

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++
- Melakukan pemrograman Device Input/Output (*open channel, getdigit, wait ring*)
- Membuat sebuah aplikasi IVR outbound

4.2. Peralatan :

Hardware :

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

Software :

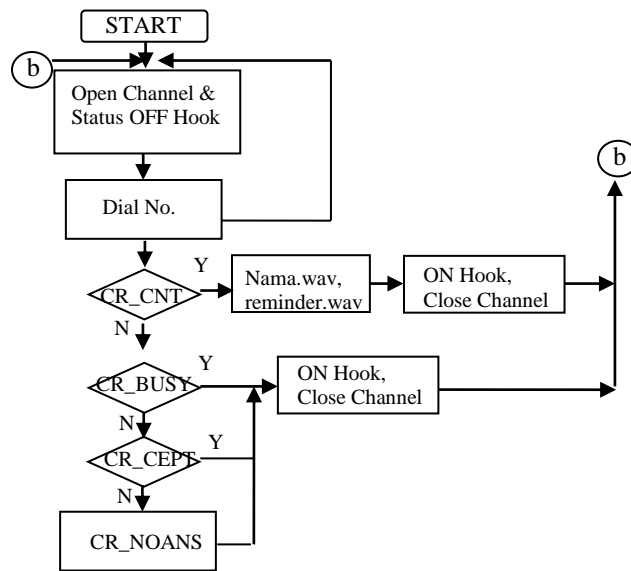
- Visual C++
- *Xampp* yang mempunyai aplikasi MySQL
- *MultiThread Sample Program*, untuk perekaman suara
(boleh menggunakan software perekaman suara yang lain, asal mempunyai format suara manusia, bukan musik)

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 sistem *Reminder* adalah sebagai berikut : Server selalu mencocokkan jam yang sudah disimpan di database dengan jam pada sistem operasi server. Saat jam pada sistem 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 periode 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 sistem 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.

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

1. Men-set lingkungan C++ menjadi lingkungan Teleponi
2. Membuat pemrograman lengkap dengan C++
3. 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. 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 |
|-----------------|---|
| siang.wav | selamat siang |
| greetings.wav | Kami dari layanan "Whatever you want". Menginformasikan event yang akan berjalan hari ini. |
| pilihan.wav | Untuk event jam 9 pagi tekan 1, untuk event jam 12 siang tekan 2 dan untuk event jam 4 sore tekan 3. Untuk kembali ke menu utama tekan 0. |
| event1.wav | Jam 9 pagi anda melaksanakan presentasi dengan PT. HM Sampoerna |
| event2.wav | Jam 12 siang ada undangan lunch dengan Sophia Latjuba |
| event3.wav | Jam 4 sore Persebaya bertanding melawan Arema di Gelora Bung Tomo |
| terimakasih.wav | terima kasih telah menggunakan layanan ini. |

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

Tab: **Link**

- [*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*.

```
#include <windows.h>
#include <string.h>
#include <fcntl.h>
#include <srllib.h>
#include <dxxlib.h>
#include <stdio.h>
```

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

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

f. Tambahkan fungsi Deteksi Digit

```
char deteksidigit(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]);
}

```

g. Tambahkan fungsi Play Suara

```

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");
}

```

h. Membuat main program.

```
//Main Program

void main(int argc, char* argv[])
{
    DX_CAP capp; int chdev;
    char dialstrg[5]="1000"; //nomer telepon yang didial//
    int cares;
    char number[3],dig[3];

    //Open channel//
    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\n");

    switch(cares)
    {
        case CR_CNCT:
            printf("answer\n");
            PlaySuara(chdev,"siang.wav");
            PlaySuara(chdev,"greetings.wav");
        lagi:
            PlaySuara(chdev,"pilihan.wav");
            number[0]=deteksidigit(chdev,dig,1);
            number[1]='\0';
            if(number[0]=='1')
            {
                PlaySuara(chdev,"event1.wav");
                goto lagi;
            }
            else if(number[0]=='2')
            {
                PlaySuara(chdev,"event2.wav");
                goto lagi;
            }
    }
}
```

```

        else if(number[0]=='3')
        {
            PlaySuara(chdev,"event3.wav");
            goto lagi;
        }
        else if(number[0]=='0')
            goto lagi;
        else
            goto keluar;
        break;
case CR_BUSY:
    printf("Busy");
    printf("\ntunggu 10 detik\n");
    if(dx_sethook(chdev,DX_ONHOOK,EV_SYNC)==-1)
    {
        printf("Error on hook\n");
        exit(1);
    }
    printf("on hook succes\n");
    sleep(10000);
    goto awal;
    break;
case CR_NOANS:
    printf("no answer");
    printf("\ntunggu 10 detik\n");
    if(dx_sethook(chdev,DX_ONHOOK,EV_SYNC)==-1)
    {
        printf("Error on hook\n");
        exit(1);
    }
    printf("on hook succes\n");
    sleep(10000);
    goto awal;
    break;
default :
    printf("else");
    printf("\ntunggu 10 detik\n");
    sleep(10000);
    goto awal;
    break;
}

keluar:
    PlaySuara(chdev,"Keluar.wav");
    if(dx_sethook(chdev,DX_ONHOOK,EV_SYNC)==-1)
    {
        printf("Error on hook\n");
        exit(1);
    }
    printf("on hook succes\n");
}

```


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. Lakukan beberapa kali percobaan untuk melihat yang terjadi pada IVR server saat melakukan reminder jika penerima sedang:
 - Kondisi sibuk
 - Tidak ada jawaban
3. Apa gunanya syntax *sleep* pada program di atas ? Coba ubah nilai *sleep*(10000), menjadi *sleep*(5000). 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), yang menerima 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 flowchart dan tabel suara yang diperlukan.