

BAB I

PENDAHULUAN

A. Latar Belakang Masalah

Software Engineering (SE) merupakan sebuah kegiatan atau bisa dikatakan project pembangunan sebuah perangkat lunak (software). SE bisa merupakan pembuatan dari nol maupun sebuah perbaikan dari sebuah software yang sudah jadi. Secara umum SE merupakan sebuah rangkaian kegiatan yang prosedural dan terstruktur untuk menghasilkan output berupa sebuah software yang baik. SE ditekankan pada tahapan-tahapan sebagai berikut:

- ❖ Analisis
- ❖ Desain
- ❖ Implementasi
- ❖ Testing dan Maintenance

Pada tahap berikutnya SE mengacu pada Manajemen Proyek pengembangan Perangkat Lunak dengan jalan memperhatikan tahapan pengembangan sebelumnya dan membentuk sebuah lingkaran kerja pengembangan software yang berkelanjutan.

Software House sebagai perusahaan penyedia aplikasi tentu akan sangat membutuhkan software-software pembantu dalam SE, baik itu dibuat sendiri atau menggunakan software buatan perusahaan lain. Software-software ini disebut sebagai CASE tools, CASE merupakan singkatan dari *Computer Added Software*

Engginering yang dapat diterjemahkan sebagai software yang membantu dalam pembuatan software atau software untuk membuat software.

Rapid Aplication Developmen (RAD) Merupakan sebuah metode pembangunan software secara cepat dan tepat menggunakan software-software pembantu yang memadai. Pembangunan software secara cepat sangat dituntut saat ini didorong oleh kebutuhan terkini akan sebuah software yang mampu membantu perusahaan dalam menyelesaikan pekerjaannya secepat mungkin demi tercapainya target yang telah ditetapkan. Pembuatan software yang ditujukan untuk penyelesaian masalah saat ini tentu akan sangat tidak berguna jika tidak diselesaikan tepat pada waktunya.

B. Perumusan Masalah

Berdasarkan uraian di atas maka dapat dirumuskan beberapa masalah yang harus dipecahkan dalam tahapan SE khususnya pada tahapan penulisan code program :

- a. Bagaimana membuat suatu perangkat lunak (*software*) komputer yang dapat digunakan untuk membantu software engineer dalam tahap penulisan code program
- b. Seberapa jauh bantuan dari software CASE tool yang akan kita bangun.
- c. Untuk fungsi yang optimal, dapatkah kita membuat CASE tools ini sesuai dengan Standarisasi penulisan code program
- d. Untuk CASE pembangkit kode, apa saja input yang dibutuhkan.
- e. Dapatkah dibuat CASE tool kusus untuk pengkodean aplikasi database.

C. Hipotesis

Sebagai tool dalam pembuatan code program yang bertujuan untuk meningkatkan produktifitas software enginner, ada beberapa hipotesa awal jika tool ini digunakan yang antara lain sebagai berikut:

1. Peningkatan Kecepatan programming. Sebagai contoh untuk membuat sebuah form master data untuk sebuah tabel dengan 10 field diperlukan waktu 3 jam untuk desain, penyusunan object-object, programming dan debugging. Diharapkan nantinya bisa dikurangi menjadi 20 menit dengan dibantu dalam tahap-tahap selain debugging.
2. Penyeragaman style programming. Dengan penggunaan tool ini diharapkan model output source code dari para programmer bisa diseragaman atau minimal tidak terlalu jauh berbeda antara satu orang dengan lainnya.
3. Memudahkan transfer Knowledge dan pekerjaan. Dengan source code yang seragam sebagai mana disebutkan dalam point 2 diharapkan jika ada programmer baru atau pemindahan pekerjaan dari satu orang ke lainnya proses adaptasinya bisa menjadi mudah, cepat dan menghindari kesalahan pemahaman dalam pembacaan alur code.
4. Meningkatkan dan menstabilkan mutu code. Perintah-perintah falidasi input yang memang cukup banyak untuk menjamin output aplikasi yang *bugs free* kadang dilewatkan oleh programmer (lupa, tidak tau atau tidak disiplin), dengan tool ini perintah-perintah tersebut dibuatkan otomatis sehingga bisa terhindar dari factor human error tersebut.

D. Batasan Masalah

Karena dalam pembuatan sebuah software, software engineer dapat menggunakan berbagai Software Developmen Kit (SDK) sesuai dengan keahlian masing-masing dan jika software tersebut merupakan software database maka akan ada begitu banyak pilihan software Relational Database Managemen System (RDBMS) yang dapat digunakan. Oleh karena itu dalam pembahasan ini hanya akan dibatasi pada:

1. Hanya akan memilih sebuah SDK dan dipadukan dengan sebuah RDBMS yang familiar.
2. Dan untuk pembatasan lingkup pembahasan maka kita akan memilih Microsoft Visual Basic 6.0 sebagai SDK dan SQL Server 2000 sebagai RDBMS.
3. Kemudian tahapan SE yang akan dibantu oleh CASE tool yang akan dibangun ini hanyalah pada tahap pengkodean program.
4. Input dari sistem ini adalah Database Schema dari database SQL Server 2000 yang diakses dengan SQL-DMO
5. Output Berupa source code Form dalam bahasa Visual Basic 6.0

Dari batasan masalah yang kemukakan maka penulis mengambil judul "CASE Tool Pembangkit Kode VB 6.0 Dari Skema Database SQL Server 2000 Dengan SQL-DMO.

E. Tujuan Penelitian

Sesuai dengan rumusan masalah yang telah dikemukakan, tujuan dari penulisan ini adalah:

- a. Menciptakan sebuah software baru, yang mampu menganalisis struktur database (db schema) dan kemudian menerjemahkannya menjadi sebuah software aplikasi database untuk database yang bersangkutan secara otomatis.
- b. Mencari sebuah metode baru dalam penulisan code program yang valid, stabil dan sesuai dengan style kita sendiri.
- c. Menemukan cara bagaimana menyimpan hasil kerja Software Engineer berupa source code yang terbaik, menjadi sebuah paket yang dapat dipergunakan kembali sebagai sebuah template yang standar yang mudah dipergunakan kembali dikemudian waktu.

Dan dari semua tujuan ini penulis berharap dapat membukakan jalan kepada pembaca dalam upaya meningkatkan kinerja sebagai software engineer sesuai dengan bahasa programming masing-masing.

F. Manfaat Penelitian

Penulis berharap penelitian yang dilakukan ini dapat memberikan manfaat-manfaat positif yang antara lain sebagai berikut,

- a. Bagi penulis. Penulis berharap dapat membuat sebuah software yang dapat membantu dalam pekerjaan sehari-hari sebagai software engineering, yang dapat menyelesaikan dengan cepat source code yang rutin dan

memberikan waktu lebih untuk berkonsentrasi pada source code *transactional* yang sifatnya kusus dan baru.

- b. Bagi STMIK AMIKOM Yogyakarta, Semoga dapat menjadi motifikasi dan contoh yang baik bagi semua mahasiswa dilingkungan STMIK AMIKOM Yogyakarta dan menambah kecintaan kita semua pada dunia IT pada umumnya dan software engineering pada khususnya.
- c. Bagi ilmu pengetahuan. Penelitian ini diharapkan dapat menjadi sebuah rujukan bagi semua pihak yang berkerja dalam bidang SE yang berkeinginan untuk meningkatkan produktifitas software engineer. Meningkatkan kecerdasan pola pikir dan merubah metode kerja seorang software engineer dari yang sebelumnya bersifat konvensional dengan menulis semua bagian code menjadi sebuah cara kerja yang lebih otomatis.

G. Metode Penelitian

Untuk memperoleh data yang akurat dan dapat dipercaya, maka metode yang dipilih adalah dengan menggunakan teknik menemukan fakta (fact finding techniques), teknik yang dapat digunakan untuk mengumpulkan data dan menemukan fakta-fakta dalam kegiatan mempelajari sistem yang ada. Teknik-teknik ini diantaranya adalah :

1. Kuisisioner

Dengan cara memberikan pertanyaan tertulis yang dijawab secara tertulis tentang hal-hal yang berkaitan dengan pembahasan.

2. Wawancara

Dengan cara tanya jawab (dialog) dengan pihak yang terkait mengenai hal-hal yang bersifat spesifik atau susah untuk digambarkan karena bersifat abstrak atau subjektif.

3. Observasi

Dengan cara melakukan pengamatan langsung terhadap obyek yang bersangkutan sesuai dengan sudut pandang yang ingin kita ambil.

4. Kepustakaan

Dengan cara mengambil data/sample dari berbagai sumber yang berhubungan, dapat berupa literatur, teori, contoh code dan aturan-aturan standarisasi penulisan program.

H. SISTEMATIKA PENULISAN

Penyusunan laporan agar lebih mudah dimengerti dan tersusun dengan baik, maka akan disajikan dalam 5 (lima) bab yang masing-masing bab adalah sebagai berikut:

BAB I : PENDAHULUAN

Berisi tentang latar belakang masalah, perumusan masalah, batasan masalah, tujuan, metode pengumpulan data, dan sistematika penulisan laporan.

BAB II : DASAR TEORI

Bab ini memaparkan tentang pengenalan sistem (sesuai judul), sistem perangkat lunak yang digunakan, metode dan model yang dijadikan sebagai rujukan.

BAB III : Analisis dan Perancangan Sistem Informasi.

Dalam bab ini akan dijelaskan tentang masalah yang terjadi dalam sistem lama dari perpustakaan tersebut mulai dari pendefinisian masalah sampai dengan identifikasi masalah yang meliputi 7 permasalahan yang timbul dan penyebab masalah dan penetapan masalah. Analisis sistem informasi yang berjalan yang meliputi analisis kelemahan sistem, analisis PIECES (Performance, Information, Economi, Control, Efisiensi, Service), analisis biaya dan manfaat serta analisis kelayakan yang meliputi kelayakan teknologi, ekonomi, hukum operasi, dan sosial, serta rancangan sistem secara umum mulai dari rancangan model sampai dengan rancangan Database serta relasi antar tabel sampai dengan rancangan input dan rancangan outputnya.

BAB IV : Testing dan Implementasi

Dalam bab ini akan dijelaskan tentang penerapan perancangan implementasi yang meliputi kegiatan implementasi, pengujian dan uji sistem serta running program.

BAB V : Kesimpulan dan Saran.

Dalam bab ini berisi kesimpulan dari bab I sampai dengan bab V dan saran yang disusun ajukan guna pengembangan dunia IT pada umumnya dan Software Engineering pada khususnya.