



Software Modularity

Erick Kurniawan, S.Kom, M.Kom

Preface

- Jika software yang dibuat semakin besar
- Banyak fungsi dan kode bertambah kompleks
- Kita harus memikirkan bagaimana maintenance-nya
- Apakah bisa diketahui modul apa saja yang kita buat?
- Apakah ada dokumentasinya?
- Apakah mudah untuk melakukan sedikit perubahan / penambahan pada program?
- Pada chapter ini akan dipelajari cara group kode menjadi modul dan memcatat documentation dari module

Grouping Code

- Setiap module dalam aplikasi berisi :
- RDBMS Table definition
- Stored Procedures yang dijalankan di database (pada oracle menggunakan PL/SQL)
- Procedure yang dijalankan dalam web atau application server program yang digunakan oleh lebih dari satu page (*shared procedures*)
- Script yang men-generate individual pages
- Memungkinkan penggunaan template
- Dokumentasi yang menjelaskan cara kerja program

Modul pada *large* OLC

- User Registration
- Articles dan Comment
- Discussion Forum (share table yang sama dengan article, tetapi berbeda pada alur kerja dan script presentation-nya)
- Chat (table yang terpisah dengan content yang lain)
- Pengaturan penjualan, penempatan, dan pendataan banner advertisements
- Calendar (personal, group)

Modul pada *large* OLC

- Pengelompokan iklan dan lelang
- E-Commerce (katalog produk, presentation product, review community member, billing, accounting)
- Email, server based email (seperti hotmail) pada community member
- Survey (polling)
- Weblog, private blog untuk tiap community member

Modul pada *large* OLC

- Cara pengelompokannya :
- Dipecah kedalam direktori yang berbeda sesuai fungsinya (contoh www.photo.net)
- Root page dari site disimpan dalam direktori /web/photonet/www/
- Library procedure yang dapat di share disimpan dalam /web/photonet/tcl/
- RDBMS table, index, dan stored procedure definition disimpan dalam /doc/sql/

Modul pada *large* OLC

- Juga dipecah berdasarkan user misal:
/module-name/ untuk user biasa dan
/module-name/admin/ untuk admin
- Dokumentasi untuk tiap modul juga disimpan dalam /doc/module-name.html
- Dokumentasi sangat penting digunakan untuk menjelaskan modul / fungsi yang dapat digunakan dalam aplikasi
- Di java ada java API, di .NET ada MSDN documentation

Shared vs Stored Proc

- Apa itu stored procedure?
- Logic program yang disimpan dalam RDBMS tertentu, yang support mis: (Oracle, SQL Server, MySQL, PostgreSQL)
- Apa itu shared procedure?
- Kumpulan method / fungsi yang digunakan pada banyak halaman
- Pada PHP (statement include), pada Java atau C# dapat berupa kumpulan class yang berisi method-method

Shared vs Stored Proc

- Bagus mana shared procedure atau stored procedure?
- Tergantung pada kebutuhan-nya, kapan menggunakan stored proc?, kapan menggunakan shared proc?
- Stored proc baik digunakan jika kode yang digunakan banyak mengambil data dari database yang sama (*same data model*)
- Shared proc baik digunakan jika kode yang digunakan tidak berhubungan dengan data model, hanya berguna pada web aplikasi yang dibangun

Documentation

- Bayangkan ada 1000 procedure tanpa dokumentasi
- Kita harus menulis *doc string* untuk tiap procedure yang dibuat
- Input apa yang dapat diterima, output yang dihasilkan
- Pada java dan .net dapat langsung digenerate setelah program selesai ditulis
- Ditambah juga dengan penjelasan dan design work pada kasus yang nyata

Memisahkan design & program

- Jika anda membuka web yang dibuat dengan PHP, ASP, atau CGI anda pasti menemukan HTML, SQL statement, IF statement (script) tercampur dalam page
- Menyulitkan jika project dibangun dalam tim yang terdiri dari designer dan programmer
- Banyak designer tidak terlalu paham terhadap program
- Banyak programmer yang tidak bisa design

Memisahkan design & program

- Selera user terhadap design berbeda (mengenai background color, font, dll)
- Bagaimana jika user menginginkan merubah color, font, header, atau footer?
- Bisa mudah bisa juga sulit, tergantung bagaimana anda membangun sistem
- Kita bisa membuat file configuration, header, footer yang terpisah dari page
- Ada beberapa metode yang biasa digunakan untuk memisahkan design dan kode

The Small Hammer

- Cara termudah untuk memisahkan design dengan program
- Dibagi 2 files untuk tiap URL
- File 1 berisi SQL statement dan beberapa procedural kode yang berisi lokal variabel atau struktur data yang berisi informasi dari RDBMS
- Statement terakhir dari file 1 memanggil procedure yang mengambil File 2
- File 2 berisi standar HTML dengan referensi data di File 1
- Yang lebih restrict (Medium Hammer): ada pembatasan permission

SQL Hammer

- Menggunakan database sebagai backend untuk versioning dan permmisioning
- Menyimpan templates pada database table

Configuration Parameter

- Dimungkinkan meski tidak disarankan untuk membangun aplikasi dibawah ini dengan *hard-coded* pada individual pages :
- RDBMS username dan password
- E-mail address dari site administrator
- IP address dari user yang tdak disukai
- Nama dari site
- Nama dari editor dan publisher
- Maximum dari ukuran attachment
- Tetapi disarankan untuk menggunakan configuration parameter

Configuration Parameter

- Dapat disimpan dalam RDBMS, pada oracle:

```
create table config_param_keys ( config_param_key_id  
integer primary key, key_name varchar(4000) not null,  
param_comment varchar(4000) );
```

```
create table config_param_values ( config_param_key_id not null references  
config_param_keys, value_index integer default 1 not  
null, param_value varchar(4000) not null );
```

Configuration Parameter

- Dapat disimpan pada config file, pada ASP.NET:

```
<configuration> <appSettings> <add  
key="publisherEmail"  
value="marketing@mycompany.com" />  
</appSettings> </configuration>
```

```
<% Dim publisherEmail as String  
publisherEmail =  
ConfigurationSettings.AppSettings(  
"publisherEmail" ) %>
```

Configuration Parameter

- Dapat disimpan pada config file, pada JSP :

```
<Context path="/myapp"
docBase="myapp" debug="0"
reloadable="true"
crossContext="true"> <Parameter
name="companyName" value="My Company,
Inc." override="false"/> </Context>
```

```
<context-param> <param-
name>companyName</param-name> <param-
value>My Company, Inc.</param-value>
</context-param>
```

```
getServletContext().getInitParameter("compa
nyName");
```