

BAB 4

4.1 Perencanaan Tahap Implementasi

Tahap Perencanaan

- Menyadari masalah
- Mendefinisikan masalah
- Menentukan tujuan sistem
- Mengidentifikasi kendala sistem
- Membuat studi kelayakan
- Menyiapkan usulan penelitian sistem
- Menyetujui atau menolak proyek penelitian
- Menetapkan mekanisme pengendalian

4.1.1 Pembagian Modul / Prosedur / Class Implementasi

Ada 8 yaitu:

Modul untuk Karyawan

'membuat sub untuk menampilkan kode Karyawan secara otomatis

Sub Kode_Karyawan()

'Memanggil Modul Koneksi_Database

Call Koneksi_Database

```
sql = "select * from Karyawan order by val(right(kode_Karyawan,len(kode_Karyawan)-2))"
```

```
rs.Open sql, db, adOpenDynamic, adLockBatchOptimistic
```

```
With rs
```

```
If .EOF = False Then
```

```
.MoveFirst
```

```
Do While Not .EOF
```

```
txtkode.Text = .Fields("kode_Karyawan")
```

```
.MoveNext
```

```
Loop
```

Modul Untuk Mengganti Password

```
Call Koneksi_Database
```

```
sql = "select * from Petugas where namaPetugas = '" & _
```

```
FrmMenuUtama.Ibluser.Caption & "' and PassPetugas = '" & _
```

```
TxtPswLama.Text & "'" & ""
```

```
rs.Open sql, db, adOpenDynamic, adLockBatchOptimistic
```

```
With rs
```

```
If .EOF = False Then
```

Modul Untuk Form Pengunjung

```
'membuatsub untuk menampilkan kode pengunjung secara otomatis
```

```
Sub no_pengunjung()
```

```
Call Koneksi_Database
```

```
sql = "select * from Pengunjung Where no_pengunjung In(Select()From Pengunjung)Order By  
no_pengunjung Desc"
```

```
rs.Open sql, db, adOpenDynamic, adLockBatchOptimistic
```

Modul Untuk form Petugas

```
Sub data_petugas()
```

```
Call Koneksi_Database
```

```
With rs
```

```
.ActiveConnection = db
```

```
.CursorLocation = adUseClient
```

```
.LockType = adLockBatchOptimistic
```

```
.Source = "Select * From petugas order by kodepetugas"
```

```
.Open  
If .EOF = True Then Exit Sub  
End With  
Set DataGrid1.DataSource = rs  
End Sub
```

Module Untuk form Transaksi

```
Call Koneksi_Database  
sql = "select * from transaksi order by kode_transaksi"  
rs.Open sql, db, adOpenDynamic, adLockBatchOptimistic  
With rs  
cmbkdtransaksi.Clear  
If .EOF = False Then  
.MoveFirst  
Do While Not .EOF  
cmbkdlapangan.AddItem .Fields("kode_transaksi")  
.MoveNext  
Loop
```

Module Untuk Form Laporan Karyawan

```
Private Sub btncetak_Click()  
Call cetak  
End Sub  
Private Sub Form_Load()  
tgl.Value = Format(Date, "dd/mm/yyyy")  
tgl1.Value = Format(Date, "dd/mm/yyyy")  
tgl2.Value = Format(Date, "dd/mm/yyyy")
```

```
Me.Top = 2500
Me.Left = 3750
Pillaporan(0).Value = False: Pillaporan(1).Value = False: _
Pillaporan(2).Value = False: Pillaporan(3).Value = False: _
tgl.Enabled = False: tgl1.Enabled = False: tgl2.Enabled = False
Call Koneksi_Database
sql = "SELECT no_Kar FROM Karyawan ORDER BY no_Karyawan"
rs.Open sql, db, adOpenDynamic, adLockBatchOptimistic
While Not rs.EOF
    cmbno.AddItem rs!no_Karyawan: rs.MoveNext: Wend
    cmbno.Enabled = False
End Sub
```

Module Untuk Form Laporan Pengunjung

```
Call Koneksi_Database
sql = "select *from Pengunjung order by no_pengunjung"
rs.Open sql, db, adOpenDynamic, adLockBatchOptimistic
While Not rs.EOF
    cmbkode.AddItem rs!no_pengunjung: rs.MoveNext: Wend
    cmbkode.Enabled = False
End Sub
```

Module Untuk form Laporan Transaksi

```
Call Koneksi_Database
sql = "SELECT kode_karyawan FROM Karyawan ORDER BY kode_Karyawan"
rs.Open sql, db, adOpenDynamic, adLockBatchOptimistic
While Not rs.EOF
```

```
cmbno.AddItem rs!no_Karyawan: rs.MoveNext: Wend
```

```
cmbno.Enabled = False
```

```
End Sub
```

4.1.2 Keterkaitan Antar Modul / Prosedur / Class

4.2 Perjalanan Tahap Implementasi (Coding)

4.2.1 Top Down / bottom Up Implementasi

4.2.2 Debugging

4.3 Ulasan Realisasi Fungsionalitas

4.4 Ulasan Realisasi User Interface Design