

Modul Training  
**Membangun Aplikasi Database Oracle**  
**Menggunakan Visual Basic**

Oleh: Achmad Basuki  
Politeknik Elektronika Negeri Surabaya

**Materi:**

- (1) Pendahuluan
- (2) Koneksi Database
- (3) Manipulasi Data
- (4) Membuat Form Entry
- (5) Membuat Data Report

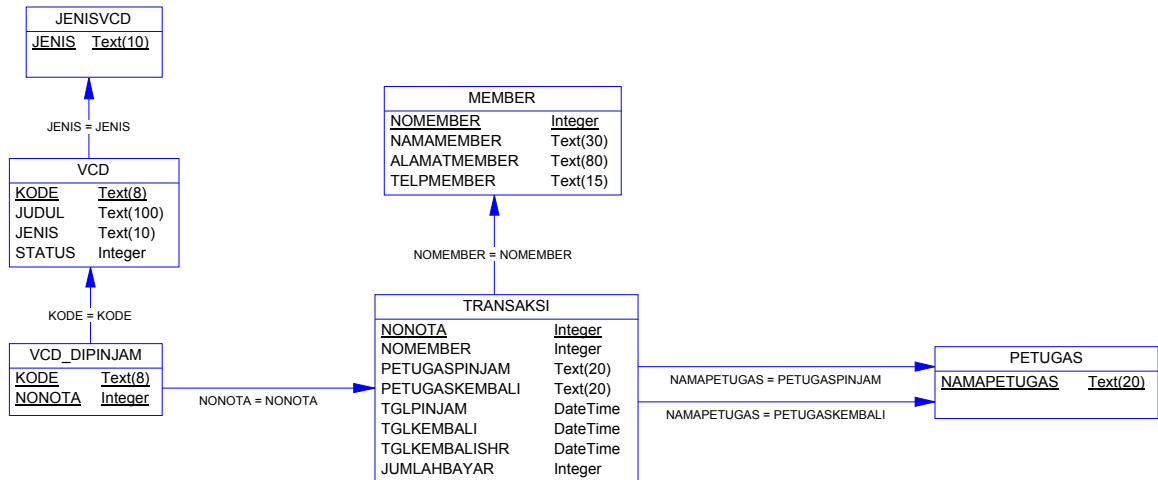
## 1. PENDAHULUAN

Modul ini menjelaskan bagaimana membuat suatu aplikasi Visual Basic dengan menggunakan database Oracle dengan teknik pemrograman yang mudah dimengerti. Tujuan dari modul ini agar peserta latihan dapat membuat suatu aplikasi database secara lengkap dengan contoh kasus tertentu. Sebagai contoh kasus yang diangkat adalah membuat aplikasi database untuk sistem informasi peminjaman VCD, yang merupakan aplikasi yang cukup mudah untuk dibuat.

Untuk kasus sistem informasi peminjaman VCD ini, *business rule* yang digunakan adalah sebagai berikut:

- Aplikasi hanya bisa dijalankan oleh satu komputer sehingga tidak membutuhkan user authentication.
- Untuk VCD baru, user harus menambahkan data VCD dan spesifikasinya. Satu judul VCD mempunyai satu nomor unik.
- Untuk pendaftaran anggota, user harus memasukkan data identitas anggota dalam form entry pendaftaran. Setiap anggota mempunyai satu nomor unik. Nomor ini yang menjadi kunci primer.
- Untuk peminjaman, user harus memasukkan data transaksi peminjaman berdasarkan nomor anggota dan nomor VCD yang dipinjam. Satu anggota bisa meminjam beberapa buah VCD dalam satu transaksi.
- Untuk pengembalian, data transaksi peminjaman dapat diaktifkan kembali dengan mencari transaksi berdasarkan nomor anggota atau nomor VCD.

Entity Relational Diagram yang digunakan untuk sistem informasi peminjaman VCD ini adalah sebagai berikut:



Gambar 1. ERD untuk Sistem Informasi Peminjaman VCD

Untuk membuat tabel di dalam database Oracle dengan menggunakan ERD di atas dapat diketikkan perintah DDL (*Data Definition Language*) berikut ini:

```

create tabel MEMBER
(
    NOMEMBER          NUMBER(8)          not null,
    NAMAMEMBER        VARCHAR2(30)       null    ,
    ALAMATMEMBER      VARCHAR2(80)       null    ,
    TELPMEMBER        VARCHAR2(15)       null    ,
    constraint PK_MEMBER primary key (NOMEMBER)
)
/

create tabel PETUGAS
(
    NAMAPETUGAS       VARCHAR2(20)       not null,
    constraint PK_PETUGAS primary key (NAMAPETUGAS)
)
/

create tabel JENISVCD
(
    JENIS             VA20                not null,
    constraint PK_JENISVCD primary key (JENIS)
)
/

create tabel TRANSAKSI
(
    NONOTA            NUMBER(10)         not null,
    NOMEMBER          NUMBER(8)          null    ,
    PETUGASPINJAM    VARCHAR2(20)       null    ,
    PETUGASKEMBALI   VARCHAR2(20)       null    ,
    TGLPINJAM        DATE                null    ,
    TGLKEMBALI       DATE                null    ,
    TGLKEMBALISHR    DATE                null    ,
    JUMLAHBAYAR      NUMBER(10)         null    ,
    constraint PK_TRANSAKSI primary key (NONOTA)
)
/

```

```

create index MEMBER_PINJAM_FK on TRANSAKSI (NOMEMBER asc)
/
create index PETUGAS_PINJAM_FK on TRANSAKSI (PETUGASPINJAM asc)
/
create index PETUGAS_KEMBALI_FK on TRANSAKSI (PETUGASKEMBALI asc)
/

create tabel VCD
(
    KODE          VARCHAR2(8)          not null,
    JUDUL         VARCHAR2(100)        null    ,
    JENIS         VA20                  null    ,
    STATUS        NUMBER(1)            null    ,
    constraint PK_VCD primary key (KODE)
)
/

create tabel VCD_DIPINJAM
(
    KODE          VARCHAR2(8)          not null,
    NONOTA        NUMBER(10)          not null,
    constraint PK_VCD_DIPINJAM primary key (KODE, NONOTA)
)
/

create index VCD_DIPINJAM_FK2 on VCD_DIPINJAM (KODE asc)
/
create index VCD_DIPINJAM_FK on VCD_DIPINJAM (NONOTA asc)
/

alter tabel TRANSAKSI
    add constraint FK_TRANSAKS_MEMBER_PI_MEMBER foreign key (NOMEMBER)
        references MEMBER (NOMEMBER)
/
alter tabel TRANSAKSI
    add constraint FK_TRANSAKS_PETUGAS_P_PETUGAS foreign key (PETUGASPINJAM)
        references PETUGAS (NAMAPETUGAS)
/
alter tabel TRANSAKSI
    add constraint FK_TRANSAKS_PETUGAS_K_PETUGAS foreign key (PETUGASKEMBALI)
        references PETUGAS (NAMAPETUGAS)
/
alter tabel VCD
    add constraint FK_VCD_REF_64_JENISVCD foreign key (JENIS)
        references JENISVCD (JENIS)
/
alter tabel VCD_DIPINJAM
    add constraint FK_VCD_DIPI_VCD_DIPIN_VCD foreign key (KODE)
        references VCD (KODE)
/
alter tabel VCD_DIPINJAM
    add constraint FK_VCD_DIPI_VCD_DIPIN_TRANSAKS foreign key (NONOTA)
        references TRANSAKSI (NONOTA)
/

```

Ketikkan perintah DDL di atas dalam suatu file teks dengan ekstensi .SQL misalkan rental.sql yang dapat dijalankan pada program aplikasi *SQLPlus* pada Oracle.

## 2. KONEKSI DATABASE

Koneksi database adalah sebuah modul (obyek) yang bekerja untuk menghubungkan aplikasi dengan sebuah DBMS (*Database Management System*). Dengan menggunakan koneksi database ini, maka aplikasi dapat melakukan manipulasi data terhadap database tertentu.

Pada umumnya database merupakan suatu sistem yang berdiri sendiri di luar lingkungan aplikasi tertentu. Demikian pula Oracle merupakan suatu database yang lengkap dan mempunyai lingkungan aplikasi sendiri. Bila ingin memanfaatkan database Oracle dalam lingkungan aplikasi Visual Basic diperlukan suatu koneksi database yang dapat menghubungkan aplikasi Visual Basic dengan database Oracle.

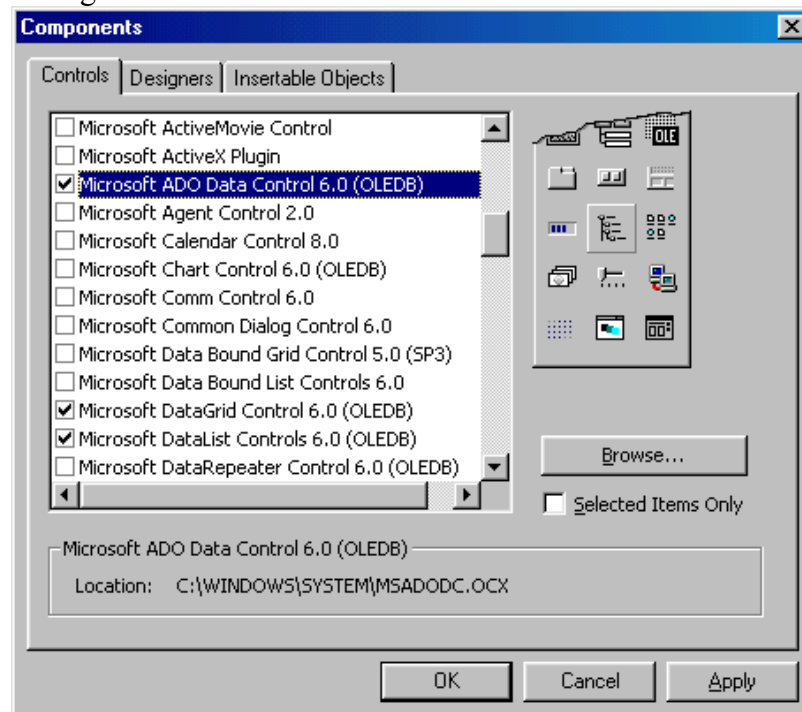
Obyek koneksi database yang dapat digunakan untuk menghubungkan Oracle dengan aplikasi Visual Basic antara lain:

- (1) ADODC (*ActiveX Data Object Data Control*)
- (2) ORADC (*Oracle Data Control*)

### 2.1. Koneksi Dengan ADODC

Untuk menggunakan koneksi ADODC ini terlebih dahulu menyiapkan atau mengaktifkan komponen obyek ini dengan memilih menu Project >> Components pada menu Visual Basic, atau dengan menekan Ctrl-T, lalu pilihlah komponen Microsoft ADO Data Control 6.0 (OLEDB) untuk koneksi database dengan ADO, Microsoft Data Grid untuk tabel grid dengan koneksi ADO, dan Microsoft Data List untuk obyek list dengan koneksi ADO seperti gambar 2 berikut ini.

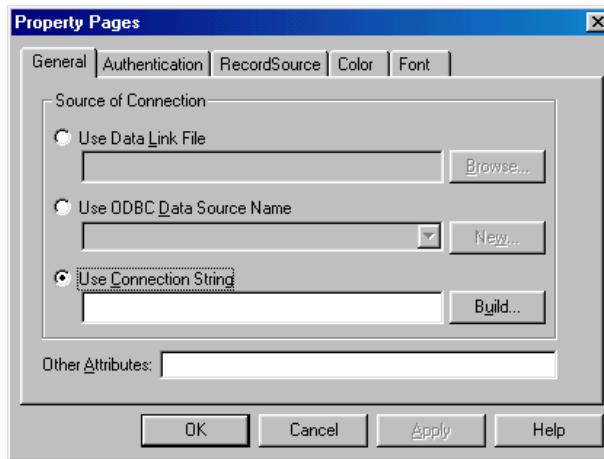
Setelah dipilih maka di bagian toolbox component akan muncul komponen dari obyek ADODC. Selanjutnya untuk menggunakan obyek ini seperti obyek yang lain tinggal menarik icon obyek ini ke dalam form yang dibuat kemudian dilengkapi isi property sesuai dengan kebutuhan.



Gambar 2. Memilih komponen ADODC

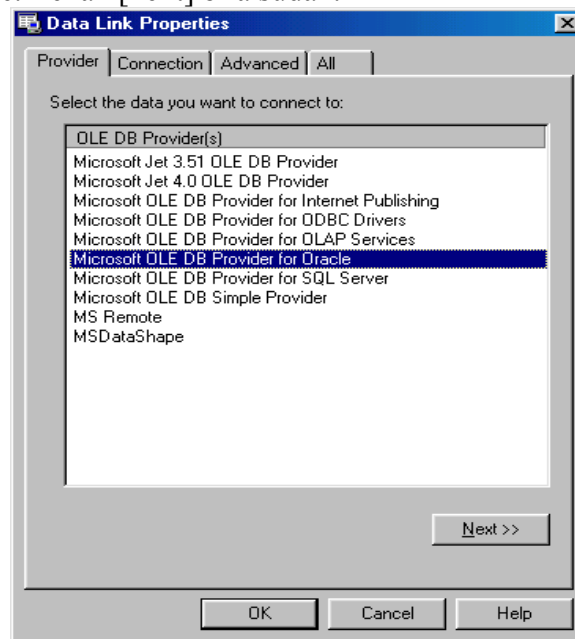
Cara menggunakan obyek ADODC untuk menghubungkan aplikasi dengan database Oracle adalah sebagai berikut:

- Tarik obyek ADODC ke dalam form. Click kanan hingga muncul tampilan seperti gambar 3 berikut ini.



Gambar 3. Properti untuk ADODC

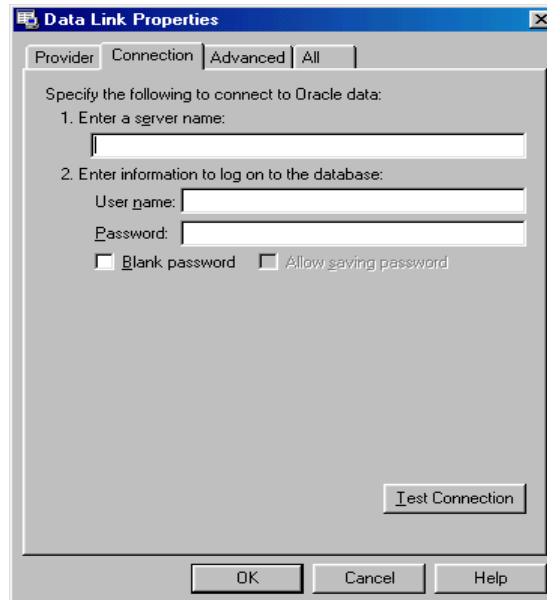
- Pilihlah Use Connection String dan tekan [**Build**], hingga muncul tampilan untuk melakukan koneksi menggunakan ODBC (Open Database Connectivity) untuk memilih provider seperti gambar 4 berikut. Untuk Oracle, pilih Microsoft OLEDB Provider for Oracle. Tekan [next] bila sudah.



Gambar 4. Pemilihan Jenis Provider

- Setelah provider dipilih, langkah berikutnya adalah memilih data link connection seperti gambar 5 berikut ini. Pada pengisian data link connection ini ada beberapa item pengisian antara lain:
  - Server name diisi dengan host name dari server database Oracle yang digunakan,
  - User name diisi dengan nama user yang sudah dibuat pada server,
  - Password diisi dengan password dari user yang bersangkutan.

- Setelah itu pilih [Test Connection] untuk memastikan apakah koneksi berhasil. Tekan [Ok] bila koneksi berhasil.



Gambar 5. Memilih data link

- Langkah berikutnya adalah mendefinisikan user dan tabel yang akan dibuka sebagai berikut:
  - Untuk mengisi user dipilih Authentication, kemudian isikan user name dan password. Ini perlu dilakukan agar saat aplikasi dijalankan tidak lagi ditanyakan identifikasi user untuk koneksi database.
  - Untuk memilih tabel yang diaktifkan dapat dipilih RecordSource, kemudian ketikkan perintah SQL pada Command Text (SQL), misalkan untuk mengaktifkan tabel anggota dapat dituliskan:

**select \* from anggota**

Untuk menguji apakah koneksi berhasil atau tidak, dapat dilakukan dengan menggunakan Data Grid. Tarik data grid ke dalam form, atur posisi dan besar dari jendela penampikan data tabel. Isi property Data Source dengan nama dari obyek ADODC yang sudah dibuat. Jalankan form, bila data muncul di tabel data grid yang dibuat maka koneksi sudah berhasil. Dalam modul ini, koneksi yang digunakan adalah ADODC.

## 2.2. Koneksi Dengan ORADC

Untuk menggunakan koneksi ORADC ini terlebih dahulu menyiapkan atau mengaktifkan komponen obyek ini dengan memilih menu Project >> Components pada menu Visual Basic, atau dengan menekan Ctrl-T, lalu pilihlah komponen Oracle Data Control untuk koneksi database dengan ORADC, Microsoft DataBound Grid untuk tabel grid dengan koneksi ADO, dan Microsoft DataBound List untuk obyek list dengan koneksi ORADC.

Setelah dipilih maka di bagian toolbox component akan muncul komponen dari obyek ORADC. Selanjutnya untuk menggunakan obyek ini seperti obyek yang lain tinggal menarik icon obyek ini ke dalam form yang dibuat kemudian dilengkapi isi property sesuai dengan kebutuhan.

Properti yang harus diisikan dalam obyek koneksi database dengan ORADC ini antara lain:

- Database, diisi dengan nama hostname dari server database Oracle yang digunakan.
- Connection, diisi dengan nama user dan password yang dipisahkan dengan tanda garis miring.
- Record Source, diisi dengan perintah SQL untuk memilih suatu query atau tabel.

### 3. MANIPULASI DATA

Proses memanipulasi (mengelola) data dalam database terdiri dari tiga macam proses yaitu:

- (1) Menambah Data
- (2) Memperbaharui Data
- (3) Menghapus Data

#### 3.1. Menambah Data

Untuk menambah data record dalam aplikasi database pada Visual Basic, dapat digunakan perintah AddNew pada recordset dalam obyek ADODC, yang dituliskan dengan :

**Nama\_ADODC.Recordset.AddNew**

Perintah ini akan menambahkan data kosong pada tabel yang ditentukan. Untuk mengisi isi tabelnya digunakan ekspresi pemindahan nilai sebagai berikut:

**Nama\_ADODC.Recordset!nama\_kolom = nilai**

Untuk semua kolom perlu diisi dengan nilai yang ditentukan dalam form aplikasi Visual Basic. Dan akhiri perintah pengisian nilai ini dengan perintah update, sehingga nilai benar-benar masuk ke dalam tabel, yaitu dengan mengetikkan:

**Nama\_ADODC.Recordset.Update**

Sebagai contoh untuk menambahkan data pada tabel anggota (nama obyek ADODC adalah anggota) dengan nama diambil dari text1, alamat diambil dari text2 dan nomor anggota diambil dari text3 dapat dilakukan dengan perintah:

```
anggota.Recordset.AddNew
anggota.Recordset!nomor=text3
anggota.Recordset!nama=text1
anggota.Recordset!alamat=text2
anggota.Recordset.Update
```

Pada proses pengelolaan data, proses penambahan data ini dilakukan ketika dapat dipastikan bahwa data tersebut belum pernah ada atau belum pernah disimpan sebelumnya. Untuk memastikan hal ini, perlu dilakukan proses pengecekan apakah data yang akan disimpan sudah pernah ada atau belum. Bila belum pernah ada maka proses penambahan data dilakukan.

#### 3.2. Memperbaharui Data

Untuk memperbaharui (meng-update) data record dalam aplikasi database pada Visual Basic, dapat digunakan perintah Update pada recordset dalam obyek ADODC, yang dilakukan dengan :

- o Untuk semua kolom perlu diisi dengan nilai yang ditentukan dalam form aplikasi Visual Basic.

**Nama\_ADODC.Recordset!nama\_kolom = nilai**

- o Dan akhiri perintah pengisian nilai ini dengan perintah update, sehingga nilai benar-benar masuk ke dalam tabel, yaitu dengan mengetikkan:

**Nama\_ADODC.Recordset.Update**

Sebagai contoh untuk memperbaharui data pada tabel anggota (nama obyek ADODC adalah anggota) dengan nama diambil dari text1, alamat diambil dari text2 dan nomor anggota diambil dari text3 dapat dilakukan dengan perintah:

```
anggota.Recordset!nomor=text3  
anggota.Recordset!nama=text1  
anggota.Recordset!alamat=text2  
anggota.Recordset.Update
```

Sebelum melakukan update data sebaiknya perlu dipastikan bahwa posisi record tidak berada di EOF (akhir file). Untuk memastikan hal ini perlu dilakukan proses pencarian data.

#### **4. Menghapus Data**

Untuk menghapus data record dalam aplikasi database pada Visual Basic, dapat digunakan perintah Delete pada recordset dalam obyek ADODC, yang dilakukan dengan:

**Nama\_ADODC.Recordset.Delete**

Seperti halnya update, sebelum menghapus data sebaiknya perlu dipastikan bahwa posisi record tidak berada di EOF (akhir file) atau dengan kata lain data yang dimaksud sudah pernah ada. Untuk memastikan hal ini perlu dilakukan proses pencarian data.

### **3. MEMBUAT FORM ENTRY**

Di dalam aplikasi sistem informasi rental VCD, ada beberapa form entry yang harus dibuat antara lain:

- (1) Form Entry VCD, untuk entry data kaset-kaset VCD yang ada pada rental.
- (2) Form Entry Member, untuk entry data anggota baru atau memperbaharui data anggota
- (3) Form Entry Peminjaman/Pengembalian selanjutnya dinamakan dengan form entry transaksi.

#### **3.1. Membuat Form Entry VCD**

Form ini dibuat untuk menambah data VCD baru, memperbaiki data VCD yang ada dan menghapus data VCD yang sudah tidak digunakan. Nama form untuk entry VCD ini adalah FormEntryVCD. Tampilan Form Entry VCD ini adalah sebagai berikut:



DATA VCD				
KODE	JUDUL	JENIS	NOSERI	STATUS
ACT00001	THE ONE	ACTION	0	1
ACT00002	ROMEO MUST DIE	ACTION	0	1
ACT00003	ZORRO	ACTION	0	1
ACT00004	THE DUEL	ACTION	0	1
HOR00001	CHINA GHOST STORY II	HOROR	0	1
HOR00002	VAMPIRE	HOROR	0	1
KAR00001	PUTRI SALJU II	KARTUN	0	1

Beri nama text atau semua isian sama dengan nama field pada tabel dengan ditambah txt di depannya, misalkan text untuk kode VCD diberi nama **TxtKode**, text untuk judul VCD diberi nama **TxtJudul**, text (DataCombo) untuk jenis diberi nama **TxtJenis** dan text untuk keadaan akhir diberi nama **TxtStatus**.

Sedangkan setiap command (button) diberi nama sama seperti teks yang tertulis dalam button ditambah dengan Cmd, misalkan nama button simpan adalah **CmdSimpan**, nama button hapus adalah **CmdHapus** dan nama button keluar adalah **CmdKeluar**.

Dalam form ini diperlukan dua koneksi database yaitu untuk tabel VCD dan tabel jenis VCD. Koneksi menggunakan ADODC. ADODC untuk koneksi dengan tabel VCD dinamakan **VCD** dan ADODC untuk koneksi dengan tabel jenis VCD dinamakan **jenisVCD**.

Beberapa aturan yang harus dilakukan dalam form ini.

- (1) Kode VCD diisi, bila kode VCD ditemukan dalam data vcd maka yang lainnya ditampilkan, dan bila tidak dianggap kode VCD baru. Keadaan akhir="Ada" bila status=1 dan keadaan akhir="Dipinjam" bila status=0. Perlu diperhatikan bahwa kode VCD tidak boleh kosong.
- (2) Jenis VCD diambil dari tabel jenisVCD, dengan cara menggunakan Data Combo dengan properties sebagai berikut: Nama diganti dengan TxtJenis, Row Source diisi dengan nama koneksi JenisVCD dan list field diisi dengan nama field pada tabel jenis VCD yang digunakan yaitu jenis.
- (3) Keadaan akhir berisi "Ada" dan "Dipinjam". Pengisian dilakukan pada property list.
- (4) Bila dipilih [**Simpan**] maka akan dicari kode VCD pada tabel VCD, bila ada dilakukan update dan bila tidak ada dilakukan insert (append).
- (5) Bila dipilih [**Hapus**] maka akan dicari kode VCD pada tabel VCD, bila ada dilakukan delete dan bila tidak ada diberikan pesan bahwa data dengan kode VCD yang ditulis tidak ada.
- (6) Bila dipilih [**Keluar**] maka form ditutup dan kembali ke menu utama.
- (7) Usahakan tabel Data VCD selalu refresh setiap terjadi perubahan data.

Implementasi dari setiap aturan di atas adalah sebagai berikut:

**(1) Function event untuk pengisian Kode VCD.**

```
Private Sub TxtKode_KeyPress(KeyAscii As Integer)
    If KeyAscii = 13 Then
        If TxtKode <> "" Then
            SQL = "select * from vcd where kode='" & Trim(TxtKode) & "'"
            VCD.RecordSource = SQL
            VCD.Refresh
            If VCD.Recordset.EOF Then
                TxtJudul = ""
                TxtJenis = ""
                TxtStatus = ""
            Else
                TxtJudul = VCD.Recordset!judul
                TxtJenis = VCD.Recordset!jenis
                If VCD.Recordset!Status = 1 Then
                    TxtStatus = 1
                Else
                    TxtStatus = 0
                End If
            End If
        Else
            MsgBox "Kode VCD tidak boleh kosong", vbOKOnly
        End If
    End If
End Sub
```

**(2) Function event untuk tombol Simpan.**

```
Private Sub CmdSimpan_Click()
    If Trim(TxtKode) <> "" Then
        If UCase(Trim(TxtStatus)) = "ADA" Then
            textstatus = 1
        Else
            textstatus = 0
        End if
        SQL = "select * from vcd where kode='" & Trim(TxtKode) & "'"
        VCD.RecordSource = SQL
        VCD.Refresh
        If VCD.Recordset.EOF Then
            VCD.Recordset.AddNew
        End If
        VCD.Recordset!kode = Trim(TxtKode)
        VCD.Recordset!judul = Trim(TxtJudul)
        VCD.Recordset!jenis = Trim(TxtJenis)
        VCD.Recordset!Status = textstatus
        VCD.Recordset.Update
        MsgBox "Data tersimpan", vbOKOnly
        SQL = "select * from vcd"
        VCD.RecordSource = SQL
        VCD.Refresh
    Else
        MsgBox "Kode VCD tidak boleh kosong", vbOKOnly
    End If
End Sub
```

**(3) Function event untuk tombol Hapus.**

```
Private Sub CmdHapus_Click()
```

```

If Trim(TxtKode) <> "" Then
    SQL = "select * from vcd where kode='" & Trim(TxtKode) & "'"
    VCD.RecordSource = SQL
    VCD.Refresh
    If VCD.Recordset.EOF Then
        MsgBox "Data tidak ada", vbCritical
    Else
        VCD.Recordset.Delete
        MsgBox "Data terhapus", vbOKOnly
    End If
    SQL = "select * from vcd"
    VCD.RecordSource = SQL
    VCD.Refresh
Else
    MsgBox "Kode VCD tidak boleh kosong", vbOKOnly
End If
End Sub

```

### (2) Function event untuk tombol Simpan.

```

Private Sub CmdKeluar_Click()
    Unload Me
End Sub

```

## 3.2. Membuat Form Entry Member

Form ini dibuat untuk menambah data anggota baru, memperbaiki data anggota yang ada dan menghapus data anggota yang sudah tidak aktif. Tampilan Form Entry Member ini adalah sebagai berikut:

DATA MEMBER			
NOMEMBER	NAMAMEMBER	ALAMATMEMBER	TELPMEMBER
1	BASUKI	SEMOLWARU UTARA IIIB/40E	6912197

Beri nama text atau semua isian sama dengan nama field pada tabel dengan ditambah txt di depannya, misalkan text untuk No anggota diberi nama **TxtNomember**, text untuk nama anggota diberi nama **TxtNamamember**, text untuk alamat diberi nama **TxtAlamatmember** dan text untuk no telepon diberi nama **TxtTelpmember**.

Sedangkan setiap command (button) diberi nama sama seperti teks yang tertulis dalam button ditambah dengan Cmd, misalkan nama button simpan adalah **CmdSimpan**, nama button hapus adalah **CmdHapus** dan nama button keluar adalah **CmdKeluar**. Dalam form ini diperlukan satu koneksi database yaitu untuk tabel member dengan menggunakan ADODC. ADODC untuk koneksi dengan tabel member ini dinamakan **Member**.

Beberapa aturan yang harus dilakukan dalam form ini.

- (1) No anggota berupa angka, bila nomor anggota yang dituliskan sudah ada di tabel member maka data member ditampilkan, dan bila tidak ada maka dianggap anggota baru dan nomor anggota secara otomatis akan melakukan counter dari nomor anggota terakhir.
- (2) Bila dipilih [**Simpan**] maka akan dicari kode VCD pada tabel VCD, bila ada dilakukan update dan bila tidak ada dilakukan insert (append).
- (3) Bila dipilih [**Hapus**] maka akan dicari kode VCD pada tabel VCD, bila ada dilakukan delete dan bila tidak ada diberikan pesan bahwa data dengan kode VCD yang ditulis tidak ada.
- (4) Bila dipilih [**Keluar**] maka form ditutup dan kembali ke menu utama.

Implementasi dari setiap aturan di atas adalah sebagai berikut:

(1) Function event untuk pengisian No anggota.

```
Private Sub TxtNomember_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
  If TxtNomember <> "" Then
    SQL = "select * from member where nomember='" & Trim(TxtNomember) &
    """"
    Member.RecordSource = SQL
    Member.Refresh
    If Member.Recordset.EOF Then
      TxtNamamember = ""
      TxtAlamatmember = ""
      TxtTelpmember = ""
    Else
      TxtNamamember = VCD.Recordset!namamember
      TxtAlamatmember = VCD.Recordset!alamatmember
      TxtTelpmember = VCD.Recordset!telpmember
    End if
  Else
    MsgBox "Kode VCD tidak boleh kosong", vbOKOnly
  End If
End If
End Sub
```

(2) Function event untuk tombol Simpan.

```
Private Sub CmdSimpan_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
  If TxtNomember <> "" Then
    SQL = "select * from member where nomember='" & Trim(TxtNomember) &
    """"
    Member.RecordSource = SQL
    Member.Refresh
    If Member.Recordset.EOF Then
      Member.Recordset.Addnew
    End if
    Member.Recordset!nomember = Trim(TxtNomember)
    Member.Recordset!namamember = Trim(TxtNamamember)
    Member.Recordset!alamatmember = Trim(TxtAlamatmember)
    Member.Recordset!telpmember = Trim(TxtTelpmember)
    Member.Recordset.Update
    SQL = "select * from member"
    Member.RecordSource = SQL
    Member.Refresh
  End If
End Sub
```

```

Else
    MsgBox "Kode VCD tidak boleh kosong", vbOKOnly
End If
End If
End Sub

```

### (3) Function event untuk tombol Hapus.

```

Private Sub CmdHapus_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
    If TxtNomember <> "" Then
        SQL = "select * from member where nomember='" & Trim(TxtNomember) &
        """"
        Member.RecordSource = SQL
        Member.Refresh
        If Member.Recordset.EOF Then
            MsgBox "Data tidak ada", vbCritical
        Else
            Member.Recordset.Delete
            MsgBox "Data terhapus", vbOKOnly
        End if
        SQL = "select * from member"
        Member.RecordSource = SQL
        Member.Refresh
    Else
        MsgBox "Kode VCD tidak boleh kosong", vbOKOnly
    End If
End If
End Sub

```

### (2) Function event untuk tombol Simpan.

```

Private Sub CmdKeluar_Click()
    Unload Me
End Sub

```

## 3.3. Membuat Form Entry Transaksi

Form ini dibuat untuk menambah transaksi peminjaman baru, menandai transaksi pengembalian, memperbaiki transaksi peminjaman atau pengembalian, dan menghapus data peminjaman atau pengembalian yang salah. Tampilan Form Entry VCD ini adalah sebagai berikut:

Beberapa aturan yang harus dilakukan dalam form ini.

- (1) No nota peminjaman berupa angka counter. Bila no nota peminjaman ditemukan di tabel transaksi maka semua item data ditampilkan, dan bila tidak ada dianggap no nota baru dan no nota bertambah secara otomatis berdasarkan no nota terakhir.
- (2) Tanggal kembali adalah tanggal pinjam ditambah dengan sejumlah judul yang dipinjam.
- (3) Bila no anggota ada di tabel member maka data member ditampilkan dan bila tidak kembali harus mengisikan no anggota. Untuk member baru sebaiknya entry member dari form entry member terlebih dahulu.

- (4) Bila dipilih [**Simpan**] maka akan dicari data transaksi pada tabel transaksi, bila ada dilakukan update dan bila tidak lakukan insert. Kemudian cari data di tabel VCD\_dipinjam berdasarkan no nota dan kode VCD, bila ditemukan dilakukan update dan bila tidak dilakukan insert.
- (5) Bila dipilih [**Hapus**] maka akan dicari data transaksi di tabel VCD\_dipinjam berdasarkan no nota dan kode VCD, bila ditemukan dilakukan delete dan bila tidak ada diberi pesan data tidak ada.
- (6) Bila dipilih [**Keluar**] maka form ditutup dan kembali ke menu utama.

#### 4. MEMBUAT DATA REPORT

Di dalam aplikasi sistem informasi rental VCD, ada beberapa laporan yang harus bisa dicetak antara lain:

- Nota peminjaman yang dicetak setiap terjadi transaksi peminjaman
- Laporan peminjaman dan pengembalian
- Rekapitulasi peminjaman dan pengembalian

Setiap laporan ini dibuat dengan menggunakan Data Report dan dihubungkan dengan satu form yang menjadikan sarana komunikasi report dan user.

- Untuk report nota peminjaman, form penghubungnya alah form entry peminjaman (transaksi)
- Untuk laporan data rekapitulasi peminjaman dan pengembalian, dibuat form penghubung tersendiri.

Selain membutuhkan form penghubung, dalam pembuatan data report diperlukan adanta data environment sebagai pengubung report dan database. Sehingga dalam pembuatan repot yang perlu dibuat adalah :

- (1) Form Penghubung
- (2) Data Environment
- (3) Data Report

#### 4.1. Membuat Nota Peminjaman

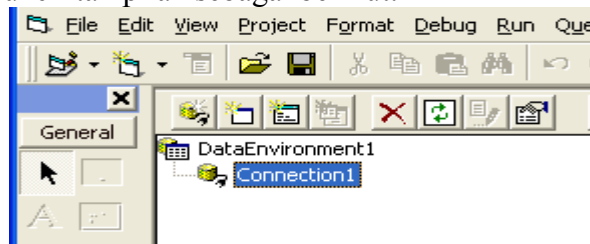
Report ini dibuat sebagai bukti peminjaman, dalam nota peminjaman selain ditampilkan informasi peminjaman secara detail, juga ditampilkan tempat untuk tanda tangan user dan peminjam. Langkah-langkah untuk membuat nota peminjaman adalah sebagai berikut:

- (1) Form penghubung menggunakan tombol [**Cetak Nota**] pada form Entry Peminjaman. Pada event dari tombol ini dituliskan :

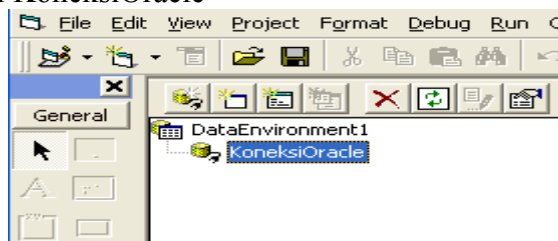
**NotaPinjam.show**

NotaPinjam adalah nama data report dari Nota Peminjaman

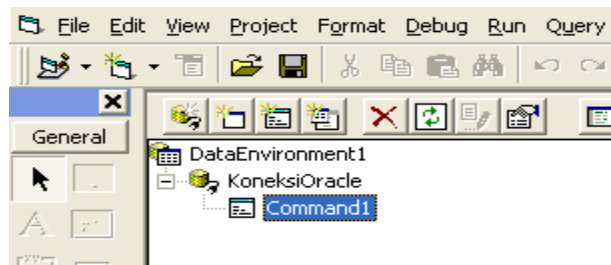
- (2) Membuat Data Environment, dilakukan dengan **Add Data Environment** pada pilihan project. Sehingga muncul tampilan sebagai berikut:



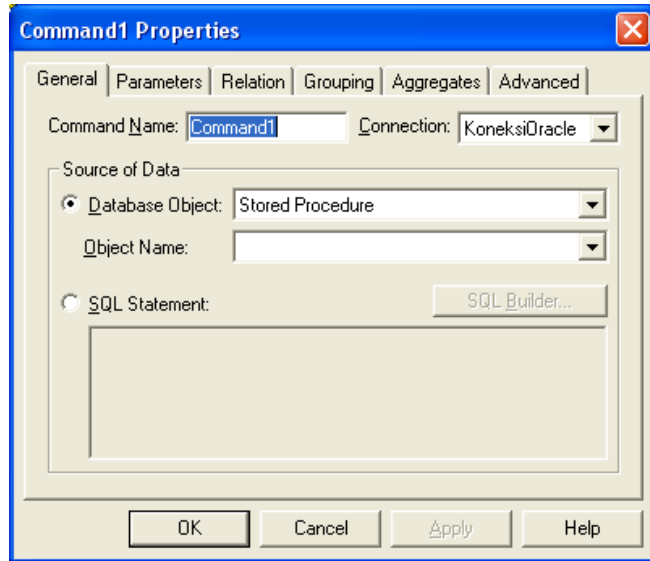
- (3) Pilih Connection1 dan click kanan, lalu pilih properties sehingga muncul tampilan seperti halnya koneksi database, cara koneksi sama dengan cara koneksi database.
- (4) Pilih lagi Connection1 dan click kanan, lalu pilih Rename, dan ubah nama Connection1 menjadi KoneksiOracle



- (5) Pada KoneksiOracle click kanan dan pilih Add Command sehingga muncul tampilan sebagai berikut:



- (6) Pilih Command1 dan click kanan, pilih properties sehingga muncul tampilan properties dari tabel atau query yang diaktifkan, sebagai berikut:



- (7) Ubah Command Name dengan NotaTransaksi, dan isi SQL statement dengan perintah query dari transaksi yang akan dicetak sebagai berikut:  
**select \* from vcd\_dipinjam**
- (8) Pilih [Ok] dan Data Environment sudah siap. Berikutnya adalah pembuatan data report dari nota peminjaman.

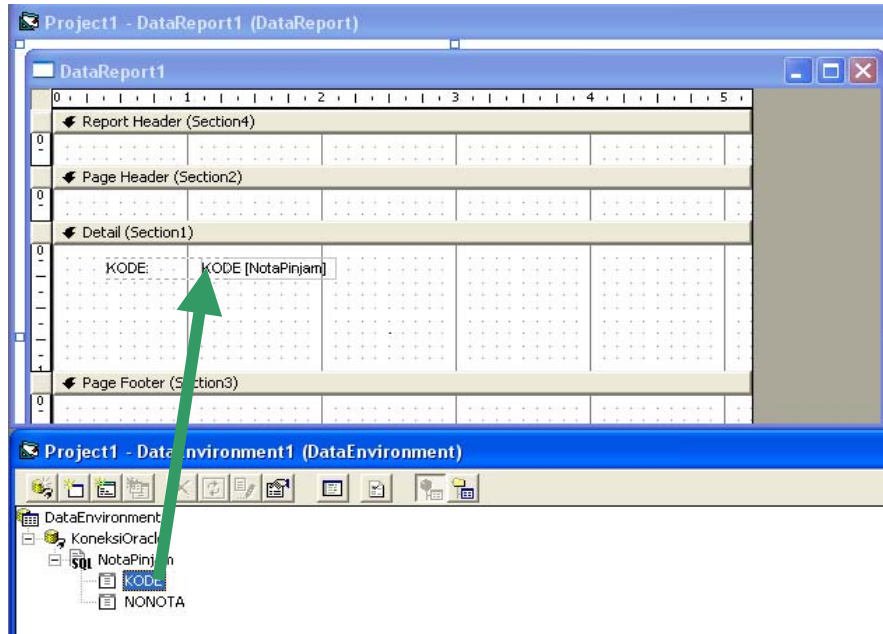
Tampilan nota peminjaman adalah sebagai berikut:

NOTA PEMINJAMAN		
Rental ABC - PENS		
Nota : _____ Tanggal : __/__/____		
Nama	: _____	No. Anggota : _____
Alamat	: _____	Telp. _____
Tanggal kembali : __/__/____		
Judul :	Jenis: No.Seri	
1.	_____	_____
2.	_____	_____
3.	_____	_____
4.	_____	_____
5.	_____	_____
Jumlah Pembayaran Rp. _____		
	Peminjam	Petugas
	( _____ )	( _____ )

Pembuatan data report dapat dilakukan dengan cara:

- (1) Pilih **Add Data Report** pada pilihan Project sehingga muncul tampilan kosong untuk pembuatan report.
- (2) Tarik variable dari field yang ada dalam data environment NotaPinjam ke dalam daerah Detail pada data report.





- (3) Kemudian atur sehingga sama dengan tampilan Nota Penjualan dan ubah nama Data Report Menjadi NotaPinjam (sama seperti yang dipanggil pada form entry peminjaman)
- (4) Isikan property Data Source dengan Data Environment1 dan Data Member dengan NotaPinjam (nama dari command pada data environment).

## 4.2. Membuat Laporan Peminjaman

Report ini dibuat menampilkan informasi peminjaman secara detail dalam periode waktu yang ditentukan misalkan harian, mingguan atau bulanan. Untuk laporan ini diperlukan pendukung form yang menyediakan tempat pengisian periode waktu yang diinginkan oleh user. Tampilan form dan laporan peminjaman adalah sebagai berikut:

LAPORAN PEMINJAMAN & PENGEMBALIAN			
Tanggal: __/__/__			
No. Nota	No. Anggota	J u d u l	Pembayaran
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____
Jumlah judul VCD		_____	Jumlah Pembayaran Rp. _____

Seperti halnya pembuatan data report dari Nota Peminjaman, proses pembuatan laporan peminjaman ini sama, hanya yang berubah adalah :

- (1) Nama Obyek dari Command pada Data Environment, gunakan nama **Peminjaman**, dan query yang dimasukkan adalah :
 

```
select transaksi.nonota,transaksi.nomember,
vcd_dipinjam.judul, transaksi.jumlahbayar
from transaksi, vcd_dipinjam
```

**where transaksi.nonota=vcd\_dipinjam.nonota**

(2) Nama Obyek dari Data Report adalah **Peminjaman**. Dengan property Data Source adalah Data Environment1 dan data member adalah Peminjaman.

### 4.3. Membuat Rekapitulasi Peminjaman

Report ini dibuat menampilkan informasi rekapitulasi peminjaman dalam periode waktu yang ditentukan misalkan harian, mingguan atau bulanan. Untuk laporan ini diperlukan pendukung form yang menyediakan tempat pengisian periode waktu yang diinginkan oleh user. Tampilan form dan rekapitulasi peminjaman dan pengembalian adalah sebagai berikut:

REKAPITULASI PEMINJAMAN & PENGEMBALIAN			
Tanggal __/__/____ sampai tanggal __/__/____			
Tanggal	Jumlah VCD Dipinjam	Jumlah VCD kembali	Pembayaran
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____
Jumlah	_____	_____	_____

Permasalahan di dalam pembuatan rekapitulasi adalah permasalahan dalam menyusun query dalam bentuk SQL. Salah satu cara yang bisa dilakukan adalah dengan memanfaatkan **view** yang ada dalam database Oracle. Terlebih dahulu dibuat view yang dapat menampilkan informasi sesuai dengan kebutuhan rekapitulasi. View ini hanya sekali saja dibuat. View yang harus dibuat adalah sebagai berikut:

View yang menggabungkan table transaksi dan table vcd\_dipinjam ditulis dalam view vpinjam sebagai berikut:

```
create view vpinjam as
select transaksi.*, vcd_dipinjam.kode from transaksi, vcd_dipinjam
where transaksi.nonota=vcd_dipinjam.nonota
```

View untuk menghitung jumlah VCD yang dipinjam setiap hari dan jumlah pembayarannya ditulis dalam view rpinjam adalh sebagai berikut:

```
create view rpinjam as
select tglpinjam, count(kode), jumlahbayar from vpinjam
group by tglpinjam
```

Seperti halnya pembuatan data report dari nota peminjaman, proses pembuatan laporan rekapitulasi peminjaman ini sama, yang berubah adalah:

(1) Nama obyek dari Command pada Data Environment, gunakan nama RekapPinjam dan query yang dimasukkan adalah:

**select \* from rpinjam**

(2) Nama obyek dari Data Report adalah RekapPinjam. Dengan properties Data Source adalah Data Environment1 dan data member adalah RekapPinjam.