

RANCANG BANGUN ERP PADA MODUL INVENTORY MANAGEMENT MENGGUNAKAN ZACHMAN FRAMEWORK

Rhiyananta Catur Yudowicitro¹⁾, Hermawan²⁾, Rika Yunitarini³⁾

Program Studi Teknik Informatika, Universitas Trunojoyo Madura

JL. Raya Telang, PO BOX 2, Kamal, Bangkalan - 69162

E-mail: ¹⁾ rhiyananta@gmail.com

ABSTRAK

Dalam perancangan *Enterprise Resource Planning* (ERP), dibutuhkan perencanaan terorganisir, dikarenakan ERP ditujukan untuk mengintegrasikan seluruh proses yang ada dalam berbagai domain fungsional perusahaan, antar departemen, maupun antar lokasi yang berbeda. Penelitian ini bertujuan merancang sebuah desain aplikasi untuk mendapatkan *blueprint* aplikasi pada modul *Inventory Management* (IM). *Zachman Framework* (ZF) merupakan salah satu kerangka kerja yang dapat memetakan artefak arsitektur informasi di sebuah organisasi. Dalam penerapannya, ZF mampu mendeskripsikan level perspektif arsitektur IT (*Planner Perspective, Owner Perspective, Designer Perspective, Builder Perspective, Implementer Perspective, Participant Perspective*) sesuai abstraksi objek (*What, How, Where, Who, When, Why*). Untuk standarisasi artefak arsitektur informasi digunakan pendekatan standart pemodelan UML 2.0, sehingga mampu meningkatkan deskripsi ZF menjadi standart *Model Driven Architecture*. Untuk pengujian hasil desain perancangan telah dikembangkan perangkat lunak ERP IM yang menerapkan *Model Driven Design* yang mampu mengkonversi artefak model menjadi aplikasi yang berkualitas. Dimana dari hasil penilaian *Stakeholder* dari pengujian sistem di PT. IGLAS (Persero), didapatkan tingkat kepuasan rata-rata 90%. Sedangkan untuk penilaian *availability* sistem secara kuantitatif dengan dibandingkan *Openerp 7.0* sistem yang dikembangkan jauh lebih kompleks dengan ketersediaan fitur 88,8% berbanding dengan *Openerp 7.0* yang hanya memiliki keterbatasan fitur 77,7%.

Kata Kunci : *Enterprise Resource Planning, Inventory Management, Zachman Framework, UML 2.0, Model Driven Design, Openerp 7.0*

ABSTRACT

In the design of Enterprise Resource Planning (ERP), it takes planning organized, since the ERP aimed to integrate the entire process that exists in different functional domains of the company, inter-departmental, or between different locations. This research aims to design a blueprint for getting the application design application in the module Inventory Management (IM). Zachman Framework (ZF) is a framework that can chart the artifact information architecture in an organization. In according to case, ZF is able to describe the architecture perspective level IT (Planner, Owner Perspective Perspective Perspective, Designer, Builder, Implementer Perspective Perspective, Participant Perspective) appropriate abstraction object (What, How, Where, Who, When, Why). To standardize information architecture artifacts used modeling standard UML 2.0 approach, so it is able to improve the description of the standard Model of ZF be Driven Architecture. for testing the results of design design has developed ERP software IM applying Model Driven Design that is able to convert a model artifact quality applications. Where the assessment of the results of the testing of the system Stakeholders in PT IGLAS (Persero), obtained an average satisfaction rate of 90%. As for the assessment of availability system quantitatively compared with Openerp 7.0 system developed much more complex with the availability features of 88.8% compared with Openerp 7.0 which only has limited features of 77.7%.

Keyword : *Enterprise Resource Planning, Inventory Management, Zachman Framework, UML 2.0, Model Driven Design, Openerp 7.0*

PENDAHULUAN

Enterprise Resource Planning (ERP) merupakan suatu cara untuk mengelola sumber daya perusahaan dengan menggunakan teknologi informasi. Penggunaan ERP yang dilengkapi dengan *hardware* dan *software* untuk mengkoordinasi dan mengintegrasikan data informasi pada setiap area *business processes* untuk menghasilkan pengambilan keputusan yang cepat karena menyediakan analisa dan laporan keuangan yang cepat, laporan penjualan yang on time, laporan produksi dan *inventory* yang dihadapi perusahaan saat ini.

Seiring dengan persaingan di dunia bisnis semakin kompleks, maka perusahaan-perusahaan mencoba untuk meningkatkan jumlah konsumennya dengan melakukan pelayanan yang cepat dan biaya yang murah dibandingkan dengan kompetitornya. Salah satu cara untuk mewujudkan kesuksesan tersebut dapat dilakukan dengan cara mengintegrasikan sistem informasi, peningkatan efisiensi dari sistem informasi untuk menghasilkan manajemen yang lebih efisien dalam *business processes*.

Modul *Inventory Management* (IM) adalah sistem manajemen dalam menentukan keseimbangan antara investasi penyimpanan persediaan dengan pelayanan pelanggan. Sistem persediaan adalah salah satu bagian *Zachman Framework* merupakan *framework* arsitektural yang paling banyak dikenal dan diadaptasi. Para arsitek data *enterprise* mulai menerima dan menggunakan *framework* ini sejak pertama kali diperkenalkan oleh *John A Zachman* di *IBM System Journal* pada tahun 1987 oleh *Zachman Institut for Framework Advancement* (ZIFA) atas pemikiran *John A Zachman* [2] [3].

UML (*Unified Modeling Language*) merupakan bahasa pemodelan sistem/perangkat lunak yang mampu mendekatkan desain dengan perangkat lunak berorientasi objek sehingga dapat diperoleh domain-domain yang diperlukan perangkat lunak [3].

Dalam penelitian ini penulis memanfaatkan *Enterprise Architectur ZF* dan pemodelan UML, yang mana UML yang di publikasikan pada tahun 1995 di sajikan kedalam *Zachman Framework* yang di

publikasikan pada tahun 1987 [4], sehingga diperoleh *blueprint* Aplikasi ERP modul *Inventory Management*.

METODE

Dalam penelitian ini digunakan *enterprise architecture Zachman Framework* yang akan menjabarkan masing-masing kolomnya yang terdiri dari *What, How, Where, Who, When dan Why*. Dan masing-masing kolom nantinya akan diuraikan dengan baris-barisnya (*scope, business model, system model, teknologi model*) Yang akan di jelaskan pada penerapan *Zachman Framework*..

Zachman Framework merupakan *framework* arsitektural yang paling banyak dikenal dan diadaptasi. Para arsitek data *enterprise* mulai menerima dan menggunakan *framework* ini sejak pertama kali diperkenalkan oleh *John A Zachman* di *IBM System Journal* pada tahun 1987 dan kemudian dikembangkan pada tahun 1992 dengan tujuan untuk menyediakan struktur dasar organisasi yang mendukung akses, integrasi, interpretasi, pengembangan, pengelolaan, dan perubahan perangkat arsitektural dari sistem informasi organisasi (*enterprise*) [1][2].

Pada Gambar 1 dijelaskan bahwa *Zachman Framework* merupakan matrik 6×6 yang merepresentasikan interseksi dari dua skema klasifikasi – arsitektur sistem dua dimensi. Pada dimensi pertama, *Zachman* menggambarkan sebagai baris yang terdiri dari 6 perspektif.

- a) *Planner Perspective (Scope Context)* : Daftar lingkup penjelasan unsur bisnis yang dikenali oleh para ahli strategi sebagai ahli teori.
- b) *Owner Perspective (Business Concept)* : Model semantik keterhubungan bisnis antara komponen-komponen bisnis yang didefinisikan oleh pimpinan eksekutif sebagai pemilik.
- c) *Designer Perspective (System Logic)* : Model logika yang lebih rinci yang berisi kebutuhan dan desain batasan sistem yang direpresentasikan oleh para arsitek sebagai desainer.
- d) *Builder Perspective (Technology Physics)* : Model fisik yang

mengoptimalkan desain untuk kebutuhan spesifik dalam batasan teknologi spesifik, orang, biaya dan lingkup waktu yang dispesifikasikan oleh *engineer* sebagai *builder*.

- e) *Implementer Perspective (Component Assemblies)* : Teknologi khusus, tentang bagaimana komponen dirakit dan dioperasikan, dikonfigurasi oleh teknisi sebagai implementator.
- f) *Participant Perspective (Operation Classes)* : Kejadian-kejadian sistem berfungsi nyata yang digunakan oleh para teknisi sebagai *participant*.

Untuk dimensi kedua, setiap isu perspektif membutuhkan cara yang berbeda untuk menjawab pertanyaan fundamental :*who, what, why, when, where and how*. Setiap pertanyaan membutuhkan jawaban dalam format yang berbeda. Zachman menggambarkan setiap pertanyaan fundamental dalam bentuk kolom/fokus.[1][2].

- a) *What* (kolom data) : material yang digunakan untuk membangun sistem (*inventory set*).

- b) *How* (kolom fungsi) : melaksanakan aktivitas (*process transformations*).
- c) *Where* (kolom jaringan) : lokasi, tofografi dan teknologi (*network nodes*).
- d) *Who* (kolom orang) : aturan dan organisasi (*organization group*).
- e) *When* (kolom waktu) : kejadian, siklus, jadwal (*time periods*).
- f) *Why* (kolom tujuan) : tujuan, motivasi dan inisiatif (*motivation reason*).

Untuk membuat desain memiliki pendekatan terhadap domain objek peneliti akan membuat penurunan *UML* dikolom *Zachman Framework* sehingga akan di peroleh *blueprint* aplikasi yang akan mempermudah pengembangan aplikasi secara berkelanjutan. *UML* merupakan singkatan dari *Unified Modelling Language* adalah sekumpulan pemodelan konversi yang digunakan untuk menentukan atau menggambarkan sebuah sistem perangkat lunak dalam kaitanya dengan objek. *UML* memiliki komponen-komponen yang mendefinisikan objek kedalam notasi-notasi.

ENTERPRISE ARCHITECTURE - A FRAMEWORK™

	DATA #/dat	FUNCTION #/fun	NETWORK #/net	PEOPLE #/peo	TIME #/tim	MOTIVATION #/mty	
SCOPE (CONTEXTUAL)	List of Things important to the Business 	List of Processes the Business Performs 	List of Locations in which the Business Operates 	List of Organizations important to the Business 	List of Events/Cycles Significant to the Business 	List of Business Goals/Strategies 	SCOPE (CONTEXTUAL)
Planner	ENTITY = Class of Business Thing	Process = Class of Business Process	Node = Major Business Location	People = Major Organization Unit	Time = Major Business Event/Cycle	Ends/Means = Major Business Goal/Strategy	Planner
BUSINESS MODEL (CONCEPTUAL)	e.g. Semantic Model 	e.g. Business Process Model 	e.g. Business Logistics System 	e.g. Work Flow Model 	e.g. Master Schedule 	e.g. Business Plan 	BUSINESS MODEL (CONCEPTUAL)
Owner	Ent = Business Entity Rel = Business Relationship	Proc = Business Process IO = Business Resources	Node = Business Location Link = Business Linkage	People = Organization Unit Work = Work Product	Time = Business Event Cycle = Business Cycle	End = Business Objective Means = Business Strategy	Owner
SYSTEM MODEL (LOGICAL)	e.g. Logical Data Model 	e.g. Application Architecture 	e.g. Distributed System Architecture 	e.g. Human Interface Architecture 	e.g. Processing Structure 	e.g. Business Rule Model 	SYSTEM MODEL (LOGICAL)
Designer	Ent = Data Entity Rel = Data Relationship	Proc = Application Function IO = User Views	Node = I/O Function (Processor, Storage, etc.) Link = Link Characteristics	People = Role Work = Deliverable	Time = System Event Cycle = Processing Cycle	End = Structural Assertion Means = Action Assertion	Designer
TECHNOLOGY MODEL (PHYSICAL)	e.g. Physical Data Model 	e.g. System Design 	e.g. Technology Architecture 	e.g. Presentation Architecture 	e.g. Control Structure 	e.g. Rule Design 	TECHNOLOGY MODEL (PHYSICAL)
Builder	Ent = Segment/Tablets Rel = Program/Process	Proc = Computer Function IO = Data Elements/Units	Node = Hardware/Systems Software Link = Link Specifications	People = User Work = Screen Format	Time = Execute Cycle = Component Cycle	End = Condition Means = Action	Builder
DETAILED REPRESENTATIONS (OUT-OF-CONTEXT)	e.g. Data Definition 	e.g. Program 	e.g. Network Architecture 	e.g. Security Architecture 	e.g. Timing Definition 	e.g. Rule Specification 	DETAILED REPRESENTATIONS (OUT-OF-CONTEXT)
Sub-Contractor	Ent = Field Rel = Address	Proc = Language Statement IO = Control Block	Node = Address Link = Protocol	People = Identity Work = Job	Time = Interval Cycle = Machine Cycle	End = Sub-condition Means = Step	Sub-Contractor
FUNCTIONING ENTERPRISE	e.g. DATA	e.g. FUNCTION	e.g. NETWORK	e.g. ORGANIZATION	e.g. SCHEDULE	e.g. STRATEGY	FUNCTIONING ENTERPRISE

© John A. Zachman, Zachman International

Gambar 1 Kerangka Kerja Zachman untuk Arsitektur Enterprise

Komponen *UML* yang digunakan dalam penelitian ini adalah sebagai berikut:

- a) *Use case Diagram*

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *usecase* merepresentasikan sebuah interaksi antara aktor dengan sistem.

b) *Class Diagram*

Class Diagram adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek. *Class* menggambarkan keadaan (*atribut/properti*) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut.

c) *Activity Diagram*

Activity Diagram menggambarkan berbagai aliran aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decisions* yang mungkin terjadi dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses parallel yang mungkin terjadi pada beberapa eksekusi.

d) *Sequence Diagram*

Sequence diagram secara grafis menggambarkan bagaimana objek berinteraksi antara satu sama lain melalui pesan eksekusi pada sebuah *usecase* atau operasi.

e) *Deployment Diagram*

Deployment diagram menggambarkan detail bagaimana komponen di *Deploy* dalam infrastruktur sistem, dimana komponen akan terletak pada mesin dan server atau perangkat, apa dan bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi server dan hal yang bersifat fisik lainnya.

Model Driven Development

Model Driven Development merupakan pendekatan disain sistem yang menekankan pada penggambaran model sistem untuk mendokumentasikan aspek teknis dan implementasi dari sebuah sistem. Pada kegiatan Pengembangannya, *Model Driven Development* memiliki tahapan rekayasa *architecture* dengan pendekatan pemodelan *UML* untuk memperoleh perancangan *software* dan rekayasa *Design* dengan

menggunakan *Framework* pemrograman untuk membangun *software*.

HASIL DAN PEMBAHASAN

Analisis hasil dan pembahasan yang diuraikan merupakan analisa definisi setiap kolom untuk penyusunan *Zachman Framework*, proses membangun aplikasi dari design, dan analisa fungsionalitas sistem untuk mengukur kualitas sistem.

Penyusunan Kolom *Zachman* dengan komponen pemodelan *UML*

Berdasarkan hasil pengumpulan data maka selanjutnya akan dilakukan proses pemetaan masalah kedalam kerangka *Zachman* untuk menghasilkan rancangan sistem yang dibutuhkan. Dalam pemetaan tersebut akan dilakukan pendekatan dengan pemodelan *UML* seperti yang di jelaskan pada Tabel 1.

Pada Tabel 1. menunjukkan beberapa komponen Pemodelan *UML* yang dipetakan kedalam sel *Zachman Framework*, diantaranya *Usecase diagram* dan *BPMN diagram* pada Kolom P2A4, *Class Diagram* pada kolom P3A1, *Activity Diagram* pada kolom P3A2, *Sequence Diagram* pada kolom P3A3 dan *Deployment Diagram* pada kolom P4A2.

Dalam pemnyusunan kolom *Zachman* perlu memahami definisi dari setiap kolom. Setiap kolom pada *zachman* memiliki definisi unik yang di pengaruhi oleh dua skema klasifikasi – arsitektur sistem dua dimensi. Dimensi pertama, *zachman* menggambarkan sebagai baris yang terdiri dari 6 perspektif. Dimensi kedua, setiap isu perspektif membutuhkan cara yang berbeda untuk menjawab pertanyaan fundamental *who, what, why, when, where and how*.

Penelitian ini, 5 kolom *Zachman* dapat diisi dengan komponen *UML* dengan analisa pendefinisian yang dijelaskan Tabel 2.

Tabel 1. Matrik Zachman Yang Telah Dipetakan sebagai komponen UML di beberapa kolomnya

Abstraksi/ Perspektif	DATA What (Things)	FUNCTION How (Process)	NETWORK Where (Location)	PEOPLE Who (People)	TIME When (Time)	MOTIVATION Why (Motivation)
Planner / Contextual (Scope)	(P1A1) Informasi Data Departemen yang terkait	(P1A2) Informasi Proses Bisnis yang terjadi	(P1A3) Informasi Lokasi terjadinya proses bisnis	(P1A4) Informasi Stakeholder yang terlibat pada proses bisnis	(P1A5) Aktifitas yang dilakukan oleh Stakeholder	(P1A6) Visi dan misi departemen terkait
Owner / Conceptual (Business Model)	(P2A1) Conceptual Data Model	(P2A2) Contextual Diagram dan Rantai Nilai Modul	(P2A3) Business Logistics Diagram	(P2A4) Use Case Diagram dan BPMN	(P2A5) Schedule Operational	(P2A6) Menentukan Alur Bisnis Proses
Designer / Logical (System Model)	(P3A1) Class Diagram	(P3A2) Activity Diagram	(P3A3) Sequence Diagram	(P3A4) RACI Matrix	(P3A5) Model Driven Development	(P3A6) Menentukan Design Artifak dari bagian-bagian software
Builder / Physical (Technology Model)	(P4A1) Physical Data Model	(P4A2) Deployment Diagram	(P4A3) Network Diagram	(P4A4) Design Interface Aplikasi	(P4A5) Dijelaskan di P3A5	(P4A6) Menentukan Rancangan Implementasi Software
Detailed Representation (Sub-Contractor)	(P5A1) Detail Struktur Model Django dengan entitas table dan utilitas fungsinya serta database	(P5A2) Model View Template dari program	(P5A3) Konfigurasi setingan router jaringan	(P5A4) Konfigurasi akses aplikasi tiap user	(P5A5) detail jadwal proses SDLC yang dibutuhkan	(P5A6) Menentukan Detail Kode Program dan konfigurasi jaringan
Function Enterprise	(P6A1) Domain Inventory	(P6A2) Pelaksanaan Implementasi	(P6A3) Implementasi	(P6A4) User yang terlibat didalam penggunaan aplikasi	(P6A5) Time schedule launching dan implementasi	(P6A6) Dokumentasi SOP Fitur-fitur aplikasi dan penggunaannya

Tabel 2. Tabel Penyusunan Kolom Zachman dengan komponen pemodelan UML

Kolom Zachman Framework	Definisi	Komponen UML Yang Sesuai
Business Model (Owner/Conceptual), People (Who)(P2A4)	Definisi tanggung jawab seperti model alur kerja (Work Flow Model) yang menggambarkan hubungan tanggung jawab dengan people.	Hubungan people dengan tanggung jawabnya akan direpresentasikan oleh Use Case Diagram, sedangkan untuk Work Flow model-nya berdasarkan bisnis proses akan direpresentasikan oleh Business Process Modelling Notation (BPMN).
System (Designer/Logical), Data (What)(P3A1)	Logical Data Model yang menunjukkan data organisasi teknologi manajemen data tertentu, dan dijelaskan dalam bahasa bisnis, hal ini biasanya direpresentasikan dengan sebuah diagram yang menggambarkan entitas dan	Class Diagram akan menggambarkan model data sistem yang di normalisasi dan dikembangkan dari Conceptual Data Modelling.

System Model (Designer/Logical), Function (How)(P3A2)	hubungannya. Merepresentasikan arsitektur aplikasi yang menangkap aktifitas/kegiatan setiap proses bisnis.	Activity Diagram yang akan menggambarkan seluruh tahapan alur kerja yang mengandung aktivitas, pilihan tindakan, perulangan dan hasil dari aktifitas tersebut.
System Model (Designer/Logical), Location (Where)(P3A3)	Arsitektur sistem pada modul Inventory Management.	Arsitektur sistem akan direpresentasikan kedalam Sequence Diagram.
Technology Model (Builder/Physical), Function (How)(P4A2)	Desain sistem akan menggambarkan bagaimana sistem secara fisik berjalan sebagai komponen Server.	Deployment Diagram akan menggambarkan komponen-komponen server yang dibutuhkan aplikasi, sehingga aplikasi dapat berjalan secara Client Server.

Proses Membangun aktifitas domain dari desain perancangan

Modul *Inventory Management*, adalah modul pengelolaan pergudangan yang efektif dengan upaya meminimasi stok dan mengoptimalkan manajemen pergudangan. Modul *Inventory Management*

Gambar 2. menunjukkan *class diagram Inventory Management* sebagai contoh pendefinisian *Zachman Framework* pada kolom P3A1 yang diperoleh *Class Diagram*.

Proses Membangun Aplikasi dari Model Driven Design

Setiap *Class* pada *Class Diagram* Gambar 2. akan difahami sebagai sebuah form aplikasi dengan tipe data sesuai perancangan. Django Framework dengan bahasa pemrograman python yang, mampu membangun aplikasi ERP sesuai perancangan dengan mentranslasikan *domain model design* pada MDD. Django Framework akan secara otomatis membuat komponen database sesuai perancangan *Class Diagram* Gambar 2. Tipe data pada *database* akan membentuk standar *form*, sehingga inputan yang diterima *form* akan menerima inputan sesuai tipe data pada perancangan *Class Diagram* dan Database.

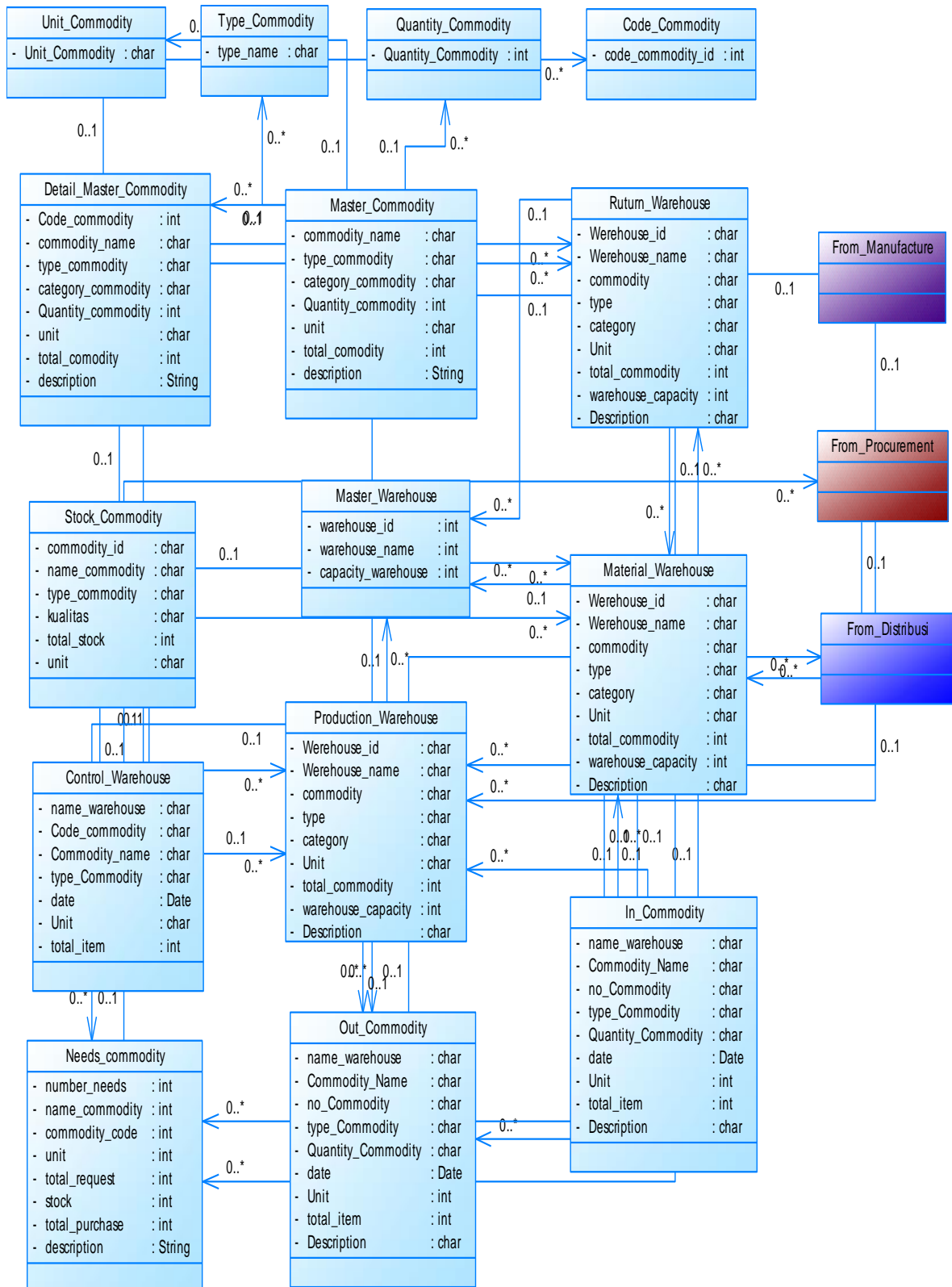
Dari *Class Diagram* Gambar 2, Django Framework akan mendeskripsikan *Class* model sebagai abstraksi *database* sesuai Gambar 3.

Dari *Class Model* Gambar 3, *django framework* akan menghasilkan Perencanaan Master Barang aplikasi sesuai Gambar 4.

Pada Gambar 5. ditampilkan *form* aplikasi entri data Master Barang yang dihasilkan dari *ClassMaster Commodity* dan *Class Master Warehouse*, dimana dalam gambar tersebut adalah gambar dari contoh perancangan pergudangan sesuai dengan *Class Diagram* pada Gambar 2

Analisis Fungsionalitas Sistem

Analisa Fungsionalitas sistem akan menilai sistem dari sudut pandang tertentu, sehingga diperoleh bobot nilai kualitas Aplikasi. Penilaian sistem dilakukan dengan dua cara yaitu dengan melakukan uji coba di PT.IGLAS (Persero) dan dengan komparasi sistem dengan *Open ERP 7.0*. Uji coba sistem dilakukan pada PT IGLAS (Persero) dengan memberikan penilaian terhadap fitur-fitur aplikasi terhadap bisnis proses perusahaan pada ujicoba tersebut dihasilkan nilai tingkat kepuasan user sebesar 90%. Berdasarkan hasil uji coba yang sesuai bisnis proses dari fungsional yang ada di dalam PT IGLAS (Persero) diperoleh kepuasan sebesar 77,7%. Pada pengujian komparasi dengan *Open ERP 7.0* dapat dilihat pada tabel 3.



Gambar 2. Class Diagram

```

class master_commodity (models.Model):
    warehouse_name = models.ForeignKey (Warehouse_material, verbose_name="Nama Gudang")
    commodity_name = models.CharField (max_length=100, verbose_name="Nama Barang")
    
```

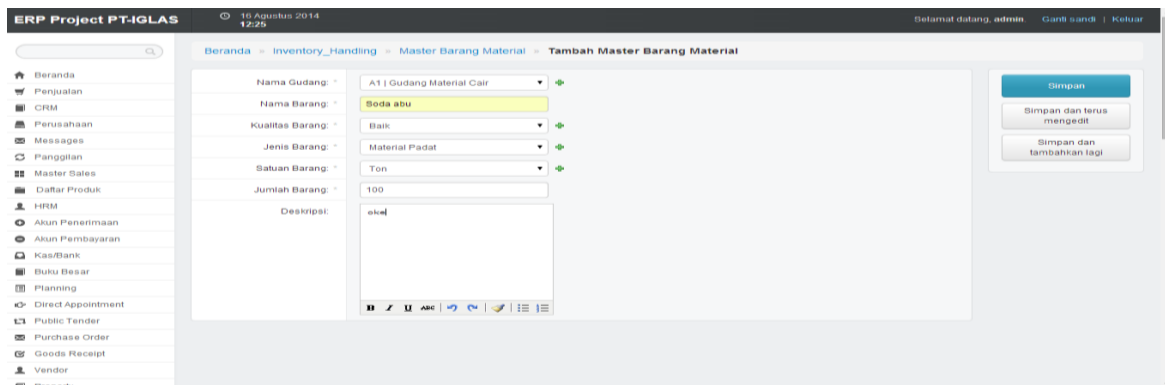
```

commodity_no = models.CharField (max_length=20, editable=False,
verbose_name="Nomer Barang")
quantity_commodity = models.ForeignKey (quantity_commodity,
verbose_name="Kualitas Barang")
type_commodity = models.ForeignKey (type_commodity, verbose_name="Jenis
Barang")
unit = models.ForeignKey (Jenis_satuan, verbose_name="Satuan Barang")
total_stock = models.IntegerField (max_length=20, verbose_name="Jumlah
Barang")
description = HTMLField (max_length=400, verbose_name="Deskripsi",blank=True)

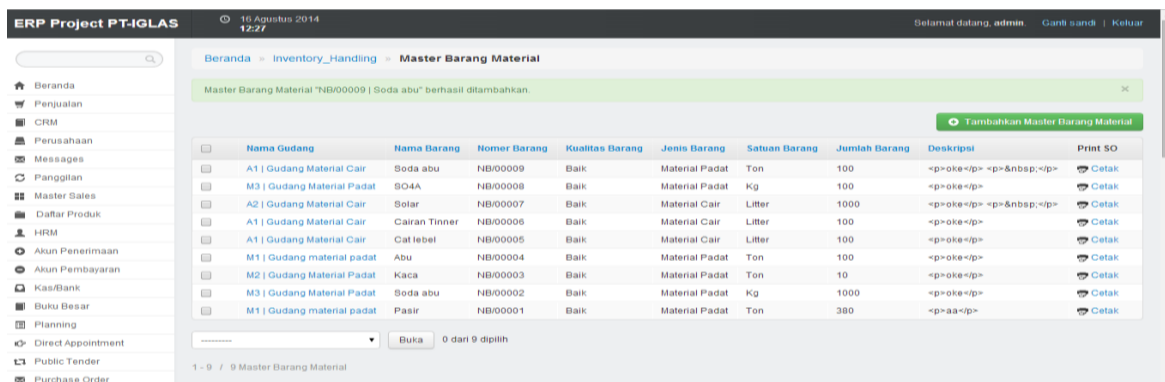
class Meta :verbose_name=_('Master Barang Produksi)
                verbose_name_plural=_('Master Barang Produksi)

def __unicode__(self):
    return u'%s' % self.id
    
```

Gambar 3. Source Code Class Model Django Framework



Gambar 4. Aplikasi Perencanaan Master Barang



Gambar 5. Perencanaan Master Barang

Tabel 3. pengujian komparasi dengan *Open ERP 7.0*

No.	Fitur	A	B
1	<i>Delivery Orders</i>		√
2	<i>Incoming Products</i>	√	√
3	<i>Physical Inventories</i>	√	√

4	<i>Run Schedulers</i>		√
5	<i>Procurement Exceptions</i>	√	√
6	<i>Products by Category</i>	√	√
7	<i>Products</i>	√	√
8	<i>Reordering Rules</i>	√	√
9	<i>Intelligent Location Assignment</i>	√	

A = Presentase cakupan fitur: $(100 / 9) * 7 = 77,7 \%$

B = Presentase cakupan fitur: $(100 / 9) * 8 = 88,8 \%$

SIMPULAN DAN SARAN

Berdasarkan hasil uji coba dapat ditarik kesimpulan, bahwa dalam penyusunan *Zachman Framework* dengan memetakan pemodelan UML dalam pengisian selnya, diperoleh *blue print* aplikasi ERP modul *Inventory Management* yang akan mempermudah dalam pengembangan aplikasi ERP secara berkelanjutan. Aplikasi ERP modul *Inventory Management* yang dibangun dengan *Model Driven Design* menghasilkan kualitas aplikasi dari pengujian sistem oleh *Stakeholder* di PT. IGLAS, didapatkan tingkat kepuasan rata-rata 90%. Sedangkan untuk penilaian *availability* sistem secara kuantitatif dengan dibandingkan *Openerp 7.0* sistem yang dikembangkan jauh lebih kompleks dengan ketersediaan fitur 88,8% berbanding dengan *Openerp 7.0* yang hanya memiliki keterbatasan fitur 77,7%.

Untuk penelitian selanjutnya sebagai penyempurnaan penelitian ini, perlu adanya penambahan Fitur *Bussiness Intelligence* untuk menentukan perencanaan pergudangan. Dalam perkembangannya, pemodelan perangkat lunak perlu dibangun dengan pendekatan *service oriented* agar sistem berjalan secara *realtime*

DAFTAR PUSTAKA

[1] Halida Ernita, Wisnu Ananta Kusuma. Pengembangan *Enterprise Resource Planning* (ERP) Untuk perusahaan Ritel. Departemen Ilmu Komputer, Fakultas Matematika dan Ilmu

- Pengetahuan Alam Institut Pertanian Bogor (2008).
- [2] Ahmad Rifai, Rianarto Sarno, Dwi Sunaryono. Rancang bangun sistem persediaan (*Inventory*) Dengan model *software as a service* menggunakan *service oriented architecture*. Jurusan Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember (2010).
- [3] Falaha, Dewi Rosmala. Penerapan *Framework Zachman* pada *Arsitektur* pengolahan data operasional. Prodi Teknik Informatika, Fakultas Teknik, Universitas Widyatama (2010).
- [4] Moriyanto, Perancangan *Arsitektur Enterprise E-Government* Bangkalan Menggunakan *Zachman Framework*. Jurusan Teknik Informatika Universitas Trunojoyo Madura (2011).
- [5] Wibisono, Setyawan., “*Enterprise Resource Planning*(ERP) Solusi Sistem Informasi Terintegrasi” *Jurnal Teknologi Informasi DINAMIK* 5,3 : 150-159. (2005).
- [6] Larman , Craigh. “storyboarding.” *In Applying UML and Pattern : an Introduction to Object-Oriented Analysis and Design and the Unified process.*, by Craig Larman, 129, Prentice Hall, 2002
- [7] Rezaei, Reza dan Sham, Fereidoon., A Methodology to Create Data Architecture in *Zachman Framework*. *Word Applied Sciences Journal*3 (Supple 2):43-49, 2008