



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB I

PENDAHULUAN

1.1 Latar Belakang

User Interface (UI) sebagai bagian dari sistem yang berinteraksi dengan *user*, diharapkan mudah untuk digunakan, dimengerti, dan memberikan hasil yang sesuai dengan keinginan *user* (Tran, 2010). UI tersebut harus jelas dan tidak ambigu untuk menggambarkan kinerja dari sistem (Hennig, 2011). Namun, sebuah UI yang semakin mudah untuk digunakan, maka semakin sulit proses pengembangannya (Schlungbaum, 1996). Proses pengembangan UI merupakan salah satu proses yang kompleks yang melibatkan pembuatan *prototype* atau *model* dari UI tersebut (Cruz, 2010). Selain itu menurut Pitkanen, merancang UI merupakan sebuah pekerjaan kompleks dan pada umumnya dibutuhkan iterasi untuk mencapai rancangan yang sesuai. Berdasarkan survey yang dilakukan oleh Myers, 48% *code* aplikasi digunakan untuk UI dan 50% dari waktu implementasi digunakan untuk uji coba *interface* tersebut (Myers, 1992).

Pengembangan UI yang sulit menyebabkan dibuatnya berbagai macam *tools* untuk membantu *developer* dalam mengembangkan UI (Schlungbaum, 1996). Salah satu jenis dari *tools* ini adalah *model-based UI generation tools*. Beberapa *model-based UI generation tools* telah dibuat seperti MECANO (Puerta, 1994), TEALLACH (Griffiths, 2001), dan DB-USE (Tran, 2010). *Model-based UI generation tools* menggunakan *declarative model* untuk

membantu dalam proses pengembangan UI (Ren, 2010). *Tools* tersebut memanipulasi satu atau lebih *declarative models* untuk membuat UI. MECANO membuat UI hanya menggunakan *domain model* (Puerta, 1994). TEALLACH menggunakan *domain model*, *task model*, dan *presentation model* (Griffiths, 2001). DB-USE menggunakan *task model*, *domain model*, dan *user model* (Tran, 2010). Model-model ini dapat diakses oleh UI pada saat *design* dan *runtime* sehingga memungkinkan *user* untuk dapat mengubah UI dengan memanipulasi model tersebut (Schlungbaum, 1996). Sebagai contoh, *data model* dibagi antara *interface design* dan *application design*, sehingga perubahan yang terjadi pada *application design* dapat dilakukan dengan mudah pada *interface design* (Puerta, 1994).

Penggunaan konsep *model-based* UI sangat menguntungkan bagi *developer*. Salah satu keuntungan yang diberikan adalah *runtime-tools* yang mendukung kustomisasi UI oleh *user* (Szekely, 1996). Salah satu UI yang paling sering berubah adalah *web* UI (Liang, 2011). Berdasarkan pengalaman Liang, banyak permintaan dari *user* untuk dapat mengatur UI dari aplikasi *web* yang mereka miliki. Karena itu, ia percaya bahwa menggunakan *end-user development* dan memberdayakan *user* untuk mengelola *web* UI adalah pendekatan yang praktis dan efektif.

Dalam mengembangkan aplikasi, *developer* juga menggunakan *framework*. *Framework* menyediakan *design* dan infrastruktur yang diperlukan dalam mengembangkan aplikasi, sehingga penggunaan *framework* menjanjikan pengembangan aplikasi yang lebih cepat (Basili, 1996). Selain itu

menurut Riehle, penggunaan *framework* bertujuan untuk mencapai produktivitas yang lebih tinggi dan waktu pengembangan yang lebih singkat dengan adanya *design* dan *code reuse*.

1.2 Rumusan Masalah

Dari latar belakang permasalahan di atas, masalah dapat dirumuskan sebagai berikut.

1. Bagaimana membangun sebuah *data model* untuk menyimpan informasi komponen UI?
2. Bagaimana membangun sebuah *framework* yang dapat membuat UI berdasarkan *data model* yang ada?

1.3 Batasan Masalah

Dalam perancangan dan pembangunan *data model* dan *framework*, diberikan batasan sebagai berikut.

1. *Data model* dan *framework* yang dibuat adalah untuk aplikasi ASP.NET dengan *database* SQL Server 2008.
2. Tipe data pada *database* yang dapat digunakan terbatas pada *char*, *nvarchar*, *datetime*, *int*, dan *decimal*.
3. Komponen UI yang dapat digunakan oleh aplikasi hanya terbatas pada *textboxes*, *labels*, *checkboxes*, *dropdown-lists*, *radiobutton-lists*, dan *column* pada *gridview*.

4. *Form* yang dibuat oleh *framework* digunakan hanya untuk sebuah *table* di *database*
5. *User* dari aplikasi yang dibuat dengan *framework* tidak dapat melakukan penambahan *form*, tetapi dapat melakukan modifikasi *form*.

1.4 Tujuan Penelitian

Penelitian ini memiliki tujuan sebagai berikut.

1. Membangun sebuah *data model* yang digunakan untuk menyimpan informasi komponen *user interface*.
2. Membangun sebuah *framework* yang dapat membuat *user interface* berdasarkan *data model* yang ada.

1.5 Manfaat Penelitian

Penelitian ini memiliki manfaat yaitu membantu proses pengembangan UI, serta meningkatkan fleksibilitas *user* dalam melakukan kustomisasi UI aplikasi.

UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA