

BAB 1

PENDAHULUAN

1.1 Latar Belakang

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*.

1.2 Rumusan Masalah

Adapun yang menjadi rumusan masalah dalam pengerjaan proyek akhir ini adalah :

- a. Bagaimana cara mengelola (membuat, mengedit, menghapus) subprogram?
- b. Bagaimana cara pengguna bisa melihat subprogram yang ada dalam *database* Oracle 11g?
- c. Bagaimana cara pengguna mengubah status subprogram (*trigger*) menjadi aktif atau non-aktif?
- d. Bagaimana cara pengguna dapat menyimpan (*back-up*) subprogram yang ada pada *database*?

1.3 Tujuan

Adapun tujuan dari pembuatan proyek akhir ini adalah membuat sebuah aplikasi yang mampu :

- a. Mengelola (membuat, mengedit, menghapus) subprogram secara GUI (*Graphic User Interface*).
- b. Sebagai alternatif untuk menampilkan informasi subprogram serta sintaksnya secara keseluruhan dengan tampilan GUI (*Graphic User Interface*), tanpa harus mengetikkan sintaks *meta-data user*.
- c. Menambahkan fitur kelola status subprogram yang dapat mengubah status aktif dan non-aktif.
- d. Melakukan *back-up* subprogram ke file .txt secara otomatis pada saat menyimpan sintaks ke *database*.

1.4 Batasan Masalah

Beberapa hal yang menjadi batasan masalah dalam pembuatan proyek akhir ini adalah :

- a. Pengelolaan subprogram meliputi membuat subprogram baru, mengedit subprogram yang sudah ada di *database*, dan menghapus subprogram.
- b. Subprogram meliputi *procedure*, *function*, dan *trigger*.
- c. User yang dapat menggunakan aplikasi ini hanya *user* yang telah mempunyai hak akses dari *database*.
- d. Jenis subprogram yang bisa diaktifkan dan dinon-aktifkan hanya *trigger*.
- e. *Back-up* data subprogram hanya ke tipe file *.txt*
- f. Subprogram tidak di *compile* di aplikasi.

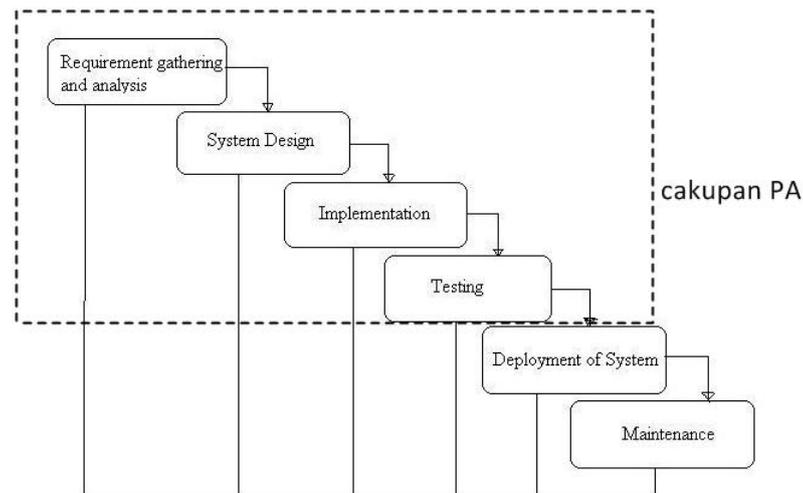
1.5 Definisi Operasional

Aplikasi yang akan dibangun merupakan aplikasi yang dapat membantu pengguna atau *user* 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 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*.

1.6 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 1-1 Pemodelan Waterfall

Berikut adalah tahapan dari penerapan metode Waterfall [6] dalam proyek akhir ini, diantaranya :

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 design 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*).

1.7 Jadwal Pengerjaan

Pengerjaan proyek akhir dimulai dari proses pendefinisian kebutuhan sampai dengan proses pengujian perangkat lunak atau aplikasi. Jadwal pengerjaan proyek akhir dapat dilihat pada tabel 1-1 berikut.

Table 1-1 Jadwal Pengerjaan

Kegiatan	Mar 2015				April 2015				Mei 2015				Juni 2015				Juli 2015			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Requirement and Analysis	█	█	█	█	█	█	█	█												
Sistem Design									█	█	█	█								
Implementation													█	█	█	█				
Testing																	█	█	█	█
Documentation	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█