

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Kemajuan teknologi informasi yang semakin berkembang berbanding lurus dengan perkembangan aplikasi yang mana aplikasi tersebut dapat meningkatkan kualitas dari informasi itu sendiri. Seiring perkembangan zaman, aplikasi untuk sistem informasi terus dikembangkan di berbagai *platform*, mulai dari aplikasi *desktop*, *web*, maupun *mobile*. Dalam hal ini, aplikasi berbasis *web* unggul dalam hal aksesibilitas, karena dapat diakses di mana-mana dan dapat berjalan di semua sistem operasi tanpa instalasi apapun.

Selain fungsional, kemudahan dan kecepatan dalam pengembangan sebuah aplikasi adalah hal yang harus diutamakan karena setiap orang ingin membuat suatu aplikasi dengan praktis dan cepat. Terkadang untuk beberapa aplikasi yang akan dibuat biasanya dibutuhkan beberapa data dimana data tersebut dapat ditambah, diubah, dicari, dan dihapus.

Saat ini sebuah mesin sudah banyak dikembangkan baik secara perorangan ataupun kelompok, dan dapat dengan mudah ditemukan melalui media internet. Pengembangan sebuah mesin tersebut dapat dengan mudah kita cari mulai dari yang sifatnya *ready-to-use* sampai dengan mesin yang masih memerlukan *code-programming* agar bisa digunakan sesuai dengan keinginan pengguna. Tentu saja kondisi tersebut sangat mudah dipahami, mengingat pengguna internet sangat

beragam, mulai dari yang hanya sekedar *browsing*, *chatting*, sampai dengan pengguna yang mempunyai keahlian dalam *programming*. Sehingga pembuat mesin tersebut tentu sudah memperkirakan sasaran dari rancangan aplikasi mesin yang akan dikembangkan. Otomatisasi adalah hal yang perlu diperhatikan dalam pengembangan sebuah aplikasi agar proses pengembangan aplikasi berjalan cepat. Oleh karena itu, perlu dibuatnya sebuah mesin dimana mesin tersebut dapat membuat program secara otomatis hingga membentuk sebuah aplikasi pengolah data dari *physical database* yang sudah tersedia. Dengan begitu, waktu yang dibutuhkan akan lebih sedikit dan tidak memakan banyak waktu yang terbuang sia-sia.

*Mesin Pembangkit Physical Data Model Menjadi Kode Program* adalah sebuah mesin berbasis 4GL (*Fourth Generation Language*) untuk membuat sebuah aplikasi *database* dengan memanfaatkan *database* fisik yang sudah ada sehingga orang dapat membuat sebuah aplikasi pengolah data hanya dengan mengkonfigurasi parameter-parameter dari aplikasi, seperti mengambil tabel-tabel yang ada di *database*, dibuatnya *kolom*, relasi basis data, dan parameter lainnya. Mesin berbasis 4GL karena aplikasi dapat dibuat tanpa menyentuh *coding* sekalipun, aplikasi akan secara otomatis membuat suatu program lengkap yang dapat menambah, membaca, mengubah, dan menghapus (*CRUD*) dan juga mencari hingga menampilkan *tabel* tertentu berdasarkan parameter-parameter yang sudah didefinisikan sebelumnya. Hal ini sangat menguntungkan bagi orang awam, karena orang awam yang bahkan sama sekali tidak mengenal bahasa

pemrograman pun dapat menggunakan aplikasi ini dengan baik untuk membuat sebuah aplikasi.

Dalam pembuatan mesin ini, diperlukan beberapa transformasi pemodelan yang diperlukan untuk mengubah *Physical Data Model* menjadi kode program. Metode ini menggunakan pendekatan *Model-Driven-Architecture (MDA)* yang mana dapat mentransformasi *Platform Independent Model (PIM)* menjadi *Platform Specific Model (PSM)* yang nantinya akan diterapkan dalam pembangkitan menjadi kode program yang bisa digunakan untuk berbagai platform.

## 1.2 Rumusan Masalah

Berdasarkan pada latar belakang di atas maka dirumuskan permasalahan sebagai berikut: Menerapkan pendekatan *Model-Driven-Architecture (MDA)* pada pembangkitan sebuah aplikasi berbasis *web* oleh mesin pembangkit kode program berbasis 4GL.

Adapun permasalahan secara rinci adalah sebagai berikut.

1. Bagaimana membangun sebuah *software* yang dapat mereduksi kegiatan memprogram (*coding*), sehingga aplikasi dapat otomatis dibuat dengan praktis dan cepat?
2. Bagaimana membangun sebuah mesin berbasis 4GL yang dapat membuat sebuah aplikasi dengan memanfaatkan *physical data model* dari pengguna yang sama sekali tidak paham bahasa pemrograman?

3. Bagaimana *Model Driven Architecture* dapat diterapkan untuk pembuatan mesin pembangkit kode program berbasis *web* dengan *design pattern Model-View-Controller*?
4. Bagaimana membangun perangkat lunak 4GL sebagai mesin pembangkit *physical data model* menjadi kode program dengan menerapkan pendekatan MDA (*Model Driven Architecture*)?

### 1.3 Batasan Masalah

Karena luasnya bidang yang dihadapi, maka dalam penelitian ini ada beberapa batasan masalah berdasarkan ruang lingkup kegiatan dalam proses pembuatan aplikasi ini. Beberapa batasan masalah tersebut adalah sebagai berikut.

1. Aplikasi yang dihasilkan berupa perangkat lunak berbasis *web*.
2. Mesin pembangkit tidak memuat bisnis proses sebagai bagian dari fitur.
3. Bahasa pemrograman untuk mesin dan hasil kode program adalah PHP dengan *database* MySQL.

### 1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah menerapkan *Model Driven Architecture* (MDA) pada pembangunan *software* berupa sebuah mesin yang dapat digunakan untuk membuat kode program secara otomatis, kumpulan program tersebut dikemas menjadi sebuah aplikasi berbasis *web* yang dapat digunakan oleh orang yang ingin membuat sebuah aplikasi secara praktis dan cepat. Sasarannya yaitu orang yang ingin membuat aplikasi tanpa perlu pengetahuan tentang pemrograman.

Adapun rincian dari tujuan penelitian ini adalah:

1. Membantu dalam proses pembangunan aplikasi yang dapat mereduksi kegiatan memprogram (*coding*) sehingga orang awam yang tidak mengenal bahasa pemrograman dapat membuat aplikasi.
2. Membuktikan bahwa pendekatan Model Driven Architecture adalah pilihan yang tepat untuk digunakan sebagai pembangkit kode sumber.
3. Menghasilkan sebuah mesin yang mampu mentransformasikan *Physical Data Model* menjadi menjadi sebuah aplikasi baru

### **1.5 Manfaat Penelitian**

1. Bagi Orang yang Tidak Bisa Bahasa Pemrograman

Orang cenderung ingin membuat aplikasi yang praktis dan cepat, dan tidak mengandung kode program di dalamnya. Diharapkan dengan mesin ini orang yang tidak mengenal bahasa pemrograman dapat membuat sebuah aplikasi dengan cepat, dengan catatan orang tersebut sudah memahami konsep basis data.

2. Bagi Peneliti

Manfaat untuk peneliti yaitu bisa dijadikan acuan dan pertimbangan untuk penelitian berikutnya yang berkaitan dengan rekayasa perangkat lunak.

### **1.6 Metode Penelitian**

Metode Penelitian yang digunakan dalam penulisan tugas akhir ini adalah sebagai berikut:

1. Tahap Pengumpulan Data

Tahap pengumpulan data yang dilakukan pada penelitian ini adalah sebagai berikut:

a. Studi Literatur

Pengumpulan data sebagai referensi yang harus dipelajari untuk memahami konsep-konsep yang berkaitan dengan penelitian ini, yaitu dengan cara mengumpulkan literatur, buku, jurnal, *internet*, *paper* dan media lainnya yang berkaitan dengan topik penelitian.

b. Observasi

Pengumpulan data berupa penelitian dan peninjauan langsung di lapangan terhadap masalah yang diambil. Dalam penelitian ini, penulis melakukan survei terhadap beberapa *software engineer* dan orang yang aktif di dalam dunia rekayasa perangkat lunak lainnya.

2. Analisis dan Perancangan

Analisis dalam penelitian ini dimulai dengan menentukan keperluan dan batasan untuk mesin dan menganalisis pendekatan MDA yang akan dipakai sebagai arsitektur dalam perangkat lunak yang akan dibuat.

Adapun perancangan dalam penelitian ini yaitu perancangan model yang akan dispesifikasi, merancang basis data yang digunakan, logika bisnis (alur) dan algoritma dari mesin dalam pembangkitan kode, dan menerapkan MDA sebagai arsitektur yang akan digunakan.

3. Implementasi dan Pengujian

Implementasi dilakukan setelah analisis berhasil dilakukan. Di sini ditentukan bahasa pemrograman apa yang akan digunakan untuk mesin

dan kode yang dibangkitkan dengan menerapkan algoritma dan MDA yang sudah dianalisis sebelumnya.

Pengujian dilakukan untuk menguji performansi dari mesin yang sudah dibuat dan kode yang dibangkitkan, supaya lebih optimal dan terhindar dari *error* yang tidak diharapkan.

#### 4. Hasil Akhir dan Penarikan Kesimpulan

Analisis hasil dilakukan untuk mengetahui sejauh mana performansi proses program dari mesin dan kode yang dihasilkan terhadap *browser* yang dipakai. Jika tidak sesuai dengan yang diharapkan, maka diperlukan analisis akhir untuk mengetahui penyebabnya. Setelah itu, baru dilakukan penarikan kesimpulan.

### 1.7 Sistematika Penulisan

Sistematika penulisan proposal ini disusun untuk memberikan gambaran umum tentang perangkat lunak yang akan dibuat. Sistematika penulisan tugas akhir ini adalah sebagai berikut :

#### **BAB I PENDAHULUAN**

Bab ini menguraikan tentang latar belakang masalah, rumusan masalah, tujuan penelitian, batasan masalah, manfaat penelitian, metode penelitian dan sistematika penulisan.

#### **BAB II TINJAUAN PUSTAKA**

Bab ini memaparkan beberapa landasan teori yang dipakai dalam penelitian ini. Adapun di sini dijelaskan beberapa teori seperti pengertian MDA, basis data, perancangan sistem, pengertian 4GL, bahasa pemrograman yang

digunakan (PHP), tinjauan perangkat lunak dan sebagainya yang mana pustaka-pustaka tersebut berperan penting dalam penelitian ini.

### **BAB III METODOLOGI PENELITIAN**

Bab ini merupakan penjabaran dari metode penelitian yang digunakan. Di dalamnya terdapat desain penelitian, analisis perancangan, analisis nonfungsional, pengembangan mesin dan sebagainya.

### **BAB IV HASIL PENELITIAN DAN PEMBAHASAN**

Pada bab ini diuraikan tentang hasil dari penelitian yang merupakan implementasi dari rumusan masalah, seperti implementasi perangkat lunak, implementasi antarmuka, dan pengujian perangkat lunak.

### **BAB V KESIMPULAN DAN SARAN**

Pada bab ini berisi tentang kesimpulan dan saran yang diajukan agar dapat menjadi bahan pertimbangan.

