

KOMUNIKASI DATA DUA APLIKASI DALAM DELPHI

Hery Heryanto

Sekolah Tinggi Manajemen Informatika dan Komputer LIKMI
Jl. Ir. H. Juanda 96 Bandung 40132

Email : hery.heryanto@gmail.com

Abstrak

Pengolahan data dalam sebuah aplikasi adalah suatu hal yang sangat biasa dalam pengolahan informasi. Berbeda dengan komunikasi data atau informasi antar dua aplikasi dalam sebuah pengolahan informasi, hal ini tidak mudah karena perbedaan struktur database dan program yang menjadi suatu kendala dalam pengiriman data tersebut. Komunikasi antar aplikasi dapat dilakukan dengan berbagai cara dan *protocol*. Baik itu melalui TCP (mis: UDP, *Socket Communication* atau cara lainnya tergantung dengan situasi penggunaannya. Salah satu cara yaitu penggunaan *Windows message* (WM_COPYDATA) bila kedua aplikasi yang bertukar informasi sama-sama ada dalam sebuah perangkat atau komputer yang sama.

Kata-kata kunci : *string, image, record, API, WM_CopyData, TcopyDataStruct*

1. PENDAHULUAN

Dalam suatu pengembangan program aplikasi ada beberapa situasi dan kondisi yang ‘memaksa’ pengembang untuk mengkomunikasikan antara dua aplikasi berbeda, baik itu secara sengaja maupun tidak sengaja. Berbagai cara dapat dilakukan untuk mengkomunikasikan kedua aplikasi tersebut, cara yang biasa digunakan adalah TCP (mis: UDP) dan *Socket Communication*.

TCP dan *Socket Communication* sangat efisien dan efektif jika dua aplikasi yang akan bertukar informasi terletak di dua perangkat atau komputer yang berbeda. Dalam satu perangkat dapat juga menggunakan cara tersebut namun ada cara lain yang lebih mudah dan praktis yaitu dengan menggunakan *Windows message*: WM_COPYDATA.

DELPHI sekarang ini sebagai sebuah *development tool* memberikan kemudahan dalam menggunakan *handling Windows message* (manipulasi pesan yang ada di sistem operasi Windows) dengan mengeluarkan sebuah *SendMessage* API untuk memanggil WM_CopyData yang diisi data yang akan dikirim ke aplikasi lain. Cara ini cukup cepat dan praktis untuk mengirimkan informasi antar dua aplikasi berbeda dalam satu perangkat yang sama.

2. WM_COPYDATA dan TCopyDataStruct

WM_COPYDATA *message* dapat menjadi sarana untuk mengirimkan data atau informasi dari satu aplikasi ke aplikasi lainnya. Aplikasi penerima menerima data yang dikirimkan berupa TCopyDataStruct *record*. TCopyDataStruct didefinisikan di dalam file Windows.pas dan membungkus struktur COPYDATASTRUCT yang berisikan data yang akan diperbolehkan untuk ditransfer.

Ini adalah contoh deklarasi dan deskripsi dari TCopyDataStruct *record* dalam DELPHI:

Type

```
TCopyDataStruct = packed record
dwData: DWORD; //sampai 32 bits data yang dapat dikirim ke aplikasi
                penerima
cbData: DWORD; //ukuran, dalam bytes, data yang ditunjuk lpData member
lpData: Pointer; //alamat data yang akan dikirim ke aplikasi penerima,
                boleh bernilai nol (nil).
end;
```

3. PROSES PERTUKARAN INFORMASI (STRING, IMAGE, RECORD)

• Pengiriman sebuah String dengan WM_CopyData

Aplikasi “Pengirim” dalam mengirimkan data ke aplikasi “Penerima” harus melakukan pengisian dan transfer dengan menggunakan fungsi SendMessage. Di bawah ini adalah cara mengirimkan sebuah nilai string dengan WM_CopyData:

```
procedure TSenderMainForm.SendString() ;
var
    stringToSend : string;
    copyDataStruct : TCopyDataStruct;
begin
    stringToSend := 'About Delphi Programming';

    copyDataStruct.dwData := 0; //digunakan untuk mengidentifikasi isi
    pesan
    copyDataStruct.cbData := 1 + Length(stringToSend) ;
    copyDataStruct.lpData := PChar(stringToSend) ;

    SendData(copyDataStruct) ;
end;
```

Fungsi SendData menyimpan data yang telah diterima dari fungsi SendString dengan memanggil FindWindow API:

```

procedure TSenderMainForm.SendData(copyDataStruct: TCopyDataStruct) ;
var
    receiverHandle : THandle;
    res : integer;
begin
    receiverHandle :=
FindWindow(PChar('TReceiverMainForm'),PChar('ReceiverMainForm')) ;
    if receiverHandle = 0 then
        begin
            ShowMessage('CopyData Receiver NOT found!') ;
            Exit;
        end;
end;

```

Dalam *code* di atas, aplikasi “Penerima” dikenali dengan menggunakan FindWindow API yang mencari nama *class* dari form utama ("TReceiverMainForm") dan judul window formnya ("ReceiverMainForm").

Note: SendMessage memiliki return values berupa integer yang dihasilkan kode di atas dan ditangani oleh the WM_CopyData message.

• Penanganan WM_CopyData – Penerimaan sebuah String yang telah dikirim

Aplikasi “Penerima” menangani WM_CopyData *message* dengan:

```

type
    TReceiverMainForm = class(TForm)
    private
        procedure WMCopyData(var Msg : TWMCopyData) ; message WM_COPYDATA;
    ...
implementation
    ...
procedure TReceiverMainForm.WMCopyData(var Msg: TWMCopyData) ;
var
    s : string;
begin
    s := PChar(Msg.CopyDataStruct.lpData) ;

```

```
//kirim balasan
msg.Result := 2006;
end;
```

TWMCopyData record dideklarasikan dengan:

```
TWMCopyData = packed record
Msg: Cardinal;
  From: HWND;//Handle dari Window yang mengantarkan data
  CopyDataStruct: PCopyDataStruct; //data lewat
  Result: Longint;//ini digunakan untuk mengirim nilai balasan ke
  "Pengirim"
end;
```

• Pengiriman String, Custom Record atau sebuah Image?

Ada banyak contoh *source code* yang memperlihatkan cara untuk mengirimkan String, Record atau Image. Berikut ini adalah salah satu contoh untuk mengirimkan sebuah gambar TBitmap:

```
procedure TSenderMainForm.SendImage() ;
var
  ms : TMemoryStream;
  bmp : TBitmap;
  copyDataStruct : TCopyDataStruct;
begin
  ms := TMemoryStream.Create;
  try
    bmp := self.GetFormImage;
    try
      bmp.SaveToStream(ms) ;
    finally
      bmp.Free;
    end;
  end;

  copyDataStruct.dwData := Integer(cdtImage) ; // identifikasi data
  copyDataStruct.cbData := ms.Size;
  copyDataStruct.lpData := ms.Memory;

  SendData(copyDataStruct) ;
finally
```

```

ms.Free;
end; end;

```

Cara untuk menerimanya:

```

procedure TReceiverMainForm.HandleCopyDataImage(
    copyDataStruct: PCopyDataStruct) ;
var
    ms: TMemoryStream;
begin
    ms := TMemoryStream.Create;
try
    ms.Write(copyDataStruct.lpData^, copyDataStruct.cbData) ;
    ms.Position := 0;
    receivedImage.Picture.Bitmap.LoadFromStream(ms) ;
finally
    ms.Free;
end;
end;

```

4. LAYOUT INTERFACE



Image hasil meng-capture Aplikasi "Sender"

Gambar di atas menggambarkan interface dari aplikasi "Sender" dan aplikasi "Receiver". Aplikasi "Sender" memiliki *radiogroup* yang di dalamnya terdapat 3 *option* yaitu String, Image dan Record. Button Send digunakan untuk men-*trigger* pengiriman data. Setelah data dikirim, *memo* dalam aplikasi "Receiver" diisi dengan keterangan id pengirim, data yang diterima dan tanggal penerimaan.

Pengiriman data dapat dilakukan jika kedua aplikasi ini dijalankan. Apabila satu aplikasi saja yang dijalankan maka aplikasi "Sender" akan menolak untuk mengirim data karena aplikasi "Sender" mengecek terlebih dahulu aplikasi "Receiver".

5. KESIMPULAN

Pertukaran informasi antar aplikasi dalam *software development* memiliki berbagai cara atau proses yang berbeda. Salah satu cara yang dapat digunakan adalah dengan menggunakan Windows *message* yang lebih praktis dan tidak memerlukan *protocol* seperti TCP,UDP dan *Socket Communication*. Proses ini dapat dilakukan apabila kedua aplikasi tersebut ada dalam satu perangkat atau komputer yang sama karena dalam satu sistem operasi yang sama yaitu Windows.

6. DAFTAR PUSTAKA

1. Delphi.about.com; “*How to Send Information (String, Image, Record) Between Two Applications*”, Zarco Gajic 2007
2. Cantu, Marco 2003. *Mastering Delphi 7*, Sybex.