DASHBOARD LIBRARY UNTUK VISUALISASI INFORMASI

Yulia[1)], Harry Timothy Tumalewa[2)], Hans Juwiantho[3)]
1,2,3) Teknik Informatika Fakultas Teknologi Industri Universitas Kristen Petra, Surabaya
Siwalankerto 121-131 Surabaya
E-mail : yulia@petra.ac.id[1)]

Abstrak

Dalam pengembangan *software* pengolahan data, kemudahan dalam mengakses informasi merupakan salah satu hal yang penting untuk diperhatikan oleh pihak pengembang *software*. Semakin praktis *software* digunakan, maka tingkat kepuasan *user* juga akan semakin bertambah. Suatu *software* yang handal adalah *software* yang mampu menyediakan informasi yang mudah dimengerti, sesuai dengan kebutuhan dari penggunanya dan informatif. Namun dalam implementasinya, diperlukan langkah pengerjaan yang cukup banyak oleh *user* untuk mengakses informasi melalui sebuah laporan pada *software* yang dikembangkan. Sehingga dibutuhkan sebuah media penampilan informasi secara visual berupa *dashboard* untuk mempermudah *user* dalam mengakses informasi yang dibutuhkan.

Kata Kunci: *dashboard, informasi visual, software*

Pendahuluan

Tujuan utama dari penggunaan program aplikasi adalah bagaimana agar data yang dimasukkan dapat diolah menjadi informasi yang berguna. Salah satu perangkat *business intelligence* (BI) yang populer saat ini adalah Informasi *Dashboard*. Informasi *Dashboard* merupakan alat untuk menyajikan informasi dari proses BI yaitu memberikan tampilan antarmuka dengan berbagai bentuk seperti diagram, laporan, indikator visual, mekanisme *alert*, yang dipadukan dengan informasi yang dinamis dan relevan

Untuk membangun sebuah informasi mengenai kinerja suatu organisasi untuk pimpinan tingkat strategis, dibutuhkan informasi yang sifatnya tidak terlalu detail dan panjang, pimpinan hanya membutuhkan informasi yang sifatnya singkat, mudah dibaca, dipahami dan tidak menimbulkan banyak persepsi. Informasi *dashboard* memberikan suatu bentuk informasi yang sesuai untuk pimpinan tingkat strategis (eksekutif) karena informasi *dashboard* disajikan dalam bentuk visual dan dapat dipahami secara sekilas. Dengan bentuk dashboard akan memudahkan pihak eksekutif untuk melakukan komparasi, menganalisa serta menggarisbawahi variabel-variabel penting untuk melihat kinerja, identifikasi kesempatan serta masalah yang terjadi (Nurwidyantoro, 2013).

Latar Belakang

Semua orang mengasumsikan bahwa *dashboard* berguna. Namun aplikasi ini dapat diimplementasikan dengan baik atau buruk. Beberapa kegagalan dari aplikasi *dashboard* ini adalah bukan dari sisi teknologi tetapi karena komunikasi. Masalah utamanya adalah karena desain dari *dashboard* kurang informatif atau masih miskin informasi. Bila dirancang dengan benar dan dengan tampilan grafik yang tepat (Brooks), maka *dashboard* akan memberikan informasi yang dapat dimengerti secara sekilas.

Dalam proses pembuatan laporan *dashboard* yang informatif dan sesuai dengan kebutuhan pengguna, diperlukan langkah pengerjaan yang cukup banyak oleh *user* untuk mengakses informasi melalui sebuah laporan pada *software* yang dikembangkan. Untuk memudahkan pihak eksekutif dalam mendapatkan informasi secara cepat, sesuai kebutuhan dan mudah dipahami, maka pada penelitian ini dibuat sebuah media penampilan informasi berupa *dashboard*.

Studi Pustaka

Dashboard

Dashboard adalah tampilan visual dari informasi yang paling penting yang dibutuhkan untuk mencapai satu atau lebih tujuan; dikonsolidasikan dan diatur pada satu layar sehingga informasi tersebut dapat dipantau dalam sekejap (Few, 2006). Gambar 1 berikut ini adalah contoh

dashboard dari Infommersion, Inc yang memberikan alat bagi pihak eksekutif dari jaringan hotel untuk mengukur kinerja salah satu hotel pada suatu waktu.

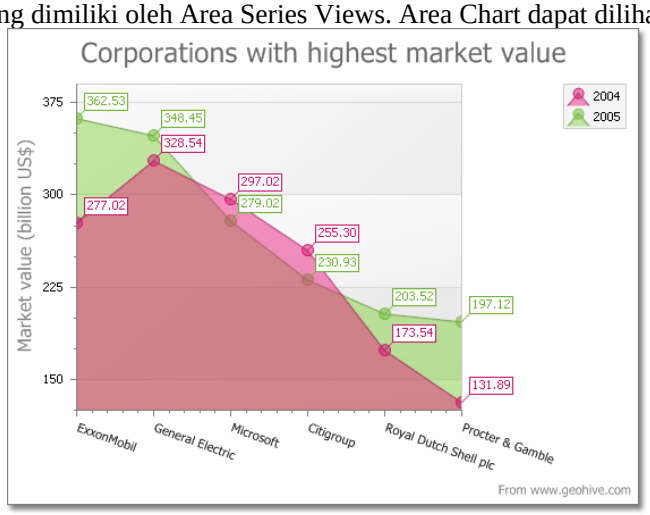
Gambar 1. Contoh Dashboard (Sumber: Few, 2006)

DevExpress Chart Control

Dalam pembuatan *dashboard* diperlukan sebuah chart control yang mampu mendukung representasi data ke dalam berbagai bentuk bagan (chart). Chart control pada penelitian ini menggunakan DevExpress WinForms Charts Suite.

DevExpress WinForms Charts Suite menyediakan lebih dari 60 tipe bagan 2D dan 3D yang berbeda-beda, mulai dari bar dan pie chart yang sederhana, hingga bagan finansial yang kompleks seperti stock dan candle stick chart. DevExpress WinForms Charts juga menyediakan fleksibilitas untuk bind ke datasource manapun yang mendukung *interface* IList atau IEnumerable. Terdapat pula Charts Wizard yang *built-in*, sehingga pengguna mampu mengkonfigurasi sebuah bagan atau kompleksitas dengan cepat dan mudah.

Berikut ini adalah salah satu contoh informasi mengenai bagan yang tersedia pada DevExpress, gambar dan cara konfigurasinya. Area Chart direpresentasikan oleh objek AreaSeriesView, yang dimiliki oleh Area Series Views. Area Chart dapat dilihat pada Gambar 2.



Gambar 2. Area Chart

View ini menampilkan series sebagai area yang diisi pada bagan, dan berguna untuk menampilkan trend terhadap beberapa series pada bagan yang sama. Segmen Program 1 berikut ini adalah cara konfigurasi Area Chart.

ScaleType merupakan cara penyajian data pada bagan. Secara default ScaleType yang digunakan adalah qualitative. Namun terdapat pula alternatif lain seperti numerical dan datetime.

Segmen Program 1. Konfigurasi Area Chart

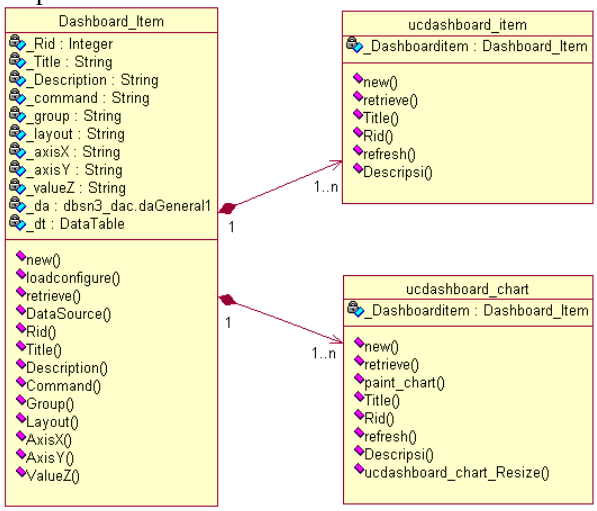
```
' Set the numerical argument scale types for the series,
' as it is qualitative, by default.
series1.ArgumentScaleType = ScaleType.Numerical
series2.ArgumentScaleType = ScaleType.Numerical

' Access the view-type-specific options of the series.
CType(series1.View, AreaSeriesView).Transparency = 80

' Access the type-specific options of the diagram.
CType(areaChart.Diagram, XYDiagram).EnableAxisXZooming = True
```

Desain Sistem

Pembuatan *dashboard* dilakukan dengan menggunakan konsep *object oriented*. Detail informasi yang akan ditampilkan pada *dashboard* disimpan pada *class Dashboard*_Item. Sedangkan representasinya dalam bentuk bagan (*chart*) dibuat menggunakan *user control*. Berdasarkan jenis informasi yang ditampilkan, *user control* dibagi menjadi dua, yaitu untuk menampilkan informasi dalam bentuk *grid* (uc*dashboard*_item) dan bagan (uc*dashboard*_chart). Desain *class* dapat dilihat pada Gambar 3.



Gambar 3. Class Diagram

Atribut _Rid pada *class Dashboard*_Item digunakan untuk menentukan *command* mana yang akan diambil dari *database* untuk dijalankan. *Command* ini merupakan *query* SELECT untuk menampilkan informasi tertentu. Melalui _Rid, judul (_Title), keterangan (_Description), penggolongan jenis informasi (_group), jenis tampilan (_layout) dan nilai-nilai yang dipakai untuk representasi bagan (_axisX, _axisY dan _valueZ) dari *command* dapat diketahui.

Operasi new() pada *class Dashboard*_Item merupakan *constructor* dengan *parameter* rid yang digunakan untuk men-*set* nilai _Rid ketika objek dibuat. loadconfigure() merupakan prosedur yang digunakan untuk mendapatkan _Title, _Description, _command dan atribut lainnya dari *database*, berdasarkan _Rid yang telah di-*set* oleh *constructor*. Retrieve() adalah fungsi untuk mengisi tabel berdasarkan _command yang telah didapatkan melalui prosedur loadconfigure(). Sedangkan operasi-operasi lainnya merupakan *property* yang digunakan sebagai *public access method*.

Pada *user control*, baik uc*dashboard*_item maupun uc*dashboard*_chart, terdapat *_Dashboard*item, yaitu objek dari *class Dashboard*_Item yang di-*composite*-kan. Seperti yang telah dijelaskan sebelumnya, *user control* digunakan untuk menampilkan informasi dalam bentuk *grid* atau *chart*. Untuk itu dibutuhkan informasi dari *class Dashboard*_Item melalui objek *_Dashboard*item.

Operasi new() pada *user control* merupakan *constructor* yang menginisialisasi objek *_Dashboard*item. retrieve() merupakan fungsi yang memanggil fungsi retrieve() pada objek *_Dashboard*item, kemudian memanggil fungsi paint_chart() (hanya berlaku pada uc*dashboard*_chart). Paint_chart merupakan prosedur untuk menampilkan bagan berdasarkan *command* pada objek *_Dashboard*item, di mana sumbu X, sumbu Y dan nilai pada bagan dimuat berdasarkan nilai _axisX, _axisY dan _valueZ. Rid() dan Descripsi() merupakan *publc access method* untuk mengganti _Rid dan _Description pada *_Dashboarditem*. Refresh() merupakan prosedur yang dibuat agar judul dan *tooltip* pada bagan tetap *up-to-date*.

Implementasi *Class*

Berdasarkan desain yang telah dibuat, diperlukan peranan *database* untuk menyimpan detail informasi dari *dashboard item*. Database yang digunakan adalah Northwind.accdb (Microsoft Access). Sedangkan tabel yang dibuat untuk menyimpan detail informasi dapat dilihat pada Gambar 4.

ID	Title	Deskripsi	Command	Group	Layout	AxisX	AxisY	ValueZ
1	Customer	Daftar Customer	select * from customers	A	grid	-	-	-
2	Supplier	Daftar Supplier	select * from suppliers	A	grid	-	-	-
3	Order	Daftar Order	select * from orders	B	grid	-	-	-
4	Shipper	Daftar Shipper	select * from shippers	C	grid	-	-	-
5	Summary	Daftar Shipping	SELECT orders.Order_Date, SUM([Order Subtotals].Subtotal) as Subtotal FROM orders INNER JOIN [Order Subtotals] ON orders.[Order ID] = [Order Subtotals].[Order ID] group by orders.Order_Date	A	bar	Order_date	Order_date	Subtotal
6	Order Customer per Tanggal	Daftar Total Order Customer per Tanggal	SELECT customers.company, orders.Order_Date, Sum([Order Subtotals].Subtotal) AS Total FROM orders, customers, [Order Subtotals] WHERE orders.[Order ID] = [Order Subtotals].[Order ID] AND customers.company IN ('Company D', 'Company F') AND orders.Customer = customers.id GROUP BY customers.company, orders.Order_Date	A	line	Company	Order_date	Total

Gambar 4. Tampilan Tabel *Dashboard*_item

Pada Gambar 4 diberikan 6 buah sampel data. Untuk penyajian data dalam bentuk *grid*, sumbu X, sumbu Y dan nilai Z tidak perlu diisi. Sedangkan untuk penyajian data dalam bentuk bagan, apabila bentuk informasi yang akan ditampilkan adalah 2 dimensi, maka nilai dari sumbu X dan Y dapat disamakan. Namun apabila informasi yang ingin ditampilkan adalah 3 dimensi (misalnya jumlah pemesanan barang pada setiap cabang perusahaan per tanggal), maka sumbu X dan Y perlu diisi dengan nama *field* yang ingin ditampilkan ke dalam bagan. Implementasi *class Dashboard*_Item dapat dilihat pada Segmen Program 3.

Segmen Program 3. Class *Dashboard*_Item

```
Public Class Dashboard_Item
    Private _Rid As Integer
    Private _Title As String
    Private _Description As String
    Private _command As String
    Private _group As String
    Private _layout As String
    Private _axisX As String
    Private _axisY As String
    Private _valueZ As String
    Private _da As dbsn3_dac.daGeneral1
    Private _dt As DataTable
    Public Sub New(ByVal rid As Integer)
        _Rid = rid
        _da = New dbsn3_dac.daGeneral1
        _dt = New DataTable
    End Sub
    Private Sub load_configure()
        Try
            _command = "select * from Dashboard_item where ID = " & _Rid
            Dim _tab As New DataTable
            _da.Fill(_tab, _command)
            If _tab.Rows.Count < 1 Then
                MsgBox("RID not found")
```

```
            Else
                _command = _tab.Rows(0)("Command")
                _Title = _tab.Rows(0)("Title")
                _Description = _tab.Rows(0)("Deskripsi")
                _group = _tab.Rows(0)("Group")
                _layout = _tab.Rows(0)("Layout")
                _axisX = _tab.Rows(0)("AxisX")
                _axisY = _tab.Rows(0)("AxisY")
                _valueZ = _tab.Rows(0)("ValueZ")
            End If
        Catch ex As Exception
            'MsgBox(ex.Message)
        End Try
    End Sub
    Public Function retrieve() As Integer
        Try
            Return _da.Fill(_dt, _command)
        Catch ex As Exception
            MsgBox(ex.Message)
        End Try
    End Function
```

Ketika prosedur New dijalankan, _Rid di-*set* sesuai dengan nilai dari parameter rid, kemudian dilakukan inisialisasi terhadap _da dan _dt. *Data adapter* untuk akses ke *database* dibuat menggunakan *library* dbsn3.dac.dll. Pada saat load_configure dijalankan, _command diisi sementara dengan perintah untuk menampilkan tabel *Dashboard*_item berdasarkan _Rid. Lalu hasilnya diisi ke dalam _dt untuk kemudian dimasukkan ke dalam variabel yang sesuai dengan hasil *query*. Fungsi retrieve digunakan untuk mengisi _dt sesuai dengan *string* yang terdapat pada variabel _command. Implementasi *user control* uc*dashboard*_item, dapat dilihat pada Segmen 4.

Segmen Program 4. User Control uc*dashboard*_item

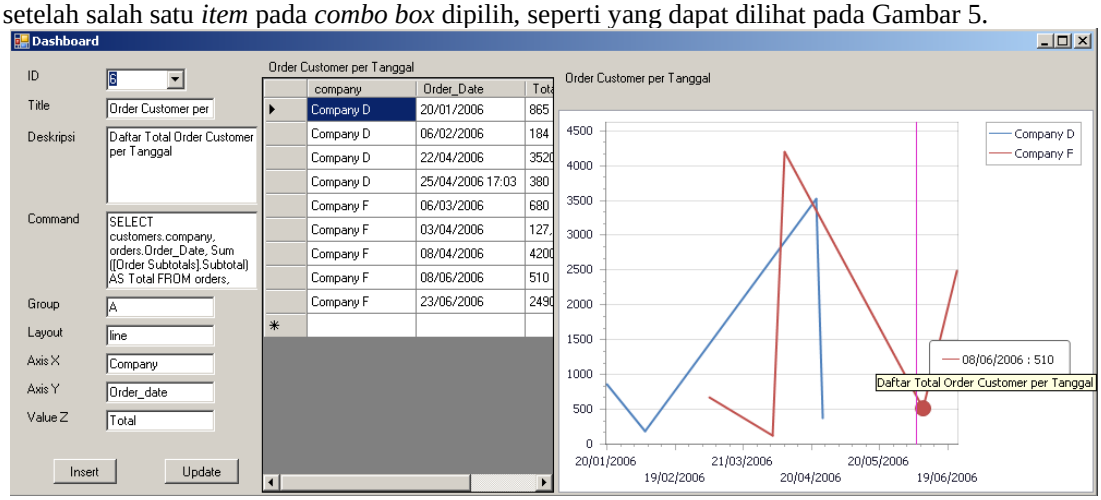
```
Public Class ucdashboard_item
    Public _Dashboarditem As Dashboard_Item
    Public Sub New()
        InitializeComponent()
        _Dashboarditem = New Dashboard_Item(1)
        C_Grid.DataSource = _Dashboarditem.DataSource
    End Sub
    Public Function retrieve() As Integer
        Return _Dashboarditem.retrieve()
    End Function
    Public Property Title() As String
        Get
            Return _Dashboarditem.Title
        End Get
        Set(ByVal value As String)
            _Dashboarditem.Title = value
            Me.refresh()
        End Set
    End Property
    Public Property Rid() As Int32
        Get
            Return _Dashboarditem.Rid
        End Get
        Set(ByVal value As Int32)
            _Dashboarditem.Rid = value
            Me.refresh()
        End Set
    End Property
    Private Sub refresh()
        C_Title.Text = _Dashboarditem.Title
        ToolTip1.SetToolTip(C_Title, _Dashboarditem.Description)
        ToolTip1.SetToolTip(C_Grid, _Dashboarditem.Description)
    End Sub
```

Pada saat prosedur New dijalankan, objek _*Dashboard*item diinisialisasi, kemudian DataSource pada *grid* disamakan dengan DataSource pada _*Dashboard*item. Fungsi retrieve diguanakan untuk memanggil fungsu retrieve pada _*Dashboard*item. Pada saat Title atau Rid di-*set*, prosedur refresh dijalankan untuk menampilkan informasi terkini pada *tooltip*.

Secara umum, uc*dashboard*_chart mirip dengan uc*dashboard*_item. Bedanya hanya pada prosedur paint_chart dan uc*dashboard*_chart_Resize.

Tampilan Program

Setelah implementasi *class* yang telah didesain, selanjutnya dibuat program untuk menguji sistem secara keseluruhan. Program ini mencakup tampilan *dashboard item* dalam bentuk *grid* dan *chart*. Pada sisi kiri *form* terdapat sebuah *combo box* untuk mengakses setiap _Rid yang terdapat di dalam *database*, dan beberapa *text box* yang memuat detail informasi berdasarkan _Rid. Pengisian _Rid dilakukan secara otomatis ke dalam *combo box* pada saat *form load*. Informasi ditampilkan setelah salah satu *item* pada *combo box* dipilih, seperti yang dapat dilihat pada Gambar 5.



Gambar 5. Pengujian Program

Setelah salah satu Rid pada *combo box* dipilih, informasi berdasarkan *command* ditampilkan ke dalam bentuk *grid* dan *chart*. Pada gambar di atas, isi dari *query* SELECT pada *command* adalah untuk menampilkan total *order* dari *company D* dan *company F* berdasarkan tanggal pemesanan (*order date*). Hasil *query* pada *grid* ditampilkan secara berurut berdasarkan *company*, kemudian tanggal lalu total *order*. Pada *chart*, dua garis yang berbeda warna menunjukkan dua *company* yang berbeda. Dengan tampilan yang menarik dan mudah dipahami, pengguna dapat dengan mudah melihat perbandingan total *order* pada tanggal tertentu antar *company* yang ada.

Kesimpulan

Dari hasil penelitian ini, dapat disimpulkan bahwa pembuatan *dashboard* membutuhkan desain *class* yang fleksibel, yaitu dengan memanfaatkan penggunaan *database* sebagai konfigurasi, sehingga memungkinkan representasi informasi ke dalam berbagai jenis *layout*. Untuk itu, diperlukan juga dukungan *chart control* yang fleksibel seperti yang telah disediakan oleh DevExpress.

Berdasarkan hasil pengujian, dapat disimpulkan bahwa baik implementasi *class*, *chart control*, maupun konfigurasi pada *database* telah berjalan dengan baik. Namun masih terdapat beberapa kekurangan pada program yang telah dibuat, di mana seharusnya setiap *text box* di-*bind* ke *data source* untuk memungkinkan pengembangan yang lebih fleksibel. Selain itu, konfigurasi di *database* juga masih belum maksimal, di mana sampel data yang disediakan masih sangat kurang, baik secara kuantitas maupun dari kompleksitasnya. Sedangkan untuk hasil pengujian yang lebih baik, dibutuhkan *command* yang berisi *query* SELECT yang lebih kompleks.

Daftar Pustaka

Brooks, Taggert J., *Representing Data Graphically*. http://www.uwlax.edu/faculty/brooks/bus230/handouts/designing%20graphs.pdf

Few, Stephen., (2006), *Information Dashboard Design: The Effective Visual Communication of Data*, O'Reilly Media; 1st edition

Nurwidyantoro, Arif., Hakim, B., Utomo, E., *SNTI 2013*, Perancangan Sistem Informasi Eksekutif, 2013

https://www.devexpress.com/ tanggal akses: 31 Oktober 2013