

Perangkat Lunak Manajemen Subprogram Untuk Oracle 11G

Iqbal Angga Dwi Putra¹, Wahyu Hidayat², Ely Rosely³
^{1,2,3}Manajemen Informatika Universitas Telkom

¹iqbalanggadwip@gmail.com, ²wahyuhidayat@tass.telkomuniversity.ac.id,

³ely.rosely@tass.telkomuniversity.ac.id

Abstrak

Pada saat ini, untuk membuat, mengedit, ataupun menghapus suatu subprogram dari Oracle diantaranya *procedure*, *function*, dan *trigger*, pengguna harus dapat mengefisienkan waktu. Selain itu, pengguna lebih menyukai interface GUI dibanding dengan console atau cmd dan pengguna juga harus teliti dalam mengetikkan sintaks subprogram tersebut. Karena, walaupun ada sedikit kesalahan pada sintaks, akan memunculkan pesan error dari sintaks tersebut. Untuk mengatasi permasalahan tersebut, aplikasi yang dibuat akan dapat membantu pengguna dalam mengatur atau mengelola subprogram yang terdapat DBMS, khususnya Oracle. Karena antar-muka GUI (Graphic User Interface), pengguna atau user dimudahkan untuk dapat membuat, mengedit, dan menghapus sebuah subprogram. Dalam aplikasi ini diberikan fitur pengecekan subprogram jika terjadi kesalahan sintaks, yaitu dengan cara menampilkan pesan error saat pengguna menekan tombol simpan. Selain itu, pada saat pengguna atau user menyimpan sintaks subprogram ke database, secara otomatis aplikasi akan langsung menyimpan sintaks tersebut ke dalam file dengan tipe *.txt* pada drive yang diinginkan. Untuk subprogram jenis *trigger*, aplikasi menyediakan fitur kelola status subprogram untuk mengaktifkan atau menon-aktifkan *trigger*.

Kata kunci: DBMS (Database Management System), GUI (Graphic User Interface), Perangkat Lunak, Subprogram

Abstract

*Recently, to create, edit, or delete a subprogram such procedures, functions, and triggers, user should be able to minimize the time. In addition, users prefer a GUI interface compared with console or cmd and users should also be careful in typing the syntax of the subprogram. Because, although there are slight errors in syntax, system will display an error message from the syntax. To solve these problems, this applications created to assist the user in organizing or managing subprogram contained DBMS, especially Oracle. Because of the interface GUI (Graphic User Interface), a user are permitted to be able to create, edit, and delete a subprogram. The application have checking feature subprogram in case of a syntax error by displaying an error message when a user presses the save button. In addition, when a user store subprograms syntax to the database, the application will automatically save the syntax directly into the files of type *.txt* on the desired drive. For subprogram type of trigger, the application provides management features subprogram status to enable or disable a trigger.*

Keywords: DBMS (Data Base Management System), GUI (Graphic User Interface), Software, Subprogram

1. Pendahuluan

Basis data Oracle adalah basis data relasional yang terdiri dari kumpulan data dalam suatu sistem manajemen basis data RDBMS. Salah satu fitur lain dari Oracle adalah membuat sebuah subprogram atau PL/SQL dimana Subprogram atau PL/SQL adalah kumpulan perintah yang *independent* (berdiri sendiri) dan dijalankan saat dipanggil oleh main program atau subprogram lain. Subprogram sendiri sudah banyak dipakai oleh pengguna atau user dalam menangani suatu proses bisnis yang terkomputerisasi.

Pada saat ini, untuk membuat, mengedit, ataupun menghapus suatu subprogram diantaranya *procedure*, *function*, dan *trigger*, pengguna harus dapat mengefisienkan waktu. Selain itu, pengguna lebih menyukai interface GUI dibanding dengan *console* atau cmd dan pengguna juga harus teliti dalam mengetikkan sintaks subprogram tersebut. Karena,

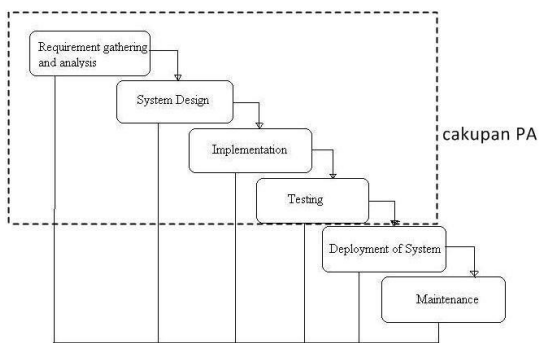
walaupun ada sedikit kesalahan pada sintaks, akan memunculkan pesan *error* dari sintaks tersebut. Khususnya untuk *trigger*, kebutuhan dari *trigger* itu sendiri seharusnya bisa ditangani, dimana agar *trigger* bisa diaktifkan atau dinonaktifkan. Selain itu, kebutuhan untuk menyimpan atau *backup* sintaks subprogram masih belum ada. Fatalnya jika subprogram yang di *database* terhapus, pengguna harus mengetikkan kembali subprogramnya.

Oleh karena itu, dalam proyek akhir ini akan dibangun sebuah aplikasi atau perangkat lunak yang dapat membantu pengguna dalam membuat, melihat, mengedit, menghapus sebuah subprogram, dan dapat menyimpan atau *backup* sintaks subprogram secara GUI (*Graphic User Interface*) sehingga pengguna tidak perlu banyak mengingat sintaks subprogram yang panjang. Selain itu, aplikasi ini juga merupakan salah satu bagian dari *tools* yang sedang

dikembangkan oleh lab *database* fakultas ilmu terapan dengan versi gratis atau *free*.

2. Metode Pengerjaan

Metode yang digunakan untuk mengerjakan proyek akhir ini adalah metode SDLC (*Software Development Life Cycle*) dengan model Waterfall. Menurut [6], Waterfall adalah metode pengembangan perangkat lunak dengan pendekatan sekuensial.



Gambar 1 Metode Waterfall

a. Requirement and Analysis

Pada tahap ini, dilakukan analisis tentang kebutuhan fungsional yang harus dimiliki oleh perangkat lunak subprogram yang akan dibangun.

b. Sistem Design

Pada tahap ini, dilakukan pembuatan desain terhadap perangkat lunak subprogram. Tahap ini mentranslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya. Desain aplikasi yang dibuat menggunakan teknik analisis dan pemrograman berorientasi objek. Rancangan tersebut menghasilkan diagram *Use Case*, diagram *sequence*, dan diagram *class*.

c. Implementation

Implementasi bahasa pemrograman pada aplikasi ini menggunakan bahasa pemrograman java. Hasil dari tahap ini adalah aplikasi yang sesuai dengan desain yang sebelumnya dibuat pada tahap desain.

d. Testing

Pengujian fokus pada perangkat lunak secara dari segi logik dan fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan. Untuk pengujian fungsionalitas

aplikasi menggunakan metode *black-box* dan untuk melihat respon pengguna menggunakan metode UAT (*User Acceptance Testing*).

3. Analisis

3.1 Kebutuhan Perangkat Keras

Pada Tabel 1 terdapat beberapa perangkat keras minimum yang dibutuhkan dalam pembangunan dan penggunaan aplikasi.

Tabel 1 Kebutuhan Perangkat Keras

No	Perangkat Keras	Spesifikasi Minimum	Spesifikasi Rekomendasi	Spesifikasi yang digunakan
1.	RAM	1GB	2GB	4GB
2.	Hardisk	20GB	100GB	320GB
3.	Processor	Dual Core	Core 2 Duo	Core i3

3.2 Kebutuhan Perangkat Lunak

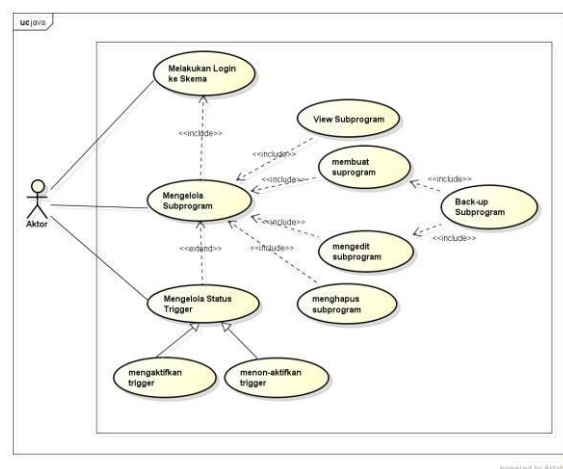
Pada Tabel 2 terdapat beberapa perangkat lunak minimum yang dibutuhkan dalam pembangunan dan penggunaan aplikasi.

Tabel 2 Kebutuhan Perangkat Lunak

No	Perangkat Lunak	Spesifikasi Minimum	Spesifikasi yang digunakan
1.	Sistem Operasi	Windows 7	Windows 7
2.	DBMS	Oracle 11g	Oracle 11g
3.	Script Editor	Netbeans IDE 7.0	Netbeans IDE 8.2
4.	Platform Java	Java 7	Java 8

3.3 Usecase Diagram

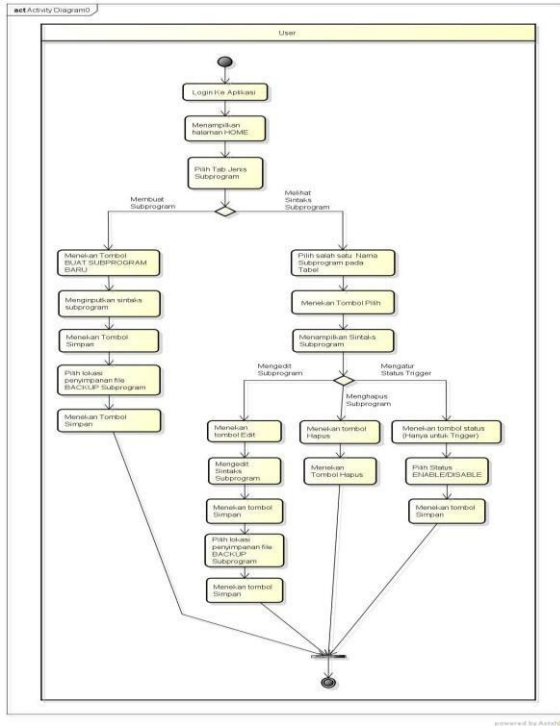
Berikut merupakan rancangan *usecase diagram*.



Gambar 2 Usecase Diagram

3.4 Diagram Aktifitas

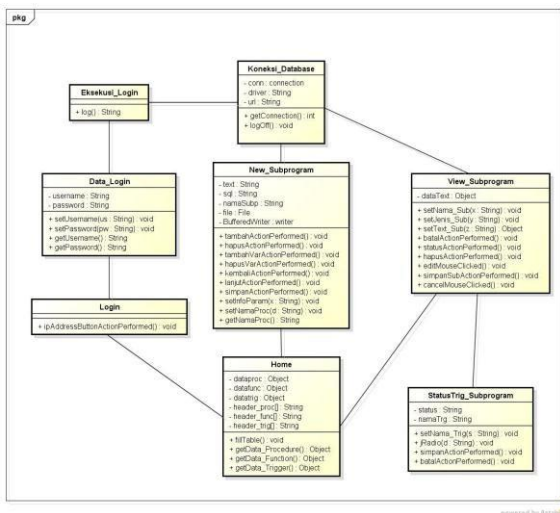
Dalam proses mengelola subprogram, terdapat beberapa tahapan yang harus dilewati. Gambar 3 menunjukkan diagram aktivitas untuk proses mengelola subprogram.



Gambar 3 Diagram Aktifitas Perangkat Lunak

3.5 Diagram Class

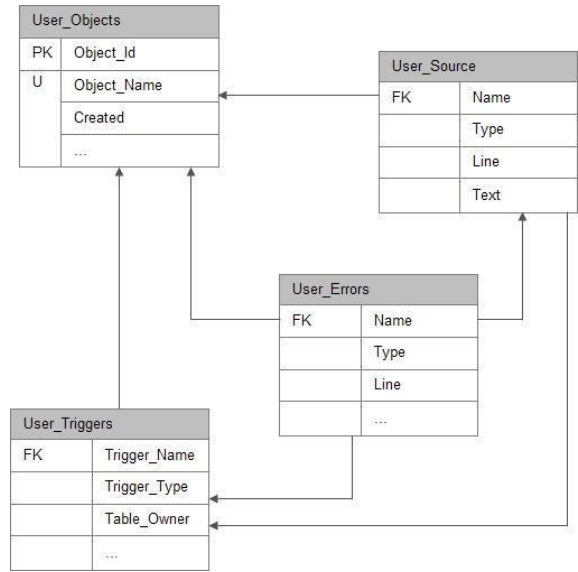
Berikut merupakan rancangan class diagram.



Gambar 4 Diagram Class

3.6 Skema Relasi Antar Tabel

Berikut merupakan skema relasi antar tabel.



Gambar 5 Skema Relasi Antar Tabel

4. Pengujian

Berikut merupakan hasil pengujian dari aplikasi..

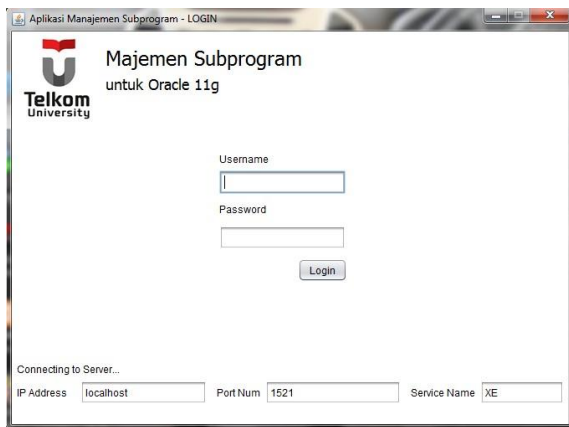
Tabel 3 Pengujian

No	Prosedur	Skenario	Status
1.	Login Ke Skema	Input <i>Username</i> dan <i>Password</i>	Success
2.	Lihat Daftar Subprogram	Input Jenis Subprogram	Success
3.	Lihat Sintaks Subprogram	Pilih Subprogram	Success
4.	Membuat Subprogram	Input Sintaks Subprogram	Success
5.	Mengedit Subprogram	Input Sintaks Subprogram	Success
6.	Menghapus Subprogram	Pilih Tombol Hapus Subprogram	Success
7.	Kelola Status Subprogram	Input Jenis Status Subprogram	Success
8.	Kelola Backup Subprogram	Pilih Tombol Simpan	Success

5. Tampilan Antarmuka

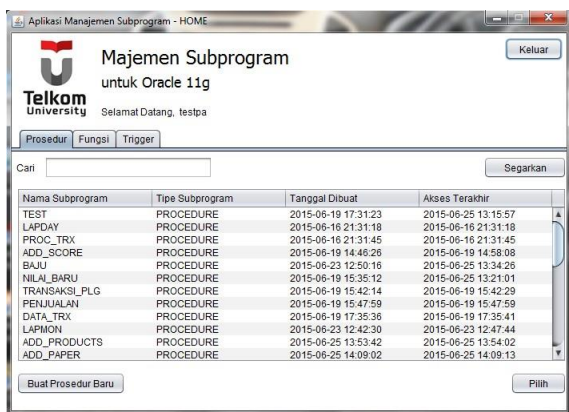
Berikut merupakan tampilan antarmuka aplikasi.

5.1 Tampilan Antarmuka Login



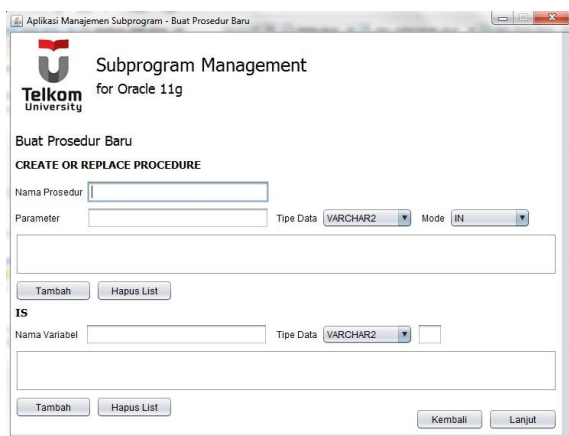
Gambar 6 Tampilan Login Perangkat Lunak

5.2 Tampilan Antarmuka Dashboard



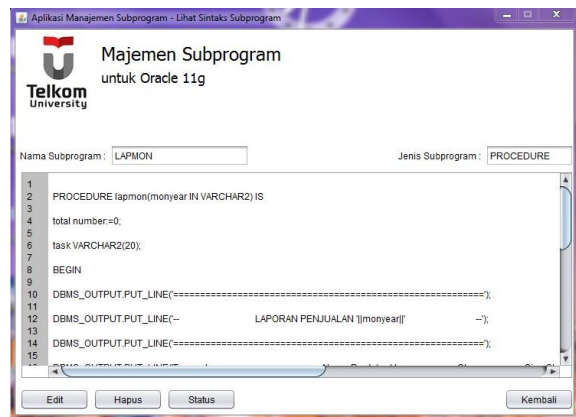
Gambar 7 Tampilan Antarmuka Dashboard

5.3 Tampilan Antarmuka Membuat Subprogram



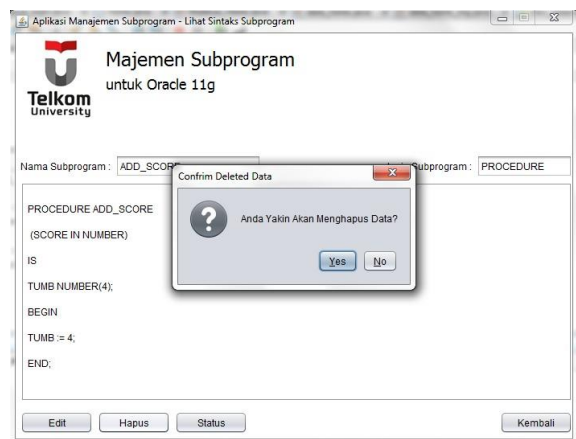
Gambar 8 Tampilan Antarmuka Membuat Subprogram

5.4 Tampilan Antarmuka Mengedit Subprogram



Gambar 9 Tampilan Antarmuka Mengedit Subprogram

5.5 Tampilan Antarmuka Menghapus Subprogram



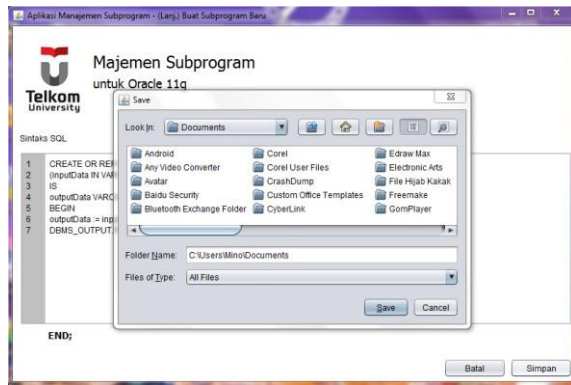
Gambar 10 Tampilan Antarmuka Menghapus Subprogram

5.6 Tampilan Antarmuka Kelola Status Subprogram



Gambar 11 Tampilan Antarmuka Kelola Status Subprogram

5.7 Tampilan Antarmuka Kelola Backup Subprogram



Gambar 12 Tampilan Antarmuka Kelola Backup Subprogram

6. Daftar Pustaka

- [1] R. A. S and M. Shalahuddin, Modul Pembelajaran Rekayasa Perangkat Lunak (Terstruktur dan Berorientasi Objek), Bandung: Modula, 2011.
- [2] R. S.Pressman, Rekayasa Perangkat Lunak : Pendekatan Praktisi (Buku II), Yogyakarta: Andi, 2002.
- [3] R. A. S., Modul Pembelajaran Pemograman Berorientasi Objek, Bandung: Modula, 2010.
- [4] A. A. Yulianto, I. G. H, K. R. Sari, R. Astuti and W. Witanti, Praktikum Analisis dan Desain Sistem Informasi, Bandung: Politeknik Telkom, 2009.
- [5] B. I. Heryanto, Menguasai Oracle SQL dan PL/SQL, Bandung: Informatika, 2009.
- [6] I. Yuniar, Sistem Manajemen Basis Data, Bandung: Politeknik Telkom, 2008.

