PERCOBAAN 2

PEMROGRAMAN DASAR IVR

(UNTUK LAYANAN INBOUND)

2.1. Tujuan :

Setelah melaksanakan praktikum ini mahasiswa diharapkan mampu :

- Mengenal fungsi-fungsi *Device Input/Output* penunjang IVR
- Mengenal fungsi Pendeteksi Digit dan Play Suara
- Menyiapkan dan merekam file-file suara untuk dimainkan
- Memprogram sebuah aplikasi layanan berbasis IVR dengan Visual C++
- Mendisain sendiri program aplikasi layanan *inbound* berbasis IVR

2.2. Peralatan :

Hardware :

- PABX untuk menyediakan jalur telepon analog
- 1 pesawat Telepon untuk mengakses layanan
- 2 jalur telepon (dimana 1 jalur analog dikoneksikan ke Voice Processing Board)
- 1 PC dilengkapi dengan Dialogic Card (D/4PCI-U) untuk IVR Server

Software :

- Visual C++ ver 6.00
- Multi Thread Program (untuk perekaman suara)

2.3. Teori :

Untuk membuat sebuah sistim layanan berbasis IVR yang dapat diakses melalui jalur telepon, diperlukan persiapan hardware maupun software. Persiapan hardware meliputi penyediaan PC sebagai IVR Server yang dilengkapi dengan *Voice Processing Board*. Tentang *Voice Processing Board* sendiri sudah dijelaskan lengkap pada petunjuk praktikum sebelumnya. Selain itu perlu disiapkan pula jalur telepon analog yang akan digunakan sebagai jalur komunikasi antara server dengan client sebagai pengakses sistim layanan IVR yang disediakan.

Persiapan software meliputi dua hal, penyiapan lingkungan Telephony dan pemrograman inti untuk menjalankan proses interaksi antara Server dengan client pengakses. Penyiapan lingkungan telephony dimaksudkan agar server yang bekerja pada lingkungan computing dapat bekerja di lingkungan Telephony, yang diperlukan oleh aplikasi IVR.

Penyiapan lingkungan Telephony meliputi penggunaan fungsi-fungsi TAPI (*Telephony Application Interface Programming*) yang sudah disediakan oleh Windows OS, dan sudah di-handel oleh fungsi-fungsi yang disediakan oleh Dialogic Driver. *Dialogic Driver* sendiri meliputi 3 hal : *Voice Driver*, *Voice Library* dan *Fungsi Voice*.

Voice Driver, digunakan untuk berkomunikasi dan mengontrol *voice hardware*, dalam hal ini Dialogic Board, sebagai pemroses *voice*. Salah satu aplikasi Voice Driver adalah *Dialogic Configuration Manager (DCM)*, yaitu *tool* untuk mengaktifkan *Voice Board*, sebelum diaplikasikan pada sistim layanan berbasis IVR.

Voice Library menyediakan interface dengan Voice Driver. Voice Library yang digunakan untuk aplikasi *single-threaded* dan *multi-threaded* terdiri dari :

- *libdxxmt.lib* → *Voice Library* utama (library untuk peng-eksekusi-an program voice)
- *libsrlmt.lib* → *Standard Run-time Library* (library untuk peng-eksekusi-an program standart,tidak tergantung dari Device yang disediakan).

Library di atas harus ditambahkan pada lingkungan pemrograman yang akan dibuat, karena akan digunakan untuk :

- utilisasi seluruh voice board
- Menulis aplikasi menggunakan model pemrograman *Single-threaded Asynchronous* atau *Multi-threaded Synchronous*
- Mengkonfigurasi peralatan
- Meng-handel kejadian-kejadian yang muncul pada device
- Mengembalikan informasi device.

a. Model Pemrograman *Single-threaded Asynchronous*

Model ini memungkinkan sebuah program tunggal dapat mengontrol berbagai kanal suara dalam satu kendali. Model ini dapat digunakan untuk pengembangan aplikasi

kompleks dimana beberapa tugas dapat dikoordinir secara simultan. Model Pemrograman Asynchronous mensupport baik manajemen *polled* maupun *callback.*

b.Model Pemrograman *Multi-threaded Synchronous*

Model ini menggunakan fungsi-fungsi yang mem-blok eksekusi aplikasi sampai seluruh fungsi selesai. Pada model ini, aplikasi mengontrol masing-masing kanal dari kendali yang berbeda. Model ini memungkinkan untuk mengatur aplikasi berbeda pada kanal yang berbeda secara dinamis dan realtime.

Driver Dialogic board sudah dilengkapi dengan ***fungsi-fungsi Voice*** khusus untuk menjalankan aplikasi-aplikasi pada Dialogic board. Fungsi ini tinggal dipanggil dengan menggunakan bahasa pemrograman C++. Beberapa fungsi-fungsi khusus tersebut adalah :

Fungsi Manajemen Device

1. ***dx_open()***

   Fungsi ini digunakan untuk membuka *channel* pada card. Fungsi lain baru dapat dijalankan setelah *channel* dibuka. Command yang digunakan adalah :

   int dx_open(namep, oflags)

   Pada konfigurasi CT Bus, sebuah board meliputi device interface digital (dtiBxTx) dan device suara (dxxBxCx), yang masing-masing mempunyai kanal yang independent. B diikuti dengan nomor board, C diikuti dengan nomor kanal suara (1 s/d 4). T diikuti nomor time slot interface digital (1 s/d 24 untuk T1, dan 1 s/d 30 untuk E1).

2. ***dx_sethook()***

   Fungsi ini digunakan untuk mengontrol status kondisi *hook* dari *channel* tertentu.

   int dex_sethook(chdev, hookstate, mode)

3. ***dx_wtring()***

   Fungsi ini digunakan untuk menunggu banyaknya jumlah *ring tone* dan merubah kondisi *channel* menjadi *On Hook* atau *Off Hook.* Digunakan untuk aplikasi inbound.

   int dx_wtring(chdev, nrings, hookstate, timeout)

Fungsi Play dan Record

Fungsi Play dan Record digunakan untuk memainkan atau merekam data suara, baik dari sebuah kanal atau lebih. Jenis-jenis fungsi Play dan Record adalah sebagai berikut :

4. ***dx_playiottdata()***

   Fungsi ini digunakan untuk memainkan data suara yang sudah direkan dari beberapa sumber pada sebuah kanal. Format file yang akan dibunyikan dispesifikasikan dalam field *wFileFormat* pada DX_XPB.

   short dx_playiottdata(chdev, iottp, tptp, xpbp, mode)

   Format file WAVE yang bisa dimainkan oleh fungsi ini adalah :

   - 6, 8, dan 11KHz linear 8-bit PCM (WAVE_FORMAT_PCM)
   - 6, 8, dan 11KHz mu-law 8-bit PCM (WAVE_FORMAT_MULAW)
   - 6, 8, dan 11KHz a-law 8-bit PCM (WAVE_FORMAT_ALAW)
   - 6 dan 8KHz 4-bit Oki ADPCM (WAVE_FORMAT_DIALOGIC_OKI_ADPCM)

5. ***dx_reciottdata()***

   Fungsi ini merekam data suara ke beberapa tujuan, kombinasi file-file data, atau peralatan-peralatan penunjang.

   short dx_reciottdata(chdev, iottp, tptp, xpbp, mode)

Fungsi I/O

Fungsi I/O bertujuan untuk men-transfer data dari dan ke kanal open idle. Fungsi ini menyebabkan kanal menjadi sibuk saat proses pen-transfer-an terjadi dan kembali *idle* saat proses pen-transfer-an sudah selesai.

6. dx_getdig()

Fungsi ini mengumpulkan digit-digit dari buffer digit kanal. Setelah fungsi ini berakhir, digit yang dikumpulkan tadi ditulis dalam format ASCIIZ ke dalam buffer local, yang disusun dalam struktur DV_DIGIT.

int dx_getdig(chdev,tptp,digitp,mode)

Pemrograman inti dilakukan dengan Visual C++, karena fungsi-fungsi Voice yang disediakan oleh Dialogic sudah berbasis C++. Pemrograman inti (main program) melakukan pemanggilan fungsi-fungsi yang sudah dijelaskan di atas berdasarkan langkah-langkah yang sudah didisain sebelumnya, dan dapat diilustrasikan dalam bentuk diagram alir.

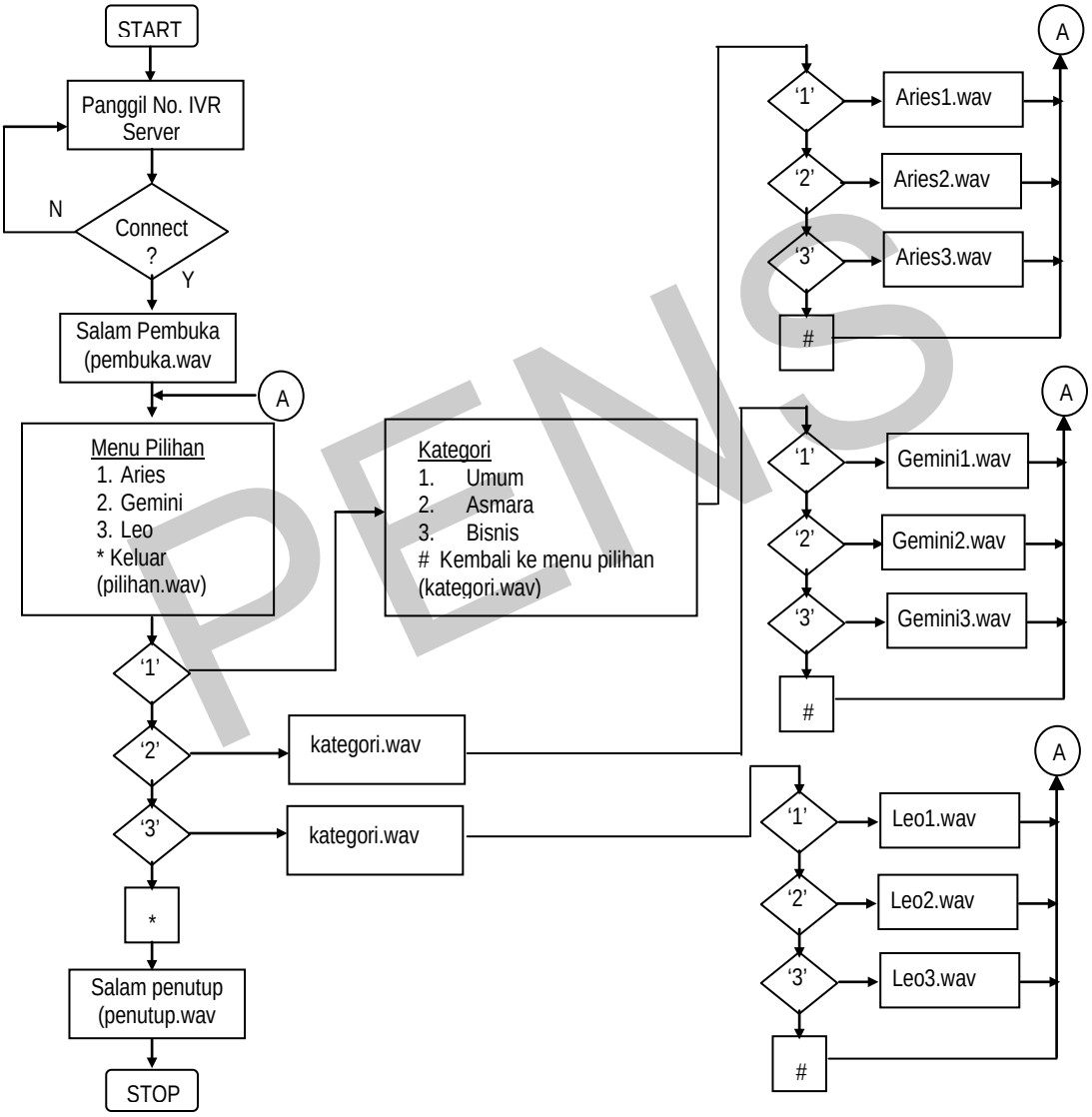
Langkah-langkah untuk menyusun pemrograman sistim layanan berbasis IVR :

1. Disain jenis layanan yang akan dibuat dan urutan pengaksesan yang akan dilakukan dalam bentuk diagram alir.
2. Aktifkan *Dialogic Configuration Manager*.
3. Siapkan dan rekam file-file WAVE yang akan dimainkan melalui pemrograman inti. Perekaman bisa menggunakan fungsi dx_recwav() atau melalui software aplikasi lain untuk perekaman WAVE. Pastikan format WAVE yang dibuat harus memenhi persyaratan yang telah ditetapkan oleh Dialogic ! (Jika tidak memenuhi syarat, file yang sudah direkam tadi tidak bisa dibunyikan melalui Dialogic Board). Untuk membantu mengingat materi yang akan direkam, buatlah dalam bentuk tulisan, yang akan dibaca sambil melakukan proses perekaman. Simpan seluruh file hasil rekaman dalam satu folder.
4. Buka Visual C++, buat Project baru (langkah selengkapnya lihat di prosedur percobaan.
5. Jalankan program yang sudah dibuat. Jangan lupa, pastikan bahwa IVR Server sudah terhubung dengan jalur telepon analog yang sudah mempunyai nomor.
6. Lakukan pengaksesan melalui pesawat telepon, dengan menghubungi nomor IVR server. Jika sudah terhubung, dengarkan urutan informasi dari file-file WAVE yang

sudah direkam sebelumnya, dengan urutan play seperti yang sudah didisain dengan diagram alir, dan lakukan penekanan digit untuk memilih informasi yang diinginkan.

2.4. Prosedur Percobaan :

Aplikasi Layanan Informasi yang akan dibuat adalah tentang Layanan Zodiak dari sebuah media penyedia jasa ramalan bintang yang bernama Dewi Fortuna. Diagram alir pengaksesan layanan ini ditunjukkan pada gambar 2.1



Gambar 2.1. Diagram Alir Sistim Layanan Zodiak

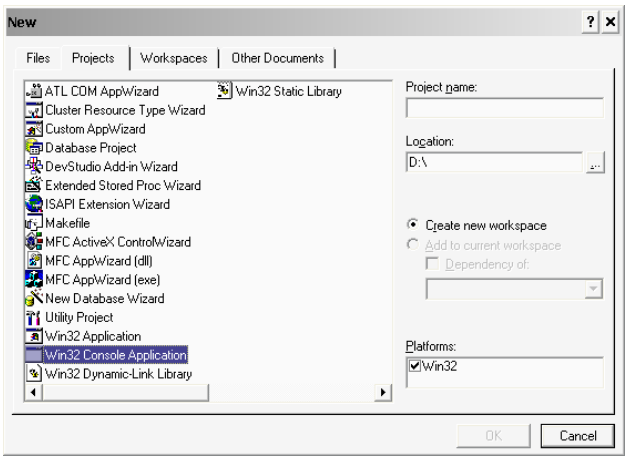
Untuk membuat Aplikasi Layanan Informasi Zodiak di atas, lakukan langkah-langkah sebagai berikut :

1. Aktifkan *Dialogic Configuration Manager*
2. Rekam file-file pada Tabel 2.1 di bawah ini dengan menggunakan *Multi Thread Mode* pada Dialogic Sample Program, dengan format perekaman WAVE Linear PCM 8 kHz. Simpan dalam sebuah folder tersendiri.

Tabel 2.1. Nama file Wav dan kalimat yang direkam

Nama File	Kalimat
pembuka.wav	Selamat Datang di Sistim Layanan Zodiak Dewi Fortuna
pilihan.wav	Untuk pilihan bintang Aries tekan 1 Untuk pilihan bintang Gemini tekan 2 Untuk pilihan bintang Leo tekan 3 Untuk keluar dari layanan tekan *
kategori.wav	Untuk kategori umum tekan 1 Untk kategori Asmara tekan 2 Untuk kategori bisnis tekan 3 Untuk kembali ke menu utama tekan #
Aries1.wav	Ada sedikit pengeluaran untuk kesehatan. Waspadalah dalam membelanjakan uang.
Aries2.wav	Tingkatkan terus hubungan dengan si dia, maju terus...
Aries3.wav	Bisnis sedang bersinar
Gemini1.wav	Bulan ini baik untuk membeli properti atau kendaraan. Kepercayaan diri di masyarakat meningkat. Bulan ini menguntungkan bagi pelajar
Gemini2.wav	Jaga selalu sikap baikmu, dia akan makin bersimpati
Gemini3.wav	Menolong orang lain dibutuhkan dalam bisnis/pekerjaan
Leo1.wav	Kesehatan butuh lebih perhatian. Nasihat sahabat berguna untuk mencari solusi masalah. Jangan berkata kasar pada siapapun
Leo2.wav	Selesainya masalah ini akan meningkatkan hubungan kalian, jalani terus...
Leo3.wav	Keuntungan bisnis menurun. Beban kerja meningkat.
penutup.wav	Terima kasih anda telah menggunakan menggunakan Sistim Layanan Zodiak Dewi Fortuna. Tetap semangat menjalani hidup anda

3. Buka Visual C++, buat Proyek baru → *File* → *New* → *Project* → pilih **Win32 Console Application,** seperti ditunjukkan pada Gambar 2.2.



Gambar 2.2. Membuat Project Baru dengan Visual C++

4. Setelah membuat project dan file *.cpp (C++ Source File), pilih *project → setting*

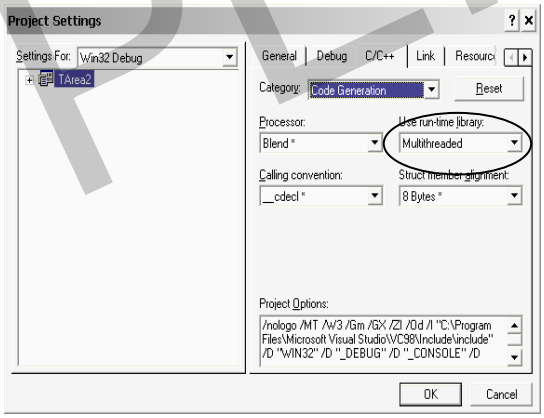
   a. Tab: C/C++

- *[Category: Code Generation]*

  Use run-time library:

  Pilih *Multithreaded*

  Hasil pengesetannya seperti pada gambar 2.3

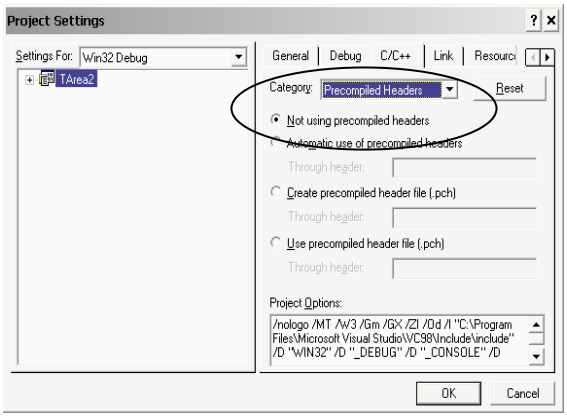


Gambar 2.3. *Setting Category Code Generation*

- *[Category: Precompiled Headers]*

  Pilih *'Not using precompiled headers'*

  Hasil pengesetannya seperti pada gambar 2.4



Gambar 2.4. *Setting Category Precompiled Headers*

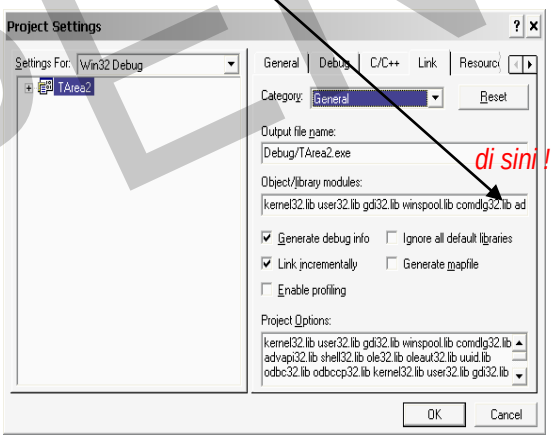
b. Tab: Link

- *[Category: General]*

  *Object/library modules: (*tambahkan*)*

  libsrlmt.lib dan **libdxxmt.lib**

  Hasil pengesetannya seperti pada gambar 2.5

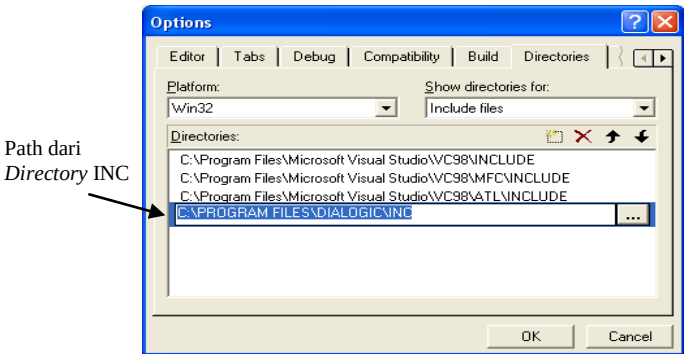


Gambar 2.5. *Setting Category General*

5. Menambahkan *Directory* INC dan LIB

Masih pada *sheet* Proyek → *Tools* → *Options* → *Directories* → *Show Directories for* : pilih *Include file* → *browse folder* dimana *Directory* INC untuk Dialogic berada, sehingga didapatkan : C:\Program Files\Dialogic\INC.

Show Directories for : pilih *Library file* → *browse folder* dimana *Directory* LIB untuk Dialogic berada, sehingga didapatkan : C:\Program Files\Dialogic\LIB → **OK**. Hal ini seperti diperlihatkan pada Gambar 8.6.



Gambar 8.6. Menambahkan *Directory* INC dari *Source* Dialogic

6. Menambah *header-header*.

Pada Project yang sudah dibuat, tambahkan header-header di bawah ini. Harus diingat, penulisan *header-header* ini harus berurutan.

```
#include <windows.h>
#include <fcntl.h>
#include <srllib.h>
#include <dxxxlib.h>
#include <stdio.h>
```

7. Membuat fungsi baru

Dua fungsi baru yang harus ditambahkan dalam Project adalah fungsi ***Deteksi Digit()*** untuk mendeteksi digit DTMF dari pesawat telepon yang ditekan oleh pengakses, serta fungsi ***Play Suara()*** untuk membunyikan suara yang sudah direkam sebelumnya

a. <u>Membuat Fungsi Deteksi Digit</u>

Buatlah fungsi DeteksiDigit(). Fungsi ini bertipe char, dimana nilai return dari fungsi ini adalah sebuah karakter yang mewakili satu digit nomor yang ditekan. Copy-kan isi fungsi DeteksiDigit() yang sudah tersedia di bawah ini ke dalam fungsi yang sudah anda buat. Hasilnya seperti ditunjukkan pada program di bawah.

```
char DeteksiDigit(int chdev,char digit[10],int x)
{
    DV_TPT tpt[3];
    DV_DIGIT digp;
```

```
    int numdigs,cnt;

    dx_clrtpt(tpt,3);
    tpt[0].tp_type=IO_CONT;
    tpt[0].tp_termno=DX_MAXDTMF;
    tpt[0].tp_length=x;
    tpt[0].tp_flags=TF_MAXDTMF;

    tpt[1].tp_type=IO_CONT;
    tpt[1].tp_termno=DX_LCOFF;
    tpt[1].tp_length=10;
    tpt[1].tp_flags=TF_LCOFF|TF_10MS;

    tpt[2].tp_type=IO_EOT;
    tpt[2].tp_termno=DX_MAXTIME;
    tpt[2].tp_length=50;
    tpt[2].tp_flags=TF_MAXTIME;

    //Get digit//
    if((numdigs=dx_getdig(chdev,tpt,&digp,EV_SYNC))==-1){
          printf("Error get digit");
          exit(1);
    }

    for(cnt=0;cnt<numdigs;cnt++){
          digit[cnt]=digp.dg_value[cnt];
    }
    return(digp.dg_value[0]);
}
```

b. Membuat Fungsi Play Suara

Langkahnya masih sama seperti cara pembuatan fungsi Deteksi Digit, namun tipe fungsi sekarang adalah *void*. Copy-kan isi fungsi PlaySuara() yang sudah tersedia di bawah ini ke dalam fungsi yang sudah anda buat. Hasilnya seperti ditunjukkan pada program di bawah.

```
void PlaySuara(int chdev,char fname[10])
{
        int fd;
        DX_IOTT iott;
        DV_TPT tpt;
        DX_XPB xpb;

        if((fd=dx_fileopen(fname,O_RDONLY|O_BINARY)) ==-1) {
        }

        tpt.tp_type             =IO_EOT;
        tpt.tp_termno       =DX_MAXDTMF;
        tpt.tp_length       =1;
        tpt.tp_flags        =TF_MAXDTMF;

        iott.io_fhandle         =fd;
        iott.io_bufp        =0;
        iott.io_offset          =0;
        iott.io_length          =-1;
        iott.io_type        =IO_DEV|IO_EOT;
```

```
        xpb.wFileFormat           =FILE_FORMAT_WAVE;
        xpb.wDataFormat           =DATA_FORMAT_DIALOGIC_ADPCM;
        xpb.nSamplesPerSec  =DRT_8KHZ;
        xpb.wBitsPerSample  =4;

        if(dx_playiottdata(chdev,&iott,&tpt,&xpb,EV_SYNC)==-1){
               printf("Error play wav file");
               exit(1);
        }
    }
```

8. Buat Prototype fungsi dibawah header-header dialogic yang sudah dibuat :

```
#include <windows.h>
#include <fcntl.h>
#include <srllib.h>
#include <dxxxlib.h>
#include <stdio.h>

char DeteksiDigit(int chdev,char digit[10],int x);
void PlaySuara(int chdev,char fname[10]);
```

Prototype fungsi

9. Membuat Main program

Main program adalah program yang berisi langkah-langkah untuk memainkan suara hasil perekaman, dan pendeteksi-an digit yang dilakukan oleh sistim IVR, sesuai dengan diagram alir yang sudah didisain sebelumnya. Sebelum menuju ke pemanggilan fungsi *PlaySuara()* dan *DeteksiDigit()*, main program perlu mendeteksi kondisi kesiapan dari Dialogic Board yang terpasang. Pendeteksian kondisi ini dilakukan dengan fungsi-fungsi : *dx_open()*, *dx-sethook()*, dan *dx_wtring()*. Main Program secara lengkap ditunjukkan di bawah :

```
void main(int argc, char* argv[])
{
   int chdev;
   char number[4], angka[4],dig[10];

mulai:
   //Open channel//
   if((chdev = dx_open("dxxxB1C1",NULL))==-1)
   {
         printf("Error open channel");
         exit(1);
   }
   printf("Open channel success\n");

   //Set on hook//
   if(dx_sethook(chdev,DX_ONHOOK,EV_SYNC)==-1)
   {
```

```
            printf("Error on hook");
            exit(1);
      }
      printf("On hook success\n");


      //Wait ring tone//
      if(dx_wtring(chdev,2,DX_OFFHOOK,-1)==-1)
      {
            printf("Error off hook");
            exit(1);
      }
      printf("Off hook success\n");


          PlaySuara(chdev,"pembuka.wav");
awal:

          PlaySuara(chdev,"pilihan.wav");
          number[0]=DeteksiDigit(chdev,dig,1);
          number[1]='\0';

    if(number[0]=='1')
      {
          PlaySuara(chdev,"kategori.wav");
          angka[0]=DeteksiDigit(chdev,dig,1);
          angka[1]='\0';
          if(angka[0]=='1')
          {
             PlaySuara(chdev,"Aries1.wav");
             goto awal;
          }
          else if(angka[0]=='2')
          {
             PlaySuara(chdev,"Aries2.wav");
             goto awal;
          }
          else if(angka[0]=='3')
          {
             PlaySuara(chdev,"Aries3.wav");
            goto awal;
          }
       else if(angka[0]=='#')
            goto awal;
      }
    else if(number[0]=='2')
      {
          PlaySuara(chdev,"kategori.wav");
          angka[0]=DeteksiDigit(chdev,dig,1);
          angka[1]='\0';
          if(angka[0]=='1')
          {
             PlaySuara(chdev,"Gemini1.wav");
             goto awal;
          }
          else if(angka[0]=='2')
          {
             PlaySuara(chdev,"Gemini2.wav");
             goto awal;
          }
          else if(angka[0]=='3')
          {
             PlaySuara(chdev,"Gemini3.wav");
```

```
            goto awal;
          }
        else if(angka[0]=='#')
           goto awal;
    }

  else if(number[0]=='3')
    {
       PlaySuara(chdev,"kategori.wav");
       angka[0]=DeteksiDigit(chdev,dig,1);
       angka[1]='\0';
       if(angka[0]=='1')
       {
           PlaySuara(chdev,"Leo1.wav");
          goto awal;
       }
       else if(angka[0]=='2')
       {
           PlaySuara(chdev,"Leo2.wav");
           goto awal;
       }
       else if(angka[0]=='3')
       {
          PlaySuara(chdev,"Leo3.wav");
          goto awal;
       }
       else if(angka[0]=='#')
          goto awal;
    }
  else if(number[0]=='*')
      PlaySuara(chdev,"penutup.wav");

    goto mulai;

}
```


2.5 Analisa :

1. Jika pada penekanan digit pilihan, dilakukan lebih dari satu kali penekanan digit, apa yang terjadi ?
2. Pada saat memilih kategori, kita memilih digit '5' (yang tidak termasuk dalam digit pilihan). Apa yang akan terjadi ?
3. Jika layanan IVR sedang berjalan, kemudian user menutup handset. Apa yang terjadi pada panggilan berikutnya ? Kapan dapat menghubungi server kembali ?
4. Jika di akhir program tidak kita sertakan *goto mulai*, apa yang terjadi ?

2.6. Pertanyaan & Tugas :

1. Buat sebuah program untuk mendeteksi digit, dengan jumlah dua digit setiap kali memasukkan digit.
2. Dengan jenis-jenis fungsi yang sudah dipelajari, buat aplikasi layanan informasi untuk sebuah agen penjualan ticket pesawat. Tentukan parameter apa saja yang bisa dijadikan layanan informasi. Buat flow chart-nya dan lengkapi dengan breakdown menu pilihan serta Tabel suara.yang diperlukan.