

## PERCOBAAN 4

### PEMROGRAMAN IVR OUTBOUND

#### 4.1. Tujuan :

Setelah melaksanakan praktikum ini mahasiswa diharapkan mampu :

- Mengerti prinsip kerja sistim reminder berbasis IVR
- Membuat program reminder dengan C++
- Membuat database untuk pendukung layanan IVR menggunakan MySQL
- Membuat koneksi antara pemrograman Database dengan C++
- Mengerti prinsip sinkronisasi waktu antara waktu server dengan waktu reminder

#### 4.2. Peralatan :

##### Hardware :

- PABX sebagai penyedia jalur analog
- 2 pesawat Telepon terminal
- 1 PC dilengkapi dengan Dialogic Card (D/4PCI-U)

##### Software :

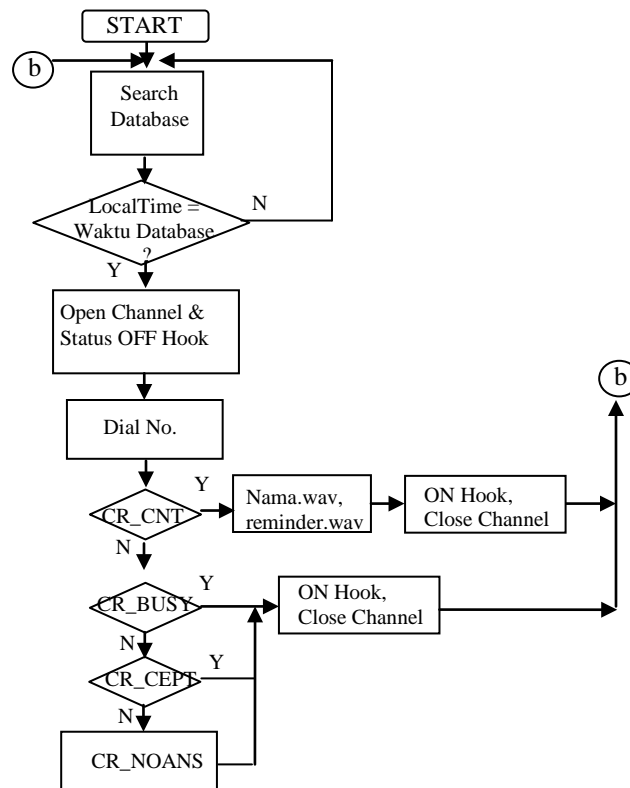
- Visual C++
- *Xampp* yang mempunyai aplikasi MySQL
- *MultiThread Sample Program*, untuk perekaman suara  
(boleh menggunakan software perekaman suara yang lain, asal mempunyai format suara manusia, bukan musik)

#### 4.3. Teori :

Sistim *Outbound* adalah aplikasi layanan IVR dimana server bersifat aktif, dengan mengirimkan rekaman suara kepada user tertentu. Biasanya aplikasi layanan ini berbentuk *reminder*, yang mengingatkan user akan adanya sebuah event yang harus dihadiri atau pengingat janji mengadakan sebuah kegiatan tertentu. *Reminder* akan dikirim ke user pada saat tertentu, dimana waktu pengiriman sudah diatur terlebih dahulu dan disimpan bersama-sama dengan identitas user pada database pelanggan.

Cara kerja sistim *Reminder* adalah sebagai berikut : Server selalu mencocokkan jam yang sudah disimpan di database dengan jam pada sistim operasi server. Saat jam pada sistim operasi server menunjukkan nilai yang sama dengan jam yang disimpan di database, server akan membaca data nomor telepon pada baris yang sama dengan baris jam di tabel pelanggan. Selanjutnya, server melakukan dial ke nomor telepon yang didapat. Apabila jalur dari nomor yang di-dial sedang tidak sibuk dan user melakukan off-hook, maka user akan mendengar suara rekaman yang berisi informasi pengingat. Namun jika jalur dari nomor yang di-dial sedang sibuk, atau user tidak segera off-hook, maka pada perioda waktu tertentu server akan melakukan dial kembali ke nomor tersebut, hingga beberapa kali. Jika sampai beberapa kali panggilan tidak diangkat, tidak dilakukan panggilan lagi, dianggap pasien tidak ditempat. Flow chart dari sistim outbound IVR ditunjukkan pada gambar 4.1

Aplikasi *outbound* ini juga dapat dikombinasi dengan aplikasi *inbound* dalam hal pemilihan menu. Sebagai contoh, pada sebuah aplikasi reminder klinik, server akan melakukan dial reminder pada seorang pasien jika pasien tersebut berkeinginan untuk diingatkan kembali saat harus menjalani kontrol perawatan. Dial dilakukan satu hari sebelum jadwal kontrol, misal jam 07.00 pagi. Apabila setelah mendengar nada panggil, pasien segera mengangkat handset, maka dia akan mendengar reminder tentang jadwal kontrol yang sudah disepakati. Pasien dapat menyetujui, membatalkan atau mengubah jadwal dengan menekan tombol-tombol keypad telepon sesuai dengan petunjuk suara rekaman. Hasil dari pemilihan pasien ini akan digunakan untuk meng-update database jadwal kontrol pasien. Apabila panggilan tidak segera diangkat, server akan mengulang panggilan sampai 3 kali pada hari tersebut (periode panggilan bisa diatur). Jika sampai tiga kali panggilan tidak diangkat, dianggap pasien tidak di rumah dan jadwal kontrol dianggap disetujui pasien.



Gambar 4.1. Flow Chart IVR Outbound

### **Fungsi Dial**

Sebagai sebuah sistem reminder, IVR Outbound ini dilengkapi dengan fungsi pemanggil yang sudah disediakan oleh Dialogic (disertakan saat kita menginstall driver Dialogic). Fungsi ini diaktifkan saat waktu sudah menunjukkan saatnya reminder dibunyikan. Syntax *dx\_dial* menjadi penggerak utama fungsi ini.

Cara kerja fungsi tersebut sebagai berikut : saat waktu sudah sesuai dengan setting di database, server membaca kolom nomor telepon, kemudian men-set off-hook dirinya dan melakukan call kepada nomor tersebut dengan syntax *dx\_dial*. Jika nomor tujuan on-hook, maka reminder akan dibunyikan. Namun jika nomor tujuan sedang sibuk, tidak ada jawaban, tidak ada *ringback tone* dan tidak ada *intercept*, maka akan kanal ditutup kembali. Program selengkapnya ditunjukkan seperti di bawah.

```

void DialNotelp(int chdev, char notelp[5])
{
dx_clrcap(&capp);
capp.ca_nbrdna=6; //bunyikan ringtone 6 kali
if((cares=dx_dial(chdev,notelp,&capp,DX_CALLP|EV_SYNC)) == -1 )
{
printf("error dial");
exit(0);
}

switch(cares)
{
case CR_CNCT:
printf("dial");
goto lanjut; //fungsi memainkan reminder
case CR_BUSY:
printf("Busy");
hook(chdev,0);
dx_close (chdev);
al=1;
break;
case CR_CEPT:
printf("intercept");
hook(chdev,0);
dx_close (chdev);
al=1;
break;
case CR_NOANS:
printf("no answer");
hook(chdev,0);
dx_close (chdev);
al=1;
break;
case CR_NODIALTONE:
printf("no dial tone");
hook(chdev,0);
dx_close (chdev);
al=1;
break;
case CR_NORB:
printf("no ringback");
hook(chdev,0);
dx_close (chdev);
al=1;
break;
default :
printf("else");
hook(chdev,0);
dx_close (chdev);
al=1;
break;
}
}

```

Langkah-langkah yang diperlukan untuk menyiapkan aplikasi outbound IVR adalah :

1. Membuat Tabel pada Database
2. Men-set lingkungan C++ menjadi lingkungan Teleponi
3. Membuat pemrograman lengkap dengan C++

4. Me-run program dan menginformasikan layanan melalui pesawat telepon

#### 4.4. Prosedur Percobaan :

Layanan yang akan dibuat adalah reminder untuk anggota sebuah perkumpulan 'Whatever You Want'. Perkumpulan ini memuat orang-orang yang ingin diingatkan jika ada event tertentu yang harus dia jalani. Untuk bisa diberi reminder, orang-orang tersebut harus mendaftar dulu ke admin.

Anggap anda adalah admin, yang menginputkan apa saja keperluan para anggota untk diingatkan. Langkah-langkah membuat program reminder ini adalah sebagai berikut :

1. Buat Tabel yang berisi informasi tentang member, hari,jam dan menit pengingat serta nomor telepon yang akan dihubungi. Gunakan MySQL (langkah-langkah membuat database dan tabel dengan MySQL sudah dibahas pada praktikum sebelumnya). Beri nama 'reminder' pada tabel yang dibuat, dan nama database 'outbound'. Struktur tabel seperti ditunjukkan pada gambar 4.2.

	Field	Type	Collation	Attributes	Null	Default	Extra
<input type="checkbox"/>	<b>Id</b>	int(11)			No		
<input type="checkbox"/>	<b>Nama</b>	varchar(40)	latin1_general_ci		No		
<input type="checkbox"/>	<b>hari</b>	char(2)	latin1_general_ci		No		
<input type="checkbox"/>	<b>jam</b>	char(2)	latin1_general_ci		No		
<input type="checkbox"/>	<b>menit</b>	char(2)	latin1_general_ci		No		
<input type="checkbox"/>	<b>notelp</b>	varchar(5)	latin1_general_ci		No		

Gambar 4.2. Stuktur Tabel 'reminder' pada database 'outbound'

2. Isi Tabel tersebut dengan informasi seperti ditunjukkan pada gambar 4.3

			<b>Id</b>	<b>Nama</b>	<b>hari</b>	<b>jam</b>	<b>menit</b>	<b>notelp</b>
<input type="checkbox"/>			1	Karina.wav	3	10	27	1001
<input type="checkbox"/>			2	Aditya.wav	3	9	37	2001

Gambar 4.3. Isi Tabel 'reminder'

3. Siapkan file-file rekaman dengan isi seperti ditunjukkan pada Tabel 4.1. (Cara merekam file suara sudah djelaskan pada praktikum sebelumnya).

Tabel 4.1. Isi dari file rekaman

Nama file	Kalimat
Karina.wav	karina
Aditya.wav	Aditya
siang.wav	selamat siang
remind1.wav	waktunya latihan basket nih
remind2.wav	waktunya makan siang dik
terimakasih.wav	terima kasih ...

#### 4. Bekerja dengan Visual C++

- a. Buat Proyek baru → *File* → *New* → *Project* → pilih **Win32 Console Application**, lalu buat file .cpp baru dengan memilih *File* → *New* → *C++ Source File*, letakkan pada project yang sudah dibuat sebelumnya → OK
- b. pilih *project* → *setting*

*Tab: C/C++*

- [*Category: Preprocessor*]

*Additional include Directories*

<cari letak folder Include MySQL pada folder Include C++>

Pastikan bahwa folder include dari MySQL sudah di-copy kan ke dalam folder include VC98 milik Visual C++ yang ada di Directory Program Files.

- [*Category: Code Generation*]
- Use run-time library: Multithreaded*
- [*Category: Precompiled Headers*]
- Pilih '*Not using precompiled headers*'

*Tab: Link*

- [*Category: Input*]
- Object/library modules: (tambahkan)*  
**wsock32.lib mysqlclient.lib libmysql.lib msys.lib**
- Ignore libraries:*  
(tuliskan) *LIBCMTD.lib*

*Additional library path:*

<cari letak folder lib MySQL pada folder lib C++>

Pastikan bahwa folder lib dari MySQL sudah di-copy kan ke dalam folder lib VC98 milik Visual C++ yang ada di Directory Program Files.

- [ Category: General ]

*Object/library modules:* (tambahkan)

**libsrmt.lib libdxxmt.lib**

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.

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

d. Menambah *header-header*.

```
#define W32_LEAN_AND_MEAN
#include <winsock2.h>
#include <include\mysql.h>
#include <windows.h>
#include <string.h>
#include <fcntl.h>
#include <srllib.h>
#include <dxxplib.h>
#include <stdio.h>
```

e. Mendefinisikan lokasi Tabel dalam Database

```
#define TABLE_OF_INTEREST "reminder"
#define SERVER_NAME "localhost"
#define DB_USER "root"
#define DB_USERPASS ""
#define DB_base "outbound"
```

f. Menambahkan *prototype* fungsi-fungsi yang diperlukan.

```
//prototypes
void hook(int chdev,int hook);
void playsuara(int chdev,char fname[10]);
```

#### g. Mendefinisikan variabel public

```
//definisi variabel public
MYSQL *hnd=NULL; // mysql connection handle
const char *sinf=NULL; // mysql server information
MYSQL_RES *res=NULL; // result of querying for all rows in table
MYSQL_ROW row; // one row returned
char sql[1024]; // sql statement used to get all rows

DWORD jm,min;
SYSTEMTIME lt;
int chdev,s,count,al,cares;
DX_CAP capp;
```

#### h. Membuat fungsi-fungsi dan main program.

```
//Fungsi sethook
void hook(int chdev,int hook)
{
    if(hook == 1)
    {
        if(dx_sethook(chdev,DX_OFFHOOK,EV_SYNC) == -1)
        {
            printf("error off hook\n");
            exit(1);
        }
        printf("off hook success\n");
    }
    else if(hook == 0)
    {
        if(dx_sethook(chdev,DX_ONHOOK,EV_SYNC) == -1)
        {
            printf("error onhook\n");
            exit(1);
        }
        printf("on hook success\n");
    }
}

//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 File Success\n");
}

//Main Program
void main(int argc, char* argv[])
{
    SYSTEMTIME lt;
    GetLocalTime(&lt);
    hari=lt.wDayOfWeek;

    fprintf(stdout, "\t\t\tPROGRAM REMINDER\n\n\n");
    fprintf(stdout, "KONEKSI DATABASE STATUS : ");

atas:
    hnd = mysql_init(NULL);

    while(1)
    {
        GetLocalTime(&lt);

        if (NULL ==
mysql_real_connect(hnd,SERVER_NAME,DB_USER,DB_USERPASS,DB_b
ase,0,NULL,0))
        {
            fprintf(stderr, "Terdapat masalah pada %s database
user %s.\n",DB_base,SERVER_NAME);
        }
        else
        {
            fprintf(stdout, "Koneksi ke %s database ke server %s dg
user '%s'.\n",DB_base,SERVER_NAME,DB_USER);
            sinf = mysql_get_server_info(hnd);
            if (sinf != NULL)
            {
                fprintf(stdout, "\n\nMendapatkan informasi:
'%s'\n",sinf);

                //Query and show
                while(1)
                {

                    DWORD hr, jm, min;
                    GetLocalTime(&lt);
                    hr=lt.wDayOfWeek;
                    jm=lt.wHour;
                    min=lt.wMinute;
                    printf("cari hari, jam dan menit
= %d %d:%d:%d\n",hr, jm, min, lt.wSecond);
                }
            }
        }
    }
}

```

```

sprintf(sql,"select * from %s where hari=%d and
jam=%d and
menit=%d",TABLE_OF_INTEREST,hr,jm,min);
fprintf(stdout,"Using sql statement: '%s' to
extract all rows from the specified
table.\n",sql);

if (!mysql_query(hnd,sql))
{
res = mysql_use_result(hnd);
if (res)
{
row = mysql_fetch_row(res);
fprintf(stdout,"Rows returned:\n\n");

while(row)
{
senin:
s=1;

if((chdev = dx_open("dxxxBlC2",NULL))== -1)
{
printf("Error open channel\n");
exit(1);
}
printf("Open channel success\n");
hook(chdev,1);
dx_clrcap(&capp);
capp.ca_nbrdna=6;

if((cares=dx_dial(chdev,row[5],&capp,DX_CALLP
|EV_SYNC)) == -1 )
{
printf("error dial");
exit(0);
}
goto bawah;
//row = mysql_fetch_row(res);
}

}
Sleep(1000);
}

printf("\nEnd of rows\n");
mysql_free_result(res);
Sleep(1000);
}
}

bawah:
switch(cares)
{
case CR_CNCT:
printf("dial nomor =>%s",row[5]);
goto lanjut;

```

```

case CR_BUSY:
    printf("Busy\n");
    hook(chdev,0);
    dx_close (chdev);
    al=1;
    break;
case CR_CEPT:
    printf("intercept\n");
    hook(chdev,0);
    dx_close (chdev);
    al=1;
    break;
case CR_NOANS:
    printf("no answer\n");
    hook(chdev,0);
    dx_close (chdev);
    al=1;
    break;
case CR_NODIALTONE:
    printf("no dial tone\n");
    hook(chdev,0);
    dx_close (chdev);
    al=1;
    break;
case CR_NORB:
    printf("no ringback\n");
    hook(chdev,0);
    dx_close (chdev);
    al=1;
    break;
default :
    printf("else\n");
    hook(chdev,0);
    dx_close (chdev);
    al=1;
    break;
}

if(al==1)
{
    count++;
    if(count>3)
    {
        goto atas;
    }
    else if(s==1)
    {
        Sleep(5000);
        goto senin;
    }
}

lanjut:

playsuara(chdev,"siang.wav");
playsuara(chdev,row[1]);
playsuara(chdev,"remind2.wav");
playsuara(chdev,"terimakasih.wav");
hook(chdev,0);

```

```

        dx_close (chdev);
        mysql_close(hnd);
        Sleep(60000);
        goto atas;
    }
} //end of while(1)
}

```

#### 4.5. Analisa :

1. Jika program sudah *free error*, pastikan bahwa tidak ada kesalahan dalam membunyikan file rekaman wav. Periksa juga kanal yang dibuka, yang akan anda pakai untuk melakukan komunikasi layanan reminder.
2. Ubah-ubah jam dan menit untuk bisa menyesuaikan dengan waktu yang ingin digunakan untuk membunyikan reminder. Dengan dua nama user yang tersedia dan dua nomor tujuan yang berbeda, pemanggilan antara kedua user tersebut memerlukan waktu tunda sekian detik. Mengapa panggilan tidak bisa dilakukan dalam waktu yang bersama-sama ?
3. Apa gunanya syntax *sleep* pada program di atas ? Coba ubah dua buah nilai `sleep(1000)`, menjadi `sleep(500)`. Apa yang terjadi ?  
Ubah juga nilai `sleep(60000)` menjadi `sleep(40000)`. Apa yang terjadi pada program di atas ? Hitung delay yang diperlukan antara pemanggilan user pertama dengan kedua.

#### Petunjuk singkat :

Program yang dibuat masih bersifat *Single Thread*, namun sudah disiapkan menjadi *Multi Thread* dengan pengesetan pada Category : *Code Generation* pada Tab C/C++.

#### 4.6. Pertanyaan & Tugas :

Buat sebuah reminder untuk guru SMA jurusan IPA (ekstensi 1000) dan IPS (ekstensi 2000), dimana jadwal UAS hari ke 1 adalah Bahasa Indonesia, hari ke 2 Bahasa Inggris, hari ke 3 Matematika (untuk IPA dan IPS), hari ke 4 Ekonomi dan Sejarah(IPS) dan Kimia dan Fisika (IPA), hari ke 5 Akuntansi (IPS) dan TIK (IPA). Jalankan pada 2 kanal berbeda, dan buatlah 2 Tabel berbeda masing-masing untuk IPA dan IPS dalam Database yang sama.