

BAB V

IMPLEMENTASI

5.1 Batasan implementasi

5.1.1 Asumsi yang dipakai

Dalam tahap implementasi digunakan asumsi-asumsi sebagai berikut:

1. Telepon yang digunakan harus dilengkapi dengan fasilitas seting tones, biasanya pesawat telepon dengan model putar belum dilengkapi fasilitas ini.
2. Karena tidak menggunakan televisi, maka hasil akhir cukup ditampilkan pada monitor PC tanpa harus di-*broadcast*.
3. Penelitian ini hanya mengembangkan sebuah *game* pencarian jalan keluar yang dibatasi oleh waktu yang telah ditetapkan.
4. Game tersebut berbentuk multiplayer yang hanya dibatasi 2 pemain.
5. Komputer yang digunakan dalam pengembangan perangkat lunak menggunakan sistem operasi Windows XP Profesional edition.

5.1.2 Lingkungan pengembangan

Antarmuka dan desain sistem dikembangkan pada lingkungan Windows.

5.1.3 Bahasa dan kompilator

Bahasa yang digunakan dalam perancangan sistem yang berorientasi objek menggunakan bahasa UML versi 2.0. sedangkan bahasa yang digunakan dalam

membangun sistem ini adalah dengan menggunakan bahasa pemrograman visual basic dan Microsoft Visual Basic 6.0 sebagai kompilatornya.

5.1.4 Perangkat lunak yang digunakan

Untuk mewujudkan sistem yang akan dibangun dibutuhkan beberapa perangkat lunak pendukung, diantaranya :

1. Sistem operasi

Sistem operasi yang digunakan dalam menjalankan sistem ini adalah sistem operasi Windows XP Profesional edition.

2. Perancangan sistem

Perancangan sistem menggunakan UML versi 2.0 dengan menggunakan Enterprise Architect 5.0 versi trial edition sebagai aplikasi pendukung.

3. Gambar peta

Peta berupa gabungan dari gambar-gambar dengan format .bmp yang dibuat dengan menggunakan aplikasi pendukung Adobe Photoshop 7.0.

5.1.5 Perangkat keras yang digunakan

Sistem pembacaan DTMF ini dijalankan dengan menggunakan perangkat keras:

1. Processor Intel Pentium III 700 Mhz
2. Memory DDRAM 256
3. Harddisk 40 GB
4. VGA card GeForce FX5200

5. Telepon PSTN yang dilengkapi dengan nada DTMF atau telepon genggam CDMA dengan menggunakan kartu telkom flexi.
6. USB modem Prolink V.92 yang mendukung voice operation.

5.2 Implementasi

Fungsi-fungsi utama dalam sistem penerjemah sinyal DTMF menjadi kode ASCII yang kemudian dijadikan sebagai parameter penggerak objek atau karakter pada permainan adalah sebagai berikut :

1. LineCallBack

Fungsi ini berada didalam modul Globals, merupakan fungsi utama dari TAPI. TAPI mengirim seluruh call state kedalam aplikasi melalui fungsi ini. Telepon dicek terlebih dahulu apakah telepon pncrima sedang sibuk atau tidak, jika tidak sibuk, maka akan ada nada yang menandakan kita dapat melanjutkan untuk memberikan parameter penggerak objek.

```
'Mendefinisikan parameter-parameter yang digunakan
Public Function LineCallBack(ByVal dwDevice As Long, ByVal dwMessage
As Long, _
ByVal dwInstance As Long, ByVal dwParam1 As Long, ByVal dwParam2 As
Long, _
ByVal dwParam3 As Long) As Long
```

```
frmBroadCast.Print "LineCallBack = " & dwMessage
Select Case dwMessage
Case TapiEvent.LINE_ADDRESSSTATE ' =0
DebugString 4, "LINE_ADDRESSSTATE"
Case TapiEvent.LINE_CALLINFO ' =1
DebugString 3, "LINE_CALLINFO"
If dwParam1 = LINECALLINFOSTATE_CALLERID Then
```

```
' Membaca informasi caller id
```

```
DebugString 4, "LINECALLINFOSTATE_CALLERID"
GetCallerInfo dwDevice
Else
```

```

DebugString 5, "LINE_CALLINFO -> " & CStr(dwParam1)
End If
Case TapiEvent.LINE_CALLSTATE                                     '=2
DebugString 3, "LINE_CALLSTATE"
LineCallStateProc dwDevice, dwInstance, dwParam1, dwParam2, dwParam3
Case TapiEvent.LINE_CLOSE                                       '=3
DebugString 4, "LINE_CLOSE"
Case TapiEvent.LINE_CREATE                                       '=19
DebugString 4, "LINE_CREATE:"
Case TapiEvent.LINE_DEVSPECIFIC                                   '=4
DebugString 4, "LINE_DEVSPECIFIC"
Case TapiEvent.LINE_DEVSPECIFICFEATURE                         '=5
DebugString 4, "LINE_DEVSPECIFICFEATURE"
Case TapiEvent.LINE_GATHERDIGITS                               '=6
DebugString 4, "LINE_GATHERDIGITS"
Case TapiEvent.LINE_GENERATE                                    '=7

' Untuk men-generate nada tone setelah tersambung yang menandakan
' kita dapat melanjutkan untuk menggerakkan objek

DebugString 3, "LINE_GENERATE"

Case TapiEvent.LINE_LINEDEVSTATE                               '=8
DebugString 3, "LINE_LINEDEVSTATE"
LineDevStateProc dwDevice, dwInstance, dwParam1, dwParam2, dwParam3
Case TapiEvent.LINE_MONITORDIGITS                             '=9

' LineMonitorDigits dipanggil saat setiap digit terdeteksi

lTollSaver = 0 'Reset the toll saver counter

' memberikan parameter kepada objek yang akan digerakkan

Select Case LoWord(dwParam1)
Case 50
frmBroadCast.move_object 38, 0
Case 52
frmBroadCast.move_object 37, 0
Case 54
frmBroadCast.move_object 39, 0
Case 56
frmBroadCast.move__object 40, 0
End Select

```

Kode Sumber 5.1 Fungsi LincCallBack

2. LineCallStateProc

Merupakan salah satu dari fungsi yang terdapat pada modul globals, yang berfungsi untuk mengecek respon dari modem apakah diterima atau ditolak. Jika modem telah siap, maka fungsi ini memberikan pesan bahwa sudah tersambung (connected).

```
Public Sub LineCallStateProc(ByVal dwDevice As Long, ByVal
dwInstance As Long, _
ByVal dwParam1 As Long, ByVal dwParam2 As Long, ByVal dwParam3 As
Long)

Select Case dwParam1
Case LINECALLSTATE_IDLE:
DebugString 3, "LINECALLSTATE_IDLE:"
Case LINECALLSTATE_OFFERING:
DebugString 3, "LINECALLSTATE_OFFERING:"
CallIDName = "UNKNOWN"
CallIDNumber = "UNKNOWN"
hCall = dwDevice
Case LINECALLSTATE_ACCEPTED:
DebugString 4, "LINECALLSTATE_ACCEPTED:"
Case LINECALLSTATE_DIALTONE:
DebugString 5, "LINECALLSTATE_DIALTONE:"
Case LINECALLSTATE_DIALING:
DebugString 5, "LINECALLSTATE_DIALING:"
Case LINECALLSTATE_RINGBACK:
DebugString 5, "LINECALLSTATE_RINGBACK:"
Case LINECALLSTATE_BUSY:
DebugString 5, "LINECALLSTATE_BUSY:"
Case LINECALLSTATE_SPECIALINFO:
DebugString 4, "LINECALLSTATE_SPECIALINFO:"
Case LINECALLSTATE_CONNECTED:
(dec256)

' saat sudah tersambung maka muncul pesan LINECALLSTATE_CONNECTED
DebugString 4, "LINECALLSTATE_CONNECTED:"

'fInTollSaver diset menjadi false
fInTollSaver = False
frmBroadCast.Print "Player connected"

Case LINECALLSTATE_PROCEEDING:
DebugString 5, "LINECALLSTATE_PROCEEDING:"
Case LINECALLSTATE_ONHOLD:
DebugString 5, "LINECALLSTATE_ONHOLD:"
Case LINECALLSTATE_CONFERENCED:
```

```

DebugString 5, "LINECALLSTATE_CONFERENCED:"
Case LINECALLSTATE_ONHOLDPENDCONF:
DebugString 5, "LINECALLSTATE_ONHOLDPENDCONF:"
Case LINECALLSTATE_ONHOLDPENDTRANSFER:
DebugString 5, "LINECALLSTATE_ONHOLDPENDTRANSFER:"
Case LINECALLSTATE_DISCONNECTED:
DebugString 4, "LINECALLSTATE_DISCONNECTED:"

'saat fInTollSaver bernilai true, maka sambungan akan secara
'otomatis terputus

If fInTollSaver = True Then
DropCall
End If
Case LINECALLSTATE_UNKNOWN:
DebugString 4, "LINECALLSTATE_UNKNOWN:"
End Select
End Sub

```

Kode Sumber 5.2 Fungsi LineCallStateProc

3. LineDevStateProc

Fungsi ini terdapat pada modul globals yang berfungsi menerima pesan dari fungsi LineCallback.

```

Public Sub LineDevStateProc(ByVal dwDevice As Long, ByVal dwInstance
As Long, _
    ByVal dwParam1 As Long, ByVal dwParam2 As Long, _
    ByVal dwParam3 As Long)

Select Case dwParam1
Case LINEDEVSTATE_OTHER:
    DebugString 5, "LINEDEVSTATE_OTHER:"
Case LINEDEVSTATE_RINGING:
    DebugString 3, "LINEDEVSTATE_RINGING:"
    DebugString 3, "Ring Count = " & CStr(dwParam3)

'jumlah dering yang dihitung sebelum kemudian melanjutkan
'ke proses selanjutnya
If dwParam3 >= 0 Then Answer
Case LINEDEVSTATE_CONNECTED:
    DebugString 5, "LINEDEVSTATE_CONNECTED:"
Case LINEDEVSTATE_DISCONNECTED:
    DebugString 5, "LINEDEVSTATE_DISCONNECTED:"
Case LINEDEVSTATE_MSGWAITON:
    DebugString 5, "LINEDEVSTATE_MSGWAITON:"
Case LINEDEVSTATE_MSGWAITOFF:

```

```

        DebugString 5, "LINEDEVSTATE_MSGWAITOFF:"
    Case LINEDEVSTATE_INSERVICE:
        DebugString 5, "LINEDEVSTATE_INSERVICE:"
    Case LINEDEVSTATE_OUTOFSERVICE:
        DebugString 5, "LINEDEVSTATE_OUTOFSERVICE:"
    Case LINEDEVSTATE_MAINTENANCE:
        DebugString 5, "LINEDEVSTATE_MAINTENANCE:"
    Case LINEDEVSTATE_OPEN:
        DebugString 5, "LINEDEVSTATE_OPEN:"
    Case LINEDEVSTATE_CLOSE:
        DebugString 5, "LINEDEVSTATE_CLOSE:"
    Case LINEDEVSTATE_NUMCALLS:
        DebugString 5, "LINEDEVSTATE_NUMCALLS:"
    Case LINEDEVSTATE_NUMCOMPLETIONS:
        DebugString 5, "LINEDEVSTATE_NUMCOMPLETIONS:"
    Case LINEDEVSTATE_TERMINALS:
        DebugString 5, "LINEDEVSTATE_TERMINALS:"
    Case LINEDEVSTATE_ROAMMODE:
        DebugString 5, "LINEDEVSTATE_ROAMMODE:"
    Case LINEDEVSTATE_BATTERY:
        DebugString 5, "LINEDEVSTATE_BATTERY:"
    Case LINEDEVSTATE_SIGNAL:
        DebugString 5, "LINEDEVSTATE_SIGNAL:"
    Case LINEDEVSTATE_DEVSPECIFIC:
        DebugString 5, "LINEDEVSTATE_DEVSPECIFIC:"
    Case LINEDEVSTATE_REINIT:
        DebugString 5, "LINEDEVSTATE_REINIT:"
    Case LINEDEVSTATE_LOCK:
        DebugString 5, "LINEDEVSTATE_LOCK:"
    Case LINEDEVSTATE_CAPSCHANG:
        DebugString 5, "LINEDEVSTATE_CAPSCHANG:"
    Case LINEDEVSTATE_CONFIGCHANGE:
        DebugString 5, "LINEDEVSTATE_CONFIGCHANGE:"
    Case LINEDEVSTATE_TRANSLATECHANGE:
        DebugString 5, "LINEDEVSTATE_TRANSLATECHANGE:"
    Case LINEDEVSTATE_COMPLCANCEL:
        DebugString 5, "LINEDEVSTATE_COMPLCANCEL:"
    Case LINEDEVSTATE_REMOVED:
        DebugString 5, "LINEDEVSTATE_REMOVED:"
    Case Else:
        DebugString 2, "LINEDEVSTATE_UNKNOWN:"
End Select

End Sub

```

Kode Sumber 5.3 Fungsi LineDevStateProc

4. myInit

Fungsi ini terdapat pada listing form broadcast, yang berfungsi untuk menginisialisasikan fungsi-fungsi TAPI (Telephony Application Programming Interface).

```

Public Sub myInit(fSetup As Boolean)
Dim nError As Long

Dim lpExtensionID As lineextensionid
Dim lUnused As Long
Dim lLineID As Long
    lLineID = GetSetting("VB-TAPI", "Settings", "DeviceID", "0")

    If m_TapiInit = False Then
        DoEvents
        nError = lineInitialize(hTAPI, App.hInstance, _
            AddressOf LineCallBack, 0, lNumLines)
        If nError <> 0 Then
            ProcsTAPIError nError
            Err.Raise nError, "Init TAPI", "Can not initialize TAPI"
        Else
            DebugString 4, "lineInitialize -> Success"
        End If
    SetupHack:
        nError = lineNegotiateAPIVersion(hTAPI, lLineID,
TAPIVERSION, _
            TAPIVERSION, lNegVer, lpExtensionID)

        If nError <> 0 Then
            If fSetup = True Then
                lLineID = lLineID + 1
                If lLineID <= lNumLines Then GoTo SetupHack
            End If
            ProcsTAPIError nError
            Err.Raise nError, "Init TAPI",
        Else
            If fSetup = True Then Exit Sub 'Just init for
dlgSetup
                nError = lineOpen(hTAPI, lLineID, hLine, lNegVer,
lUnused, lUnused, _
                    LINECALLPRIVILEGE_OWNER,
LINEMEDIAMODE_AUTOMATEDVOICE, 0)
                If nError <> 0 Then ProcsTAPIError nError
            End If

            lpLineDevCaps.dwTotalSize = Len(lpLineDevCaps)
    
```



```

        nError = lineGetDevCaps(hTAPI, lLineID, lNegVer, lUnused,
lpLineDevCaps)
        If nError <> 0 Then ProcesSTAPIError nError

        nError = lineSetStatusMessages(hLine,
lpLineDevCaps.dwLineStates, 0)
        If nError <> 0 Then
            ProcesSTAPIError nError
            Err.Raise nError, "Init TAPI", "Can not setup for status
messages"
        End If

        m_TapiInit = True

        DebugString 3, "m_TapiInit=" & CStr(m_TapiInit)

    Else
        If hTAPI <> 0 Then
            nError = lineShutdown(hTAPI)
            If nError <> 0 Then ProcesSTAPIError nError
            hTAPI = 0
            m_TapiInit = False
        End If
    End If
End Sub

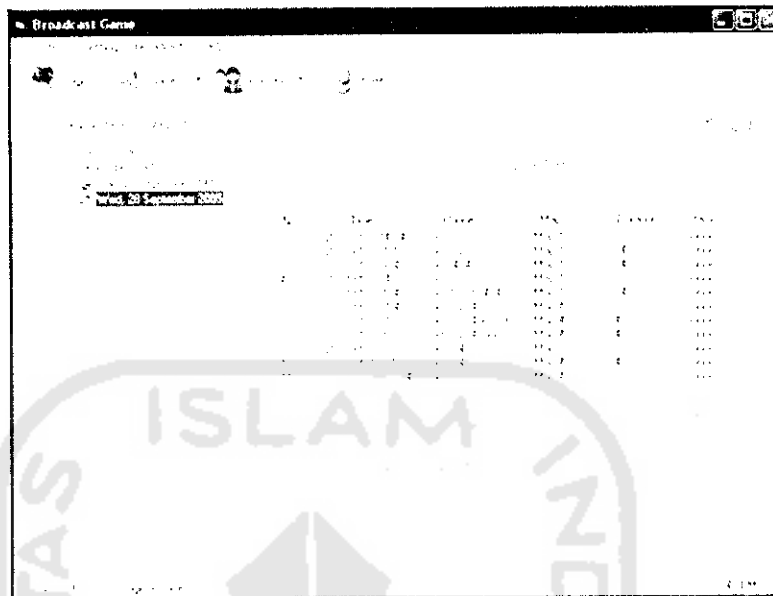
```

Kode Sumber 5.4 Fungsi mylnit

5.2.1 Implementasi antarmuka sistem

5.2.1.1 Tampilan menu utama

Berikut adalah tampilan menu utama sistem saat dijalankan. Gambar dapat dilihat pada gambar 5.1.



Gambar 5.1 Menu utama

5.2.1.2 Tampilan form login

Form login berfungsi untuk memasukkan data operator atau admin. Jika login berhasil, maka user berhak untuk mengakses sistem.








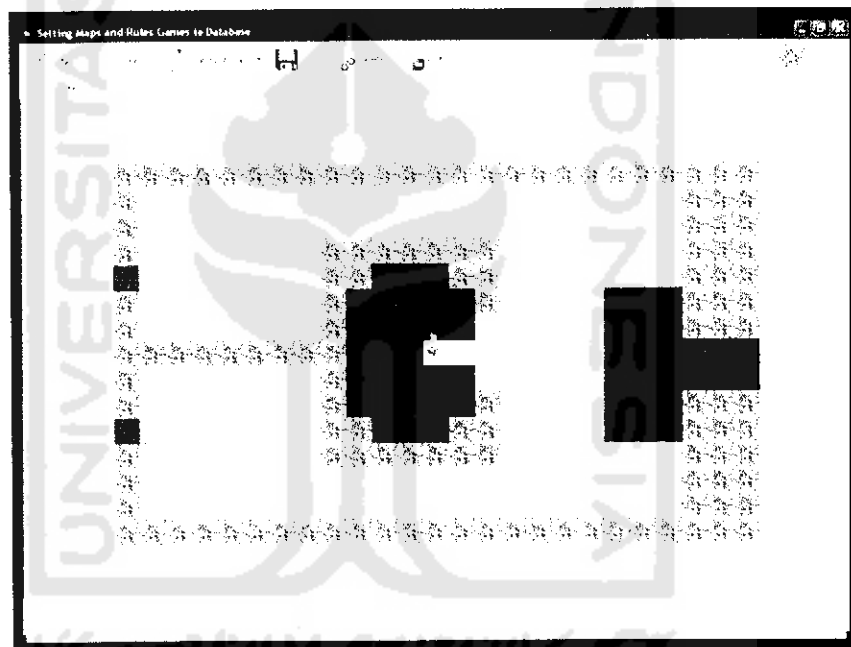
Gambar 5.2 Form login

5.2.1.3 Tampilan form seting peta

Form seting peta digunakan untuk merancang gambar peta yang diinginkan, termasuk membuat rules peta. Rules peta berbentuk file text yang ditampung pada file

tamp.txt yang kemudian disimpan didalam database. File tersebut berisikan kode-kode pada setiap objek. Kode-kode tersebut adalah sebagai berikut :

-  : Jalan diset pada angka 0  : Tujuan akhir diset angka 3
 : Tanaman diset pada angka 1  : Awal pemain diset angka 4
 : Air diset angka 2



Gambar 5.3 Form seting peta

5.2.1.4 Tampilan seting input data pemain

Form ini untuk menyimpan sejarah pemain. Data pemain juga dapat dicari dan dilihat kembali berdasarkan kriteria yang kita inginkan. Gambar dapat dilihat pada Gambar 5.4.

Gambar 5.4 Form seting input data pemain

5.2.1.5 Tampilan set peta manual

Form ini digunakan untuk memilih gambar yang akan dijadikan peta dalam permainan secara manual. Gambar yang dapat dipilih adalah gambar yang telah diset melalui form seting peta.

Gambar 5.5 Form set peta manual

5.2.1.6 Tampilan set durasi permainan

Form ini berfungsi untuk mengeset lama waktu didalam permainan. jika batasan waktu yang telah diset pada menu ini habis, maka permainan berakhir.

Gambar 5.6 Form set durasi permainan

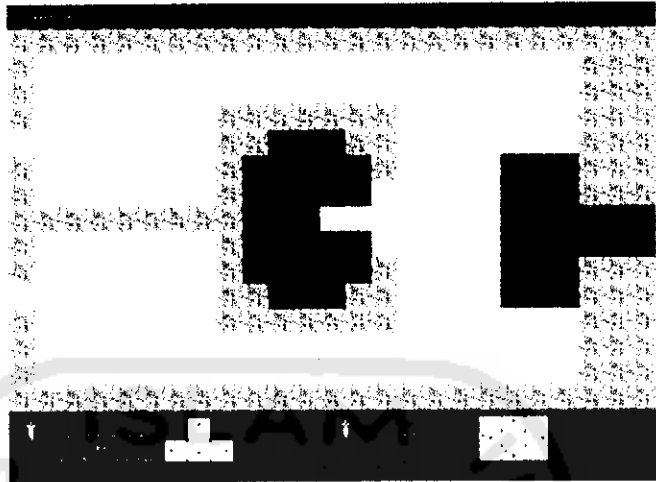
5.2.1.7 Tampilan set hadiah

Form ini digunakan untuk menginputkan informasi bentuk hadiah yang akan diberikan kepada pemenang.

Gambar 5.7 Form set hadiah

5.2.1.8 Tampilan saat game dibroadcast

Saat game telah dibroadcast dan permainan telah dimulai, maka diketahuilah bahwa objek dari player 2 dapat digerakkan dengan menggunakan keypad telepon. Berikut adalah tampilan pada saat game telah dibroadcast.



Gambar 5.8 Tampilan saat game dibroadcast

