

PERANGKAT LUNAK MANAJEMEN USER UNTUK ORACLE 11G DENGAN JAVA

Riyani Aprilia Rustandi ¹, Wahyu Hidayat ST.,MT.,OCA², Ely Rosely Ir.,M.B.S³

¹²³ Program Studi D3 Manajemen Informatika, Fakultas Ilmu Terapan, Universitas Telkom
¹riyani1904@gmail.com, ²why@tass.telkomuniversity.ac.id, ³ely.rosely@tass.telkomuniversity.ac.id

Abstrak

Manajemen *user* merupakan mekanisme atau cara untuk melakukan pengelolaan terhadap *user* login. Pengelolaan yang dilakukan berupa penambahan atau penghapusan *user*, pengelolaan *role*, pemetaan *role* dan hak akses dengan *user* maupun objek-objek di *database* dan *profile user*. Pengelolaan *user* masih dilakukan melalui *console* pada *command line* dengan menggunakan sintaks. Hal ini cukup rawan terjadi kesalahan dalam penulisannya sehingga menghambat seorang DBA atau pengguna *database* lainnya dalam manajemen *user*. Oleh karena itu, dibangun sebuah aplikasi GUI berbasis desktop untuk manajemen *user* dengan bahasa pemrograman Java dan DBMS Oracle11g. Metode yang digunakan dalam pembuatan aplikasi ini adalah metode *Waterfall*. Perangkat lunak ini diuji fungsionalitasnya dengan metode *black box*. Selain itu dilakukan pula uji kepuasan terhadap perangkat lunak melalui kuisisioner yang melibatkan 14 responden. Hasilnya 93% responden merasa bahwa perangkat lunak ini mudah digunakan dan menyukai menggunakan perangkat lunak dibandingkan menggunakan *console*. Hasil pengujian menunjukkan bahwa perangkat lunak dapat digunakan sebagai alternatif *tools* untuk mempermudah pengguna *database* dalam melakukan manajemen *user* pada Oracle 11g.

Kata kunci: manajemen *user*, *console*, aplikasi desktop, Oracle 11g

Abstract

User management is a mechanism or a way to take over management of user login. Managed by the addition or deletion of user, role management, role mapping and privileges to users and objects in the database and user profile. User management console is done at the command line using the syntax and quite prone to mistakes in writing thus hampering a DBA or other database users in the user management. Therefore, a desktop-based GUI application for user management is built using Java programming language and DBMS Oracle11g. The method used in the development of this application is the Waterfall method. This software was tested functionality with a method black box. In addition it also conducted a test of satisfaction with the software through a questionnaire involving 14 respondents. The results are 92% of respondents felt that the software is easy to use and prefer to use the software than using the console. The test results show that the software can be used as an alternative tool to facilitate users in performing user management database in Oracle 11g.

Keywords: *user management, console, desktop-based application, Oracle 11g.*

1. Pendahuluan

1.1 Latar Belakang

Manajemen *user* adalah suatu mekanisme atau cara untuk melakukan pengelolaan terhadap *user* agar dapat login. Pada sebuah *database* Oracle, manajemen *user* digunakan untuk melakukan pengelolaan terhadap *user* agar dapat melakukan aktivitas terhadap objek-objek yang ada pada *database* Oracle. Pengelolaan yang dilakukan berupa penambahan atau penghapusan *user*, pengelolaan atribut *user* seperti nama, *password* dan hak akses, pemetaan *role* dengan *user* maupun objek-objek di *database* dan juga mengelola *profile user*.

Saat ini untuk dapat mengelola *user* tersebut biasanya dilakukan secara *console* pada *command line* dan menggunakan sintaks. Namun jika pengelolaan *user* dilakukan dengan menggunakan sintaks maka cukup rawan terjadi kesalahan dalam penulisan sintaksnya

dan dinilai kurang praktis. Seiring dengan berjalannya waktu, para pengguna *database* mulai beralih pada *tools* komersial berbasis GUI untuk memudahkan dalam manajemen *user*. Namun *tools* komersial tersebut berbayar, sehingga membatasi pengguna *database* untuk menggunakan *tools* tersebut. Pada beberapa *tools* GUI yang gratis seperti *Oracle Database Express Edition* untuk manajemen *user* hanya terdapat fitur untuk pembuatan *user* sampai pemberian *system privileges*, namun pemetaan hak akses terhadap objek-objek *database* dan pengelolaan *profile user* belum tersedia.

Oleh karena itu, dilakukan perancangan dan pembangunan sebuah perangkat lunak manajemen *user* berbasis GUI yang merupakan salah satu bagian dari *tools* yang dikembangkan oleh Laboratorium *Database* Fakultas Ilmu Terapan dengan versi *free*. Perangkat lunak ini ditujukan untuk digunakan oleh DBA, *programmer*, dan pengguna *database* lainnya serta menjadi alternatif *tools* untuk mempermudah

pengguna dalam manajemen *user* terutama pada Oracle 11g .

1.2 Rumusan Masalah

Adapun rumusan masalah dalam proyek akhir ini adalah sebagai berikut.

- Bagaimana melakukan pengelolaan *user* pada Oracle 11g tanpa menggunakan sintaks?
- Bagaimana melakukan pengelolaan *role* pada Oracle 11g tanpa menggunakan sintaks?
- Bagaimana memetakan antar *role* dengan *user* dan objek-objek di *database* tanpa menggunakan sintaks?
- Bagaimana melakukan pengelolaan *profile user* tanpa menggunakan sintaks?

1.3 Tujuan

Adapun tujuan dalam pembuatan proyek akhir ini adalah membuat perangkat lunak yang menyediakan antarmuka *Graphical User Interface* (GUI) untuk :

- Mengelola *user* pada Oracle 11g.
- Mengelola *role* pada Oracle 11g.
- Mengelola hak akses/*role* terhadap *user* dan objek-objek di *database*.
- Mengelola *profile user* pada Oracle 11g.

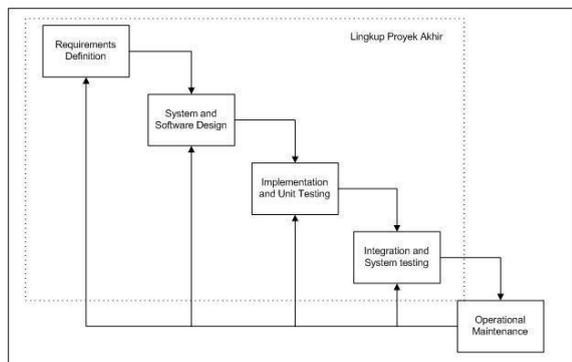
1.4 Batasan Masalah

Adapun batasan masalah dalam proyek akhir ini adalah sebagai berikut.

- Pengelolaan *profile user* meliputi batas jumlah koneksi konkuren, batas waktu koneksi dan *idle time*.
- DBMS yang digunakan adalah Oracle 11g.
- Tidak menangani *dragging pointer* dan *lead memory*.
- Proyek akhir ini tidak sampai pada *maintenance*.

1.5 Metode Pengerjaan

Metode yang digunakan dalam pembuatan proyek akhir ini adalah metode *waterfall*. Menurut [1] *waterfall* adalah metode pengembangan klasik yang sistematis dimana setiap hasil dari suatu tahap akan menjadi masukan awal untuk tahapan selanjutnya.



Gambar 1 Metode Waterfall

2. Tinjauan Pustaka

2.1 Manajemen User

Manajemen *user* adalah suatu mekanisme atau cara untuk melakukan pengelolaan terhadap *user* login. Manajemen *user* digunakan untuk melakukan pengelolaan *user* agar dapat melakukan aktivitas terhadap objek-objek yang ada pada *database* Oracle.

2.1.1 Mengelola User

Untuk dapat koneksi ke Oracle pengguna harus memiliki akun di *database* Oracle. Karakteristik dari akun *user* termasuk *password*, *profile*, *default tablespaces* dan kuota ruang penyimpanan [2]. Untuk dapat membuat *user* biasanya yang memiliki hak akses sebagai DBA. Pengelolaan yang dilakukan antara lain menambah *user* baru, mengedit atribut *user*, membatasi koneksi *user* ke *database*(lock dan unlock) serta menghapus *user*.

2.1.2 Memberikan Hak Akses/Role Terhadap User

Hak akses (*Privilege*) merupakan hak untuk dapat menjalankan statement SQL atau untuk dapat mengakses obyek *database* yang dimiliki *user* lainnya [2].

Dalam Oracle, *privileges* dibagi menjadi 2 :

1. System Privileges

System privileges diberikan oleh seorang DBA atau *user* yang diberikan hak akses sebagai DBA. *System privileges* memungkinkan *user* untuk dapat menjalankan perintah atau akses ke *database* [2].

System privileges antara lain : CREATE USER, DROP USER, DROP ANY TABLE, BACKUP ANY TABLE, SELECT ANY TABLE, CREATE ANY TABLE, CREATE SESSION, CREATE TABLE, CREATE SEQUENCE, CREATE VIEW, CREATE PROCEDURE, dsb.

Statements WITH ADMIN OPTION digunakan saat pemberian *system privileges* dan *user* yang diberi hak akses tersebut dapat menurunkan atau memberikan hak akses yang sama kepada *user* lain.

2. Object Privileges

Object privileges memungkinkan *user* untuk dapat melakukan akses pada obyek spesifik dalam *database*, seperti tabel, view, sequence, procedure, function atau package [2].

Statements WITH GRANT OPTION digunakan saat pemberian *object privileges* dan *user* yang diberi hak akses tersebut dapat menurunkan atau memberikan hak akses yang sama kepada *user* lain.

2.1.3 Membuat Role

Role merupakan kumpulan dari beberapa hak akses (*group privileges*) yang dapat diberikan kepada *user* [2].

2.1.4 Menghapus Hak Akses/Role

Mencabut hak akses/*role* yang telah diberikan terhadap *user* menggunakan sintaks REVOKE.

2.1.5 Mengelola Profile

Profile adalah sebuah *named-sets* batasan *resource* dan *password*. *Profile* di-assign ke pengguna melalui operasi CREATE *USER* dan ALTER *USER*. Pembatasan *resource* bisa dilakukan pada *level session* dan (*operation*) *call*. Batasan *resource* dan *password* mencakup : waktu CPU, operasi I/O, lama *idle*, lama koneksi, memori, *concurrent sessions*, rentang waktu *password*, *password history*, kompleksitas *password* dan penguncian pemakai [3].

2.2 Metadata Views

Beberapa metadata *views* yang digunakan adalah sebagai berikut.

Tabel 1 Metadata Views [3]

Nama	Isi
DBA_users	Semua <i>users</i> di <i>database</i>
DBA_profiles	Semua <i>profiles</i> di <i>database</i>
DBA_roles	Semua <i>roles</i> di <i>database</i>
DBA_role_privs	Semua <i>role</i> yang di-grant ke <i>user</i> atau <i>role</i> lain
DBA_sys_privs	System <i>privileges</i> yang digrant ke <i>user</i> atau <i>role</i>
DBA_tab_privs	Objek <i>privileges</i> yang digrant ke <i>user</i> atau <i>role</i>

2.3 Flowmap

Menurut [4] *Flowmap* adalah diagram yang menggambarkan aliran dokumen pada suatu prosedur kerja di organisasi. Diagram yang menunjukkan arus dari dokumen, aliran data fisis, entitas dan kegiatan operasi yang berhubungan dengan aplikasi.

2.4 Unified Modeling Language (UML)

Menurut [5] UML adalah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik pemrograman berorientasi objek. UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun dan dokumentasi dari sistem perangkat lunak. Berikut ini adalah macam-macam diagram UML.

2.5 Oracle

Oracle adalah basis data relasional yang terdiri dari kumpulan data dalam suatu sistem manajemen basis data RDBMS. Perusahaan perangkat lunak Oracle memasarkan jenis basis data ini untuk bermacam-macam aplikasi yang bisa berjalan pada banyak jenis dan merk perangkat keras (*platform*). Basis data Oracle ini dikembangkan oleh Larry Ellison, Bob Miner dan Ed Oates melalui perusahaan konsultanya

bernama *Software Development Laboratories* (SDL) pada tahun 1977. Pada tahun 1983, perusahaan ini berubah nama menjadi Oracle Corporation sampai sekarang [7].

2.6 Java

Java adalah bahasa pemrograman yang dikembangkan oleh Sun Microsystems pada tahun 1991 sebagai *internal corporate research project Java*. Java merupakan bahasa pemrograman yang dikembangkan dari bahasa C++ oleh James Gosling [8]. Java merupakan bahasa pemrograman objek murni karena semua kode programnya dibungkus dalam kelas. Java disebut sebagai bahasa pemrograman yang *portable* karena dapat dijalankan pada berbagai sistem operasi asalkan terdapat JVM (Java Virtual Machine) [5].

2.7 Swing

Swing adalah bentuk implementasi *user interface* yang menambahkan komponen-komponen dalam sistem GUI. Terdapat tiga buah *container* tingkatan paling atas dalam pengembangan GUI pada Java yaitu *Applet*, *Dialog*, dan *Frame*. *Frame* yang diimplementasikan dengan kelas *JFrame* digunakan untuk membuat *window* dan *form* dalam aplikasi desktop [9].

2.8 Black Box Testing

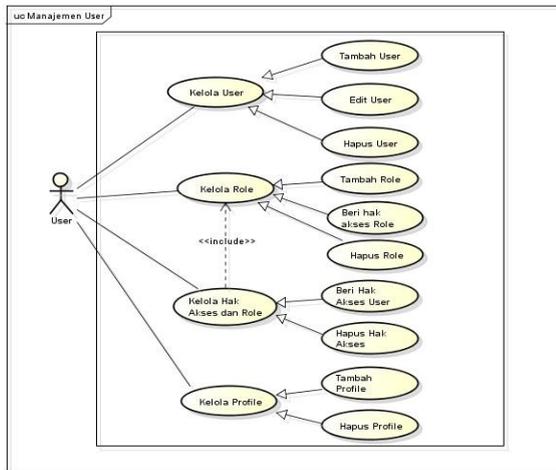
Black box merupakan kotak yang menentukan tingkah laku sistem atau suatu bagian dari suatu sistem. Sistem atau bagian yang merespon stimulus (*event*) spesifik dengan mengaplikasikan serangkaian transisi yang memetakan stimulus tersebut ke dalam suatu respon [10]. *Black box* testing adalah suatu pengujian yang berfokus terhadap fungsionalitas yang dibuat suatu sistem.

3. Analisis Kebutuhan dan Perancangan

3.1 Analisis Kebutuhan Fungsionalitas

a. Use case Diagram

Berikut adalah Use Case Diagram untuk aplikasi yang akan dibangun.



Gambar 2 Use Case Diagram

b. Deskripsi Use case

Berikut adalah deskripsi dari pendefinisian use case diatas.

Tabel 2 Deskripsi Use case

Nama Use Case	Deskripsi
Kelola User	Mengelola data <i>User</i>
- Tambah <i>User</i>	Menambahkan <i>user</i> baru
- Edit <i>User</i>	Mengedit atau mengubah atribut-atribut <i>user</i>
- Hapus <i>User</i>	Menghapus <i>user</i> yang telah dibuat
Kelola Role	Mengelola data <i>Role</i>
- Tambah <i>Role</i>	Menambahkan <i>role</i> baru
- Beri hak akses <i>role</i>	Memberikan hak akses (<i>privileges</i>) terhadap <i>role</i>
- Hapus <i>Role</i>	Menghapus <i>role</i> yang telah dibuat
Kelola Hak Akses	Mengelola hak akses terhadap <i>user</i>
- Beri hak akses <i>user</i>	Memberikan hak akses (<i>privileges</i>) terhadap <i>user</i>
- Hapus hak akses <i>user</i>	Menghapus atau mencabut hak akses dari <i>user</i>
Kelola Profile	Mengelola data <i>profile</i>

- Tambah <i>profile</i>	Menambah <i>profile</i> baru
- Hapus <i>profile</i>	Menghapus <i>profile</i> yang telah dibuat

3.2 Analisis Kebutuhan Perangkat Keras

Berikut ini adalah perangkat keras yang digunakan dalam perancangan dan pengembangan aplikasi.

Tabel 3 Kebutuhan Perangkat Keras

No	Perangkat Keras	Spesifikasi yang Digunakan
1.	Laptop	Acer Travelmet
2.	Processor	Intel Core i3
3.	RAM	4 GB
4.	Harddisk	500 GB

3.3 Analisis Kebutuhan Perangkat Lunak

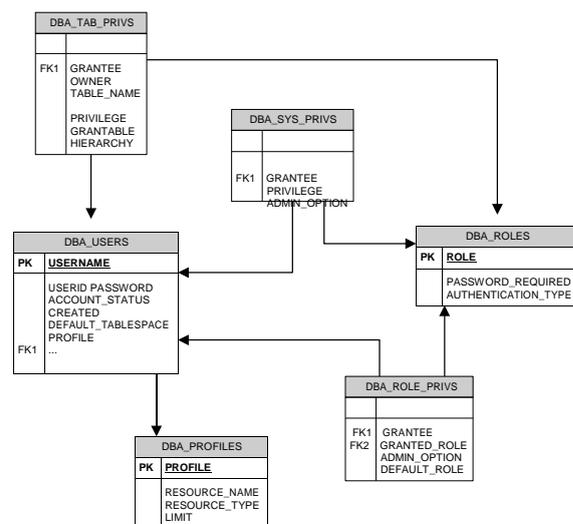
Berikut ini adalah perangkat lunak yang digunakan dalam perancangan dan pengembangan aplikasi.

Tabel 4 Kebutuhan Perangkat Lunak

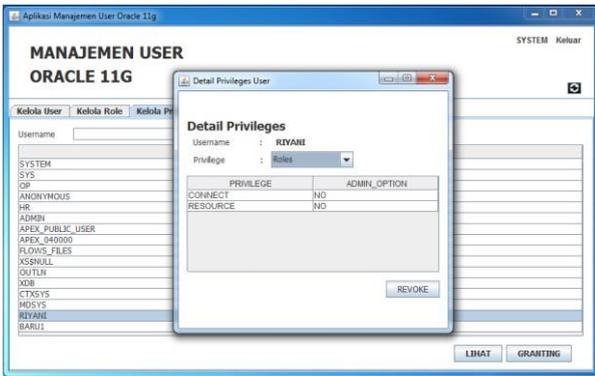
No	Perangkat Lunak	Spesifikasi yang Digunakan
1.	Sistem Operasi	Windows 7
2.	Editor	Netbeans IDE 7.2 , Astah Professional
3.	Database	Oracle 11g

3.4 Skema Relasi Metadata

Berikut ini adalah relasi antara metadata yang digunakan.



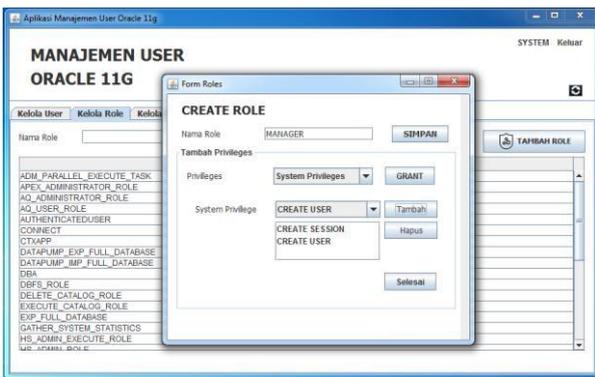
Gambar 3 Skema Relasi Metadata



Gambar 9 Halaman detail Privileges

4.4 Halaman Kelola Roles

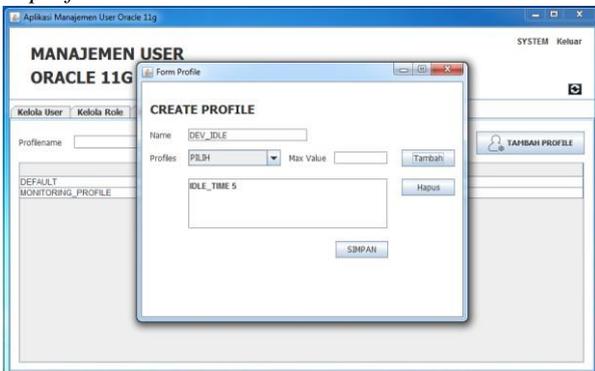
Untuk mengelola *roles* maka memilih menu Kelola Roles. Pada halaman menu tersebut tampil informasi data *roles* yang telah ada. Untuk menambahkan *roles* maka memilih *button* "TAMBAH ROLE" dan akan tampil halaman untuk *form roles*. Kemudian menekan *button* "SIMPAN" untuk menyimpan *role* tersebut. Setelah itu kan tampil form untuk Tambah Privileges dan terdapat *button* "GRANT" untu memberikan *privilege* yang telah dipilih untuk *role* tersebut.



Gambar 10 Halaman Form Roles

4.5 Halaman Kelola Profiles

Untuk mengelola *profiles* maka memilih menu Kelola Profile. Pada halaman menu tersebut tampil informasi data *profiles* yang telah ada. Untuk menambahkan *profile* maka memilih *button* "TAMBAH PROFILE"



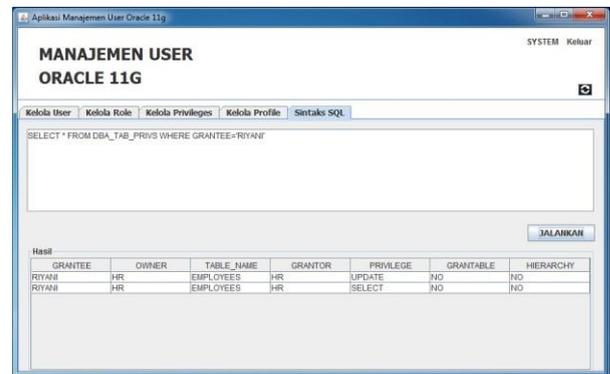
Gambar 11 Halaman Form Profiles

4.6 Halaman Sintaks SQL

Dalam melakukan manajemen user dengan menggunakan sintaks disediakan halaman untuk mengeksekusi sintaks tersebut yaitu pada menu Sintaks SQL. Untuk mengeksekusi sintaks yang telah diketikan dengan menekan *button* "JALANKAN".



Gambar 12 Halaman Sintaks SQL(1)



Gambar 13 Halaman Sintaks SQL(2)

4.7 Evaluasi Perbandingan Manajemen User antara Perangkat Lunak Manajemen User dan Oracle Database Express Edition

Berikut ini adalah tabel perbandingan antara perangkat lunak yang telah dibangun dengan Oracle Database Express Edition berdasarkan fungsionalitas untuk manajemen user.

Tabel 5 Perbandingan Manajemen User Berdasarkan Fungsionalita

Fungsionalitas	Perangkat Lunak Manajemen User	Oracle Database Express Edition
1. Mengelola User		
- Tambah	✓	✓
- Edit	✓	✓
- Hapus	✓	✓
2. Mengelola Role		
- Tambah	✓	✓

- Grant Privileges	✓	✓
- Hapus	✓	✓
3. Mengelola Privileges		
- Grant Privileges		
a. System Privileges	✓	✓
b. Object Privileges	✓	-
c. Roles	✓	✓
- Hapus Privileges	✓	✓
d. Mengelola Profile		
- Tambah	✓	-
- Hapus	✓	-
e. Koneksi ke remote server	✓ (tanpa harus mengedit TNSname.ora)	✓ (dengan mengedit TNSname.ora)

5. Penutup

5.1 Kesimpulan

Dalam Proyek akhir ini telah berhasil dibangun sebuah aplikasi untuk manajemen *user* yang menyediakan antarmuka *Graphical User Interface* (GUI) untuk mengelola *user*, *roles*, pemetaan antara *roles* dan *privileges* dengan *user* dan *object database* lainnya serta mengelola *profile user* dengan tidak harus menuliskan *syntax*. Selain itu, masih disediakan pula *SQL Command* untuk melakukan pengelolaan dengan *syntax*.

5.2 Saran

Dalam pembangunan perangkat lunak diperlukan juga sebuah pengembangan lebih lanjut yang diharapkan dapat :

1. Memperbaiki tampilan agar aplikasi lebih *user friendly*.
2. Memperbanyak varian dari *object privileges* dan *resource* pada *profile*.
3. Menangani *dragging pointer* dan *lead memory* untuk meningkatkan kinerja perangkat lunak.

Daftar Pustaka

- [1] I.Sommerville, Software Engineering (Rekayasa Perangkat Lunak)/Edisi 6/Jilid I, Jakarta: Erlangga, 2003.
- [2] B. Bryla, Oracle Database Foundations, London: Sybex, 2003.
- [3] B. I. Hutabarat, Panduan Oracle 8i Untuk Administrator Database, Jakarta: PT. Elex Media Komputindo, 2003.
- [4] Jogianto, Analisis dan Desain Sistem Informasi, Yogyakarta: Andi Offset, 1990.
- [5] Rosa A.S and M. Shalahuddin, Modul Pembelajaran Rekayasa Perangkat Lunak (Terstruktur dan Berorientasi Objek), Bandung: Modula, 2011.
- [6] A. Agung Yulianto, I. Gartina, R. Astuti, S. Dewi, S. Komala Sari and W. Witanti, Analisis Desain dan Sistem Informasi, Bandung: Politeknik Telkom, 2009.
- [7] R. Greenwald, Beginning Oracle Application Express, Canada: Wiley Publishing, 2009.
- [8] A. Nugroho, Rational Rode untuk Pemodelan Berorientasi Objek, Bandung: Informatika, 2005.
- [9] B. Raharjo, I. Heryanto and A. Haryono, Mudah Belajar Java, Bandung: Informatika, 2012.
- [10] R. S. Pressman, Rekayasa Perangkat Lunak: Pendekatan Praktisi (Buku II), Yogyakarta: Andi, 2002.

