

BAB 1

PENDAHULUAN

Pada bab ini menjelaskan mengenai latar belakang, identifikasi masalah, tujuan tugas akhir, lingkup tugas akhir, metodologi tugas akhir, dan sistematika penulisan tugas akhir.

1.1. Latar Belakang

Aplikasi sistem informasi pusat data desa (SIPUDES) merupakan aplikasi yang berfungsi untuk mengelola data penduduk yang menguni suatu desa serta dapat digunakan untuk menentukan peluang investasi bagi desa tersebut berdasarkan sumber daya manusia yang dimiliki oleh penghuni desa. Saat ini aplikasi sistem informasi pusat data desa telah digunakan di Kelurahan Ciumbuleuit, namun demikian pada aplikasi SIPUDES ditemukan beberapa *error* pada beberapa fungsi perangkat lunak, selain itu ada permintaan dari user untuk dilakukan penambahan fungsi pada aplikasi SIPUDES sehingga perlu dilakukan perawatan perangkat lunak.

Perawatan perangkat lunak berfokus pada perubahan yang dihubungkan dengan kesalahan, penyesuaian yang dibutuhkan ketika lingkungan perangkat lunak berkembang, serta perubahan sehubungan dengan perkembangan yang disebabkan oleh perubahan kebutuhan pelanggan [NUG10]. Menurut Boehm, biaya *maintenance* akan lebih besar dibandingkan dengan biaya *development* [BOE17]. Sehingga faktor *maintence* perlu diperhatikan. Salah satu teknik yang dilakukan dalam pencegahan terhadap kesalahan ataupun kemudahan kode yang mudah dibaca dan dipahami oleh *programmer* adalah dengan melakukan *refactoring*. Dalam melakukan *maintenance* suatu aplikasi biasanya akan timbul masalah yang disebabkan oleh struktur kode program buruk atau biasanya disebut dengan *bad smell code*. *Bad smell code* mengindikasikan adanya kelemahan struktur desain program yang dapat memperlambat proses *maintenance* dan menambah resiko *bug*. Dari sisi *programmer*, adanya *Bad smell code* akan menyebabkan kebingungan dalam memahami program dan bahkan seringkali *programmer* harus melakukan banyak hal serupa secara berulang-ulang. *Refactoring* mengizinkan *programmer* melakukan perbaikan-perbaikan pada kode program walaupun hal tersebut tidak direncanakan dalam tahap perancangan sistem. Perbaikan pada proses *refactoring* pada umumnya ditujukan untuk optimalisasi kualitas kode program dengan syarat tidak mengubah output dan proses dari perangkat lunak tersebut.

Refactoring merupakan salah satu aspek paling khas dari *Extreme Programming*. *Refactoring* dapat diartikan sebagai suatu proses memperbaiki struktur internal sebuah *sistem* perangkat lunak dengan tetap mempertahankan fungsionalitas dari sebuah *sistem* [FOW02]. Tujuan dari *refactoring* adalah membuat perangkat lunak lebih mudah dimengerti karena struktur diperbaiki, duplikasi kode bisa diperbaiki, membantu menemukan *bugs*, meningkatkan pemahaman terhadap kode dan pemahaman ini sangat membantu dalam menemukan dan mengantisipasi *bugs*, membuat program pada iterasi berikutnya menjadi lebih cepat [GAR00]. Hal ini secara luas diyakini bahwa *refactoring* meningkatkan kualitas dan produktivitas *software*. Dengan adanya *refactoring*, *programmer* dapat mempertahankan

fungsi dan kinerja *software* dan memahami kode dengan lebih mudah. Banyak yang mempercayai bahwa kurangnya pemanfaatan *refactoring* pada kode, suatu saat dapat menimbulkan permasalahan teknis dalam melakukan *maintenance*. Hal ini tentu saja berdampak pada biaya dan waktu *maintenance*. Oleh karena itu, pentingnya dilakukan *refactoring* agar kondisi diatas bisa dihindari dan *maintenance* perangkat lunak pun menjadi lebih cepat dan terarah.

Berdasarkan permintaan user di Kelurahan Ciumbuleuit untuk menyelesaikan permasalahan yang ada pada aplikasi sistem informasi pusat data desa, penulis mengusulkan untuk melakukan perawatan terhadap aplikasi sistem informasi pusat data desa dengan menggunakan teknik *refactoring*.

1.2. Identifikasi Masalah

Berdasarkan latar belakang yang telah dijelaskan sebelumnya, maka identifikasi masalah yang akan dibahas dalam pengerjaan tugas akhir ini adalah bagaimana melakukan *maintenance* perangkat lunak pada studi kasus aplikasi sistem informasi pusat data desa dengan menerapkan teknik *refactoring*.

1.3. Tujuan Tugas Akhir

Berdasarkan identifikasi masalah yang telah dipaparkan sebelumnya, maka tujuan yang ingin penulis capai pada pengerjaan tugas akhir ini adalah sebagai berikut:

1. Memperbaiki *bugs* pada aplikasi sistem informasi pusat data desa.
2. Memperbaiki struktur kode program menggunakan teknik *refactoring* agar memudahkan perawatan aplikasi dimasa yang akan datang.

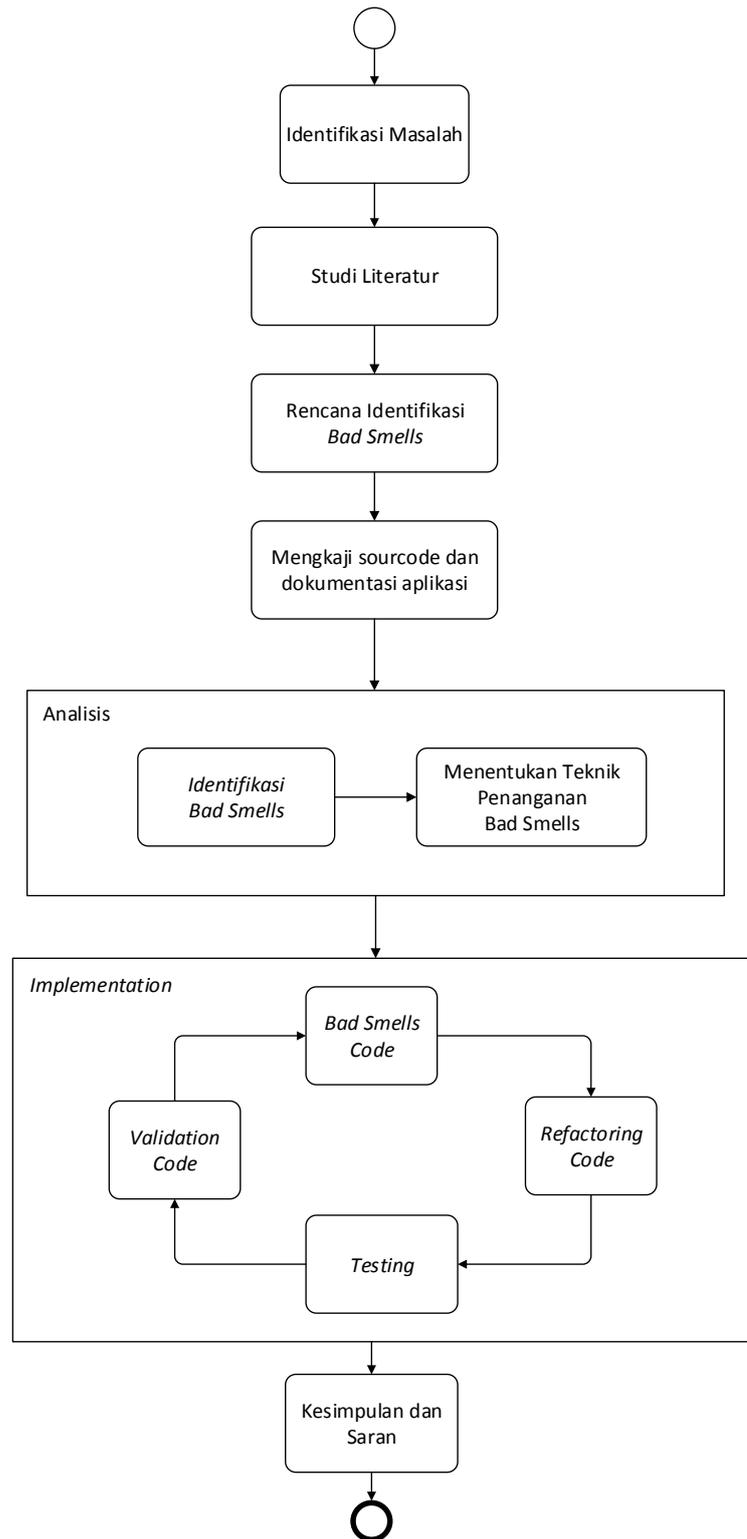
1.4. Lingkup Tugas Akhir

Adapun lingkup dan batasan materi yang akan dibahas dalam pengerjaan tugas akhir ini yaitu dapat dijelaskan sebagai berikut:

1. Tidak sampai melakukan penambahan fungsi pada aplikasi.
2. Pengkajian *source code* dari aplikasi yang sudah ada.
3. Menentukan bagian-bagian dari aplikasi yang akan dilakukan *refactoring*.
4. Dokumentasi yang dikaji hanya *user manual* dan dokumentasi internal seperti beberapa *comment* pada *source code*.
5. Aplikasi berbasis web dan menggunakan bahasa pemrograman PHP dan mysql sebagai DBMS-nya.

1.5. Metodologi Tugas Akhir

Untuk pengerjaan laporan tugas akhir ini penulis menentukan metode yang akan digunakan untuk pengerjaan tugas akhir. Gambar 1.1 menjelaskan mengenai metodologi yang digunakan penulis dalam mengerjakan tugas akhir.



Gambar 1.1 Metodologi Tugas Akhir

Dari gambar diatas dapat dijelaskan bahwa penelitian ini dimulai dari melakukan identifikasi masalah dan kemudian melakukan studi literatur yang berkaitan dengan *refactoring*, setelah itu membuat rencana identifikasi *bad smells* lalu mulai mengkaji *source code* dan dokumentasi dari aplikasi yang akan berguna untuk melakukan analisis mengenai identifikasi *bad smells code* dan pemilihan

teknik penangan dari *bad smells code* yang teridentifikasi. Setelah melakukan analisis kegiatan penelitian berlanjut pada implementasi penerapan *refactoring* pada *maintenance* perangkat lunak yaitu melakukan *list bad smells code*, kemudian *refactoring bad smells code* yang telah di list. Setelah dilakukan *refactoring* kemudian dilakukan testing pada program hingga menghasilkan *validation code* yang mudah dimengerti dan di-*maintenance* untuk kepentingan dimasa yang akan datang. Tahapan terakhir dari penelitian ini adalah penarikan kesimpulan berdasarkan analisis dan implementasi yang telah dilakukan serta saran bagi penelitian selanjutnya.

1.6. Sistematika Penulisan Tugas Akhir

Adapun sistematika di dalam penulisan tugas akhir ini akan dijelaskan sebagai berikut:

BAB 1 PENDAHULUAN

Bab pertama menjelaskan mengenai latar belakang dalam memilih topik bahasan, ruang lingkup dalam pembahasan, dan tujuan, metode penelitian yang digunakan, dan sistematika penulisan dalam tugas akhir.

BAB 2 LANDASAN TEORI

Bab kedua menjelaskan mengenai teori yang diperlukan untuk pengerjaan tugas akhir. Dalam hal ini memuat teori yang berkaitan dengan penerapan *refactoring* pada perangkat lunak, seperti pengertian rekayasa perangkat lunak, perawatan perangkat lunak, rekayasa ulang perangkat lunak, dan *refactoring*.

BAB 3 SKEMA PENELITIAN

Bab ini menjelaskan mengenai alur penyelesaian tugas akhir, peta analisis, kerangka pemikiran teoritis dan objek penelitian.

BAB 4 GAMBARAN UMUM SISTEM

Bab ini menjelaskan mengenai gambaran umum sistem secara menyeluruh seperti deskripsi sistem, manfaat sistem, karakteristik sistem serta struktur internal program.

BAB 5 MAINTENANCE DAN REFACTORING

Bab ini menjelaskan mengenai tahapan *maintenance dan refactoring* pada aplikasi yang menjadi studi kasus tugas akhir serta menjelaskan testing aplikasi tersebut

BAB 6 KESIMPULAN DAN SARAN

Bab ini akan menjelaskan mengenai kesimpulan dari hasil analisis dan implementasi yang telah dilakukan dan memberikan saran untuk mengatasi permasalahan yang terjadi pada proses *refactoring*.