

ST-RK-1.16-082-007/R-



Modul Praktikum

PEMGROGRAMAN VISUAL II

Mahasiswa dapat mengerti tentang konsep OOP (Pemrograman Berorientasi Object), Membuat Control pada Windows Form, Pengenalan Database Shcema (XSD), Mencetak Aplikasi pada Windows Form dan Mendeploy sebuah aplikasi pada Visual Basic .Net



Join Us

**THE
CYBER
campus**
<http://www.stikom.edu>

Laboratorium Komputer

DAFTAR ISI

DAFTAR ISI.....	1
PENGANTAR	3
MODUL 1	4
1.1. Obyek Koneksi	5
Konsep	5
Membuat Obyek Connection.....	6
1.2. Obyek Command	10
Pengertian Aplikasi terkoneksi	10
Obyek Command.....	10
1.3. Obyek DataReader	11
Latihan	13
MODUL 2	15
2.1. DML pada Obyek Command.....	16
2.2. ExecuteNonQuery.....	18
Latihan	19
MODUL 3	21
3.1. Cara Membuat DataSet.....	23
3.2. DataTable.....	23
3.3. DataColumn	24
3.4. Constraint	24
3.5. DataRelation	25
3.6. Manipulasi Data.....	25
3.7. AcceptChanges and RejectChanges.....	26
3.8. DataGridView	26
Latihan	28
MODUL 4	31
4.1. Pencarian dengan metode Find, Contains dan Select pada DataTable.....	32
4.2. DataView	33
4.3. Aplikasi Terputus.....	35
Latihan	38
MODUL 5	43
5.1. Pengertian Data Binding	44
5.2. Aliran data pada Data Binding.....	44
5.3. Kontrol Data-Bound.....	45
5.4. Event Format dan Parse pada Object Binding.....	45
5.5. Object BindingSource dan BindingNavigator	46
Latihan	48

MODUL 6	52
6.1. ADO.NET dan Arsitektur XML.....	53
6.2. XML Schema.....	54
6.3. XML File	58
6.4. Pengaplikasian XML pada Form	60
Latihan	62
MODUL 7	65
7.1. Membuat Crystal Report.....	66
7.2. Formula Fields.....	72
7.3. Parameter Fields.....	73
7.4. Control pada Crystal Report.....	75
7.5. Menampilkan Crystal Report pada Aplikasi Window	75
Latihan	78

PENGANTAR

Praktikum Pemrograman Visual 2 merupakan versi praktek dari kuliah Pemrograman Visual 2 (PV2). Disini kita bisa belajar bagaimana mengimplementasikan materi yang telah dipelajari di kuliah PV2. Walaupun tidak semua materi kuliah bisa dicover oleh praktikum, setidaknya esensi dari seluruh materi kuliah ada di praktikum ini.

Seperti yang telah kita ketahui bersama, sistem praktikum menjadikan waktu latihan digunakan untuk trouble shooting. Ibarat sebuah forum/FAQS, praktikan diberi HAK untuk bertanya kepada pengajar. Tanpa adanya pertanyaan, rasa-rasanya tidak akan pernah ada jawaban. Dan tidak ada jawaban berarti tidak ada forum.

Ada 7 modul dalam buku ini, yang menjadikan modul ini begitu special dikalangan modul-modul lain. Namun jangan khawatir, Anda sudah bayar untuk 8x pertemuan, jadi untuk pertemuan ke-8 Anda akan dihadapkan pada sebuah tugas project.

Tugas ini akan menghabiskan waktu 80 menit sehingga waktu latihan benar-benar ditiadakan. Tugas ini merupakan rangkuman dari materi-materi yang pernah kita pelajari di 7 pertemuan sebelumnya. Mulai dari bagaimana membuat koneksi, memilih jenis environment yang tepat untuk permasalahan tertentu, sampai dengan membuat laporan dan nota (jika waktunya cukup tentunya, hehehe). Dalam pengerjaan tugas ini, Anda diberi HAK untuk bertanya kepada pengajar lho, dan kesempatan ini hanya ada di Praktikum PV2. Mengapa? Karena jika Anda PAHAM tentang konsep teori dan logika pemrograman dalam mengerjakan tugas project tersebut, bisa dipastikan Anda bisa mengerjakan Ujian Praktikum PV2 yang notabene selama ini tingkat kelulusan mahasiswa kurang dari 50%.

Semua aturan-aturan yang berkaitan dengan pengerjaan soal seperti properti koneksi dan lain-lain, akan dilaksanakan dengan cara seksama dan dilakukan langsung di TKP. Ada pertanyaan? Tunggu kami para pengajar di kelas Anda.

MODUL 1

KONEKSI DAN APLIKASI TERKONEKSI

What is important is to keep learning, to enjoy challenge, and to tolerate ambiguity. In the end there are no certain answers.

- Martina Horner -



Tujuan:

Mahasiswa paham dan mengerti tentang Koneksi
Mahasiswa paham dan mengerti tentang Aplikasi Terkoneksi

Materi:

SQLConnection, SqlCommand dan SqlDataReader

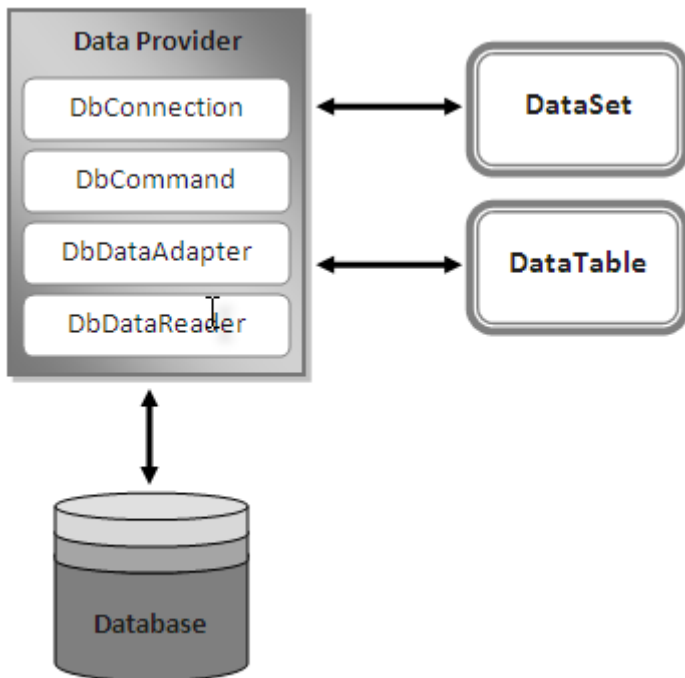
Referensi:

1. Amelia, Tan, 2008, Pemrograman Visual II Panduan Sertifikasi MCTS (Exam 70-526), Tanpa Kota, Tanpa Penerbit
2. Huddleston, James, 2007, Beginning VB 2005 Databases From Novice to Professional, New York, Springer-Verlag New York Inc.
3. Willis, Thearon, 2006, Beginning Visual Basic 2005 Databases, Indianapolis, Wiley Publishing, Inc.

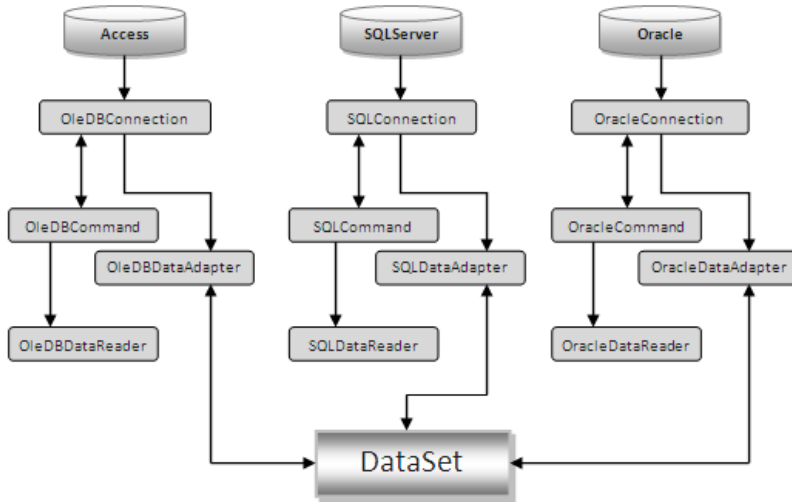
1.1. Obyek Koneksi

Konsep

Arsitektur ADO .NET dibagi menjadi 3 kelompok besar, yaitu Data Provider, DataSet dan DataTable (Thearon Willis, 2006, *Beginning Visual Basic 2005 Database*). Data Provider merupakan penyedia layanan agar DataSet dan DataTable dapat terhubung dengan Database.



Seperti yang dapat dilihat pada gambar, Data Provider itu sendiri dibagi menjadi 4 bagian yaitu Connection, Command, DataAdapter dan DataReader. Data Provider juga dibagi kedalam beberapa tipe berdasarkan jenis database seperti ODBC Data Provider, SQL Server Data Provider, Oracle Data Provider dan OLEDB Data Provider.



Untuk selanjutnya, pembahasan akan dilakukan untuk SQL Server Data Provider. Sehingga namespace yang dipakai adalah System.Data.SqlClient. Dan pada dasarnya, **yang membedakan masing-masing Data Provider adalah pada jalur koneksi yang digunakan**. Sedangkan untuk obyek-obyek lain dalam Data Provider mempunyai cara yang sama untuk membaca dan mengeksekusi perintah SQL.

Untuk melakukan koneksi, seperti yang bisa dilihat pada gambar, diperlukan SqlConnection. Dan ada 2 cara yang bisa dilakukan untuk membuat obyek Connection, secara visual dengan menambahkan kontrol Connection dan secara koding dengan mengetikkan obyek Connection secara manual. Yang perlu diperhatikan adalah, **obyek Connection merupakan sebuah jalur** untuk mengeksekusi perintah SQL dan menerima hasil dari query, bukan obyek yang melakukan eksekusi perintah SQL atau menampung hasil query.

Membuat Obyek Connection

Seperti yang sudah dijelaskan sebelumnya, untuk membuat sebuah koneksi ke dalam database SQLServer diperlukan sebuah SqlConnection dan koneksi inilah yang membedakan antara sebuah Data Provider dengan Data Provider lain. Pembedanya terletak pada ConnectionString. Untuk macam-macam ConnectionString itu

sendiri, dapat Anda lihat pada situs www.connectionstrings.com. Dan Anda harus ingat bahwa pada praktikum PV2, Anda **hanya diperbolehkan untuk melakukan koneksi dari koding**.

1. ConnectionString

Sebuah `ConnectionString` mendefinisikan berbagai parameter yang dibutuhkan oleh `SqlConnection` untuk membuat koneksi kedalam database `SQLServer`. Parameter-parameter ini meliputi:

- **Data Source**
Berisi nama atau lokasi (IP address) server yang dituju diikuti nama instance dari `SQLServer` (jika ada). Namun untuk database `SQLServer` yang ada pada 1 komputer yang sama dengan aplikasi Anda, Anda hanya perlu menuliskan tanda “.” atau `localhost` atau (`local`) diikuti nama *instance* dari `SQLServer` (jika ada).
- **Initial Catalog**
Berisi nama database yang akan Anda gunakan dalam aplikasi Anda. Database ini harus sudah terdaftar didalam `SQLServer`.
- **Integrated Security**
Bernilai `TRUE` atau `FALSE`. Dengan mengaktifkan parameter ini (`TRUE`), maka `SqlConnection` menggunakan user id dan password (otentikasi) windows yang saat itu sedang `LOGIN`, dan Anda tidak perlu memakai parameter `User Id` dan `Password`. Biasanya parameter ini dipakai jika `SQLServer` berada pada 1 komputer yang sama dengan aplikasi Anda.
- **User Id**
Berisi nama `LOGIN` yang ada `SQLServer`.
- **Password**
Berisi password dari `User Id` yang bersangkutan. Anda harus menggunakan `User Id` dan `Password` jika Anda tidak menggunakan parameter `Integrated Security`, atau jika nilainya `FALSE`. Biasanya `User Id` dan `Password` dipakai jika `SQLServer` berada pada komputer yang berbeda dengan aplikasi Anda.

Contoh (studi kasus) membuat `ConnectionString` untuk `SqlConnection`:

```
Server      : WEB-SERVER
Database   : VISUAL2
User ID     : USER
Password   : PASSWORD
```

Maka `ConnectionString` yang dipakai adalah:

```
"Data Source=WEB-SERVER;Initial " & _  
"Catalog=VISUAL2;User Id=USER; " & _  
"Password=PASSWORD"
```

Untuk membuat instance dari `SqlConnection` itu sendiri, Anda tinggal menyalin baris kode dibawah ini. Namun sebelumnya Anda harus import dulu `Namespacenya`.

```
Imports System.Data.SqlClient  
  
Dim conn As SqlConnection()  
Private Sub Form_Load(...) handles  
MyBase.Load  
    Dim strcon As String = _  
        "Data Source=WEB-SERVER;" & _  
        "Initial Catalog=VISUAL2;User Id=" & _  
        "USER;Password=PASSWORD"  
  
    conn = New SqlConnection(strcon)  
End Sub
```

2. Method Open dan Close

Method `Open` dalam obyek `SqlConnection` digunakan untuk membuka koneksi kedalam `SQLServer`. Sedangkan method `Close` digunakan untuk menutup koneksi. Pada dasarnya, kedua method ini dilakukan hanya ketika terjadi transaksi kedalam database. Dengan begitu, Anda memperkecil error yang terjadi pada aplikasi Anda karena penggunaan obyek `SqlConnection` yang berulang-ulang.

Sehingga kode program untuk melakukan koneksi bertambah menjadi:

```
Dim conn As SqlConnection ()  
Private Sub Form_Load(...) handles  
MyBase.Load  
    Dim strcon As String = _  
        "Data Source=WEB-SERVER;" & _  
        "Initial Catalog=VISUAL2;User Id=" & _  
        "USER;Password=PASSWORD"  
  
    conn = New SqlConnection(strcon)
```

```
conn.open
'lakukan transaksi
conn.close
End Sub
```

3. Mengatasi kegagalan Koneksi

Seperti yang sudah dijelaskan sebelumnya, koneksi tidak luput dari error sehingga koneksi gagal dilakukan, dan pada akhirnya aplikasi terpaksa harus *debug* dan *exit*. Tujuan mengatasi kegagalan koneksi sebenarnya hanya untuk meng-*handle* agar aplikasi tidak *exit*, selain itu juga untuk memberi tahu error apa yang terjadi dan apa penyebabnya.

Anda hanya menggunakan *try...catch* untuk mengatasi kegagalan koneksi. Namun informasi error yang terjadi dan penyebabnya, tentunya dalam bahasa Inggris. Untuk membuat *custom information*, Anda harus tahu macam-macam Exception yang bakalan terjadi ketika ada kegagalan koneksi. Itu jika Anda mau direpotkan oleh itu semua.

Simply, kode program koneksi setelah ditambahkan *try...catch* menjadi seperti berikut:

```
Dim conn As SqlConnection ()
Private Sub Form_Load(...) handles
MyBase.Load
    Dim strcon As String = _
        "Data Source=WEB-SERVER;" & _
        "Initial Catalog=VISUAL2;User Id=" & _
        "USER;Password=PASSWORD"

    Try
        conn = New SqlConnection(strcon)
        conn.open
        'lakukan transaksi
    Catch ex As Exception
        MessageBox.Show(ex.Message)
    Finally
        conn.close
    End Try
End Sub
```

1.2. Obyek Command

Pengertian Aplikasi terkoneksi

Aplikasi terkoneksi pada ADO .NET adalah sebuah skenario tentang jalannya aplikasi yang selalu terhubung dengan database. Jadi selama aplikasi tersebut dijalankan, maka koneksi ke database harus selalu dalam keadaan terbuka. Namun pada pengaplikasiannya, seperti yang sudah dijelaskan pada materi sebelumnya, **koneksi hanya dilakukan ketika ada transaksi**, entah itu hanya sebuah Select, DML(Data Manipulation Language, Insert-Update-Delete) ataupun DDL(Data Definition Language, Create-Alter-Drop).

Obyek Command

Setelah melakukan koneksi melalui SqlConnection, Anda pasti ingin segera berinteraksi dengan database, seperti Select, Insert, Update dan Delete, atau mungkin memodifikasi database Anda sendiri. Apapun yang akan Anda lakukan berkaitan dengan interaksi Anda dengan database, melibatkan sebuah obyek Command yaitu SqlCommand.

Agar lebih memudahkan pemahaman tentang pemakaian SqlCommand, lihat baris kode berikut:

```
Dim comm As New SqlCommand _  
("perintah QUERY", conn)
```

Constructor dari SqlCommand ada 4 macam, namun hanya 2 yang umum dipakai. Salah satunya seperti contoh kode program dibawah ini:

```
Dim comm As New SqlCommand()  
comm.Connection = conn  
comm.CommandText = "perintah QUERY"
```

Setelah itu, Anda harus mengeksekusi perintah tersebut menggunakan salah satu dari method-method yang ada dibawah ini:

1. ExecuteReader

Method ini menghasilkan sekumpulan data yang berbentuk DataReader yang mempunyai sifat *read-only* dan *forward-only*. Method ini hanya dipakai jika query Anda adalah Select.

2. ExecuteScalar

Method ini menghasilkan sebuah nilai (dalam bentuk 1 baris dan 1 kolom – biasanya disebut dengan 1 cell). Method ini biasanya dipakai ketika query Anda adalah Select Group Function.

3. ExecuteNonQuery

Method ini menghasilkan sebuah nilai integer yang mengindikasikan berapa jumlah baris yang dipengaruhi oleh sebuah query berbentuk DML.

4. ExecuteXMLReader

Method ini menghasilkan versi XML dari DataReader

1.3. Obyek DataReader

DataReader merupakan sebuah obyek yang digunakan untuk membaca data dari database secara cepat. Seperti namanya, sifat dari DataReader adalah *read-only* dan *forward-only* sehingga data yang ada pada obyek DataReader **tidak dapat dimanipulasi** (hanya untuk keperluan **pembacaan data** saja), dan sekali dibaca maka **tidak dapat kembali** lagi untuk membaca data sebelumnya.

DataReader yang digunakan dalam SQL Server Data Provider adalah SqlDataReader. SqlDataReader berasosiasi langsung dengan SqlCommand ketika sang Command melakukan perintah ExecuteReader (perintah query yang diassignkan ke Command adalah sebuah *select query*).

```
Dim dr as SqlDataReader = _
comm.ExecuteReader()
```

Seperti yang dilihat pada contoh, SqlDataReader tidak menggunakan constructor untuk penginisialisasiannya. Tetapi dia langsung di-set dengan menggunakan method ExecuteReader dari SqlCommand. Pembacaan data dalam SqlDataReader adalah per baris data dan menggunakan proses looping (perulangan) mengingat sifatnya yang berupa *forward-only*.

```
While dr.Read()
    MessageBox.Show(dr(0))
End While
```

Sebagai contoh, tabel Mahasiswa dengan struktur:

Nama Kolom	Tipe Data
NIM	Char(11)
Nama	Varchar(50)
Tgl_Lahir	DateTime
Semester	Numeric(18,0)

Untuk mengambil datanya dan ditampung kedalam control interface, diperlukan kode program seperti berikut:

```
Dim conn As SqlConnection()
Private Sub Form_Load(...) handles
MyBase.Load
    Dim strcon As String = _
        "Data Source=WEB-SERVER;" & _
        "Initial Catalog=VISUAL2;User Id=" & _
        "USER;Password=PASSWORD"

    Try
        conn = New SqlConnection(strcon)
        conn.open
        Dim comm As New SqlCommand _
            ("select * from mahasiswa", conn)

        Dim dr as SqlDataReader = _
            comm.ExecuteReader()
        While dr.Read()
            ComboBox1.Items.Add(dr("NIM"))
            Listbox1.Items.Add(dr("Nama"))
            DateTimePicker1.Value = dr(2)
            TextBox1.Text = dr.Item(3)
        End While
    Catch ex As Exception
        MessageBox.Show(ex.Message)
    Finally
        conn.close
    End Try
End Sub
```

Latihan

1. Parameter ConnectionString:
Data Source=WEB-SERVER;Initial Catalog=Visual2;
User Id=Pnimpanjang;Password=Pnimpanjang;
2. Script database:
<http://web-server/pv2/script>
Jalankan script melalui Query window pada SQL Server Management Studio.
3. Contoh/Demo:
<http://web-server/pv2/Latihan>
4. Buat interface form:

Setting Awal:

5. Property FormBorderStyle = FixedToolWindow.
6. Max length dari txtKode adalah 5.
7. Max length dari txtNama adalah 50.
8. cboJurusan berisi semua nama jurusan. Datanya diambil dari tabel Jurusan. Ubah property DropDownStyle menjadi DropDownList.
9. numSKS: NumericUpDown dengan min 1 dan max 10.
10. numSemester: NumericUpDown dengan min 1 dan max 8.
11. btnSimpan, btnUbah dan btnHapus nonaktif (Enable = FALSE).
12. Untuk mengatur icon pada Button, tanyakan pada pengajar.

Cara Kerja:

13. Ketika form Load, cboJurusan terisi.
14. btnCari digunakan untuk mencari data Mata Kuliah sesuai dengan inputan pada txtKode. Jika data yang dicari tidak ada, berikan sebuah pesan.
15. Contoh query untuk pencarian:
`select * from mata_kuliah where kode_mk = '1018'`
16. Jika data yang dicari ada, langsung tampilkan data-datanya pada form sesuai dengan control interfacenya.
17. Yang harus diperhatikan adalah, tabel Mata_Kuliah tidak mempunyai kolom Nama_Jurusan. Sedangkan cboJurusan hanya menampilkan Nama_Jurusan. Untuk mengatasi masalah ini, tanyakan pada pengajar.
18. btnBatal digunakan untuk mengembalikan form kedalam keadaan semua (FORM pertama kali LOAD)

MODUL 2

DATA MANIPULATION LANGUAGE (DML) PADA APLIKASI TERKONEKSI

*Learning is not attained by chance, it must be
sought for with ardor and attended to with
diligence.*
- Abigail Adams -



Tujuan:

Mahasiswa paham dan mengerti bagaimana melakukan DML

Materi:

SqlConnection, SqlCommand

Referensi:

Amelia, Tan, 2008, Pemrograman Visual II Panduan Sertifikasi
MCTS (Exam 70-526), Tanpa Kota, Tanpa Penerbit
Huddleston, James, 2007, Beginning VB 2005 Databases From Novice
to Profesional, New York, Springer-Verlag New York Inc.
Willis, Thearon, 2006, Beginning Visual Basic 2005 Databases,
Indianapolis, Wiley Publishing, Inc.

Data Manipulation Language merupakan sebuah language yang digunakan untuk melakukan insert, update dan delete data dalam object database.

Sebagai contoh, tabel Mahasiswa dengan struktur:

Nama Kolom	Tipe Data
NIM	Char(11)
Nama	Varchar(50)
Tgl_Lahir	DateTime
Semester	Numeric(18,0)

Contoh DML:

Insert

```
Insert into Mahasiswa values
('05410104003', 'Adi', '3-Apr-1986', 6)
-- atau
Insert into Mahasiswa(NIM, Nama) values
('05410104003', 'Adi')
```

Update

```
Update Mahasiswa set Nama='Budi'
Where NIM='05410104003'
```

Delete

```
Delete Mahasiswa Where NIM='05410104003'
```

2.1. DML pada Obyek Command

Pada praktikum minggu sebelumnya, Anda belajar tentang gimana caranya menampilkan data dari tabel. Konsep DML pada SQLCommand sama seperti konsep SELECT. Anda hanya mengubah parameter query ketika Anda meng-instance-kan SQLCommand.

```
Dim query as String = _
"select * from mahasiswa"
Dim comm as New SqlCommand(query, conn)
```

Menjadi

```
Dim query as String = _
"Insert into Mahasiswa(NIM>Nama) values
('05410104003','Adi')"
Dim comm as New SqlCommand(query, conn)
```

Menurut James Huddleston dalam bukunya *Beginning VB 2005 Databases From Novice to Professional*, mengatakan bahwa ketika kita melakukan DML, kode-kode SQL kita tulis pada VB .NET 2005 kita sesuaikan dengan kode-kode SQL pada SQL Editor (Query Analyzer, Query Window dari Management Studio). Walaupun cara ini valid, cara ini bukanlah cara yang baik karena untuk setiap transaksi, Anda harus mengubah lagi kode programnya. Kode SQL yang baik adalah kode SQL yang mampu melakukan DML kapanpun. Ada 2 cara untuk melakukan hal ini. Salah satunya kurang efektif bila dibandingkan dengan cara lainnya.

Cara yang kurang efektif adalah Anda harus membuat kode SQL dinamis pada kode program Anda. Sebagai contoh:

```
Dim conn as new SqlConnection(strcon)
Dim vnim as String = "05410104003"
Dim vnama as String = "Adi"
Dim values as String = _
("'" & vnim & "','" & vnama & "'")
Dim query as String = _
"Insert into Mahasiswa(NIM>Nama) values"
& values

conn.open()
Dim comm as New SqlCommand(query, conn)
comm.ExecuteNonQuery()
```

Cara yang jauh lebih baik dari cara tersebut adalah dengan menggunakan command parameters, sebuah placeholder dalam obyek command untuk menyimpan nilai-nilai yang akan diproses oleh query. Dimulai dengan karakter @ dan diikuti oleh nama parameter tanpa ada spasi. Contoh kode program:

```
Dim conn as new SqlConnection(strcon)
Dim vnim as String = "05410104003"
Dim vnama as String = "Adi"

Dim query as String = _
```

```
"Insert into Mahasiswa (NIM, Nama)
values ('@vnmim', '@vnama' )"
Dim comm as New SqlCommand(query, conn)
comm.Parameters.Add("@vnmim", SqlDbType.NVarChar, 11)
comm.Parameters.Add("@vnama", SqlDbType.NVarChar, 50)

Try
    conn.Open()
    comm.Parameters("@vnmim").Value = vnmim
    comm.Parameters("@vnama").Value = vnama
    comm.ExecuteNonQuery()
Catch ex as Exception
    MessageBox.Show(ex.Message)
Finally
    conn.close()
End Try
```

2.2. ExecuteNonQuery

Jika Anda perhatikan, beberapa contoh kode program diatas menggunakan `comm.ExecuteNonQuery`. Ini adalah sebuah method yang digunakan oleh `SqlCommand` untuk mengeksekusi query DML yang diberikan kepadanya.

```
Dim hasil as Integer
hasil = comm.ExecuteNonQuery()
```

Seperti yang Anda lihat, method ini mengembalikan sebuah nilai Integer. Nilai ini berisi jumlah baris yang terpengaruh oleh query DML. Misal update 2 baris data mengembalikan nilai 2, delete 5 baris data mengembalikan nilai 5. Namun, nilai ini jarang digunakan.

Latihan

1. Buat interface form:

Setting Awal:

2. Property `FormBorderStyle = FixedToolWindow`.
3. Max length dari `txtKode` adalah 5.
4. Max length dari `txtNama` adalah 50.
5. `cboJurusan` berisi semua nama jurusan. Datanya diambil dari tabel Jurusan. Ubah property `DropDownStyle` menjadi `DropDownList`.
6. `numSKS`: `NumericUpDown` dengan min 1 dan max 10.
7. `numSemester`: `NumericUpDown` dengan min 1 dan max 8.
8. `btnSimpan`, `btnUbah` dan `btnHapus` aktif (`Enable = TRUE`).
9. Untuk mengatur icon pada Button, tanyakan pada pengajar.

Cara Kerja:

10. Ketika form Load, `cboJurusan` terisi.
11. `btnCari` digunakan untuk mencari data Mata Kuliah sesuai dengan inputan pada `txtKode`. Jika data yang dicari tidak ada, berikan sebuah pesan.
12. Contoh query untuk pencarian:
`select * from mata_kuliah where kode_mk = '1018'`
13. Jika data yang dicari ada, langsung tampilkan data-datanya pada form sesuai dengan control interfacenya.

14. Yang harus diperhatikan adalah, tabel Mata_Kuliah tidak mempunyai kolom Nama_Jurusan. Sedangkan cboJurusan hanya menampilkan Nama_Jurusan. Untuk mengatasi masalah ini, tanyakan pada pengajar.
15. btnSimpan digunakan untuk melakukan INSERT INTO kedalam tabel Mata_Kuliah.
16. btnUbah digunakan untuk melakukan UPDATE kedalam tabel Mata_Kuliah.
17. btnHapus digunakan untuk melakukan DELETE kedalam tabel Mata_Kuliah.
18. btnBatal digunakan untuk mengembalikan form kedalam keadaan semua (FORM pertama kali LOAD).

MODUL 3

DATASET

*Begitu banyak pelajaran penting yang bisa
kamu kuasai di SEKOLAH KEHIDUPAN
Semakin banyak yang kamu pelajari dan
praktekkan, maka semakin kaya hidupmu
- anonim -*



Tujuan:

Mahasiswa paham dan mengerti bagaimana membuat DataSet
Mahasiswa paham dan mengerti bagaimana membuat DataTable
Mahasiswa paham dan mengerti bagaimana DML dalam DataTable

Materi:

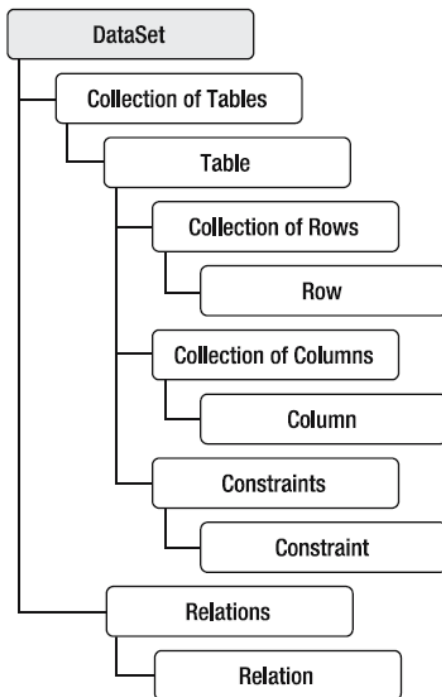
DataSet, DataTable, DataColumn, DataRow

Referensi:

Amelia, Tan, 2008, Pemrograman Visual II Panduan Sertifikasi
MCTS (Exam 70-526), Tanpa Kota, Tanpa Penerbit
Huddleston, James, 2007, Beginning VB 2005 Databases From Novice
to Profesional, New York, Springer-Verlag New York Inc.
Willis, Thearon, 2006, Beginning Visual Basic 2005 Databases,
Indianapolis, Wiley Publishing, Inc.

Pada 2 materi sebelumnya, kita semua belajar tentang koneksi database menggunakan *connected environment* yang bersifat *read-only* dan *forward-only*. Pada modul ini, kita akan belajar tentang cara lain untuk berkomunikasi dengan database, yaitu dengan menggunakan obyek DataSet. DataSet merupakan obyek dari class System.Data.DataSet yang bisa dipakai oleh semua data provider, independen, dan bisa digunakan baik secara terkoneksi atau tidak terkoneksi dengan *data source*.

DataSet mampu menggambarkan sekumpulan data yang tersimpan dalam berbagai DataTable, constraint serta hubungan antar tabel. Selain itu, dia juga mampu menyimpan begitu banyak data dan memprosesnya di secara offline, dengan kata lain, *disconnected* dari database. Tujuan utama adanya DataSet adalah sebagai penyedia layanan database secara virtual yang disimpan dalam *memory cache*.



Seperti yang bisa kita lihat di gambar, DataSet terdiri dari sekumpulan tabel yang biasa disebut sebagai DataTable. DataTable

ini dibentuk dari sekumpulan kolom yang biasa disebut sebagai DataColumn dan sekumpulan baris yang biasa disebut sebagai DataRow. Selain terdiri dari sekumpulan DataTable, DataSet juga mampu mengatur Constraint dan hubungan antar table yang biasa disebut sebagai DataRelation.

3.1. Cara Membuat DataSet

Ada beberapa cara untuk membuat DataSet:

Membuat dari RDBMS dengan menggunakan SqlDataAdapter (modul 4).

Membaca dari file XSD (modul 6).

Membaca dari file XML (modul 6).

Membangunnya dari DataTables, DataRelations dan Constraints lalu mempopulasikan datanya.

Membuat dari DataReader.

Yang akan kita pelajari untuk modul minggu ke-3 hanyalah membuat DataSet dari DataTables, DataRelations dan Constraints. Cara ini akan kita lakukan dengan menggunakan kode program. Dan hal pertama yang perlu dilakukan adalah meng-instance-kan DataSet tersebut.

```
Dim ds As New DataSet("MyDataSet")
```

3.2. DataTable

Seperti yang udah kita ketahui kalo DataSet dibangun dari DataTables, DataRelations ama Constraints. Pada ADO.NET, DataTables merepresentasikan tabel-tabel pada DataSet. Semua data yang ada pada DataTable hanya bisa digunakan ketika program *running*, namun bisa juga dikaitkan langsung dengan *data source* misal SQL Server dengan menggunakan SqlDataAdapter.

```
Dim dt As New DataTable("MyDataTable")
```

Untuk menambahkan DataTable kedalam DataSet:

```
ds.Tables.Add(dt)
```

3.3. DataColumn

Dalam DBMS, sebuah tabel dibangun dari sekumpulan kolom. DataTable menggunakan konsep yang sama, namun menggunakan DataColumn.

```
Dim dc As New DataColumn("No_Nota")
dc.DataType = GetType(String)
```

Untuk nambahin kolom tersebut kedalam DataTable:

```
dt.Columns.Add(dc)
```

Trus gimana kalo kolomnya banyak? Biasanya penulis memakai cara seperti ini:

```
Dim str() as String = _
new String() {"No_Nota", "Tanggal", "IDSup"}

For i As Integer = 0 to str.Length - 1
Dim dc As New DataColumn(str(i))
    dt.Columns.Add(dc)
Next
```

Dari looping itu, setidaknya kita udah mbuat tabel dengan 3 kolom. Untuk mengatur tipe kolomnya, yang perlu dilakukan hanyalah mengambil kolom tersebut langsung dari DataTable.

```
dt.Columns("No_Nota").DataType = _
GetType(String)
dt.Columns("Tanggal").DataType = _
GetType(Date)
dt.Columns("IDSup").DataType = _
GetType(Integer)
```

3.4. Constraint

Ada 2 macam berdasarkan levelnya, column constraints dan table constraints. Untuk column constraints, bisa ditulis langsung pada DataColumn milik DataTable.

```
`kolom boleh null
dt.Columns("IDSup").AllowDBNull = True

`mengatur panjang kolom
```

```
dt.Columns("No_Nota").MaxLength = 13
`nilai kolom tidak dapat diubah
dt.Columns("Tanggal").ReadOnly = True
`nilai kolom harus unique
dt.Columns("No_Nota").Unique = True
```

Untuk table constraints:

Unique Constraints

```
dt.Constraints.Add("No_Nota_UQ",
dt.Columns("No_Nota"), False)
```

Primary Key

```
Dim pk() As DataColumn =
new DataColumn() {dt.Columns("No_Nota")}
dt.PrimaryKey = pk
```

Foreign Key

```
dtDetil.Constraints.Add("No_Nota_FK", _
dt.Columns("No_Nota"), _
dtDetil.Columns("No_Nota"))
```

3.5. DataRelation

Terkadang, tabel-tabel dalam database saling berhubungan untuk memastikan apakah data-data yang dientrikan pada tabel-tabel transaksi sesuai dengan data-data pada tabel master. Pada dasarnya, DataRelation membuat Foreign Key Constraints. Untuk membuat DataRelation, kita butuh obyek dari DataSet.

```
ds.Relations.Add("rel_Nota", _
dt.Columns("No_Nota"), _
dtDetil.Columns("No_Nota"), True)
```

3.6. Manipulasi Data

Untuk membuat baris-baris data dalam sebuah DataTable dibutuhkan DataRow.

Menambah baris data.

```
Dim row As DataRow = dt.NewRow
row("No_Nota") = "NO.0408.0001"
row("Tanggal") = Now
```

```
row("IDSup") = 40
```

Untuk menambahkan baris datanya kedalam DataTable:

```
dt.Rows.Add(row)
```

Mengubah baris data.

```
Dim row As DataRow =  
dt.Rows.Find("NO.0408.0001")  
  
If Not IsNothing(dr) Then  
    row.BeginEdit()  
    row.IDSup = "50"  
    row.EndEdit()  
End If
```

Untuk membatalkan perubahan bisa pake method row.CancelEdit

Menghapus baris data.

```
Dim row As DataRow =  
dt.Rows.Find("NO.0408.0001")  
  
If Not IsNothing(dr) Then  
    dt.Rows.Remove(row)  
End If
```

3.7. AcceptChanges and RejectChanges

Digunakan sebagai cara untuk mengontrol transaksi. Sama seperti perintah Commit dan Rollback pada Oracle. Namun , method ini hanya berlaku di DataSet, DataTable dan DataRow.

```
dt.AcceptChanges()  
dt.RejectChanges()
```

dt.RejectChanges akan mengembalikan keadaan data pada DataTable sesuai dengan data ketika terakhir kali dt.AcceptChanges dijalankan.

3.8. DataGridView

Untuk menampilkan data-data apa saja yang ada pada DataTable, kita bisa menggunakan bantuan control DataGridView dengan menjalankan kode berikut

```
DataGridView1.DataSource = dt
```

Atau

```
DataGridView1.DataSource = dt.DefaultView
```

Latihan

1. Buat desain interface:



Kotak abu-abu di bawah numSemester adl DataGridView.

Setting Awal:

Property FormBorderStyle = FixedToolWindow.

MaxLength dari txtKode = 5

MaxLength dari txtNama = 50

numSKS = NumericUpDown dengan min 1 dan max 10

numSemester = NumericUpDown dengan min 1 dan max 8

Semua button aktif

DataGridView → AutoSizeColumnsMode = AllCells

Cara Kerja:

Tidak ada koneksi ke database.

Buat sebuah DataTable dengan nama dtMataKuliah

```
Dim dtMataKuliah As New DataTable("Mata  
Kuliah").
```

Tambahkan DataColumn pada DataTable tersebut.

Column Name	Type and Length	Constraint
Kode_MK	String(5)	Primary Key
Nama	String(50)	
Kode_Jurusan	String(4)	
SKS	Integer	
Semester	Integer	

Contoh kode program untuk membuat kolom Kode_MK:

```
dtMataKuliah.Columns.Add(new
DataColumn("Kode_MK"))
With dtMataKuliah.Columns("Kode_MK")
.DataType = GetType(String)
.MaxLength = 5
End With
```

Untuk Primary Key, bisa Anda lihat dimodul/buku ajar Anda.

- Setelah Anda membuat dtMataKuliah beserta kolom dan constraint-nya, Anda set Data Source dari DataGridView Anda ke dtMataKuliah. Lihat materi di modul/buku ajar.
- Kode program untuk no. 11 dan 12 dikerjakan pada event dengan Handles MyBase.Load.

cboJurusan diisi manual melalui kode program dengan isi:

D3KA
D3MI
D4MM
S1SI
S1SK

- Melalui event dengan Handles DataGridView1.Click tulis kode ini:

```
Dim row As DataRow = _
dtMataKuliah.Rows(DataGridView1.CurrentRow
.Index)
txtKode.Text = row("Kode_MK")
txtNama.Text = row("Nama")
```

```
cboJurusan.SelectedItem =  
row("Kode_Jurusan")  
numSKS.Value = row("SKS")  
numSemester.Value = row("Semester")
```

5. Melalui event dengan Handles `btnSimpan.Click`, Anda melakukan penambahan data kedalam `dtMataKuliah` (bukan `DataGridView`).
6. Melalui event dengan Handles `btnUbah.Click`, Anda melakukan perubahan data kedalam `dtMataKuliah` (bukan `DataGridView`).
7. Melalui event dengan Handles `btnHapus.Click`, Anda melakukan penghapusan data dari `dtMataKuliah` (bukan `DataGridView`).
8. Melalui event dengan Handles `btnBatal.Click`, Anda membersihkan semua inputan yang ada didalam `TextBox`, `ComboBox` dan `NumericUpDown`.
9. Melalui event dengan Handles `btnKeluar.Click`, program keluar.

MODUL 4

DATASET DAN APLIKASI TERPUTUS

Kecenderungan untuk mengembangkan yang terbaik pada diri kita, akan menghasilkan yang terbaik untuk kita dan orang lain.

Begitu juga sebaliknya. Kita tinggal pilih :

SUKSES ATAU TERJUNGKAL

- anonim -



Tujuan:

Mahasiswa paham dan mengerti tentang pencarian data

Mahasiswa paham dan mengerti tentang pengurutan data

Mahasiswa paham dan mengerti tentang DataView

Mahasiswa paham dan mengerti tentang DataAdapter

Materi:

Sorting, Filtering, DataColumn, DataRow

Referensi:

Amelia, Tan, 2008, Pemrograman Visual II Panduan Sertifikasi
MCTS (Exam 70-526), Tanpa Kota, Tanpa Penerbit

Huddleston, James, 2007, Beginning VB 2005 Databases From Novice
to Professional, New York, Springer-Verlag New York Inc.

Willis, Thearon, 2006, Beginning Visual Basic 2005 Databases,
Indianapolis, Wiley Publishing, Inc.

Jika minggu yang lalu kita udah bahas tentang gimana bikin DataTable, DataColumn ama gimana nambahin baris-baris data pada DataTable, sekarang kita akan bahas tentang fasilitas yang HARUS ada pada kumpulan data... yaitu gimana caranya NYARI ama NGURUTIN data. Karena kedua fasilitas ini nantinya berhubungan ama tampilan data pada FORM.

4.1. Pencarian dengan metode Find, Contains dan Select pada DataTable

Pada dasarnya, proses pencarian dilakukan dengan bantuan Primary Key. Namun hal itu terkadang tidak berlaku untuk beberapa kasus. Metode Find dan Contains membutuhkan bantuan PK, sedangkan metode Select tidak.

Proses yang diperlukan untuk melakukan Find dan Contains adalah SAMA. Yang berbeda hanyalah hasil dari keduanya. Metode Find menghasilkan DataRow sedangkan metode Contains menghasilkan Boolean.

Find dengan single PK

```
Dim row As DataRow =  
dt.Rows.Find("01410100077")  
If Not IsNothing(row) then  
    'berarti datanya ketemu  
    MessageBox.Show("Data Ada.")  
    Tampil(row)  
Else  
    MessageBox.Show("Data Tidak Ada.")  
End If
```

Find dengan multiple PK

```
Dim krit(2) as Object  
krit(0) = "01410100077"  
krit(1) = "Tegar"  
Dim row As DataRow = dt.Rows.Find(krit)  
If Not IsNothing(row) then  
    MessageBox.Show("Data Ada.")  
    Tampil(row)  
Else  
    MessageBox.Show("Data Tidak Ada.")  
End If
```

Contains dengan single PK

```
Dim statusCari as Boolean =
dt.Rows.Contains("01410100077")
If statusCari = True Then
    MessageBox.Show("Data Ada.")
Else
    MessageBox.Show("Data Tidak Ada.")
End If
```

Contains dengan multiple PK

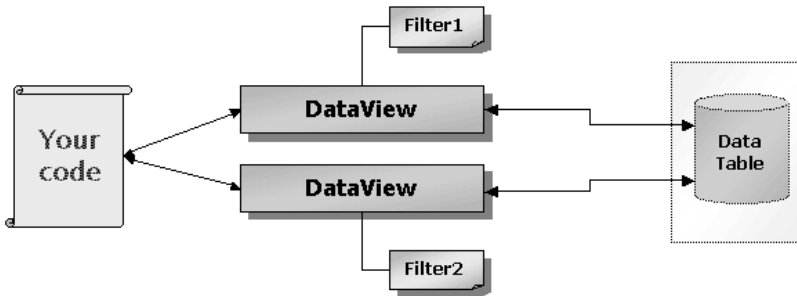
```
Dim kriteria(2) as Object
kriteria(0) = "01410100077"
kriteria(1) = "Tegar"
Dim statusCari as Boolean =
dt.Rows.Contains(kriteria)
If statusCari = True Then
    MessageBox.Show("Data Ada.")
Else
    MessageBox.Show("Data Tidak Ada.")
End If
```

Untuk metode Select, hasil berupa array of DataRow.

```
Dim row() as DataRow
row = dt.Select(" Kota like 'S%' " & _
"AND Sts='M' ")
If row.Length > 0 Then
    MessageBox.Show("Data Ada.")
    For i As Integer = 0 To row.Length-1
        Tampil(row(i))
    Next
Else
    MessageBox.Show("Data Tidak Ada.")
End If
```

4.2. DataView

Nggak usah bingung ama yang satu ini. Pada dasarnya, DataView sama seperti View dalam object Database (selain Table, Index, User, de-el-el). Keuntungan dari penggunaan DataView adalah pencarian dan pengurutan yang langsung *ter-binding* dengan data bound control (contoh: DataGridView). Selain itu, pencarian data dapat dilakukan tanpa bantuan PK.



Kita bisa membuat DataView dari DataTable dengan menggunakan kode program berikut:

```
Dim dv As DataView = dt.DefaultView
```

Pencarian data pada DataView bisa menggunakan RowFilter, Find dan FindRows. Sedangkan untuk pengurutan data menggunakan Sort.

RowFilter

```
dv.RowFilter = " Kota='Surabaya' "
```

Untuk Find dan FindRows, DataView harus di-Sort terlebih dahulu. Metode Find menghasilkan nilai Integer (index baris atau -1 jika tidak ada), sedangkan FindRows menghasilkan array of DataRowView.

Sort

```
dv.Sort = "NIM ASC"
```

Find dengan single kriteria

```
Dim hasilCari As Integer =  
dv.Find("01410100077")  
If hasilCari >= 0 Then  
    MessageBox.Show("Data Ada.")  
Else  
    MessageBox.Show("Data Tidak Ada.")  
End If
```

Find dengan multiple kriteria, DataView harus di-Sort berdasarkan kriteria.

```
dv.Sort = "NIM ASC, Kota ASC"  
Dim kriteria(2) as String  
kriteria(0) = "01410100077"
```

```
kriteria(1) = "Surabaya"  
Dim hasilCari As Integer =  
dv.Find(kriteria)  
If hasilCari >= 0 Then  
    MessageBox.Show("Data Ada.")  
Else  
    MessageBox.Show("Data Tidak Ada.")  
End If
```

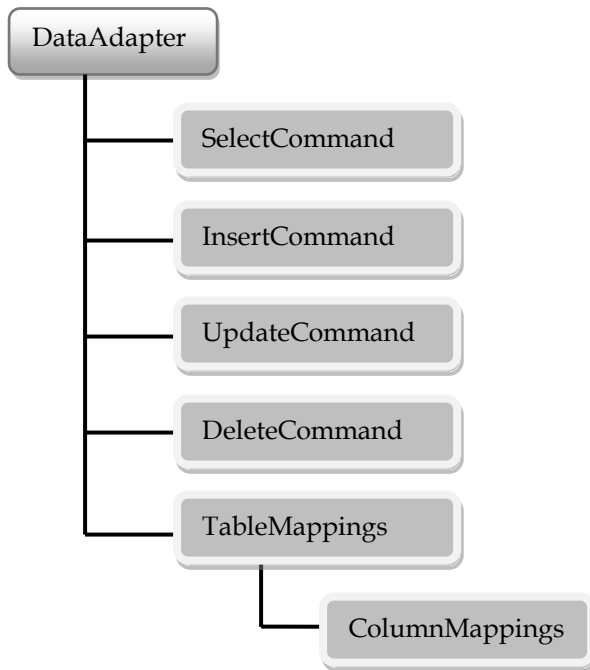
FindRows

```
dv.Sort = "Kota"  
Dim rowView() As DataRowView  
rowView() = dv.FindRows("Surabaya")  
If rowView.Length > 0 Then  
    MessageBox.Show("Data Ada.")  
    For i as Integer =0 To rowView.Length-1  
        Tampil(rowView(i))  
    Next  
Else  
    MessageBox.Show("Data Tidak Ada.")  
End If
```

4.3. Aplikasi Terputus

Penggunaan memori komputer untuk mengolah data dari database, sehingga koneksi pada database tidak dilakukan terus menerus tetapi hanya pada saat mengambil data dan menyimpan perubahan data saja.

Seperti yang telah dijelaskan sebelumnya bahwa data yang diambil membutuhkan sebuah object DataSet/DataTable. Proses pengambilan ini membutuhkan object perantara berupa DataAdapter.



Untuk membuat obyek DataAdapter, kita tinggal nulis baris-baris kode berikut kedalam VB

```
Dim conn As New SqlConnection(strcon)
Dim da As New SqlDataAdapter("Select *
From Mahasiswa", conn)
```

Untuk pengisian kedalam DataTable gunakan kode berikut

```
da.Fill(dt)
```

Metode Fill hanya mengambil data-data sesuai dengan query yang diberikan pada SqlDataAdapter. Untuk mengambil struktur tabelnya, tambahkan perintah berikut sebelum Fill.

```
da.MissingSchemaAction =
MissingSchemaAction.AddWithKey
```

Ada 4 properti sebenarnya, mulai dari Add, AddWithKey, Error, dan Ignore. Namun yang umum dipakai adalah AddWithKey.

Untuk menampilkan DataTable yang sudah terisi oleh data dari database, coba lihat modul 3 materi tentang DataGridView.

Untuk menyimpan perubahan Dataset/DataTable kedalam Database, kita hanya perlu metode Update dari bject DataAdapter. Namun sebelumnya, kita harus membuat InsertCommand, UpdateCommand dan DeleteCommand.

```
Dim cb As New SqlCommandBuilder(da)
da.Update(dt)
```

Untuk langkah-langkah detil dari proses Update itu sendiri sebenarnya ada 3 proses:

```
'Delete
da.Update(dt.Select(Nothing, Nothing,
DataRowState.Deleted))
'Update
da.Update(dt.Select(Nothing, Nothing,
DataRowState.ModifiedCurrent))
'Insert
da.Update(dt.Select(Nothing, Nothing,
DataRowState.Added))
```

Latihan

1. Buat desain interface:

Kotak abu-abu di bawah numSemester adl DataGridView.

Setting Awal:

Property FormBorderStyle = FixedToolWindow.

MaxLength dari txtKode = 5

MaxLength dari txtNama = 50

numSKS = NumericUpDown dengan min 1 dan max 10

numSemester = NumericUpDown dengan min 1 dan max 8

Semua button aktif

DataGridView → AutoSizeColumnsMode = AllCells

Cara Kerja:

Kode_MK	Nama	Kode_Jurusan
1000	Test	D3MI
1001		
1018	Bahasa Inggris	S1SI
102	Test2	S1SI
105		
106	Test6	

Buat koneksi ke dalam database.

```
Dim conn As New SqlConnection("...;...;...")
```

Buat sebuah DataTable dengan nama dtMataKuliah

```
Dim dtMataKuliah As New DataTable("Mata  
Kuliah").
```

2. Buat sebuah DataAdapter untuk mengambil struktur dan semua data pada tabel Mata_Kuliah untuk kemudian ditampung dalam dtMataKuliah.

```
Dim daMataKuliah As New SqlDataAdapter  
  
daMataKuliah = New SqlDataAdapter("select  
* from mata_kuliah", conn)  
daMataKuliah.Fill(dtMataKuliah)
```

3. set Data Source dari DataGridView ke dtMataKuliah.
4. Kode program untuk no. 11 dan 12 dikerjakan pada event dengan Handles MyBase.Load.

5. cboJurusan diisi sesuai dengan data yang ada di tabel Jurusan. Gunakan DataTable untuk mengambil semua data dari tabel Jurusan. Lalu untuk memasukkan datanya, gunakan kode berikut:

```
cboJurusan.DataSource = dtJurusan
cboJurusan.DisplayMember = "Nama_Jurusan"
cboJurusan.ValueMember = "Kode_Jurusan"
```

6. Melalui event dengan Handles DataGridView1.Click tulis kode ini:

```
Dim row As DataGridViewRow =
DataGridView1.Rows(DataGridView1.CurrentRow.Index)

txtKode.Text = row.Cells("Kode_MK").Value

If Not IsDBNull(row.Cells("Nama").Value)
Then txtNama.Text = row.Cells("Nama").Value

If Not IsDBNull(row.Cells("Kode_Jurusan").Value)
Then cboJurusan.SelectedValue = row.Cells("Kode_Jurusan").Value

If Not IsDBNull(row.Cells("SKS").Value)
Then numSKS.Value = row.Cells("SKS").Value

If Not IsDBNull(row.Cells("Semester").Value)
Then numSemester.Value = row.Cells("Semester").Value
```

7. Melalui event dengan Handles btnCariKode.Click, Anda melakukan pencarian berdasarkan kode. Jika datanya ada, langsung tampilkan data hasil pencarian kedalam control yang bersesuaian. Jika tidak, tampilkan pesan.

8. Melalui event dengan Handles `btnCariNama.Click`, Anda melakukan pencarian berdasarkan nama (gunakan wildcards). Hasil pencarian langsung mempengaruhi `DataGridView`.

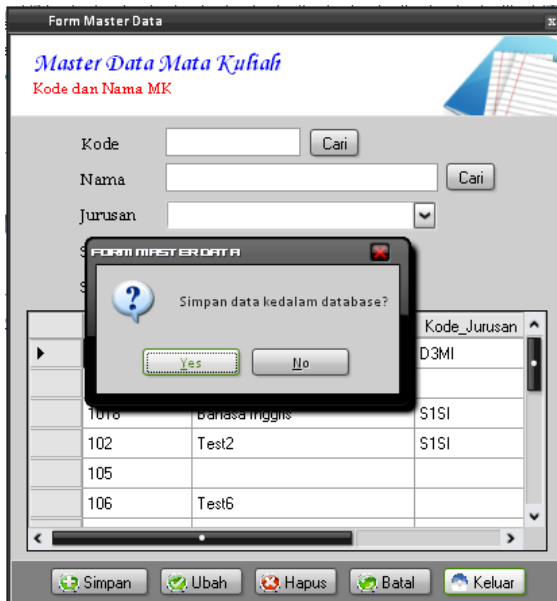
The screenshot shows a Windows application window titled "Form Master Data". The window contains a search form and a DataGrid. The search form has fields for "Kode", "Nama" (with "test" entered), "Jurusan", "SKS", and "Semester". Below the form is a DataGrid with columns: Kode_MK, Nama, Kode_Jurusan, SKS, and Semest. The first row is selected and highlighted in black. At the bottom of the window are buttons for "Simpan", "Ubah", "Hapus", "Batal", and "Keluar".

	Kode_MK	Nama	Kode_Jurusan	SKS	Semest
▶	1000	Test	D3MI	2	2
	102	Test2	S1SI	3	5
	106	Test6		4	3
	107	Test7	S1SI	4	3
*					

Setelah salah satu data pada `DataGridView1` di-click, kembalikan data pada `DataGridView1` menjadi seperti sebelum proses pencarian berdasarkan Nama dilakukan.

9. Melalui event dengan Handles `btnSimpan.Click`, Anda melakukan penambahan data kedalam `dtMataKuliah` (bukan `DataGridView`).
10. Melalui event dengan Handles `btnUbah.Click`, Anda melakukan perubahan data kedalam `dtMataKuliah` (bukan `DataGridView`).
11. Melalui event dengan Handles `btnHapus.Click`, Anda melakukan penghapusan data dari `dtMataKuliah` (bukan `DataGridView`).
12. Melalui event dengan Handles `btnBatal.Click`, Anda membersihkan semua inputan yang ada didalam `TextBox`, `ComboBox` dan `NumericUpDown`.

- 13. Melalui event dengan Handles btnKeluar.Click, sebelum keluar dari aplikasi, lakukan pengecekan apakah user mau menyimpan data pada database atau tidak. Jika user memilih untuk menyimpan kedalam database, lakukan perintah Update



```
Dim cb As New  
SqlCommandBuilder(daMataKuliah)  
  
daMataKuliah.Update(dtMataKuliah)
```

MODUL 5

DATA BINDING

Kesulitan bukan ditimbulkan oleh orang yang menceritakan segala yang diketahuinya, tetapi orang yang lebih banyak bicara daripada yang diketahuinyalah yang menjadi penyebabnya
- anonim -



Tujuan:

Mahasiswa paham dan mengerti tentang DataBinding

Materi:

DataBinding

Referensi:

- Amelia, Tan, 2008, Pemrograman Visual II Panduan Sertifikasi MCTS (Exam 70-526), Tanpa Kota, Tanpa Penerbit
- Huddleston, James, 2007, Beginning VB 2005 Databases From Novice to Profesional, New York, Springer-Verlag New York Inc.
- Willis, Thearon, 2006, Beginning Visual Basic 2005 Databases, Indianapolis, Wiley Publishing, Inc.

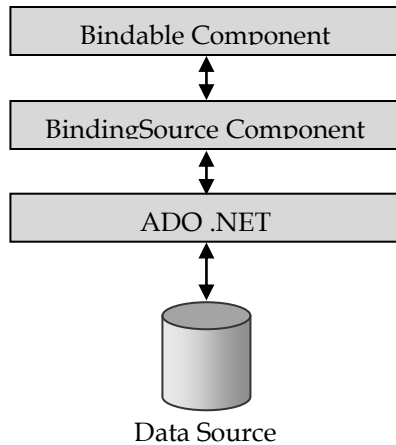
5.1. Pengertian Data Binding

Data Binding adalah sebuah fasilitas yang tersedia dalam .NET Framework untuk menghubungkan kontrol yang ada pada form dengan data source, seperti pada DataSet, DataTable, DataView atau Array.

Dengan adanya koneksi antara kontrol pada form dengan data source maka setiap kali ada perubahan data pada data source, otomatis tampilan data pada kontrol juga ikut berubah. Sebaliknya, jika kita menambah, mengubah atau menghapus data langsung dari kontrol yang ada pada form, otomatis data yang ada pada data source juga ikut berubah.

5.2. Aliran data pada Data Binding

Mengerti dengan baik konsep dari aliran data yang terjadi antara kontrol pada form dengan data source sangat dibutuhkan untuk mengetahui cara kerja Data Binding. Berikut ini diagram untuk menjelaskan aliran data yang terjadi pada data binding:



Dalam .NET, kontrol yang ada pada form mempunyai banyak properti yang bisa dihubungkan pada sebuah data source. Setiap properti yang bersifat data-bound berkoneksi dengan sebuah object binding.

5.3. Kontrol Data-Bound

Kontrol yang tersedia pada Toolbox seperti TextBox, ComboBox, ListBox dan DataGridView dikenal juga dengan sebutan kontrol data-bound. Hal tersebut dikarenakan pada kontrol-kontrol ini, sudah tersedia properti untuk melakukan data binding pada sebuah data source.

Berikut ini syntax yang digunakan untuk melakukan data binding pada beberapa kontrol data-bound:

Data binding pada kontrol TextBox

```
tbNama.DataBindings.Add("Text",
dsPerdagangan, "Supplier>NamaSupplier")
```

Data binding pada kontrol ComboBox

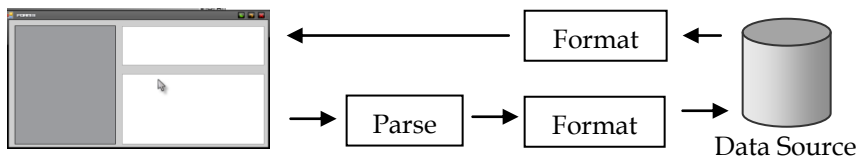
```
cbSatuan.DataSource =
dsPerdagangan.Tables("Satuan")
cbSatuan.DisplayMember = "Nama_Satuan"
cbSatuan.ValueMember = "Kode_Satuan"
```

Data binding pada kontrol DataGridView

```
gridPembelian.DataSource = dsPerdagangan
gridPembelian.DataMember = "Pembelian"
```

5.4. Event Format dan Parse pada Object Binding

Dalam banyak kasus, seting terjadi bahwa data yang ada didalam data source kadangkala ingin ditampilkan berbeda pada kontrol. Hal ini biasanya terjadi pada data dengan tipe data datetime dan decimal.



Untuk mengakomodasi kebutuhan akan format data pada kontrol yang telah di binding, maka pada object binding terdapat dua event fungsi tersebut: Format dan Parse.

Event Format digunakan untuk memformat data dari data source sebelum ditampilkan pada kontrol. Sedangkan event Parse digunakan untuk menghapus format yang telah dilakukan pada kontrol sebelum dikembalikan lagi ke data source.

Untuk memanggil event tersebut, kita harus membuat object Binding dengan keyword WithEvents agar terdapat event untuk format dan parse. Berikut contoh syntax-nya:

```
Dim WithEvents binTanggal As Binding
binTanggal = new Binding("Text",
dsPerdagangan, "Pembelian.Tanggal")
dtpTanggal. DataBindings.Add(binTanggal)

Sub Tanggal_Format(ByVal sender as
Object, ByVal e As ConvertEventArgs)
Handles binTanggal.Format
    e.Value = Format(e.Value, "dd/MM/yyyy")
End Sub

Sub Tanggal_Parse(ByVal sender as Object,
ByVal e As ConvertEventArgs) Handles
binTanggal.Parse
    e.Value = DateTime.Parse(e.Value)
End Sub
```

5.5. Object BindingSource dan BindingNavigator

BindingSource adalah sebuah object yang digunakan untuk menghubungkan komponen-komponen dalam sebuah form dengan data source.



Berikut ini contoh penggunaan object BindingSource:

```
Dim mhsBindingSource as BindingSource
mhsBindingSource = new
BindingSource(dsAkademik, "Mahasiswa")
```



```
DataGridView1.DataSource =
mhsBindingSource
tbNim.DataBindings.Add("Text",
mhsBindingSource, "Nim")
```

Pada komponen DataGridView, kita bisa menambahkan object BindingNavigator untuk dapat berpindah antar record dengan lebih mudah.

Berikut ini contoh menghubungkan object BindingNavigator dengan object BindingSource

```
mhsBindingNavigator.BindingSource =
mhsBindingSource
```

Tabel dibawah ini merupakan daftar properti oada BindingSource yang dapat digunakan:

Properti	Keterangan
Count	Untuk mengetahui jumlah record
Position	Mencari index record yang sedang aktif
MoveFirst	Berpindah ke record pertama
MoveLast	Berpindah ke record terakhir
MoveNext	Berpindah ke record selanjutnya
MovePrevious	Berpindah ke record sebelumnya

Untuk memanipulasi data dari object BindingSource:

Metode	Keterangan
AddNew	Untuk menambah record baru
CancelEdit	Membatalkan perubahan record
EndEdit	Menyimpan perubahan record
RemoveAt	Menghapus record

Latihan

Buat desain interface:



Kotak abu-abu di bawah numSemester adl BindingNavigator.

Setting Awal:

- Property FormBorderStyle = FixedToolWindow.
- MaxLength dari txtKode = 5
- MaxLength dari txtNama = 50
- numSKS = NumericUpDown dengan min 1 dan max 10
- numSemester = NumericUpDown dengan min 1 dan max 8
- btnUbah dan btnHapus enable = False. Sisanya enable = True
- Tambahkan control BindingSource kedalam form

Cara Kerja:

The screenshot shows a Windows application window titled "Form Master". The window contains a form titled "Master Data Mata Kuliah" with the subtitle "Kode dan Nama MK". The form has several input fields: "Kode" with the value "1000", "Nama" with "Test", "Jurusan" with a dropdown menu showing "D4 Multimedia", "SKS" with a spinner box set to "3", and "Semester" with a spinner box set to "2". Below the form is a navigation bar with buttons for "Simpan", "Ubah", "Hapus", "Batal", and "Keluar". There is also a status bar showing "1 of 18" and some navigation icons.

Buat koneksi ke dalam database.

```
Dim conn As New SqlConnection("...;...;...")
```

Buat sebuah DataTable dengan nama dtMataKuliah

```
Dim dtMataKuliah As New DataTable("Mata  
Kuliah").
```

1. Buat sebuah SqlDataAdapter untuk mengambil struktur dan semua data pada tabel Mata_Kuliah untuk kemudian ditampung dalam dtMataKuliah.

```
Dim daMataKuliah As New SqlDataAdapter  
daMataKuliah = New SqlDataAdapter("select  
* from mata_kuliah", conn)  
daMataKuliah.Fill(dtMataKuliah)
```

2. BindingSource dan BindingNavigator:

```
BindingSource1.DataSource = dtMataKuliah
BindingNavigator1.BindingSource =
BindingSource1
```

3. Anda buat DataBindings untuk masing-masing control

```
txtKode.DataBindings.Add("Text",
BindingSource1, "Kode_MK")

txtNama.DataBindings.Add("Text",
BindingSource1, "Nama")

cboJurusan.SelectedValue =
dtMataKuliah.Rows(BindingSource1.Position
)("Kode_Jurusan")

numSKS.DataBindings.Add("Text",
BindingSource1, "SKS")

numSemester.DataBindings.Add("Text",
BindingSource1, "Semester")
```

4. Kode program untuk no. 11, 12, 13 dikerjakan pada event dengan Handles MyBase.Load.

5. cboJurusan diisi sesuai dengan data yang ada di tabel Jurusan. Gunakan DataTable untuk mengambil semua data dari tabel Jurusan. Lalu untuk memasukkan datanya, gunakan kode berikut:

```
cboJurusan.DataSource = dtJurusan
cboJurusan.DisplayMember = "Nama_Jurusan"
cboJurusan.ValueMember = "Kode_Jurusan"
```

6. Melalui event dengan Handles BindingSource1.PositionChanged, ketik kode berikut:

```
cboJurusan.SelectedValue =  
dtMataKuliah.Rows (BindingSource1.Position  
) ("Kode_Jurusan")
```

7. Melalui event dengan Handles btnSimpan.Click, Anda melakukan update data kedalam tabel Mata_Kuliah (dalam Database).
8. Melalui event dengan Handles btnBatal.Click, Anda membersihkan semua inputan yang ada didalam TextBox, ComboBox dan NumericUpDown.
9. Melalui event dengan Handles btnKeluar.Click aplikasi keluar.

MODUL 6

DATASET & XML

If there are no stupid questions, then what kind of questions do stupid people ask? Do they get smart just in time to ask questions?

- Scott Adams -



Tujuan:

Mahasiswa paham dan mengerti XML

Materi:

XML Schema

XML File

Referensi:

Amelia, Tan, 2008, Pemrograman Visual II Panduan Sertifikasi MCTS (Exam 70-526), Tanpa Kota, Tanpa Penerbit

Huddleston, James, 2007, Beginning VB 2005 Databases From Novice to Professional, New York, Springer-Verlag New York Inc.

Willis, Thearon, 2006, Beginning Visual Basic 2005 Databases, Indianapolis, Wiley Publishing, Inc.

XML (eXtensible Markup Language) merupakan bahasa markup untuk mendeskripsikan berbagai macam data/informasi dengan lebih akurat dan fleksibel. Berbeda dengan HTML, XML gak punya format baku karena merupakan metalanguage sehingga kita bisa melakukan kustomisasi desain markup language yang akan kita gunakan. Satu hal yang menonjol pada XML adalah tersedianya perintah-perintah dasar yang dapat digunakan untuk berbagi informasi dengan mudah antara komputer, aplikasi dan sistem operasi yang berbeda.

Beberapa keunggulan XML:

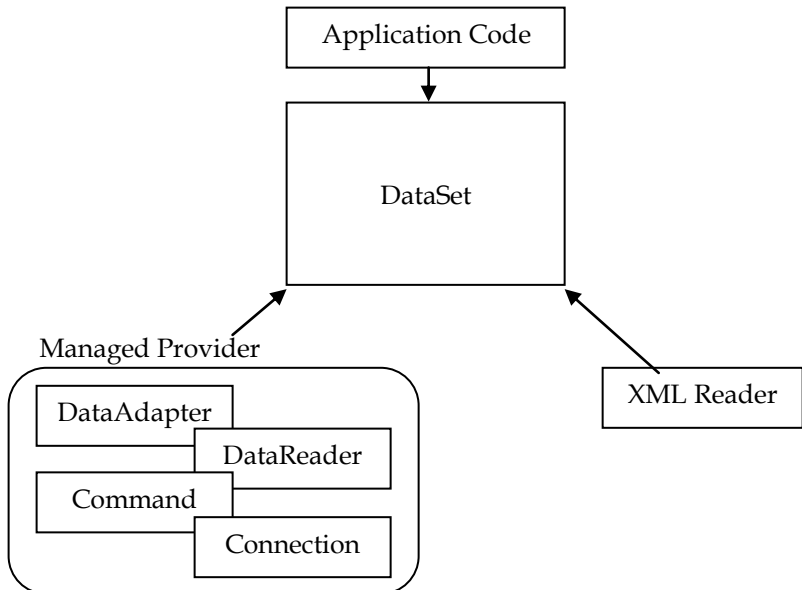
Simplicity → Mudah untuk dibaca oleh programmer dan komputer.

Extensibility → Kemampuan untuk menambah styles, linking dan referencing.

Interoperability → Dapat digunakan diberbagai macam platform dan tool.

6.1. ADO.NET dan Arsitektur XML

ADO.NET tidak hanya mendukung data relasional seperti dalam database, melainkan dapat juga terintegrasi dengan data hirarki seperti XML.



Object DataSet dapat diisi data-nya melalui object-object pada Managed Provider jika mengambil data-datanya langsung dari database atau dapat juga dari koding. Selain melalui dua hal tersebut, untuk mengisi DataSet dapat menggunakan object XMLReader jika data-nya dalam bentuk XML.

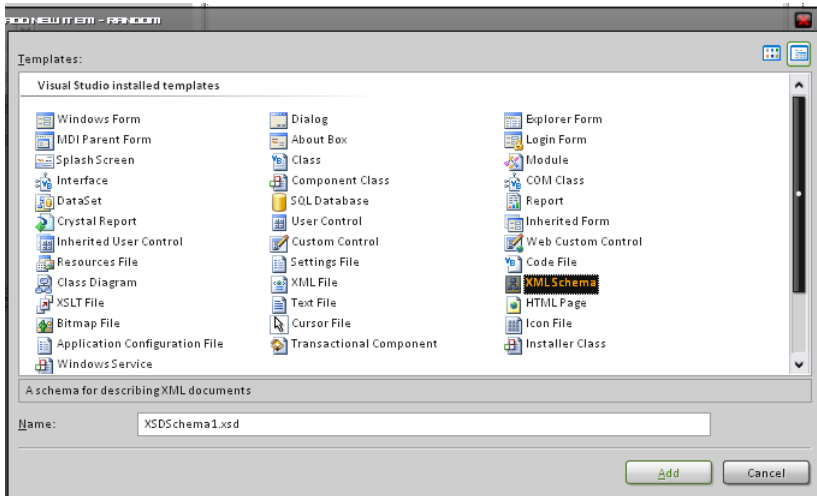
Setelah data berada dalam DataSet, maka tidak ada perlakuan yang berbeda lagi terhadap data tersebut, tidak peduli darimana data itu berasal, apakah dari database atau dari dokumen XML. Sehingga lewat DataSet, kita bisa ambil data dari tabel lalu disimpan dalam XML atau sebaliknya.

6.2. XML Schema

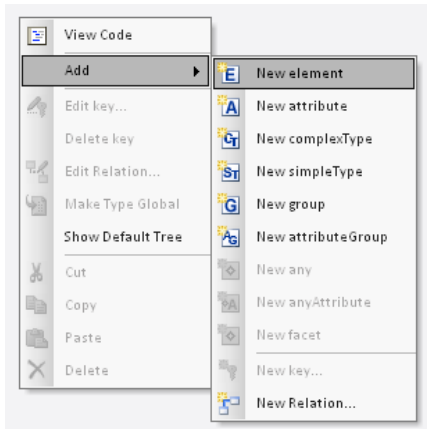
Sebuah format XML yang digunakan untuk menyimpan struktur XML layaknya sebuah database. Sehingga pada XML Schema ini, berisi informasi tentang Tabel (nama tabel beserta field dan tipe datanya), Constraint dan Relationship.

Untuk membuatnya, ikuti langkah-langkah berikut:

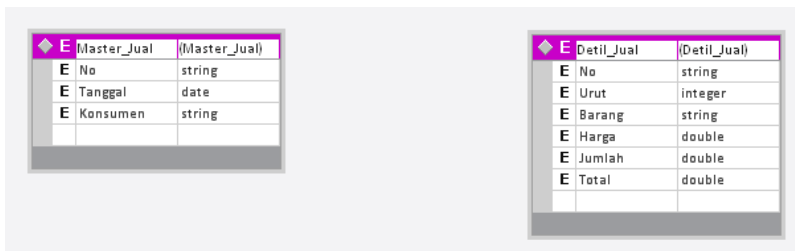
1. Add New Item



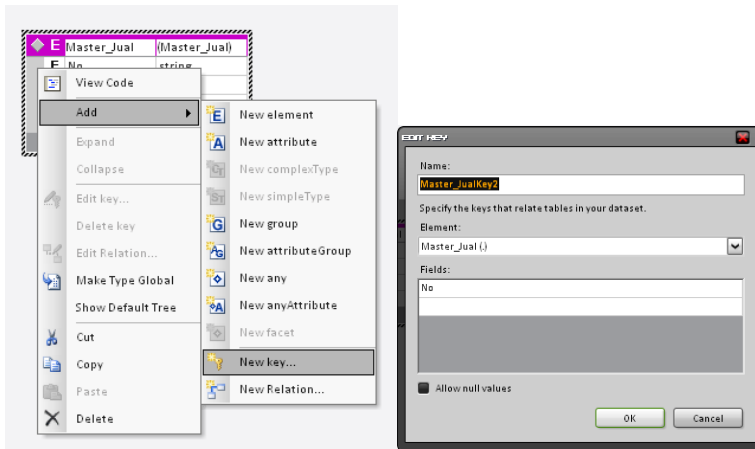
2. Add New Element



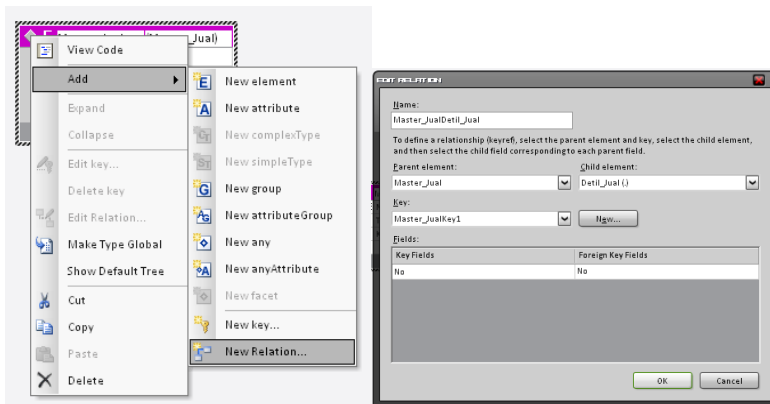
3. Tambahkan nama dan kolom serta tipe data kolomnya



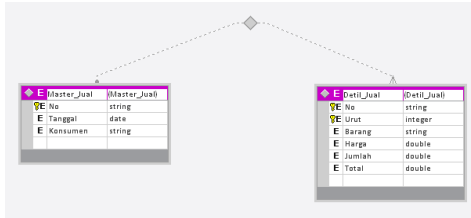
4. Add New Key untuk membuat Primary Key



5. Add New Relation untuk membuat Foreign Key



6. Hasil akhir

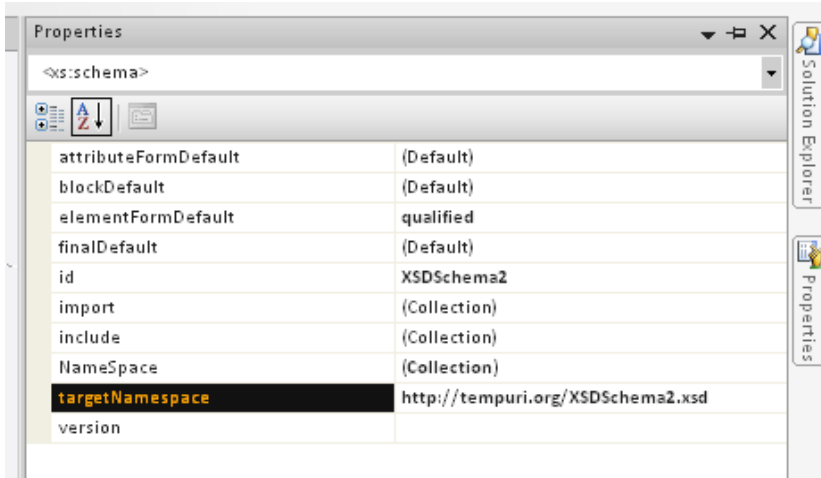


Generated Code:

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <xsi:schema id="XSISchema2.xsd" namespace="http://tempuri.org/XSISchema2.xsd" elementFormDefault="qualified" />
3 <xsi:include schemaLocation="XSISchema2.xsd" />
4 <xsi:complexType base="xsd:string" />
5 <xsi:sequence />
6 <xsi:element name="No" type="xsd:string" />
7 <xsi:element name="Tanggal" type="xsd:date" />
8 <xsi:element name="Konsumen" type="xsd:string" />
9 </xsi:sequence>
10 </xsi:complexType>
11 <xsi:key name="Master_JualKey1" />
12 <xsi:selector xpath="." />
13 <xsi:field xpath="masterNo" />
14 </xsi:key>
15 </xsi:element>
16 <xsi:element name="Detil_Jual" />
17 <xsi:complexType />
18 <xsi:sequence />
19 <xsi:element name="No" type="xsd:string" />
20 <xsi:element name="Urut" type="xsd:integer" />
21 <xsi:element name="Barang" type="xsd:string" />
22 <xsi:element name="Harga" type="xsd:double" />
23 <xsi:element name="Jumlah" type="xsd:double" />
24 <xsi:element name="Total" type="xsd:double" />
25 </xsi:sequence>
26 </xsi:complexType>
27 <xsi:key name="Detil_JualKey1" />
28 <xsi:selector xpath="." />
29 <xsi:field xpath="masterNo" />
30 <xsi:field xpath="masterUrut" />
31 </xsi:key>
32 <xsi:keyref name="Master_JualDetil_Jual" refer="Master_JualKey1" />
33 <xsi:selector xpath="." />
34 <xsi:field xpath="masterNo" />
35 </xsi:keyref>
    
```

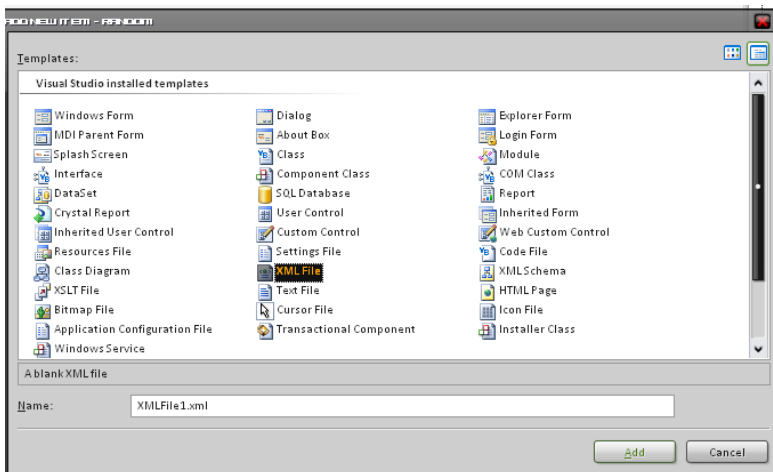
Perhatikan properti targetNamespace pada file XML Schema:



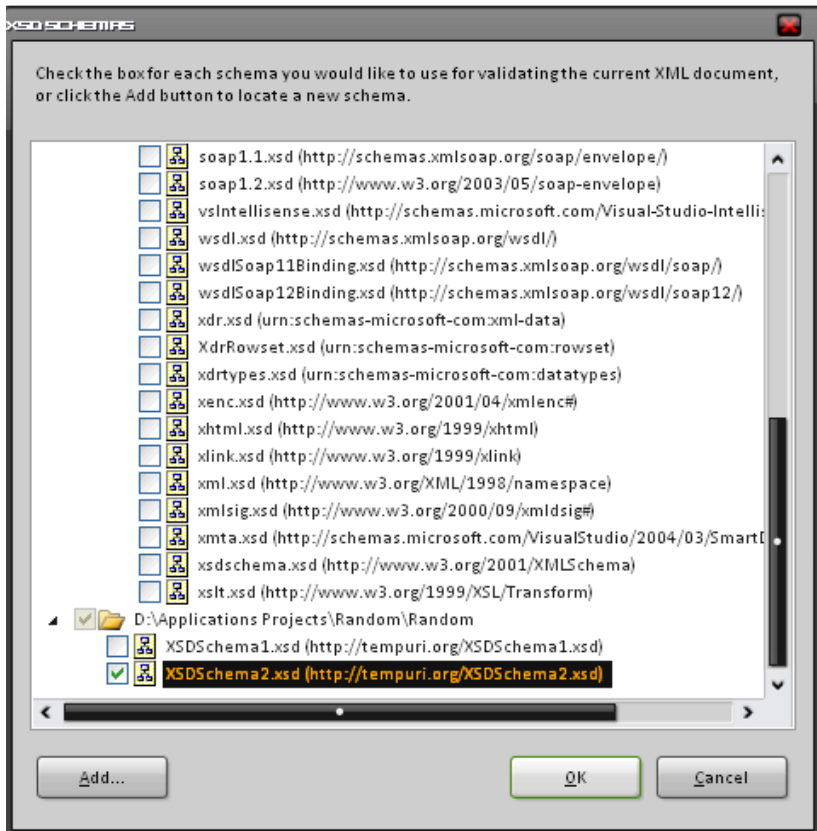
6.3. XML File

Bila XML Schema digunakan untuk membuat struktur database, berbeda halnya dengan XML File. XML File digunakan untuk melakukan DML. Semua data disimpan di file ini. Namun ada kemungkinan juga bila kita melakukan penyimpanan struktur database (dari dataset) disini.

1. Add New Item



2. Ubah properti Schemas pada file XML File.



Pilih Schema dari file XML Schema yang baru saja dibuat.

3. Pada Code Area, tulis kode berikut:

```
<XSDSchema2
xmlns="http://tempuri.org/XSDSchema2.xsd"
></XSDSchema2>
```

XSDSchema2 didapat dari nama file XML Schema yang dicentang di point no.2

4. Coba ke Grid Area. Anda bisa menambahkan beberapa data disini.



6.4. Pengaplikasian XML pada Form

```
Dim ds As New DataSet

Private Sub Form2_Load(ByVal sender As
System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    ds.ReadXmlSchema
    ("..\..\XSDSchema2.xsd")

    ds.ReadXml ("..\..\XMLFile1.xml")

    DataGridView1.DataSource =
    ds.Tables("Detil_Jual")
End Sub
```

Kode “..\..\namafile.ekstensi” merupakan letak file. Maksud dari kode itu adalah bahwa file berada 2 tingkat diatas file executable.



File executable berada di folder Random, sedangkan file executable berada di folder Random\bin\Debug.

Untuk proses datanya, sama seperti memproses data pada DataSet karena semua data dari XML File ditampung oleh DataSet sesuai dengan XML Schemanya.

Untuk melakukan penyimpanan kedalam XMLFile, tulis kode berikut:

```
ds.WriteXml ("..\..\XMLFile1.xml")
```

Latihan

Buat desain interface:

The screenshot shows a Windows application window titled "Form Master Data". The window contains a header with the text "Master Data Mata Kuliah" in blue and "Kode dan Nama MK" in red. Below the header are several input fields: "Kode" (text box), "Nama" (text box), "Jurusan" (dropdown menu), "SKS" (numeric up-down control), and "Semester" (numeric up-down control). The "Kode" and "Nama" fields have "Cari" buttons next to them. Below the input fields is a large grey rectangular area representing a DataGrid. At the bottom of the window is a toolbar with five buttons: "Simpan", "Ubah", "Hapus", "Batal", and "Keluar".

Kotak abu-abu di bawah numSemester adl DataGridView.

Setting Awal:

- Property FormBorderStyle = FixedToolWindow.
- MaxLength dari txtKode = 5
- MaxLength dari txtNama = 50
- numSKS = NumericUpDown dengan min 1 dan max 10
- numSemester = NumericUpDown dengan min 1 dan max 8
- Semua Button, enable = True.

Cara Kerja:

Tidak ada koneksi ke database

Buat sebuah file XML Schema dengan nama XSDAkademik.

MataKuliah

Kolom	TipeData
Kode_Kuliah	String
Nama_MK	String
Kode_Jurusan	String
SKS	Integer
Semester	Integer

Buat Primary Key di kolom Kode_Kuliah

Jurusan

Kolom	TipeData
Kode_Jurusan	String
Nama_Jurusan	String

Buat Primary Key di kolom Kode_Jurusan

Buat Relation di kolom Kode_Jurusan. Jurusan sebagai Parent Element, Mata_Kuliah sebagai Child Element.

1. Buat sebuah file XML File dengan nama XMLAkademik. Hubungkan XML File dengan XML Schema. Lalu isikan data-data sesuai dengan data-data yang ada pada tabel Mata_Kuliah dan tabel Jurusan di database SQLServer (minimal 5 data masing-masing tabel).

Menghubungkan XML dengan Form:

2. Buat sebuah variabel bertipe DataSet
3. Melalui event dengan Handles MyBase.Load, tuliskan kode program untuk membaca XML Schema dan XML File yang telah Anda buat. Lalu set DataSource dari DataGridView ke tabel Mata_Kuliah yang ada pada DataSet.
4. Ketika salah satu cell pada DataGridView disorot, tampilkan datanya pada control-control yang bersesuaian.
5. cboJurusan diisi sesuai dengan data yang ada di tabel Jurusan (dalam DataSet). Gunakan cboJurusan.DataSource.

6. Melalui event dengan Handles `btnCariKode.Click`, Anda melakukan pencarian data berdasarkan kode matakuliah. Jika data yang dicari ada, tampilkan datanya pada control-control yang bersesuaian.
7. Melalui event dengan Handles `btnCariNama.Click`, Anda melakukan pencarian data berdasarkan substring nama matakuliah. Pencarian ini langsung mempengaruhi jumlah data yang tampil pada `DataGridView`.
8. Melalui event dengan Handles `btnSimpan.Click`, Anda melakukan insert into kedalam tabel `Mata_Kuliah` (dalam `DataSet`).
9. Melalui event dengan Handles `btnUbah.Click`, Anda melakukan update kedalam tabel `Mata_Kuliah` (dalam `DataSet`).
10. Melalui event dengan Handles `btnHapus.Click`, Anda melakukan delete kedalam tabel `Mata_Kuliah` (dalam `DataSet`).
11. Melalui event dengan Handles `btnBatal.Click`, kembalikan keadaan form seperti ketika pertama kali form di-load.
12. Melalui event dengan Handles `btnKeluar.Click` aplikasi keluar. Sebelum aplikasi keluar, ada sebuah pesan "Apakah Data ingin disimpan kedalam XML?". Jika Ya, lakukan proses `WriteXML`.

MODUL 7

CRYSTAL REPORT

If you have a talent, use it in every which way possible. Don't hoard it. Don't dole it out like a miser. Spend it lavishly like a millionaire intent on going broke.
- Brendan Francis -



Tujuan:

Mahasiswa paham dan mengerti tentang bagaimana membuat laporan melalui Crystal Report

Materi:

Crystal Report

Referensi:

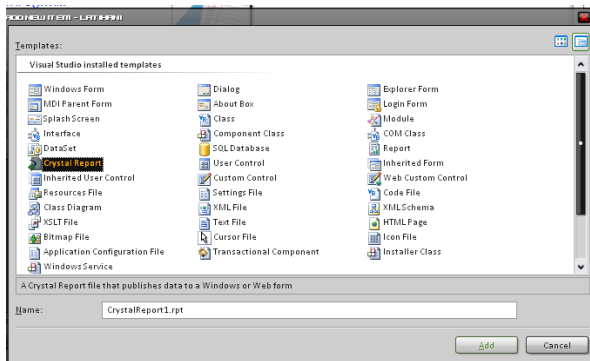
Amelia, Tan, 2008, Pemrograman Visual II Panduan Sertifikasi MCTS (Exam 70-526), Tanpa Kota, Tanpa Penerbit
Huddleston, James, 2007, Beginning VB 2005 Databases From Novice to Profesional, New York, Springer-Verlag New York Inc.
Willis, Thearon, 2006, Beginning Visual Basic 2005 Databases, Indianapolis, Wiley Publishing, Inc.

Crystal Report untuk Visual Studio .NET merupakan tool standar untuk membuat laporan. Memberikan kemudahan dalam membuat laporan yang interaktif dan presentatif bagi semua platform .NET, baik melalui WEB maupun aplikasi desktop.

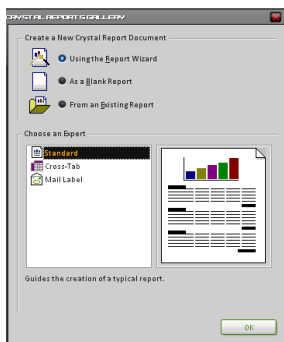
7.1. Membuat Crystal Report

Untuk membuat laporan dengan menggunakan Crystal Report, ikuti langkah-langkah berikut:

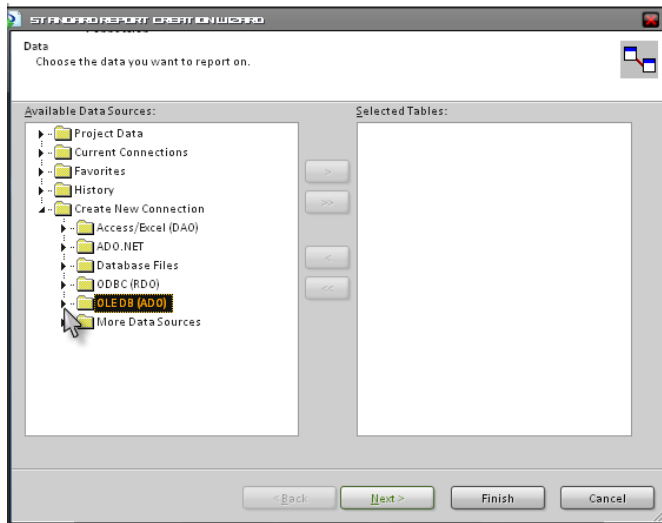
1. Klik kanan pada project Anda, pilih Add dan klik New Item.
2. Pada window Add New Item, pilih templates Crystal Report dan isikan nama laporan yang Anda inginkan.



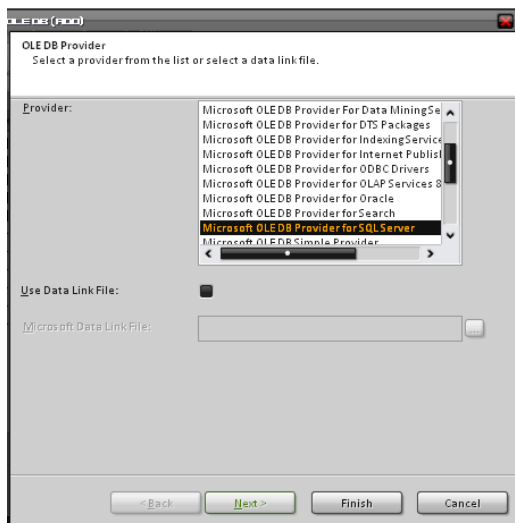
3. Pada window Crystal Report Gallery Anda dapat memilih Report Wizard atau Blank Report untuk mulai membuat laporan. Sebagai awal, pilih Using the Report Wizard.



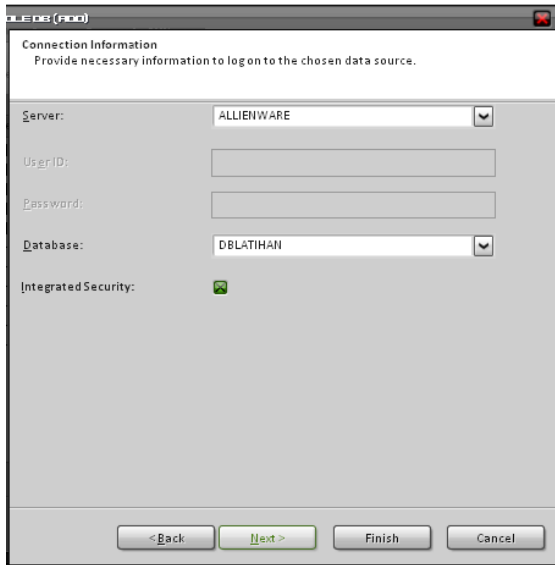
- Setelah itu Anda akan dihadapkan pada pemilihan data source dari Crystal Report. Untuk dapat melakukan koneksi ke SQL Server, pilih Create New Connection - OLE DB (ADO).



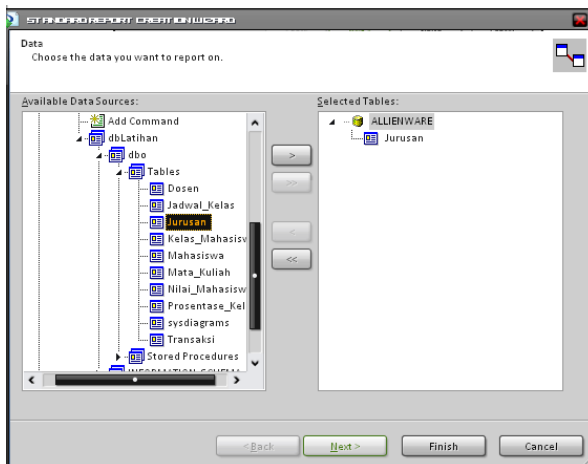
- Pada window OLE DB Provider, pilih Microsoft OLE DB Provider for SQL Server lalu klik tombol NEXT.



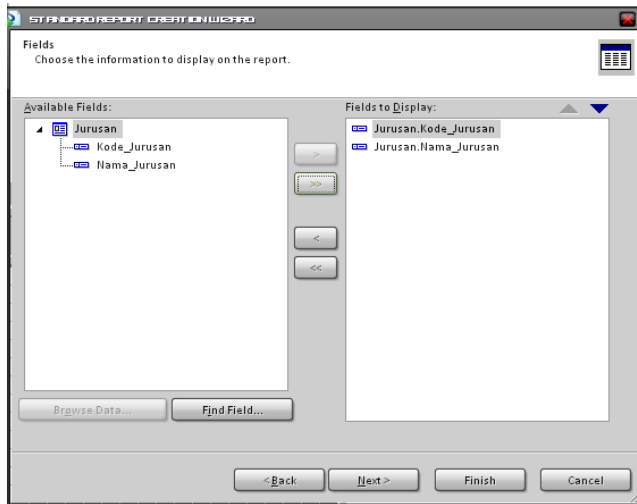
6. Pada window Connection Information, Anda isikan berbagai properti koneksi ke SQL Server Anda lalu klik tombol FINISH.



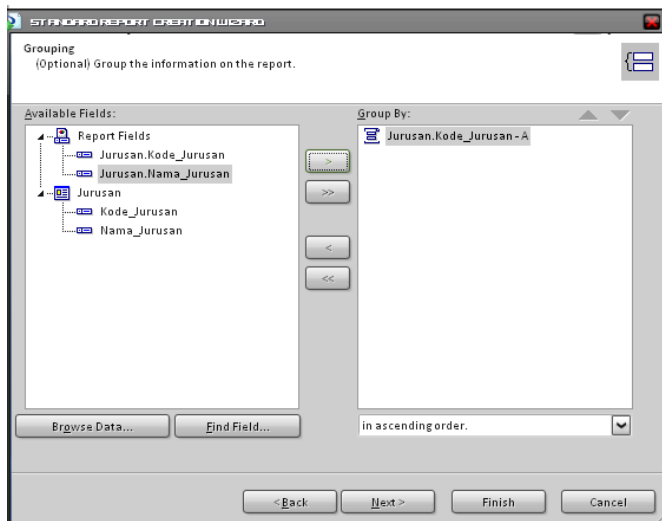
7. Kembali ke window Standard Report Creation Wizard, Anda masih dihadapkan pada pemilihan data tetapi langsung memilih tabel-tabel yang akan Anda gunakan dalam laporan. Jika sudah, klik tombol NEXT.



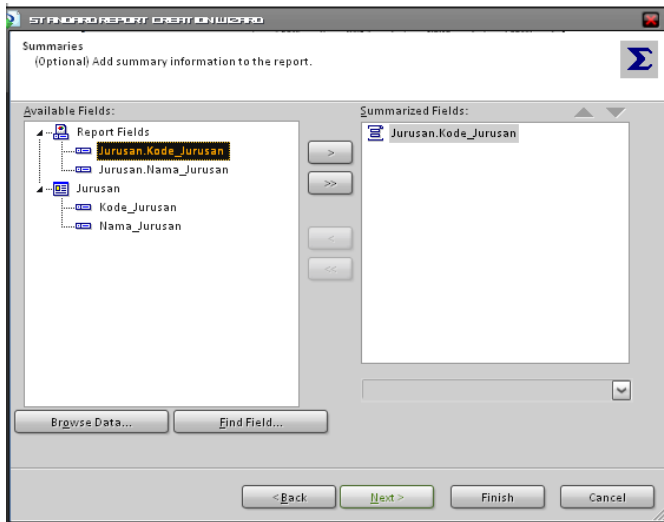
- Anda tinggal memilih field-field apa aja yang akan Anda tampilkan di laporan Anda. Field-field ini diambil langsung dari tabel-tabel yang Anda pilih pada window sebelumnya.



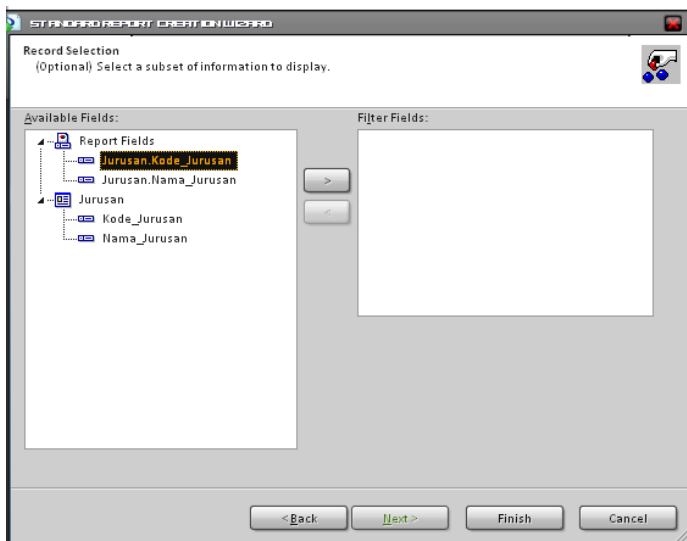
- Setelah itu, jika Anda membutuhkan pengelompokan data, Anda tinggal pilih ingin mengelompokkan berdasarkan kolom apa pada window yang satu ini.



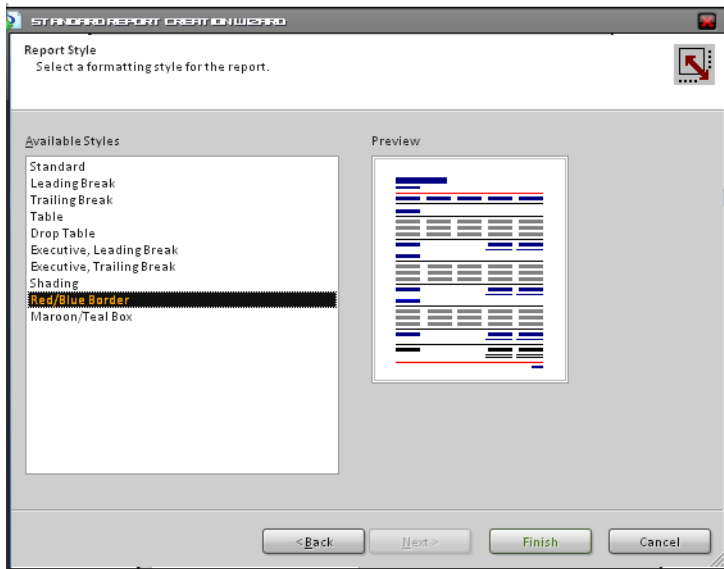
- 10. Pengelompokan Data pasti membutuhkan summary. Anda tinggal memilih summary-nya pada window ini.



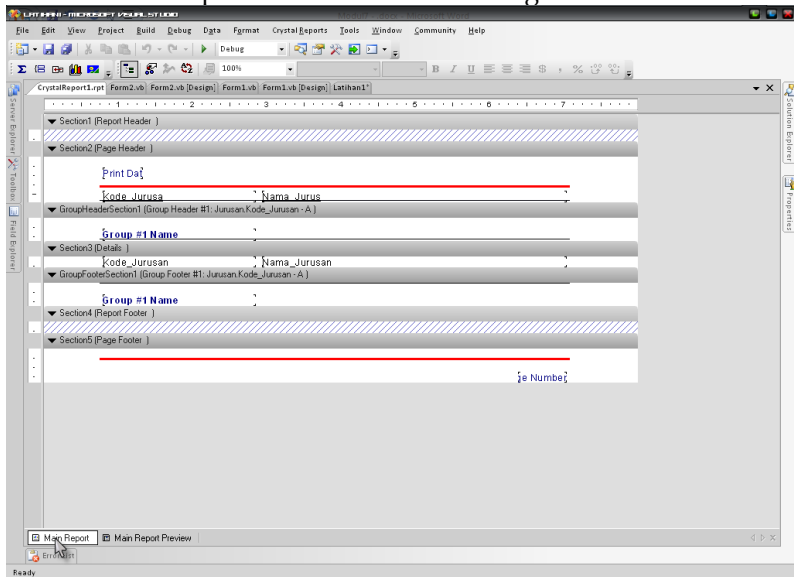
- 11. Window yang satu ini digunakan untuk melakukan pembatasan data yang akan ditampilkan. Ibaratnya seperti klausa WHERE pada syntax SQL.



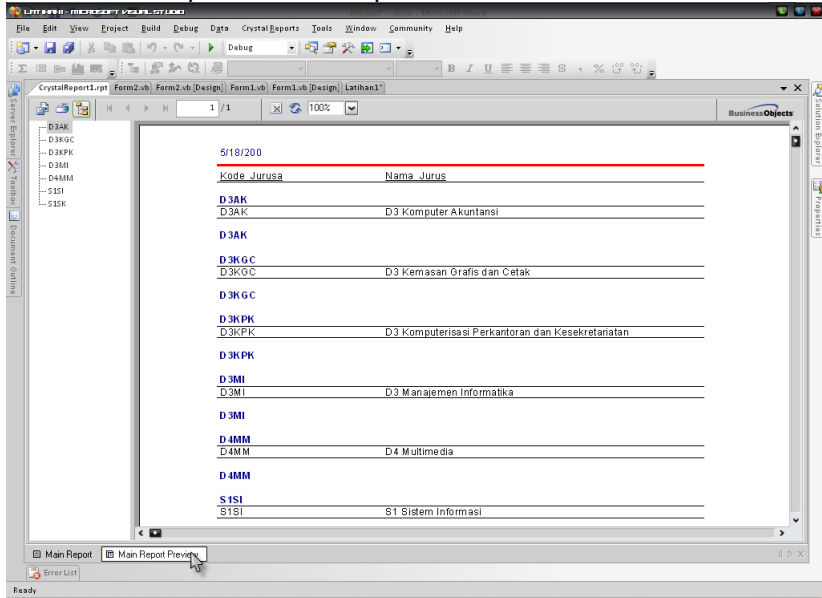
- Yang terakhir adalah memilih tampilan laporan dari template-template yang sudah disediakan oleh Crystal Report.



Hasil dari ke-10 proses tersebut adalah sebagai berikut:



Preview dari laporan adalah seperti ini:



7.2. Formula Fields

Formula adalah hasil sebuah rumusan yang dapat ditentukan sendiri sesuai dengan kebutuhan. Kita bisa menggunakan formula ini untuk membuat perhitungan. Berikut ini langkah-langkah dalam membuat formula:

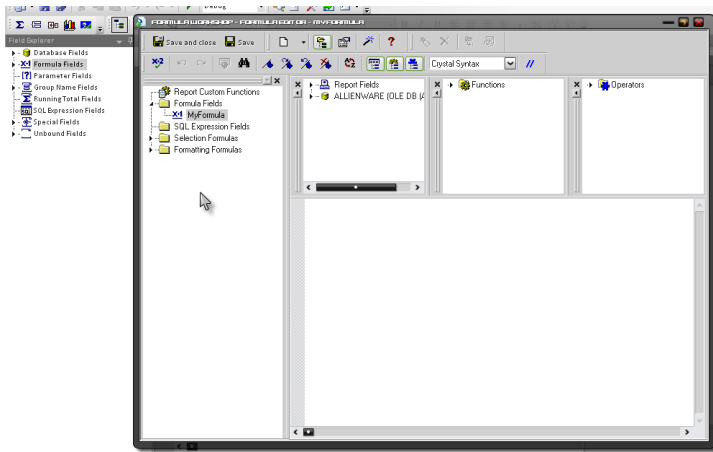
Klik kanan pada Formula Fields (pada Toolbox), pilih **New**.

Isi nama formula yang akan dibuat, klik **OK**.

Pada **Formula Editor**, ketikkan formula yang Anda inginkan.

Setelah formula diisikan, pilih toolbar **Save and Close** atau tekan tombol **CTRL+S**.

Preview dari window Formula Editor bisa dilihat di halaman selanjutnya.



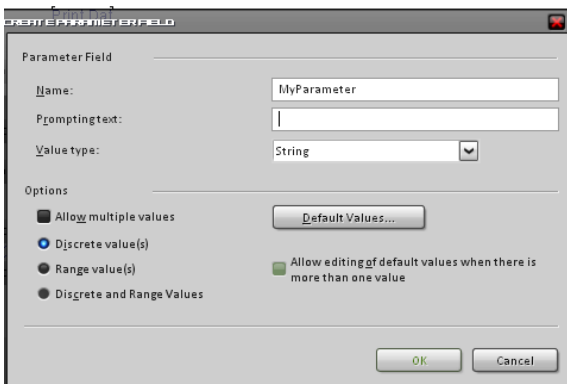
7.3. Parameter Fields

Parameter digunakan untuk menampung nilai dari User untuk digunakan pada laporan. Penggunaan parameter paling sederhana misalnya untuk menampilkan teks sebagai judul laporan sampai pada penggunaan yang lebih kompleks lagi seperti untuk melakukan pemilihan data tertentu (filtering). Berikut ini cara untuk membuat parameter:

Klik kanan pada Parameter Fields, pilih **New**.

Pada tampilan **Create Parameter Field**, isi parameternya dan tipe data dari parameter tersebut.

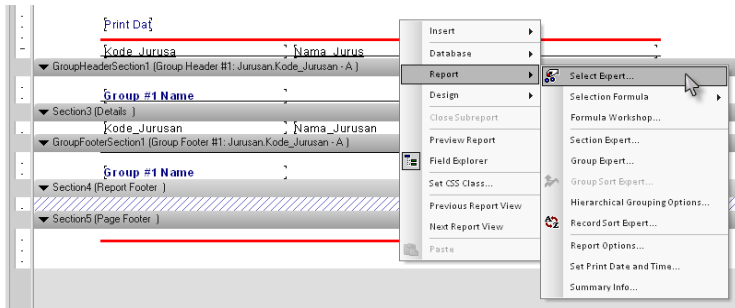
Klik tombol **OK**.



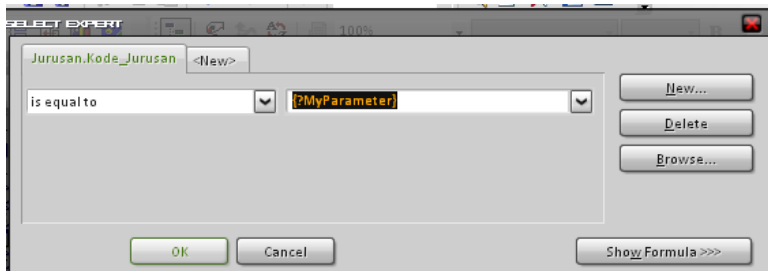
Setelah parameter berhasil dibuat, langkah selanjutnya adalah Anda melakukan **drag and drop** parameter tersebut kedalam laporan. Jika laporan tersebut ditampilkan, maka akan muncul window **Enter Parameter Values**, yang digunakan untuk mengisi nilai dari parameter tersebut.

Untuk penggunaan parameter dalam filtering, ikuti langkah-langkah berikut:

1. Pada design area Crystal Report, klik kanan pada area kosong lalu pilih Reports – Select Experts.



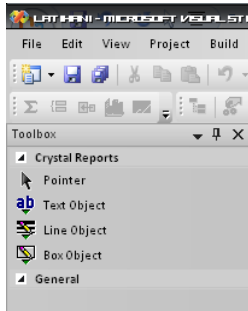
2. Pada window Choose Field, pilih nama field yang akan di filter, lalu klik OK.
3. Muncul tampilan Select Expert, Anda dapat memilih operator yang akan digunakan, kemudian pilih parameternya.



4. Klik OK.

7.4. Control pada Crystal Report

Ada 3 control pada Crystal Report yang dapat digunakan untuk mengatur format tampilan laporan. Semua control ini ada di toolbox tab Crystal Report.



Berikut penjelasan dari masing-masing control:

1. Text Object

Control yang digunakan untuk menampilkan tulisan pada laporan. Biasanya digunakan untuk menampilkan judul laporan dan informasi lain yang bersifat permanen.

2. Line Object

Control yang digunakan untuk membuat garis vertikal atau horisontal, tetapi tidak bisa digunakan membuat garis diagonal.

3. Box Object

Control yang digunakan untuk membuat tampilan kotak, persegi panjang atau ellips (jika sudutnya rounded).

7.5. Menampilkan Crystal Report pada Aplikasi Window

Untuk dapat menampilkan laporan yang sudah kita buat kedalam form, ikuti langkah-langkah berikut:

1. Drag and drop control CrystalReportViewer pada form.
2. Atur prop. CrystalReportViewer tersebut seperti dibawah ini:

Properti	Nilai	Keterangan
DisplyGroupTree	False	Tidak menampilkan daftar group yang ada pada laporan.
Dock	Fill	Menampilkan laporan pada keseluruhan form.
EnableDrillDown	False	Tidak menampilkan Drill Down yang ada pada CrystalReportViewer.

3. Ketik kode berikut ini:

```
Imports _
    CrystalDecisions.CrystalReports.Engine
Imports CrystalDecisions.Shared
Imports System.Data.SqlClient

Dim objRepDoc As New ReportDocument
Dim objConInfo As New ConnectionInfo
Dim objTableLogOnInfo As New
TableLogOnInfo

objRepDoc = New CrystalReport1
objTableLogOnInfo = _
    objRepDoc.Database.Tables(0).LogOnInfo
objConInfo.ServerName = "(local)"
objConInfo.UserID = "myUser"
objConInfo.Password = "myPassword"
objConInfo.DatabaseName = "dbLatihan"
objTableLogOnInfo.ConnectionInfo =
objConInfo
objRepDoc.Database.Tables(0). _
    ApplyLogOnInfo(objTableLogOnInfo)

CrystalReportViewer1.ReportSource = _
    objRepDoc
```

4. Untuk menambahkan formula sebagai filtering, tulis kode berikut:

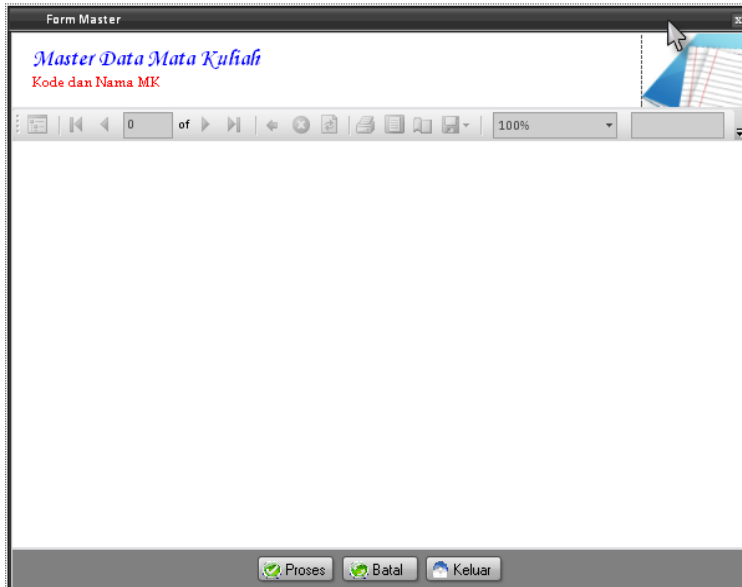
```
objRepDoc.RecordSelectionFormula = _
    "{Jurusan.Kode_Jurusan} = 'S1SI'"
```

5. Sedangkan untuk mengisi parameter, tulis kode berikut:

```
objRepDoc.SetParameterValue("myParameter"  
, "SISI")
```

Latihan

Buat desain interface:



Kotak putih di tengah form adalah CrystalReportViewer.

Setting Awal:

CrystalReportViewer1, Dock: Fill.

Semua Button, enable = True.

Cara Kerja:

1. Buat sebuah Crystal Report untuk menampilkan seluruh data Mata Kuliah. Data diambil dari tabel Mata_Kuliah pada SQL Server 2005.
2. Buat KOP Laporan dengan menampilkan logo, nama dan alamat lengkap Perusahaan. KOP ini tampil di tiap-tiap halaman.
3. Dibawah KOP Laporan, tampilkan tanggal sistem.
4. Tampilkan Judul Laporan hanya di halaman pertama saja.
5. Layout Detail terserah Anda.
6. Untuk Kode Jurusan, tampilkan Nama Jurusannya.

7. Data yang ditampilkan dikelompokkan berdasarkan Kode Jurusannya.
8. Tampilkan jumlah Mata Kuliah per jurusannya.
9. Tampilkan juga Jumlah seluruh Mata Kuliah.
10. Dibagian bawah laporan, beri tulisan:
Printed by [Nama User] on [Jam Sistem]

Ket:

Nama User diambil dari parameter fields yang diisi dari form yang memanggil Crystal Report ini. Tulisan ini hanya tampil satu kali dalam satu laporan (bukan per halaman!!!).

Menghubungkan Crystal Report dengan Form:

11. Ketika Form pertama kali dijalankan, tampilkan sebuah INPUTBOX untuk menginputkan nama user.
12. Laporan tampil ketika btnProses diklik. Ingat pada event ini, Anda harus mengisi parameter field yang ada di Crystal Report.
13. Melalui event dengan Handles btnBatal.Click, kembalikan keadaan form seperti ketika pertama kali form di-load.
14. Melalui event dengan Handles btnKeluar.Click aplikasi keluar.