

Percobaan 6

Sistem Voice Mail Server

6.1. Tujuan :

Setelah melaksanakan praktikum ini mahasiswa diharapkan mampu :

- Merekam suara yang akan disimpan dalam *voice mail server* dengan menggunakan *Dialogic Card D/4 PCI-U*.
- Membuat sebuah sistem *voice mail* yang relatif mudah di akses dengan menggunakan *Dialogic Card D/4 PCI-U*.

6.2. Peralatan :

- PABX
- 2 pesawat Telepon
- 1 PC dilengkapi dengan Dialogic Card (D/41JCT-LS atau D/4PCI-U)

6.3. Teori :

6.3.1 IVR

Salah satu jenis teknologi CTI yang mendukung layanan umum adalah IVR (*Interactive Voice Response*). IVR banyak digunakan untuk sistem layanan berbasis telepon, seperti layanan tagihan PLN, tagihan telepon, minimarket, informasi nilai sampai cerita anak anak via telepon. Teknologi ini memanfaatkan database suara yang disimpan di dalam komputer. Database ini dipanggil melalui telepon, selanjutnya dimainkan sesuai dengan pilihan yang didial oleh pelanggan melalui keypad telepon.



Gambar 1. Blok Diagram Sistem IVR

6.3.2 Voice Mail Box

Layanan *voice mailbox* ini merupakan suatu layanan yang diaplikasikan untuk kondisi dimana seseorang sering *mobile*, sehingga dengan adanya layanan ini penelpon dapat meninggalkan pesan jika seseorang yang dihubungi tidak mengangkat telepon. Layanan ini bisa digunakan oleh pelanggan perumahan atau pelanggan perkantoran. Contoh aplikasi untuk pelanggan perkantoran adalah:

Jaringan telepon dibuat berada dalam line network PABX yang diasumsikan sebagai sebuah sentral pesawat telepon dalam jaringan telepon yang sebenarnya. Jaringan telepon yang digunakan juga ada dua macam yaitu jaringan telepon internal (*internal network*), yang merupakan jaringan telepon pada kantor atau instansi tempat pekerja bekerja dan jaringan telepon eksternal (*external network*), yang merupakan jaringan di luar kantor atau instansi tempat orang lain menghubungi pekerja.

Fungsi divert atau pengalihan diaktifkan oleh karyawan sebelum dia meninggalkan kantor sehingga jika ada panggilan masuk yang menghubungi karyawan, maka setelah bunyi *ring tone* akan dialihkan ke nomor *voice server* untuk mem-*play* pesan suara yang telah di tinggalkan oleh karyawan. Pada saat panggilan diterima oleh *dialogic*, maka penelpon akan diminta memasukkan nomor tujuan sehingga pesan yang ditinggalkan karyawan akan di-*play*-kan. Hal ini dapat dilakukan karena adanya proses pengecekan pada database *voice mail* yang ada pada server yang dapat mengetahui apakah ada pesan yang harus disampaikan kepada penelpon ataukah tidak.

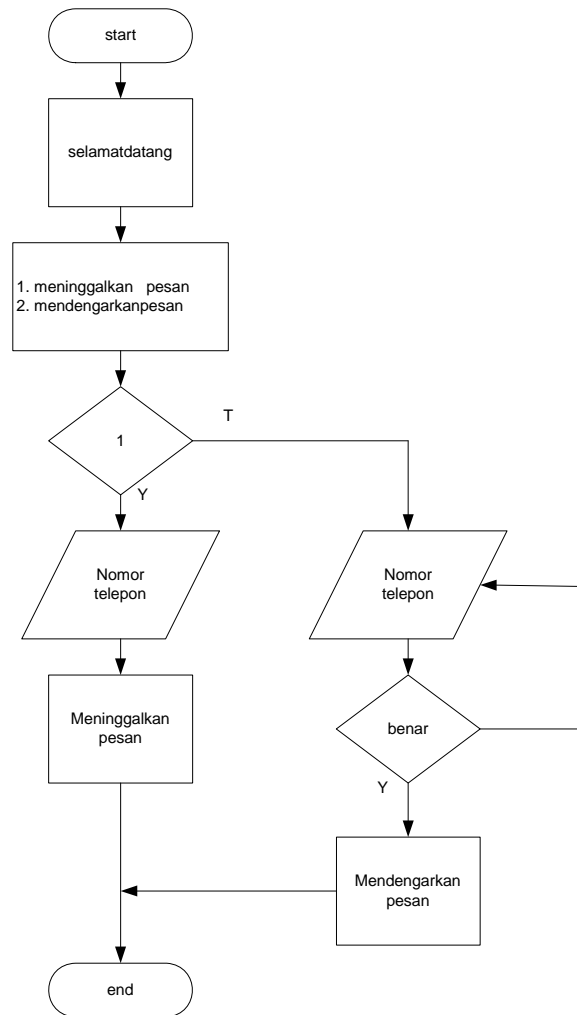
Dari sisi karyawan sendiri dapat menggunakan beberapa fitur layanan yang disediakan, dimana dari dalam jaringan kantor, karyawan dapat menghubungi nomor *voice mail* yang hanya diketahui oleh para karyawan dalam kantor sendiri sebagai nomor *intern*. Setelah tersambung, maka karyawan akan diminta memasukkan password member agar bisa mengakses *voice mail*. Jika pengecekan berhasil, maka member dapat menggunakan menu pembuatan *voice mail* baru dan penggantian password member dengan prosedur yang telah ditentukan sebagai parameter untuk menggunakan menu tersebut.

6.3.3 Aplikasi Layanan Informasi Inbound dan Outbound

Aplikasi ini merupakan integrasi dari praktikum IVR inbound dan outbound yang telah dilakukan sebelumnya. Dimana pada praktikum kali ini akan dibuat sebuah sistem

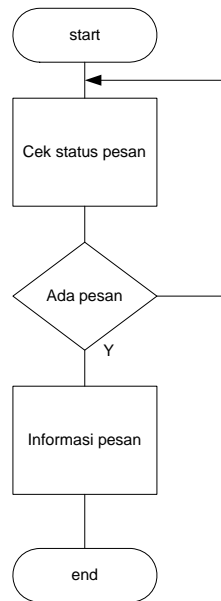
voice mail untuk meninggalkan pesan dan penerima telepon akan diberi informasi jika ada pesan yang masuk .

Diagram Alir pengaksesan voice mail ditunjukkan pada Gambar 2



Gambar 2. Diagram Alir cara pengaksesan Sistem Voice Mail yang Terkoneksi dengan Database

Diagram Alir pemberitahuan pesan ditunjukkan pada Gambar 3



Gambar 3. Diagram Alir cara pemberitahuan pesan yang Terkoneksi dengan Database

Jenis file wav yang akan dibunyikan adalah seperti pada Tabel 1

Tabel 1. Tabel Isi file wav

Nama File Wav	Kalimat
selamatdatang	Selamat datang di layanan voice mail server
pilihan pesan	Tekan 1 untuk meninggalkan pesan tekan 2 untuk mendengarkan pesan
notelp	Masukkan nomor telepon anda
tinggalkanpesan	Silahkan tinggalkan pesan
bunyitelepon	Telepon yang anda masukkan adalah
1000	seribu
satu	satu
dua	dua
tiga	tiga
empat	empat
konfirmasi	Tekan 1 jika benar tekan 2 jika salah
pesan1	Pesan 1
pesan2	Pesan 2
adapesan	Ada pesan yang masuk

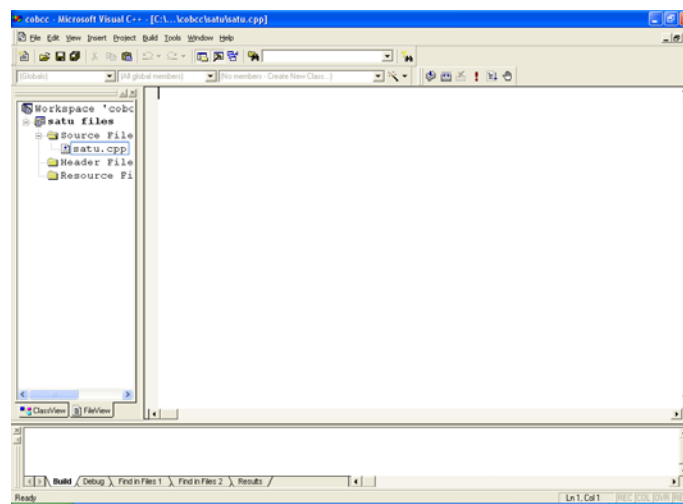
6.4. Prosedur Percobaan :

1. Aktifkan DCM(*Dialogic Configuration Manager*) sehingga dialogic dalam posisi running. Siapkan file-file wav yang akan digunakan, dengan cara merekam melalui SAMPLE PROGRAM . Isi dari file wav seperti pada Tabel 1
2. Menyiapkan aplikasi database. Dimana field-field yang disiapkan untuk database adalah :
 1. id_pesan (int)
 2. nama (varchar(20))
 3. telp (varchar(20))
 4. pesan(char(20))

Tabel 2. Field-field yang disiapkan untuk database

id_pesan	nama	telp	pesan
1	sita	1001	0
2	Rini	1002	0

3. Buka C++, buat Project baru untuk Sistim Layanan ini
 - a. Buka Visual C++ , kemudian buat workspace, project, dan file baru sehingga didapat tampilan seperti Gambar 4



Gambar 4. Membuat workspace baru dengan Visual C++

b. Setelah membuat project dan file *.cpp, pilih *project* → *setting*

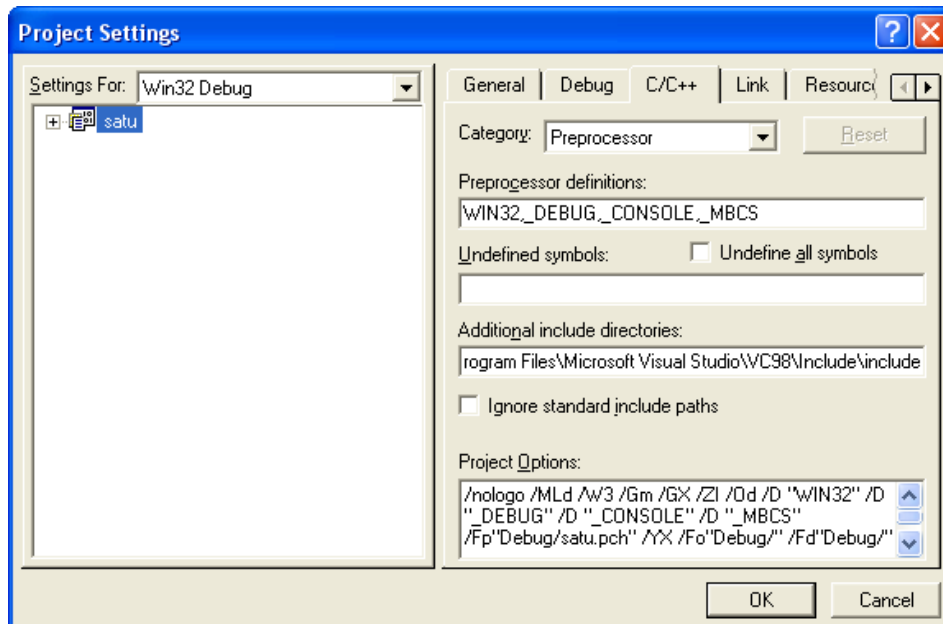
- Tab: C/C++

a. [Category Preprocessor]

Addition include directories

< path to your Mysql installation > \include

Hasil pengesetannya seperti pada gambar 5



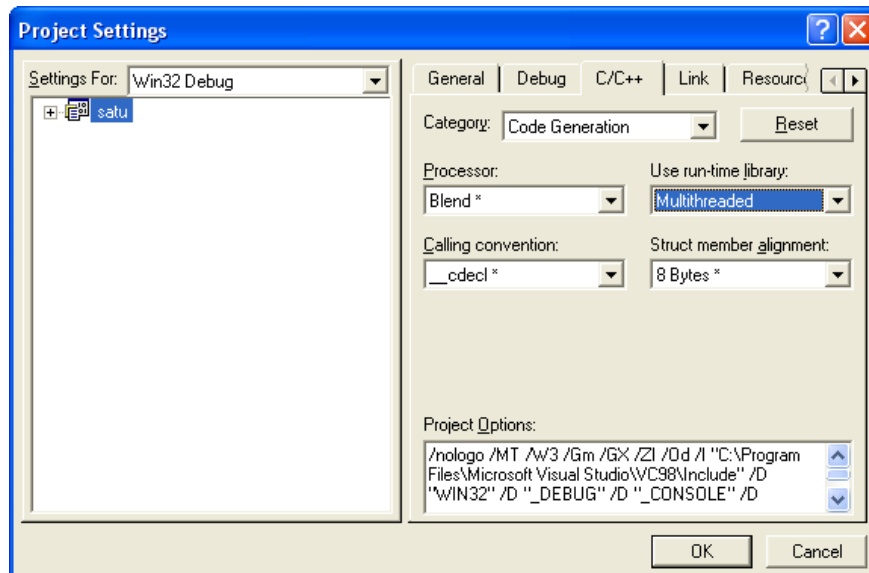
Gambar 5. Setting Category Preprocessor

b. [Category:Code Generation]

use run-time library:

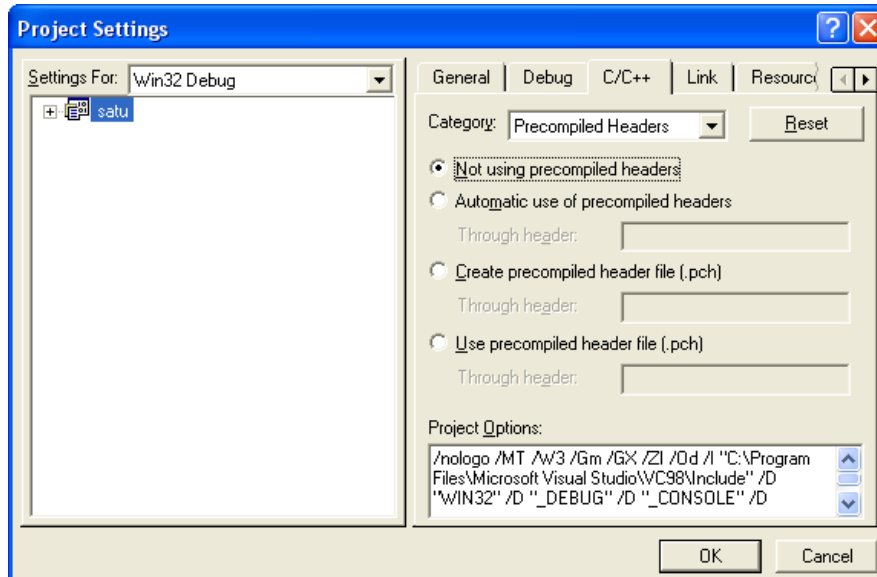
Multithreaded

Hasil pengesetannya tampak seperti gambar 6.



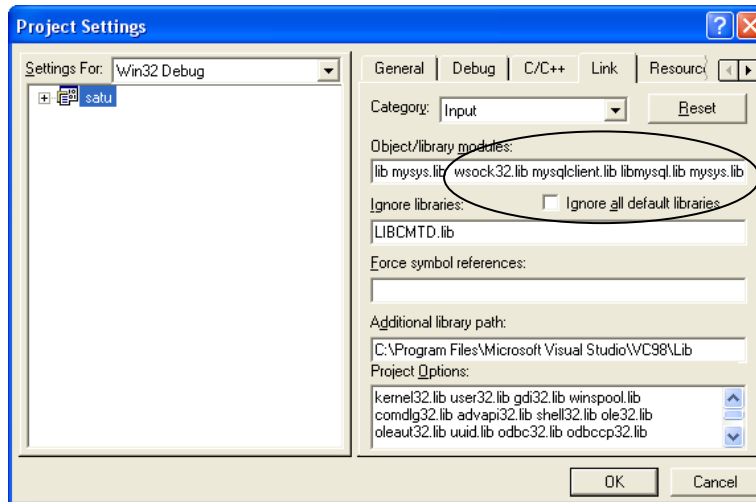
Gambar 6. Setting Category Code Generation

- c. [Category:Precompiled Headers]
 Select 'not using Precompiled Headers'
 Hasil pengesetannya tampak pada gambar 7



Gambar 7. Setting Category Precompiled Headers

- Tab:Link
 - a. [Category:Input]
 - Object/library modules: (tulis)
 wsock32.lib mysqlclient.lib libmysql.lib mysys.lib
 - Ignore libraries:
 LIBCMTD.lib
 - Additional library path:
 <path to your MySql installation>\lib\debug
- Hasil pengesetannya seperti gambar 8



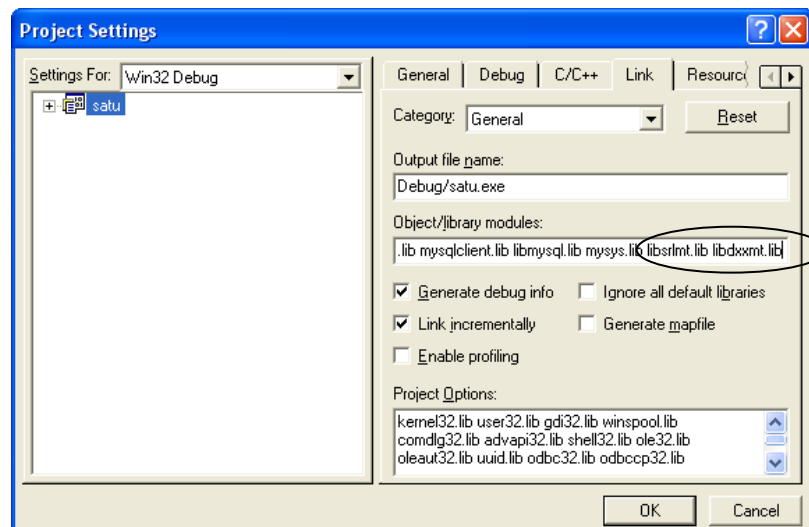
Gambar 8. Setting Category Input

b. [Category:General]

- o Object/library modules:(tuliskan)

libshlmt.lib dan libdxmt.lib

Hasil pengesetannya tampak seperti gambar 9

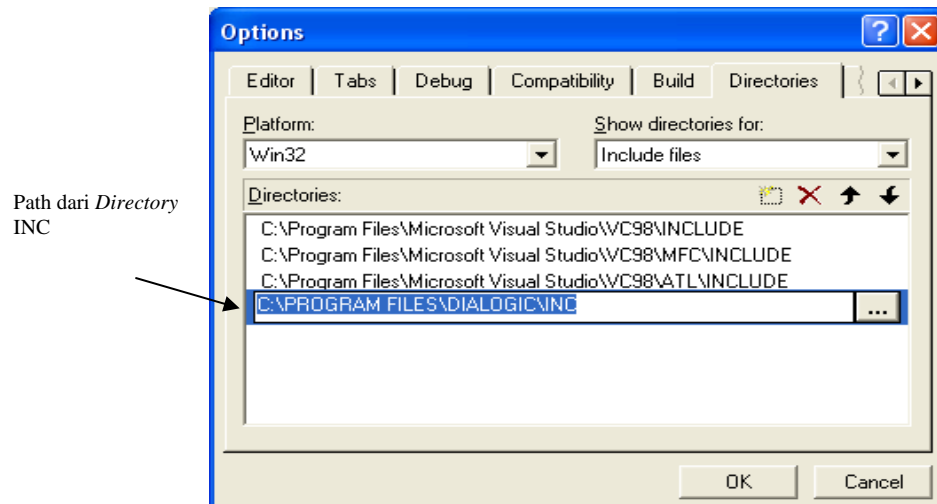


Gambar 9. Setting Category Input

- c. 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. Hal ini diperlihatkan seperti pada Gambar 11.

Show Directories for : pilih *Library file* → *browse folder* dimana *Directory* LIB untuk Dialogic berada, sehingga didapatkan : C:\Program Files\Dialogic\LIB → *Ok*



Gambar 10. Project Setting Box untuk Menambahkan File Library

- d. Menambah *header-header*. Harus diingat, penulisan *header-header* ini harus berurutan.

```
#define W32_LEAN_AND_MEAN
#include <winsock2.h>
#include "mysql.h"
#include <windows.h>
#include <fcntl.h>
#include <srllib.h>
#include <dxxlib.h>
#include <stdio.h>
#include <string.h>
#include <time.h>
```

e. Tambahkan fungsi PlaySuara.

```
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\n");
        exit(1);
    }

    printf("play wav succes\n");
}
}
```

f. Tambahkan fungsi deteksi digit

```
char DetectDigit(int chdev, char digit[10], int x)
{
    DV_TPT tpt[3];
    DV_DIGIT digp;
    int numdigs, cnt;
    dx_clrtp(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\n");
            exit(1);
        }
        printf("get digit succes\n");
        for(cnt=0;cnt<numdigs;cnt++)
        {
            digit[cnt]=digp.dg_value[cnt];
        }
        return(digp.dg_value[0]);
    }
}

```

f. Tambahkan fungsi record

```

void record(int chdev,char fname[10])
{
    int fd; /* file descriptor for file to be played */
    DX_IOTT iott; /* I/O transfer table */
    DV_TPT tpt; /* termination parameter table */
    DX_XPB xpb; /* I/O transfer parameter block */
    tpt.tp_type = IO_EOT;
    tpt.tp_termno = DX_MAXDTMF;
    tpt.tp_length = 1;
    tpt.tp_flags = TF_MAXDTMF;
    if((fd = dx_fileopen(fname,O_RDWR|O_BINARY)) == -1)
    {
        printf("File open error\n");
        exit(2);
    }
    /* clear digit buffer */
    dx_clrdigbuf(chdev);
    /* Set up DX_IOTT */
    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_MULAW;
    xpb.nSamplesPerSec = DRT_8KHZ;
    xpb.wBitsPerSample = 8;

    if (dx_reciottdata(chdev,&iott,&tpt,&xpb,PM_TONE|EV_SYNC) == -1)
    {
        printf("Error recording file - %s\n", ATDV_ERRMSGP(chdev));
        exit(4);
    }
}
}

```

g. Buat main program yang dilengkapi dengan prototype fungsi untuk aplikasi IVR

inbound!

```
#define host "localhost"
#define user "root"
#define paswd ""
#define db "mailbox"

void PlaySuara(int chdev, char fname[10]);
void record(int chdev, char fname[10]);
char DetectDigit(int chdev, char digit[10], int x);

void main(int argc, char* argv[])
{
    char number[10], dig[2];
    int chdev, no_telp;
    char query[100], terupdate[100], query1[100], terupdate1[100];

    MYSQL mysql;
    MYSQL_RES *result;
    MYSQL_ROW row;

    MYSQL *hnd=NULL;
    MYSQL *hnd1=NULL;
    MYSQL *hnd2=NULL;
    MYSQL *hnd3=NULL;

    //koneksi ke mysql

    hnd = mysql_init(NULL);

    if(!mysql_connect(hnd, host, user, paswd))
    {
        printf(mysql_error(hnd));
    }

    if(mysql_select_db(hnd, db))
    {
        printf(mysql_error(hnd));
    }

    hnd1 = mysql_init(NULL);

    if(!mysql_connect(hnd1, host, user, paswd))
    {
        printf(mysql_error(hnd1));
    }
}
```

```

if(mysql_select_db(hnd1,db))
{
    printf(mysql_error(hnd1));
}

hnd2 = mysql_init(NULL);

if(!mysql_connect(hnd2,host,user,paswd))
{
    printf(mysql_error(hnd2));
}

if(mysql_select_db(hnd2,db))
{
    printf(mysql_error(hnd2));
}

hnd3 = mysql_init(NULL);

if(!mysql_connect(hnd3,host,user,paswd))
{
    printf(mysql_error(hnd3));
}

if(mysql_select_db(hnd3,db))
{
    printf(mysql_error(hnd3));
}

//Open channel//
if((chdev = dx_open("dxxxB1C1",NULL))== -1)
{
    printf("Error open channel");
    exit(1);
}
printf("open channel succes\n"); //Set on hook//
if(dx_sethook(chdev,DX_ONHOOK,EV_SYNC)== -1)
{
    printf("Error on hook");
    exit(1);
}
printf("ONhook succes\n"); //Wait ring tone//
if(dx_wtring(chdev,2,DX_OFFHOOK,-1)== -1)
{
    printf("Error off hook");
    exit(1);
}
printf("off hook succes\n");

PlaySuara(chdev, "selamatdatang.wav");

PlaySuara(chdev, "pilihanpesan.wav");

```

```

number[0]=DetectDigit(chdev,dig,1);
number[1]='\0';

if(number[0]=='1')
{
    PlaySuara(chdev,"notelp.wav");
    number[0]=DetectDigit(chdev,dig,1);
    number[1]=DetectDigit(chdev,dig,1);
    number[2]=DetectDigit(chdev,dig,1);
    number[3]=DetectDigit(chdev,dig,1);
    number[4]='\0';

    no_telp=atoi(number);

    PlaySuara(chdev,"tinggalkanpesan.wav");

    sprintf(terupdate,"update message set pesan=%s where
telp=%d", "1",no_telp);

    mysql_query(hndl,terupdate);
    result=mysql_use_result(hndl);
    mysql_close(hndl);

    sprintf(query,"select telp from message where
telp=%d",no_telp);
    mysql_query(hnd,query);
    result=mysql_use_result(hnd);

    if(row=mysql_fetch_row(result))
    {

        if(strcmp(row[0],"1003")==0)
        {
            record(chdev,"pesan1.wav");
        }

        else if(strcmp(row[0],"1002")==0)
        {
            record(chdev,"pesan2.wav");
        }

    }
    mysql_close(hnd);
}

else if(number[0]=='2')
{
    balik:

    PlaySuara(chdev,"notelp.wav");
    number[0]=DetectDigit(chdev,dig,1);
    number[1]=DetectDigit(chdev,dig,1);

```

```

number[2]=DetectDigit(chdev,dig,1);
number[3]=DetectDigit(chdev,dig,1);
number[4]='\0';

no_telp=atoi(number);

PlaySuara(chdev,"bunyi telepon.wav");
if(number[0]=='1')
{
    PlaySuara(chdev,"1000.wav");
    if(number[3]=='1')
    PlaySuara(chdev,"satu.wav");
    else if(number[3]=='2')
    PlaySuara(chdev,"dua.wav");
    else if(number[3]=='3')
    PlaySuara(chdev,"tiga.wav");
    else if(number[3]=='4')
    PlaySuara(chdev,"empat.wav");
}

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

if(number[0]=='1')
{
    sprintf(terupdate1,"update message set pesan=%s
where telp=%d", "0",no_telp);

    mysql_query(hnd2,terupdate1);
    result=mysql_use_result(hnd2);
    mysql_close(hnd2);

    sprintf(query1,"select telp from message where
telp=%d",no_telp);
    mysql_query(hnd3,query1);
    result=mysql_use_result(hnd3);

    if(row=mysql_fetch_row(result))
    {
        if(strcmp(row[0],"1003")==0)
        {
            PlaySuara(chdev,"pesan1.wav");
        }

        else if(strcmp(row[0],"1002")==0)
        {
            PlaySuara(chdev,"pesan2.wav");
        }

    }

    mysql_close(hnd3);
}

```



```

        }
        else if(number[0]=='2')
        {
            goto balik;
        }
    }
}

```

- h. Buat main program yang dilengkapi dengan prototype fungsi untuk aplikasi IVR outbound!

```

#define host "localhost"
#define user "root"
#define paswd ""
#define db "mailbox"

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

void main(int argc, char* argv[])
{
    char number[10],dig[2],dialstrg[5];
    int chdev,no_telp;
    int cares;
    char query[100];

    MYSQL mysql;
    MYSQL_RES *result;
    MYSQL_ROW row;

    MYSQL *hnd4=NULL;
    hnd4 = mysql_init(NULL);

    if(!mysql_connect(hnd4,host,user,paswd))
    {
        printf(mysql_error(hnd4));
    }

    if(mysql_select_db(hnd4,db))
    {
        printf(mysql_error(hnd4));
    }

    balik:
    printf("cari pesan");

    sprintf(query,"select telp from message where pesan=%d",1);
    mysql_query(hnd4,query);
    result=mysql_use_result(hnd4);

    if(row=mysql_fetch_row(result))
    {
        strcpy(dialstrg,row[0]);
    }
}

```

```

printf("%s\n",dialstrg);

    if((chdev = dx_open("dxxxB1C1",NULL))== -1)
    {
        printf("Error open channel");
        exit(1);
    }
    printf("open channel succes\n");
awal:

    if(dx_sethook(chdev,DX_OFFHOOK,EV_SYNC)==-1)
    {
        printf("Error off hook");
        exit(1);
    }
    printf("off hook succes\n");

    dx_clrcap(&capp);
    capp.ca_nbrdna=4;
    if((cares=dx_dial(chdev,dialstrg,&capp,DX_CALLP|EV_SYNC))
== -1 )
    {
        printf("error dial");
        exit(0);
    }

    printf("proses dial");

    switch(cares)
    {
        case CR_CNCT:
            printf("answer\n");
            PlaySuara(chdev,"adapesan.wav");
        case CR_BUSY:
            printf("Busy");

            if(dx_sethook(chdev,DX_ONHOOK,EV_SYNC)==-
1)
            {
                printf("Error on hook\n");
                exit(1);
            }
            printf("on hook succes\n");
            printf("\ntunggu 10 detik\n");
            Sleep(10000);
            goto awal;
            break;
        case CR_NOANS:
            printf("no answer");
            if(dx_sethook(chdev,DX_ONHOOK,EV_SYNC)==-
1)
            {
                printf("Error on hook\n");
                exit(1);
            }
            printf("on hook succes\n");

```

```

        printf("\ntunggu 10 detik\n");
        Sleep(10000);
        goto awal;
        break;
    default :
        printf("else");
        printf("\ntunggu 10 detik\n");
        Sleep(10000);
        goto awal;
        break;
    }

}
else
    goto balik;

mysql_close(hnd4);
}

```

Analisa:

1. Hitunglah berapa rata-rata waktu eksekusi yang dibutuhkan untuk aplikasi IVR inbound, lakukan pengujian sebanyak 10 kali!
2. Hitunglah berapa rata-rata waktu eksekusi yang dibutuhkan untuk aplikasi IVR outbound, lakukan pengujian sebanyak 10 kali!

6.5. Pertanyaan & Tugas :

Buatlah aplikasi yang menggabungkan IVR inbound dan outbound. Lengkapi dengan flow chart pengaksesan serta breakdown menu pilihan.