

## BAB 6

### PROGRAM UNITS DAN STORED PROGRAMS UNITS

Oracle Form menyediakan mekanisme untuk menyimpan prosedur (disebut dengan *Program Units*) yang ditulis dalam bahasa PL/SQL. *Program Units* dapat digunakan untuk menambahkan fungsionalitas yang tidak mungkin dapat dilakukan oleh SQL. Beberapa contoh program units telah diberikan pada saat membahas pembuatan Master Detail Form pada bab sebelumnya. Dengan menyimpan Program Units pada suatu form, maka beberapa block pada form dapat memakai prosedur yang sama. Dengan demikian akan dapat mereduksi penulisan kode yang sama dan dapat memudahkan pemeliharaan kode yang ditulis

Procedure yang ditulis dalam PL/SQL juga dapat disimpan dalam Oracle Database sebagai suatu obyek dalam sebuah schema. *Stored Program Units* (disebut juga dengan *Stored Procedures*) sangat ideal untuk situasi dimana aturan standart bisnis atau logika aplikasi harus diimplementasi pada beberapa form atau aplikasi. User dapat membuat *stored procedures* dengan baris perintah pada Oracle SQL\*PLUS.

Oracle Forms dapat memanggil Program Units (secara internal) maupun Stored Program Units. Fleksibilitas ini memungkinkan desainer aplikasi untuk memperluas kontrol atas jalannya aplikasi dan fasilitas yang disediakan.

Pada bab ini akan dibahas contoh pembuatan *Program Units* dan *Stored Program Units* pada Oracle Forms.

#### 5.1. Membuat Program Units dalam Form Oracle

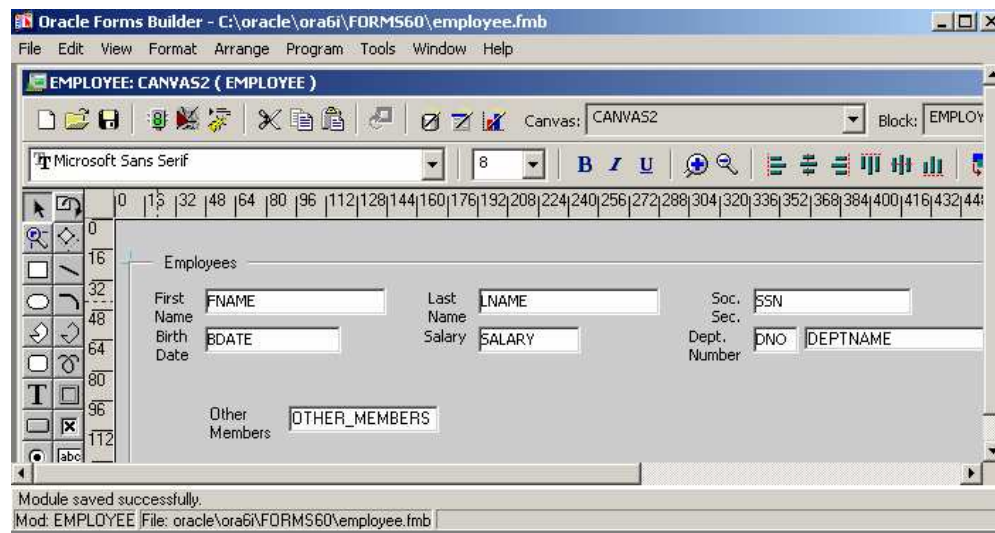
Pada bab berikut, kita akan menambahkan satu field untuk menghitung jumlah anggota department yang sama dengan seorang employee yang datanya ditampilkan pada block EMPLOYEE. Field ini disebut dengan OTHER\_MEMBERS.

### 5.1.1. Membuat Item Baru pada Block EMPLOYEE

Tambahkan item baru ke dalam block EMPLOYEE. Beri nama dengan OTHER\_MEMBERS, ubah propertinya sebagai berikut :

**Enabled = Yes**  
**Navigabled = No**  
**Database Item = No**  
**Query Only = No**  
**Primary Key = No**  
**Insert Allowed = No**  
**Query Allowed = No**  
**Update Allowed = No**

Tampilan block EMPLOYEE setelah diberi tambahan satu item baru :



### 5.1.2. Membuat Prosedur u/ menghitung Anggota dari Departemen Lain

Selanjutnya kita akan membuat *Program Units (prosedur)* dalam Oracle Forms yang dapat menghitung jumlah anggota yang lain dari department employee. Untuk keperluan ini, kita perlu melewati parameter yang diberi nama IN\_EMPLOYEE\_SSN yang berisi nomer SSN dari employee. Prosedur akan mengembalikan nilai pada parameter OUT\_COUNT\_OTHER\_MEMBERS yang berisi jumlah employee lain yang bekerja pada department yang sama.

Untuk membuat prosedur ini, pilih *Program Units* pada Object Navigator. Kemudian klik menu *Navigator -> Create*. Kemudian beri nama dengan OTHER\_DEPARTMENT\_MEMBERS.

Ketik kode berikut :

```
PROCEDURE OTHER_DEPARTMENT_MEMBERS(IN_EMPLOYEE_SSN
IN NUMBER, OUT_COUNT_OTHER_MEMBERS OUT NUMBER) IS
  TEMP_COUNT NUMBER;
BEGIN
  TEMP_COUNT:=0;
  SELECT COUNT(*)
  INTO TEMP_COUNT
  FROM EMPLOYEE
  WHERE EMPLOYEE.DNO =
  (SELECT EMPLOYEE.DNO
  FROM EMPLOYEE
  WHERE EMPLOYEE.SSN=IN_EMPLOYEE_SSN);

  IF (NVL(TEMP_COUNT,0) <>0) THEN
    OUT_COUNT_OTHER_MEMBERS:=(TEMP_COUNT - 1);
  ELSE
    OUT_COUNT_OTHER_MEMBERS:=0;
  END IF;
END;
```

Klik tombol *Compile* untuk memastikan kode sudah benar.

### 5.1.3. Membuat Trigger untuk Memanggil Prosedur

Kemudian buat trigger untuk memanggil prosedur yang baru dibuat. Buat trigger POST\_CHANGE pada field SSN. Kemudian ketik kode berikut :

```
declare
  return_count NUMBER;
BEGIN
  OTHER_DEPARTMENT_MEMBERS(:EMPLOYEE.SSN,
return_count);
  :EMPLOYEE.OTHER_MEMBERS:=return_count;
END;
```

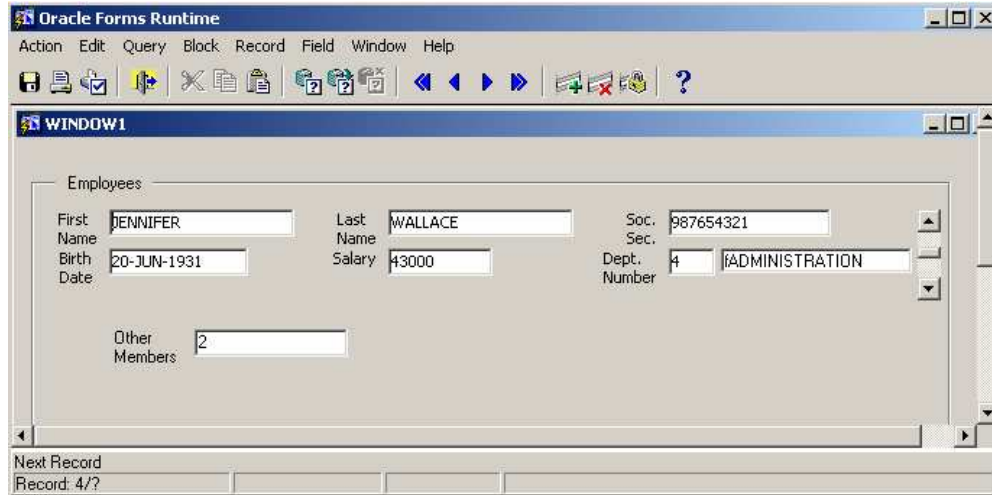
Klik tombol *Compile* untuk memastikan kode sudah benar.

#### 5.1.4. Menyimpan, meng-Compile/Build dan Menjalankan Form

Simpan form dengan nama *emp\_memb.fmb*

Untuk itu klik pada menu *File -> Save As*, kemudian beri nama *emp\_memb*

Setelah disimpan, form dijalankan dan akan memberikan hasil seperti ini :



The screenshot shows the Oracle Forms Runtime window titled 'WINDOW1'. The form is titled 'Employees' and contains several input fields: First Name (JENNIFER), Last Name (WALLACE), Birth Date (20-JUN-1931), Salary (43000), Soc. Sec. Number (987654321), Dept. Number (4), and Other Members (2). The status bar at the bottom indicates 'Next Record' and 'Record: 4/?'.

## 5.2. Membuat Stored Procedures dalam SQL Plus

Pada contoh sebelumnya, telah dibuat prosedur (*Program Units*) pada Oracle Forms yang digunakan untuk membentuk query pada database. Prosedur ini disimpan sebagai bagian dari source code dan di-compile pada saat form di-built. Proses yang terjadi pada saat pemanggilan prosedur :

1. Pemanggilan dari trigger POST\_CHANGE ke procedure yang melewati nomer ssn dari employee.
2. Remote access dari prosedur ke database dengan melewati SQL query
3. Remote return dari database dalam bentuk perintah *count*
4. Sejumlah proses pengolahan dalam prosedur (IF..THEN)
5. Local return dari prosedur ke trigger.

Untuk mereduksi trafik jaringan yang diperlukan untuk pergi dan kembali diantara client dan server (nomer 2 dan 3). Cost paling rendah diperoleh pada pemanggilan local antara trigger dan prosedur dan pengolahan local (nomer 1,4,5).

Pada contoh yang dibuat, data yang diquery tidak terlalu banyak. Bagaimanapun, pada persoalan real, sejumlah data yang cukup besar akan di-return, dan ini membutuhkan waktu proses yang cukup lama.

Satu solusi untuk mengatasi hal ini adalah dengan memindah query lebih dekat ke data yang ada dalam database. Untuk itu cara yang dapat dilakukan adalah dengan mengimplementasikan prosedur OTHER\_DEPARTMENT\_MEMBERS sebagai prosedur dalam database. Caranya adalah dengan menggunakan perintah Oracle SQL\*PLUS untuk membuat *stored procedure*.

Login ke SQL\*PLUS, kemudian jalankan perintah berikut pada SQL\*PLUS.

```
CREATE PROCEDURE OTHER_DEPARTMENT_MEMBERS
  (IN_EMPLOYEE_SSN IN NUMBER,
   OUT_COUNT_OTHER_MEMBERS OUT NUMBER) AS
  TEMP_COUNT NUMBER;
BEGIN
  TEMP_COUNT:=0;
  SELECT COUNT(*)
  INTO TEMP_COUNT
  FROM EMPLOYEE
  WHERE EMPLOYEE.DNO =
  (SELECT EMPLOYEE.DNO
  FROM EMPLOYEE
  WHERE EMPLOYEE.SSN=IN_EMPLOYEE_SSN);

  IF (NVL(TEMP_COUNT,0) <>0) THEN
    OUT_COUNT_OTHER_MEMBERS:=(TEMP_COUNT - 1);
  ELSE
    OUT_COUNT_OTHER_MEMBERS:=0;
  END IF;
END;
/
```