Model yang menyimpan data dari kategori barang yang nantinya akan di akses oleh model <i>product</i> .
<b>Store</b>	Model yang menyimpan data dari gudang, dimana data dari gudang ini akan digunakan oleh staf gudang yang akan melakukan penerimaan barang yang masuk.
<b>Product_instore</b>	Model dari <i>usecase</i> manajemen penerimaan barang, dimana penerimaan barang ini bermaksud untuk setiap barang yang masuk kedalam gudang maka data yang diperlukan oleh model <i>product_instore</i> ini adalah model <i>product</i> dan model <i>store</i> .
<b>Order</b>	Model dari <i>usecase</i> barang keluar yaitu barang yang keluar dari gudang, maka data yang diperlukan oleh model <i>order</i> untuk mencatat data dari pelanggan yang meminta barang.

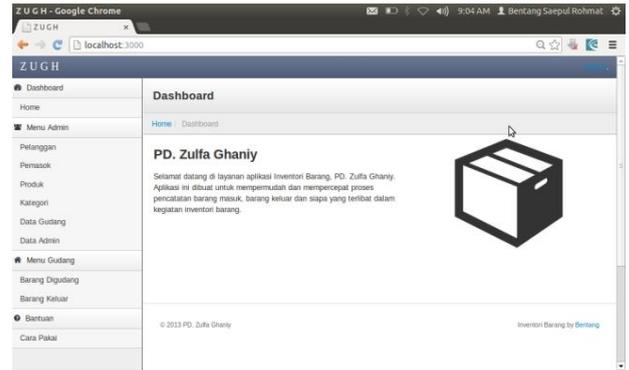
**Implementasi**

Implementasi sistem dilakukan dengan menggunakan bahasa pemrograman *Ruby On Rails 3.0.1* dengan basis data yang digunakan adalah *MySQL*. Aplikasi tersebut dapat dijalankan pada berbagai *platform* sistem operasi dan perangkat keras, tetapi implementasi sepenuhnya dilakukan diperangkat keras PC (*personal computer*) dengan sistem operasi *Ubuntu*.



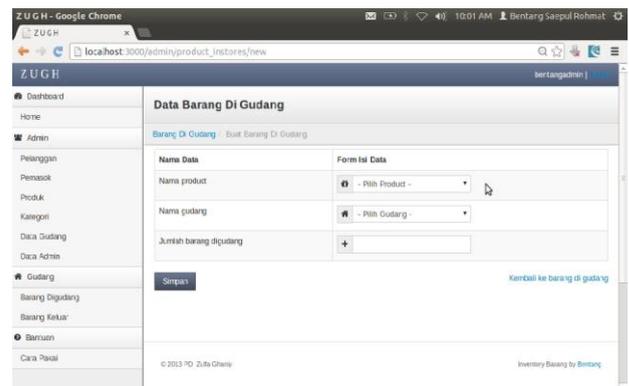
**Gambar 8.** Tampilan form login

Gambar 8 merupakan tampilan Form Login untuk validasi users dan admin dimana atribut yang diminta yaitu username dan password.



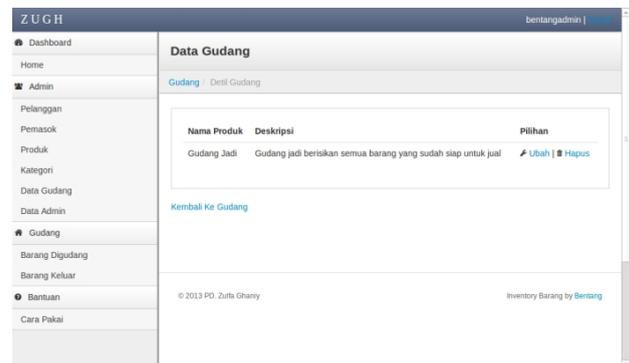
**Gambar 9** Tampilan halaman utama (*index*)

Gambar 9 merupakan tampilan halaman utama yang memperlihatkan profil perusahaan dan juga menampilkan menu-menu yang dapat digunakan dalam sistem ini. Menu admin terdiri dari data pelanggan, pemasok, produk, kategori, data gudang, dan data admin, sedangkan menu gudang terdiri dari barang di gudang dan barang keluar. Terdapat juga menu bantuan yang menjelaskan cara pemakaian sistem ini. Pada implementasi sistem ini dibagi menjadi tiga tampilan yaitu tampilan *form*, tampilan *show*, dan tampilan *index*. Berikut contoh implementasi tersebut untuk data barang di Gudang yang ditunjukkan masing-masing pada Gambar 10, 11, dan 12.



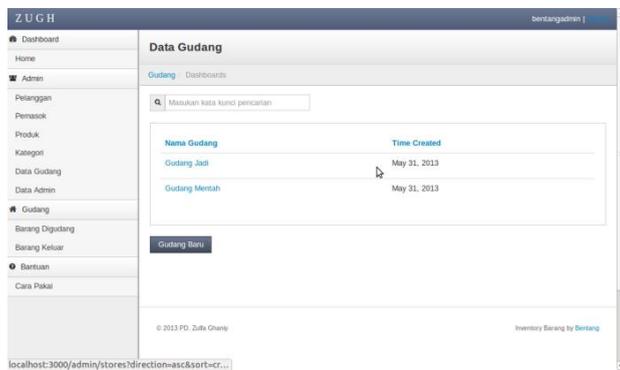
**Gambar 10.** Tampilan Form Barang di Gudang

Pada Tampilan *Form* yang diperlihatkan pada Gambar 10 terdapat tombol simpan untuk menyimpan data ke dalam *database* setelah selesai menginputkan data.



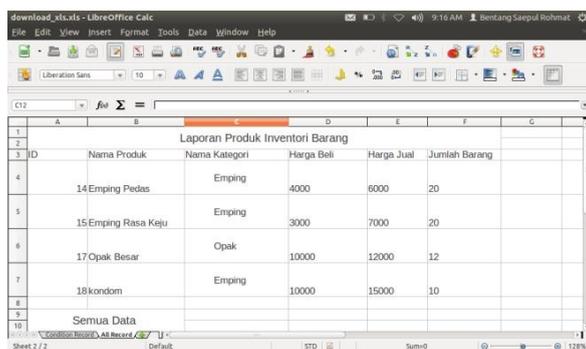
**Gambar 11** Tampilan *show* barang di gudang

Pada Tampilan *show* seperti yang diperlihatkan pada Gambar 11 terdapat label ubah untuk mengedit atau mengupdate data ke dalam *database* dan label hapus untuk menghapus data yang tidak diperlukan.



Gambar 12 Tampilan *index* barang digudang

Pada Tampilan *index* seperti yang ditunjukkan Gambar 12 terdapat tombol gudang baru untuk menambahkan data ke dalam *database* dan textbox pencarian untuk mencari data yang diperlukan.



Gambar 13. Hasil laporan semua data *record*

Tampilan hasil laporan seluruh data diperlihatkan pada Gambar 13 dimana hasil laporan di export ke dalam *Microsoft Excel*. Laporan berupa ID, nama produk, nama kategori, harga beli, harga jual, dan jumlah barang.

### 3. Kesimpulan

Berdasarkan hasil penelitian dan identifikasi masalah serta pembahasan sistem yang dianalisis, maka dapat diambil beberapa kesimpulan sebagai berikut:

1. Pengolahan Data Inventori Barang yang ada di PT. XXX sering tidak valid, dan tidak efisien dalam proses pencarian data dan pelaporan sehingga diperlukan perancangan sistem informasi yang baru.
2. Telah dibuat perancangan sistem informasi inventori barang dengan menggunakan bahasa pemodelan UML dan ORM serta implementasi menggunakan Ruby On Rails
3. Metode OOP cocok untuk menyelesaikan permasalahan diatas dan cocok untuk diterapkan pada Ruby On Rails
4. Perancangan ini telah diimplementasikan di perusahaan tempat penelitian, dan telah memberikan manfaat bagi PT. XXX

### Daftar Pustaka:

- [1] Burd, Barry, 2007. *Ruby on Rails For Dummies*. Indianapolis: Wiley Publishing, Inc.
- [2] Gordon B. Davis, 1995. *Sistem Informasi Manajemen*, Gramedia, Jakarta
- [3] Jogiyanto. 2008. *Analisis dan Desain Sistem Informasi*, Andi, Yogyakarta.
- [4] Koher, Eric L.A. 2006. *Produksi dan Bahan Baku*, Gramedia, Jakarta
- [5] Meidya, Devin. 2007. *Perbandingan Cross Site Scripting Pada Framework PHP Codeigniter dan Ruby on Rails*. Universitas Telkom, Bandung
- [6] Rajshekhar, A.P, 2008. *Building Dynamic Web 2.0 with Ruby on Rails*. Packt Publishing, Birmingham
- [7] Shalahuddin, Muhammad dan Rosa A.S. 2011. *Modul Pembelajaran Rekayasa Perangkat Lunak*, Modula, Bandung.

### Biodata Penulis

**RR. Isni Anisah**, memperoleh gelar Sarjana Teknik (S.T), Jurusan Teknik Informatika Universitas Pasundan Bandung, lulus tahun 2007. Memperoleh gelar Magister Komputer (M.Kom) Program Pasca Sarjana Magister Rekayasa Sistem Informasi STMIK LIKMI Bandung, lulus tahun 2012. Saat ini menjadi Dosen di Politeknik Piksi Ganesha Bandung

**Candra Mecca Sufyana**, memperoleh gelar Sarjana Sains (S.Si), Jurusan Fisika Komputasi Institut Teknologi Bandung (ITB), lulus tahun 2008. Memperoleh gelar Magister Teknik (M.T) Program Pasca Sarjana Magister FTTM ITB, lulus tahun 2013. Saat ini menjadi Dosen di Politeknik Piksi Ganesha Bandung.

**Bentang Saepul Rohmat**, memperoleh gelar Ahli Madya (A.Md), Jurusan Manajemen Informatika Politeknik Piksi Ganesha Bandung, lulus tahun 2013. Saat ini menjadi paraktisi IT di salah satu perusahaan swasta di Bandung.