

BAB V

IMPLEMENTASI PERANGKAT LUNAK

5.1 Implementasi Sistem

Pada tahap implementasi ini akan diharapkan bahwa sistem yang telah dirancang dapat dioperasikan pada keadaan yang sebenarnya.

5.1.1. Rencana Implementasi

Rencana implementasi sebuah sistem dilakukan untuk merencanakan hal-hal yang berhubungan dengan sarana terhadap pengadaan dan pemeliharaan sistem yang direncanakan, misalnya biaya yang diperlukan. Pada tahap pengimplementasian sistem, semua biaya akan dikeluarkan untuk kegiatan implementasi dan pemeliharaan. Waktu yang digunakan untuk kegiatan implementasi tersebut dibuat dalam sebuah rencana atau jadwal waktu melalui jadwal waktu ini akan dapat digambarkan mengenai tahapan-tahapan pekerjaan yang akan dilaksanakan.

5.1.2. Kegiatan Implementasi

Kegiatan implementasi dilaksanakan atas dasar rencana implementasi yang telah disusun. Kegiatan implementasi meliputi :

1. Mempersiapkan dan memilih tempat instalasi sistem.
2. Pemrograman dan Pengujian sistem

5.2. Pembahasan Sistem

Pembuatan sistem, database merupakan prioritas utama yang harus didahulukan sebelum mulai menuju ke pemrograman. Hal ini dikarenakan, database merupakan landasan dari bentuk pemrograman. Database Paradox merupakan sistem database yang dijadikan sasaran penyimpanan data. Dengan integrasi yang cukup kuat dengan Borland Delphi (karena satu produk dengan Borland Group), database Paradox diharapkan mampu untuk mendukung hasil akhir yang memadai dari terbentuknya sistem.

Sistem yang terbagi dalam delapan form disajikan dalam bentuk yang sederhana agar mudah digunakan oleh user. Pembagian ketiga bagian user juga memberikan perlindungan akses ke dalam menu-menu yang tidak berhak. Pada sistem telah dirancang, untuk user Administrasi hanya dapat membuka menu Administrasi. Untuk user Penjualan juga hanya dapat membuka menu Penjualan. Namun, hanya user Root (super user) yang dapat membuka semua menu serta menginputkan data pameran dan mengisikan stok barang awal. Perintah-perintah SQL yang dipakai dalam kebanyakan sistem antara lain :

1. select

Digunakan untuk membaca jumlah record atau menampilkan record ke dalam grid data (T Data Grid) atau ke laporan (Quick Report)

Contoh diletakkan dalam event Form Show pada saat membuka jendela Pembeli (Kustomer):

```

q1.Close;
q1.SQL.Clear;
q1.SQL.Add('select * from pembeli where kode_pameran=:kode_pameran order by
nama');
q1.Params[0].AsString := FUtama.kode_pameran;
q1.Open;
StatusBar1.Panels[0].Text := inttostr(q1.RecordCount)+' data ...';

```

2. insert dan update

Dipakai dalam operasi penambahan atau penyimpanan data yang diedit.

Contoh diletakkan pada saat user menekan tombol Simpan di form

```

if edit_data then
begin
q1.Close;
q1.SQL.Clear;
q1.SQL.Add('select * from users where nama=:nama_temp');
q1.Params[0].AsString := nama_temp;
q1.Open;
if not q1.IsEmpty then
begin
q1.Close;
q1.SQL.Clear;
q1.SQL.Add('delete from users where nama=:nama_temp');
q1.Params[0].AsString := nama_temp;
q1.ExecSQL;
end;
q1.Close;
q1.SQL.Clear;
q1.SQL.Add('insert into users(nama,passwd,lvl) ');
q1.SQL.Add('values (:nama,:passwd,:lvl)');
q1.Params[0].AsString := nama;
q1.Params[1].AsString := passwd;
q1.Params[2].AsString := lvl;
q1.ExecSQL;
end
else
begin
q1.Close;
q1.SQL.Clear;
q1.SQL.Add('select * from users where nama=:nama');
q1.Params[0].AsString := nama;
q1.Open;
if q1.IsEmpty then
begin
q1.Close;
q1.SQL.Clear;
q1.SQL.Add('insert into users(nama,passwd,lvl) ');
q1.SQL.Add('values (:nama,:passwd,:lvl)');
q1.Params[0].AsString := nama;

```

```

        q1.Params[1].AsString := passwd;
        q1.Params[2].AsString := lvl;
        q1.ExecSQL;
    end
else
    begin
        Application.MessageBox('Nama User ini telah ada dalam database.
Ulangi pengisian Data User.', 'Perhatian', MB_ICONWARNING);
    end;
end;
end;

```

3. delete

Perintah SQL ini akan dieksekusi pada saat user menekan tombol delete keyboard pada grid data (user menghapus data pameran).

```

Q1.Close;
q1.SQL.Clear;
q1.SQL.Add('delete from pameran where kode_pameran=:kode_pameran_temp');
q1.Params[0].AsString := kode_pameran_temp;
q1.ExecSQL;

```

Transaksi data pada setiap table, selalu ditangani pertama kali oleh identitas pameran. Jika pada table pameran terdapat record pameran dengan field status bernilai aktif, maka pada judul caption akan ditampilkan nama pameran tersebut. Namun, apabila dalam table pameran tidak ada record pameran dengan field status bernilai aktif, maka sistem akan menolak untuk melakukan transaksi data. Sehingga sistem hanya dapat menyimpan data-data pameran yang sebelumnya telah direncanakan. Pada laporan, hanya akan terbagi ke dalam 3 modul antara lain :

1. Laporan Stok Barang

Pada laporan ini, kategori dibagi ke dalam 2 filter, yaitu tampilkan semua data dan tampilkan data berdasarkan jenis barangnya. Filter ini dimasukkan ke

5.3. Evaluasi dan pemeliharaan Sistem

5.3.1. Evaluasi dan Pengujian Sistem

Sebelum sebuah sistem akan diterapkan pada kegiatan yang sebenarnya maka diperlukan suatu kegiatan untuk mengevaluasi dan pengujian terhadap berbagai aspek. Supaya program aplikasi tersebut dapat dikatakan *error free* atau bebas dari berbagai kesalahan, maka pengujian dilakukan untuk dapat menemukan berbagai kesalahan yang mungkin terjadi. Program aplikasi tersebut diuji untuk setiap modul-modulnya, kemudian dilanjutkan dengan pengujian terhadap semua modul yang telah dirangkai. Tahapan pengujian sistem ini meliputi :

1. Pengujian Pengkodean

Pengujian program aplikasi dilakukan supaya kemungkinan terjadinya berbagai kesalahan pada program tersebut dapat diidentifikasi atau diketahui sejak awal. Dalam sebuah program, tanpa pandang bulu mengenai ukuran program maupun pembuatnya, selalu terdapat kemungkinan adanya kesalahan (*error*). Dalam dunia komputer kesalahan (*error*) juga diistilahkan dengan *bug* dan proses pencarian terhadap kesalahan-kesalahan tersebut dinamakan *debugging*. *Error* merupakan problem di dalam pekerjaan pemrograman, karena kehadirannya menyebabkan program tidak dapat bekerja dengan semestinya.

a. Kesalahan Bahasa (*language error*) dan Kesalahan Penulisan Sintax Program (*syntax error*)

dalam satu ComboBox dengan perintah SQL yang lengkap adalah sebagai berikut :

```
q1.Close;
q1.SQL.Clear;
if (edJenis_Barang.ItemIndex=0) then
    q1.SQL.Add('select * from barang where
kode_pameran='+quotedStr(FUtama.kode_pameran)+' order by jenis_barang desc')
else
    q1.SQL.Add('select * from barang where
kode_pameran='+quotedStr(FUtama.kode_pameran)+'
jenis_barang='+quotedStr(edJenis_Barang.Text)+' order by jenis_barang desc');
q1.Open;
```

2. Laporan Penjualan

Laporan penjualan terbagi menjadi 2 kriteria pencarian, yaitu tampilkan semua data dan tampilkan data berdasarkan tanggal harian. Syntax SQL-nya adalah :

```
q1.Close;
q1.SQL.Clear;
if RadioButton1.Checked then
    q1.SQL.Add('select * from trans_beli where
kode_pameran='+quotedStr(FUtama.kode_pameran)+' order by no_nota')
else
    q1.SQL.Add('select * from trans_beli where
kode_pameran='+quotedStr(FUtama.kode_pameran)+'
tgl_beli='+quotedStr(FormatDateTime('M/d/yy',edTanggal.Date))+
no_nota');
q1.Open;
```

3. Laporan Kustomer

Laporan kustomer tidak dibagi dalam kategori apapun. Pada laporan, seluruh data customer akan ditampilkan sesuai dengan kode pameran yang aktif

Merupakan kesalahan di dalam penulisan *source code* program yang tidak sesuai dengan yang disyaratkan. Kesalahan jenis ini dapat terjadi karena salah di dalam menuliskan ejaan suatu kode, lupa dalam memberikan penutup *blok kode* program, tidak menyertakan suatu argumen yang diminta dan berbagai kesalahan sintak lainnya. Delphi menjumpai kesalahan semacam ini, maka ia akan menghentikan kegiatan dan menunjukkan dimana letak sintak *error* tersebut atau dengan cara menyoroti kode program yang bersintak salah, kemudian harus mengoreksi semua sintak *error* sampai program dapat berjalan, berarti kode telah bebas dari kesalahan sintak.

b. Kesalahan Pada Waktu Proses (*run time error*)

Merupakan kesalahan yang terjadi pada waktu proses *eksekutable* atau eksekusi terhadap program saat dijalankan. Komputer tidak pernah salah, dalam arti ia mengerjakan segala sesuatu yang diperintahkan padanya. Tetapi pemrogram sering kali tidak sengaja memberikan perintah yang salah. Terjadilah *exception!* Yang menyebabkan proses terhadap program aplikasi tersebut berhenti sebelum selesai pada saatnya karena *kompiler* menemukan kondisi-kondisi yang belum terpenuhi yang tidak dapat dikerjakan.

atau mutu suatu program sebelum sebuah sistem dapat diterapkan pada lingkungan yang sebenarnya.

5.3.2. Pemeliharaan Sistem

Pemeliharaan sistem dilakukan untuk menjamin supaya daya guna dari sistem yang telah jadi dan bebas dari berbagai kesalahan tersebut dapat dimanfaatkan atau digunakan semaksimal mungkin, baik dari segi kegunaannya maupun dari segi pemanfaatan waktu penggunaannya. Pemeliharaan terhadap sebuah sistem dapat dilakukan secara rutin dan berkala supaya apabila terjadi kerusakan atau hal-hal yang tidak diinginkan terjadi pada program aplikasi, maka hal tersebut dapat terdeteksi dengan cepat.

