

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Dewasa ini penggunaan aplikasi Basis data dalam proses bisnis sudah kian populer. Ini terlihat dari semakin banyak digunakannya aplikasi Basis data dalam membantu kegiatan bisnis sehari - hari. Mulai dari aplikasi POS (*Point Of Sale*) hingga Aplikasi Basis data Perbankan. Tentunya ini semua bukan tanpa sebab, karena di zaman yang serba instan dan cepat ini informasi untuk keperluan bisnis harus bisa didapat secara *realtime* dan akurat guna mendukung keputusan bisnis.

Adanya permintaan akan Aplikasi Basis data yang cukup tinggi membuat vendor – vendor software bermunculan, tak terkecuali vendor lokal. Menurut laporan research IDC (*International Data Corporation*) di Indonesia tercatat Jumlah *software house* atau *independent software vendor* (ISV) ada sekitar 250 pada tahun 2006, dan terus berkembang hingga mencapai 500 di tahun 2010. Vendor software ini nantinya akan bertugas untuk merancang aplikasi basis data untuk bisnis baik dalam skala kecil, menengah, hingga besar.

Dalam merancang aplikasi basis data biasanya vendor Software akan menerapkan pola desain tertentu guna memudahkan perawatan (*maintenance*), perbaikan (*bug fixing*), dan penambahan fitur (*update*) di masa – masa yang akan datang. Penerapan pola desain tertentu juga memudahkahkan komunikasi antar programmer karena adanya kesamaan *mindset*, dan juga membuat peralihan tugas menjadi lebih mudah.

Selain memberikan segudang manfaat, menerapkan pola desain tertentu juga membuat pekerjaan programmer menjadi lebih banyak dan rumit. Ini disebabkan programmer harus memecah kode – kode yang mereka tulis menjadi beberapa kelas - kelas terpisah yang saling berhubungan. Tentunya jika di kerjakan secara manual akan memakan waktu yang lama. Hal ini juga akan meningkatkan *development cost* bagi perusahaan karena harus memperkerjakan lebih banyak programmer jika ingin pekerjaan lebih cepat selesai.

Dengan melihat permasalahan di atas, penulis merasa perlu membuat sebuah aplikasi yang berfungsi men-*generate* kode – kode program yang selalu berulang dan mengikuti pola tertentu sehingga dapat menghemat waktu *programmer* dalam membuat aplikasi basis data. Dalam penelitian ini akan lebih dispesifikkan dalam membangun aplikasi form builder yang berfungsi men-*generate* kode – kode untuk Form operasi CRUD (*CREATE READ UPDATA DELETE*) pada sebuah aplikasi basis data dengan menggunakan pola desain MVP (*Model View Presenter*) Varian *Passive View*.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang masalah yang telah diuraikan sebelumnya, maka pokok permasalahan pada penelitian ini adalah sebagai berikut :

1. Bagaimana mendesain kode Form operasi CRUD pada aplikasi basis data dengan menggunakan pola arsitektur MVP (*Model View Presenter*) Varian *Passive View*.
2. Bagaimana membuat aplikasi *code generator* dengan Microsoft Visual Basic.net.

3. Bagaimana aplikasi yang dibuat dapat menghasilkan kode – kode Form operasi CRUD dengan menggunakan pola arsitektur MVP Varian *Passive View* secara otomatis sehingga dapat menghemat waktu *programmer* dalam membuat aplikasi basis data.
4. Bagaimana agar kode yang dihasilkan dapat terintegrasi dengan file *solution* Microsoft Visual Basic.net.

### 1.3 Batasan Masalah

Agar penelitian ini dapat dilakukan lebih mendalam dan tidak meluas, maka penulisan skripsi hanya akan difokuskan pada :

1. Aplikasi hanya menghasilkan kode program form operasi CRUD untuk satu tabel yaitu kode untuk menambah, merubah, dan menghapus data dari satu tabel dalam sebuah basis data.
2. Aplikasi tidak menghasilkan kode program form operasi CRUD untuk lebih dari satu tabel seperti pada form transaksi ataupun master detail.
3. Aplikasi juga tidak menghasilkan kode form laporan, login, atau form – form lain yang tidak mengandung operasi CRUD.
4. Aplikasi hanya menghasilkan kode untuk mengakses data dari *database* MySQL Server.
5. Kode yang dihasilkan hanya kode dalam bahasa pemrograman visual basic.net
6. File *solution* yang dihasilkan hanya compatible dengan microsoft visual studio versi 2010 ke atas.

7. Pola desain (*Design Pattern*) yang digunakan adalah MVP (*Model view Presenter*) Varian *Passive View*.

#### 1.4 Tujuan

Tujuan pembuatan skripsi ini adalah :

1. Membuat aplikasi Form Builder yang dapat mengenerate kode - kode Form operasi CRUD pada sebuah aplikasi basis data dengan menggunakan pola desain MVP (*Model View Presenter*) Varian *Passive View* secara otomatis.
2. Sebagai syarat kelulusan program "S1" serta memperoleh gelar "Sarjana" di Sekolah Tinggi Manajemen Informatika dan Komputer "AMIKOM" Yogyakarta pada jurusan Teknik Informatika.

#### 1.5 Manfaat

1. Secara teoritis penelitian ini diharapkan dapat menambah serta memperkaya khasanah ilmu pengetahuan, terutama pada bidang *Object Oriented Programming* dan *Software Design Pattern*.
2. Secara praktis penelitian ini diharapkan dapat menghasilkan solusi berupa sebuah perangkat lunak yang dapat membantu otomatisasi terutama pada pembuatan aplikasi basis data.

#### 1.6 Metode Penelitian

Metode penelitian merupakan salah satu hal yang sangat penting dalam penyusunan skripsi ini. Untuk mendapatkan hasil sesuai yang diinginkan maka metode penelitian yang benar, akurat dan lengkap sangat diperlukan dalam

penyusunan skripsi ini. Penelitian ini dilakukan dengan melalui tahapan-tahapan sebagai berikut:

1. Metode Kepustakaan (*Library*). Metode yang digunakan untuk mendapatkan konsep- konsep yang akurat menggunakan buku-buku sebagai bahan referensi dalam mendapatkan informasi-informasi yang dibutuhkan.
2. Analisis Data. Menganalisis lebih mendalam dari data-data yang telah didapat.
3. Perancangan Program. Dilakukan sebagai gambaran dan acuan dalam desain program selanjutnya.
4. Pembuatan Laporan. Menyusun laporan skripsi dan permasalahan yang sudah ada secara sistematis, diambil dari permasalahan yang sudah di analisa.
5. Ujicoba Program. Untuk memastikan bahwa aplikasi yang dibuat tersebut sudah berjalan dengan baik sesuai dengan yang diharapkan.

### 1.7 Sistematika Penulisan

Untuk memberikan gambaran dan kerangka yang jelas mengenai pokok bahasan dalam setiap bab dalam penelitian ini maka diperlukan sistematika pembahasan. Berikut gambaran sistematika pembahasan pada masing – masing bab :

## BAB I : PENDAHULUAN

Bab ini berisi latar belakang masalah, rumusan masalah, batasan masalah, tujuan, manfaat penelitian, dan sistematika penulisan.

## BAB II : LANDASAN TEORI

Bab ini menjelaskan tentang teori yang berhubungan dengan permasalahan penelitian.

## BAB III : ANALISIS DAN PERANCANGAN SISTEM

Pada bab ini dibahas mengenai analisis rancangan program dan langkah pembuatan perangkat lunak atau aplikasi.

## BAB IV : IMPLEMENTASI DAN PEMBAHASAN

Meliputi hasil yang dicapai dari perancangan sistem dan implementasi program. Sehingga dapat ditarik suatu kesimpulan.

## BAB V : PENUTUP

Berisi kesimpulan dan saran dari hasil yang dicapai sehingga dapat digunakan sebagai bahan pertimbangan bagi pihak – pihak yang berkepentingan serta kemungkinan pengembangannya.