

1. Pendahuluan

1.1 Latar belakang

MDB adalah aplikasi berbasis web yang berfungsi untuk manajemen *appointment*, *task*, dan penjadwalan. Untuk mengatasi kemungkinan pengembangan system, MDB harus memiliki desain yang fleksibel, *reusable*, dan *extendable*.

Maka harus dipikirkan bagaimana sistem akan berubah. Yaitu desain yang tidak memerlukan *redesign* pada masa yang akan datang. Perubahan itu meliputi *redefinition*, *reimplementation* dan *retesting*. Intinya *redesign* yang memerlukan biaya yang mahal harus diminimalkan.

Dibawah ini adalah beberapa hal yang kurang optimal pada MDB yang memperbesar kemungkinan MDB akan didesain ulang pada masa yang akan datang.

- 1 *Readability* dari program rendah, sehingga sulit untuk dikembangkan secara estafet.
- 2 Implementasi MDB masih menggunakan metode struktural sehingga desainnya tidak modular antara data dan *behaviour*.
- 3 Sulit dirubah menjadi *web* yang lebih dinamik, karena beberapa implementasi masih statis. Hal ini meliputi susunan menu hirarki dan layout web.
- 4 Sulit untuk dievolusikan untuk memenuhi *requirement* yang kemungkinan besar akan muncul, seperti mendukung *user specific task/modul*.
- 5 Tidak mendukung *customizable layout* sesuai dengan keinginan *user*.

1.2 Perumusan masalah

Masalah yang akan dibahas dalam TA ini adalah mendesain MDB dengan desain *pattern*. *Pattern-pattern* yang akan dipakai sebagian dari *pattern-pattern*:

- 1 **Creational Pattern**. *Abstract Factory*, *Builder*, *Factory Method*, *Prototype*, dan *Singleton*
- 2 **Structural Pattern**. *Adapter*, *Bridge*, *Composite*, *Decorator*, *Facade*, *Flyweight*, dan *Proxy*
- 3 **Behavioral Pattern**. *Chain of Responsibility*, *Command*, *Interpreter*, *Iterator*, *Mediator*, *Memento*, *Observer*, *State*, *Strategy*, *Template Method*, dan *Visitor*

1.3 Tujuan

Tujuan dari pembuatan TA ini adalah:

- 1 Menjadikan kode program MDB memiliki tingkat *reusability* yang tinggi sehingga mudah berevolusi sesuai dengan kebutuhan sistem
- 2 Menjadikan MDB kode program lebih *readable* sehingga dapat diprogram secara estafet
- 3 Menganalisis performansi setelah dilakukan desain ulang

1.4 Batasan masalah

- 1 Tidak membahas tentang analisis sistem atau dengan kata lain tidak menganalisis ulang sistem.
- 2 Evolusi yang dimaksudkan dibatasi pada mendukung *user specific task*/modul dan mendukung *customizable layout* sesuai dengan keinginan *user*
- 3 Performansi yang akan dianalisis dibatasi pada waktu eksekusi aplikasi

1.5 Metodologi penyelesaian

Metodologi penyelesaian masalah yang digunakan dalam TA ini terdiri dari tahapan-tahapan berikut ini :

- 1 **Studi Literatur.** Pada tahap ini dilakukan pencarian dan pengumpulan literatur-literatur yang berhubungan dengan masalah yang sedang dikerjakan dari buku-buku referensi yang menunjang.
- 2 **Analisis.** Tahap ini terdiri dari proses *mapping* dari kode program MDB ke *diagram* dan menganalisa kode program MDB untuk menemukan kandidat *pattern*.
- 3 **Redesign.** Desain ulang MDB dengan menerapkan desain *pattern*.
- 4 **Reimplementation**
- 5 **Testing**
- 6 **Dokumentasi**