

Pemrograman dengan C++ Builder

17.1. Bagaimana Cara Menampilkan Teks Yang Dapat Bergerak?

Untuk dapat menampilkan teks yang seolah-olah bisa berjalan, dapat dilakukan dengan mudah, yaitu dengan memberikan perintah geser kiri atau geser kanan. Contoh berikut ini adalah menampilkan tulisan seolah-olah tulisan itu bergerak ke kanan dan bergerak ke atas layar. Seperti kalau habis nonton Film atau TV, sudah ada daftar nama para artis, sutradara dan sebagainya.

Caranya:

1. Buat Form Seperti berikut:
2. Terdiri dari komponen Image, Panel, Memo, Label, Button, dan Timer



Gambar 17.1. Desain Form

3. Masukan Teks yang ingin ditampilkan pada property Memo Lines String
4. Masukan kode berikut pada Ttimer

```
void __fastcall TForm1::Timer1Timer(TObject *Sender)
{
    Label1->Left=Label1->Left+1;
    if(Label1->Left==270)
        Label1->Left=-355;
    Memo1->Top=Memo1->Top-1;
    if (Memo1->Top== - 585)
        Memo1->Top=225;
}
```

5. Masukan kode berikut pada Form Create
- ```
void __fastcall TForm1::FormCreate(TObject *Sender)
{
 ShowWindow (GetWindow (Handle, GW_OWNER), SW_HIDE);
}
```

6. Masukan kode berikut pada Tombol Close
- ```
void __fastcall TForm1::BitBtn1Click(TObject *Sender)
{
    Application->Terminate();
}
```

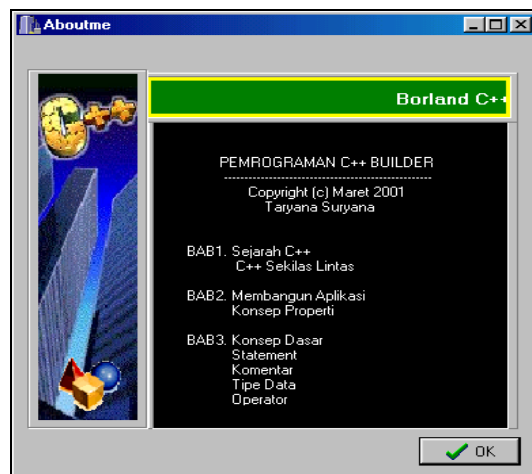
Kode Program Selengkapnya:

```
//-----
#include <vcl.h>
#pragma hdrstop
```

Pemrograman dengan C++ Builder

```
#include "ulain.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm1 *Form1;
//-----
__fastcall TForm1::TForm1(TComponent* Owner)
: TForm(Owner)
{
}
//-----
void __fastcall TForm1::Timer1Timer(TObject *Sender)
{
    Label1->Left=Label1->Left+1;
    if(Label1->Left==270)
        Label1->Left=-355;
    Memo1->Top=Memo1->Top-1;
    if (Memo1->Top== - 585)
        Memo1->Top=225;
}
//-----
void __fastcall TForm1::FormCreate(TObject *Sender)
{
    ShowWindow (GetWindow (Handle, GW_OWNER), SW_HIDE);
}
//-----
void __fastcall TForm1::BitBtn1Click(TObject *Sender)
{
    Application->Terminate();
}
//-----
```

Jika sudah selesai coba jalankan, maka Anda akan melihat tulisan yang bergerak sangat cantik.



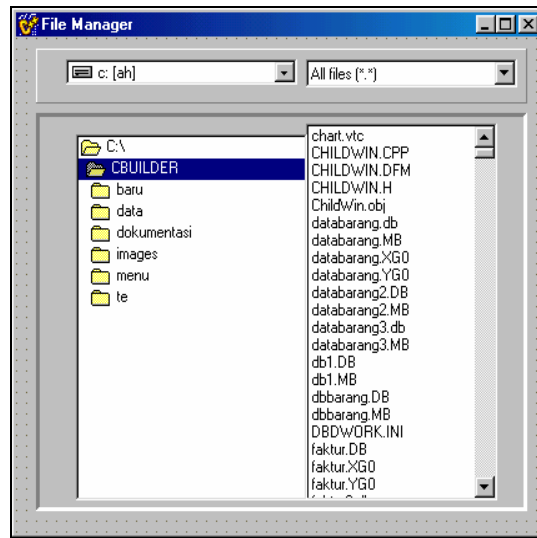
Gambar 17.2. Hasil RUN

17.2. Membuat File Manager

Jika Anda mempunyai sebuah program dan Anda menginginkan agar user dapat berpindah direktori dari satu direktori ke direktori lain, atau melihat daftar file yang ada di direktori aktif, maka dapat dilakukan dengan menggunakan beberapa komponen tanpa menuliskan sebaris kode program.

Berikut adalah contoh, bagaimana membuat File Manager.

1. Buat Form seperti berikut:
2. Terdiri dari Panel1, FileListBox1, DirectoryListBox1, DriveComboBox1 dan FilterComboBox1

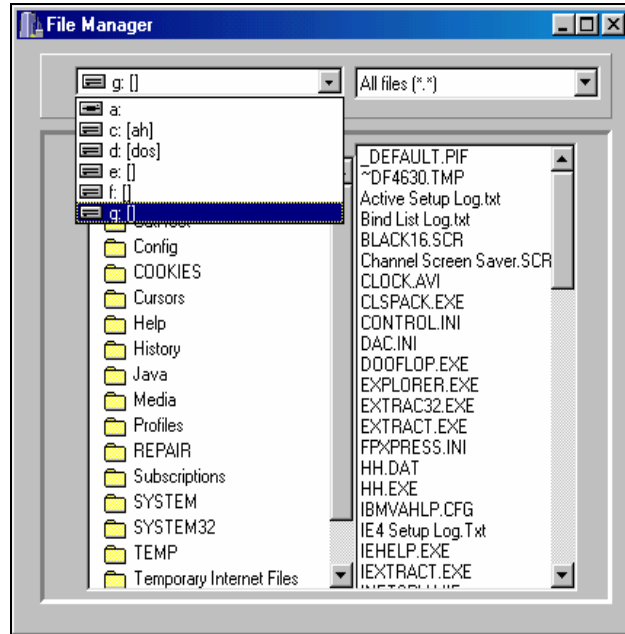


Gambar 17.3. Desain Form File Manager

Atur Properti masing-masing komponen seperti berikut:

KOMPONEN	PROPERTI	NILAI
DirectoryListBox1	FileList	FileListBox1
DriveComboBox1	DirList	DirectoryListBox1
FilterComboBox1	FileList	FileListBox1
	Filter	All Files *.* Program Files *.exe

Jika sudah diatur Properti masing-masing komponen, coba anda jalankan, jika berhasil, maka sekarang anda telah memiliki sebuah file manager, yang dapat digunakan untuk mengelola file sederhana.



Gambar 17.3. Hasil Run Program File Manager

17.3. Menjalankan Program Lain

Untuk menjalankan atau mengeksekusi program lain dari program aplikasi kita, dapat dilakukan dengan menggunakan teknik ShellExecute dan CreateProcess dari Win32 API.

Berikut adalah Definisi dari Struktur SHELLEXECUTEINFO

```
typedef struct _SHELLEXECUTEINFO { // sei
    DWORD    cbSize;
    ULONG    fMask;
    HWND     hwnd;
    LPCSTR   lpVerb;
    LPCSTR   lpFile;
    LPCSTR   lpParameters;
    LPCSTR   lpDirectory;
    int      nShow;
    HINSTANCE hInstApp;

    // Optional members
    LPVOID lpIDLList;
    LPCSTR lpClass;
    HKEY   hkeyClass;
    DWORD  dwHotKey;
    HANDLE hIcon;
    HANDLE hProcess;
} SHELLEXECUTEINFO, FAR *LPSHELLEXECUTEINFO;
```

Cara membuatnya:

1. Buat form seperti File Manager diatas.
2. Masukkan kode program berikut pada Even DoubleClick dari FileListBox1.

```
void __fastcall TForm1::FileListBox1DbClick(TObject *Sender)
{
    SHELLEXECUTEINFO execinfo;
    memset(&execinfo,0,sizeof(execinfo));
```

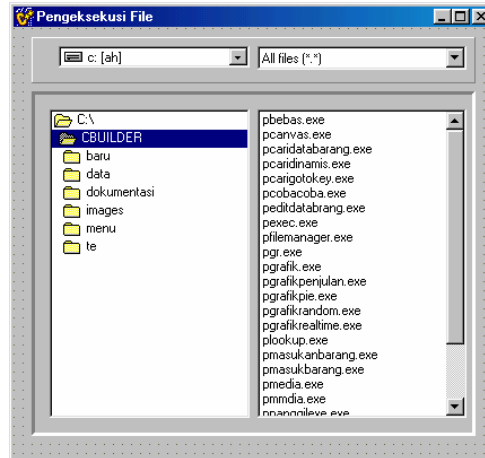
```
execinfo.cbSize=sizeof(execinfo);
execinfo.lpVerb = "open";
execinfo.lpFile = FileListBox1->FileName.c_str();
execinfo.lpParameters = "";
execinfo.fMask = SEE_MASK_NOCLOSEPROCESS;
execinfo.nShow = SW_SHOWDEFAULT;
if(!ShellExecuteEx(&execinfo))
{
    ShowMessage("Tidak bisa Menjalankan Proses");
    return;
}
}
```

Program Selengkapnya:

```
//-----
#include <vcl.h>
#pragma hdrstop
#include "ufexec.h"
#include "shellapi.h"
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm1 *Form1;
//-----
__fastcall TForm1::TForm1(TComponent* Owner)
    : TForm(Owner)
{
}
//-----
void __fastcall TForm1::FileListBox1DbClick(TObject *Sender)
{
    SHELLEXECUTEINFO execinfo;
    memset(&execinfo,0,sizeof(execinfo));
    execinfo.cbSize=sizeof(execinfo);
    execinfo.lpVerb = "open";
    execinfo.lpFile = FileListBox1->FileName.c_str();
    execinfo.lpParameters = "";
    execinfo.fMask = SEE_MASK_NOCLOSEPROCESS;
    execinfo.nShow = SW_SHOWDEFAULT;
    if(!ShellExecuteEx(&execinfo))
    {
        ShowMessage("Tidak bisa Menjalankan Proses");
        return;
    }
}
```

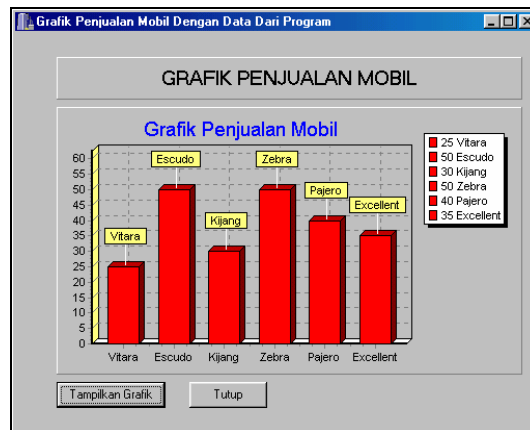
Jalankan program tersebut, maka Anda akan dapat melihat output seperti gambar berikut:

Pemrograman dengan C++ Builder



Gambar 17.4. Hasil Run Pengeksekusi File

Dalam contoh hasil run gambar 3 terdapat file pgrafikpenjualan.exe, jika di-klik dua kali, maka secara langsung file tersebut akan dieksekusi, berikut adalah contoh outputnya:



Gambar 17.5. Program ini dijalankan dari Pengeksekusi File

17.4. Bagaimana Me-Restart Windows

Jika Anda membuat program komputer, dan Anda menginginkan agar komputer tersebut melakukan proses booting kembali (restart), maka anda memampatkan dan menggunakan fungsi yang telah disediakan oleh Win32 API.

Adapun fungsi Win32 API yang digunakan adalah fungsi ExitWindowsEx.

Bentuk penulisan Fungsi ExitWindowsEx, adalah sebagai berikut:

```
BOOL ExitWindowsEx(  
    UINT uFlags, // shutdown operation  
    DWORD dwReserved // reserved  
);
```

Contoh Program

1. Buat Form seperti berikut:



Gambar 17.6. Desain Form Shutdown Windows

2. Masukkan kode berikut pada Tombol Shutdown

```
void __fastcall TForm1::Button1Click(TObject *Sender)
{
    UINT flags;
    if(RadioButton1->Checked)
        flags = EWX_LOGOFF;
    else if(RadioButton1->Checked)
        flags = EWX_SHUTDOWN;
    else
        flags = EWX_REBOOT;
    ExitWindowsEx(flags,0);
}
```

Program Selengkapnya:

```
//-----
#include <vcl.h>
#pragma hdrstop

#include "ureboot.h"
#include "shellapi.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm1 *Form1;
//-----
__fastcall TForm1::TForm1(TComponent* Owner)
    : TForm(Owner)
{
}
//-----

void __fastcall TForm1::Button1Click(TObject *Sender)
{
    UINT flags;
    if(RadioButton1->Checked)
        flags = EWX_LOGOFF;
    else if(RadioButton1->Checked)
        flags = EWX_SHUTDOWN;
    else
        flags = EWX_REBOOT;
    ExitWindowsEx(flags,0);
}
//-----
```

17.5. Menampilkan Direktori Windows dan Windows System

Biasanya jika anda menginstall program baru ke dalam sistem komputer anda, maka yang pertama dicari oleh program tersebut adalah direktori Windows, karena biasanya akan lebih aman jika file-file system yang mendukung program Anda disimpan di direktori Windows, atau Windows System. Nah bagaimana sih, cara mengetahui dimana sebenarnya direktori Windows itu berada, ikuti pembahasan berikut ini.

Untuk mencari direktori dimana Windows berada, dapat dilakukan dengan menggunakan fungsi GetWindowsDirectory, Jika Fungsi ini dipanggil maka dia akan menampilkan path dari Windows tersebut.

```
UINT GetWindowsDirectory(  
    LPTSTR lpBuffer,           // address of buffer for Windows directory  
    UINT uSize                 // size of directory buffer  
);
```

Sedangkan untuk menampilkan path atau direktori dimana Windows\System berada, dapat dilakukan dengan menggunakan fungsi GetSystemDirectory.

```
UINT GetSystemDirectory(  
    LPTSTR lpBuffer,           // address of buffer for system directory  
    UINT uSize                 // size of directory buffer  
);
```

Berikut adalah contoh programnya:

1. Buat Form seperti berikut:



Gambar 17.7. Desain Form Tampilkan Direktori Windows

2. Masukkan kode berikut pada Form Create

```
void __fastcall TForm1::FormCreate(TObject *Sender)  
{  
    AnsiString path;  
    path.SetLength(MAX_PATH);  
    path.SetLength(GetWindowsDirectory(path.c_str(), path.Length()));  
    Label1->Caption=AnsiString ("Direktori Windows : ") + path;  
    path.SetLength(MAX_PATH);  
    path.SetLength(GetSystemDirectory(path.c_str(), path.Length()));  
    Label2->Caption=AnsiString ("Direktori System : ") + path;  
}
```

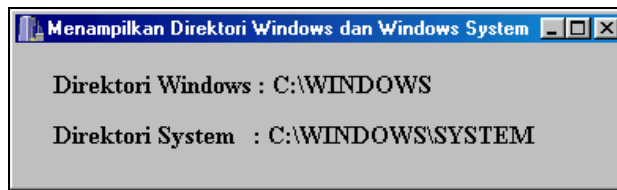
Kode Selengkapnya sebagai berikut:

```
//-----  
#include <vcl.h>  
#pragma hdrstop  
  
#include "ugetwindows.h"  
#include "shellapi.h"  
//-----  
#pragma package(smart_init)
```



```
#pragma resource "*.dfm"
TForm1 *Form1;
//-----
__fastcall TForm1::TForm1(TComponent* Owner)
    : TForm(Owner)
{
}
//-----
void __fastcall TForm1::FormCreate(TObject *Sender)
{
    AnsiString path;
    path.SetLength(MAX_PATH);
    path.SetLength(GetWindowsDirectory(path.c_str(), path.Length()));
    Label1->Caption=AnsiString ("Direktori Windows : ") + path;
    path.SetLength(MAX_PATH);
    path.SetLength(GetSystemDirectory(path.c_str(), path.Length()));
    Label2->Caption=AnsiString ("Direktori System : ") + path;
}
//-----
```

Jalankan program tersebut, jika tidak ada kesalahan, maka akan ditampilkan output seperti gambar berikut:



Gambar 17.8. Informasi Direktori Windows dan Sytem

17.6. Menampilkan Informasi Sistem

Biasanya jika kita ingin mengetahui berapa jumlah memori yang terpakai oleh sistem, kita akan selalu menggunakan fasilitas System Informasi yang diberikan oleh Windows. Nah bagaimana jika seandainya Anda ingin membuatnya sendiri, agar dapat ditempel didalam aplikasi yang dibuat.

- Untuk menampilkan informasi nama komputer, dapat digunakan fungsi `GetComputerName`.

```
BOOL GetComputerName(
    LPTSTR lpBuffer,    // address of name buffer
    LPDWORD nSize       // address of size of name buffer
);
```

- Untuk menampilkan informasi jenis tipe komputer, dapat digunakan fungsi `GetSystemInfo`.

```
VOID GetSystemInfo(
    LPSYSTEM_INFO lpSystemInfo    // address of system information structure
);
```

- Untuk menampilkan Versi dari Windows yang sedang digunakan dapat digunakan fungsi `OSVERSIONINFO`

```
typedef struct _OSVERSIONINFO{
    DWORD dwOSVersionInfoSize;
```

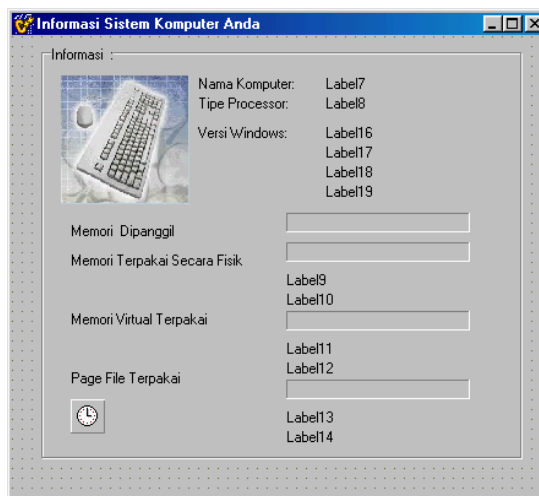
```

        DWORD dwMajorVersion;
        DWORD dwMinorVersion;
        DWORD dwBuildNumber;
        DWORD dwPlatformId;
        TCHAR szCSDVersion[ 128 ];
    } OSVERSIONINFO;
d. Untuk menampilkan informasi mengenai
    ☐ Jumlah memori yang tersedia
    ☐ Jumlah memori terpakai
    ☐ Jumlah memori secara fisik
    ☐ Jumlah memori tidak terpakai
    ☐ Jumlah memori virtual
    ☐ Dan jumlah Page File
    dapat digunakan fungsi GlobalMemoryStatus.
    VOID GlobalMemoryStatus(
        LPMEMORYSTATUS lpBuffer // pointer to the memory status structure
    );

```

Untuk mencoba melihat hasilnya, ikuti langkah berikut:

1. Buat Form seperti berikut:



Gambar 17.9. Desain Form Informasi Sistem.

2. Pada Form OnCreate, masukan kode berikut:

```

void __fastcall TForm1::FormCreate(TObject *Sender)
{
    AnsiString computername;
    computername.SetLength(MAX_COMPUTERNAME_LENGTH + 1);
    DWORD size;
    size= computername.Length();
    if(GetComputerName(computername.c_str(),&size))
        Label7->Caption=computername;
    else
        Label7->Caption="No Name";

    //Tampilkan jenis processor
    SYSTEM_INFO systeminfo;

```

```
GetSystemInfo(&systeminfo);
switch(systeminfo.dwProcessorType)
{
case PROCESSOR_INTEL_386:
    Label8->Caption="i386";
    break;
case PROCESSOR_INTEL_486:
    Label8->Caption="i486";
    break;
case PROCESSOR_INTEL_PENTIUM:
    Label8->Caption="Pentium";
    break;
case PROCESSOR_MIPS_R4000:
    Label8->Caption="MIPS";
    break;
case PROCESSOR_ALPHA_21064:
    Label8->Caption="Alpha";
    break;
default:
    Label8->Caption="Tidak Terdeteksi";
    break;
}
OSVERSIONINFO info;
info.dwOSVersionInfoSize = sizeof(info);
GetVersionEx(&info);
switch(info.dwPlatformId)
{
case VER_PLATFORM_WIN32s:
    Label16->Caption = "Windows Win 32s";
    break;
case VER_PLATFORM_WIN32_WINDOWS:
    Label16->Caption = "Windows 95";
    break;
case VER_PLATFORM_WIN32_NT:
    Label16->Caption = "Windows NT";
    break;
default:
    Label16->Caption = "Windows 2000 +";
    break;
}
Label17->Caption = AnsiString("Version:") + String((int) info.dwMajorVersion) + "." +
String((int)info.dwMinorVersion);
Label18->Caption = AnsiString("Build :") + String((int) info.dwBuildNumber
&0xFFFF);
Label19->Caption = AnsiString("System Info:") + info.szCSDVersion + """;
}
```

3. Pada Ttimer, masukan kode berikut:

```
void __fastcall TForm1::Timer1Timer(TObject *Sender)
{
    MEMORYSTATUS memory;
    memory.dwLength = sizeof(memory);
    GlobalMemoryStatus(&memory);
    ProgressBar1->Position = (float) memory.dwMemoryLoad;

    //memory terpakai secara fisik
    ProgressBar2->Position = 100.0 * (memory.dwTotalPhys - memory.dwAvailPhys) /
```

```
(float) memory.dwTotalPhys;
Label9->Caption = AnsiString ((int)(memory.dwTotalPhys / 1024)) + " K Total";
Label10->Caption = AnsiString ((int)(memory.dwAvailPhys / 1024)) + " K Free";

//virtual memory terpakai
ProgressBar3->Position = 100.0 * (memory.dwTotalVirtual - memory.dwAvailVirtual) /
(float) memory.dwTotalVirtual;
Label11->Caption = AnsiString ((int)(memory.dwTotalVirtual / 1048576.0)) + " MB
Total";
Label12->Caption = AnsiString ((int)(memory.dwAvailVirtual / 1048576.0)) + " MB
Total";

//page file usage
ProgressBar4->Position = 100.0 * (memory.dwTotalPageFile -
memory.dwAvailPageFile) / (float) memory.dwTotalPageFile;
Label13->Caption = AnsiString ((int)(memory.dwTotalPageFile / 1048576.0)) + " MB
Total";
Label14->Caption = AnsiString ((int)(memory.dwAvailPageFile / 1048576.0)) + " MB
Total";
}
```

Program Selengkapnya:

```
//-----
#include <vcl.h>
#pragma hdrstop

#include "umemory.h"
#include "shellapi.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm1 *Form1;
//-----
__fastcall TForm1::TForm1(TComponent* Owner)
: TForm(Owner)
{
}
//-----
void __fastcall TForm1::FormCreate(TObject *Sender)
{
    AnsiString computername;
    computername.SetLength(MAX_COMPUTERNAME_LENGTH + 1);
    DWORD size;
    size= computername.Length();
    if(GetComputerName(computername.c_str(),&size))
    Label7->Caption=computername;
    else
    Label7->Caption="No Name";

    //Tampilkan jenis processor
    SYSTEM_INFO systeminfo;
    GetSystemInfo(&systeminfo);
    switch(systeminfo.dwProcessorType)
    {
    case PROCESSOR_INTEL_386:
        Label8->Caption="i386";
        break;
    case PROCESSOR_INTEL_486:
```

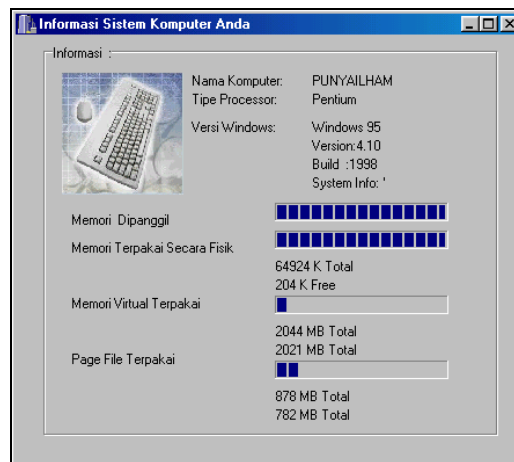
```
Label8->Caption="i486";
break;
case PROCESSOR_INTEL_PENTIUM:
Label8->Caption="Pentium";
break;
case PROCESSOR_MIPS_R4000:
Label8->Caption="MIPS";
break;
case PROCESSOR_ALPHA_21064:
Label8->Caption="Alpha";
break;
default:
Label8->Caption="Tidak Terdeteksi";
break;
}
OSVERSIONINFO info;
info.dwOSVersionInfoSize = sizeof(info);
GetVersionEx(&info);
switch(info.dwPlatformId)
{
case VER_PLATFORM_WIN32s:
Label16->Caption = "Windows Win 32s";
break;
case VER_PLATFORM_WIN32_WINDOWS:
Label16->Caption = "Windows 95";
break;
case VER_PLATFORM_WIN32_NT:
Label16->Caption = "Windows NT";
break;
default:
Label16->Caption = "Windows 2000 +";
break;
}
Label17->Caption = AnsiString("Version:") + String((int) info.dwMajorVersion) + "." +
String((int)info.dwMinorVersion);
Label18->Caption = AnsiString("Build :") + String((int) info.dwBuildNumber
&0xFFFF);
Label19->Caption = AnsiString("System Info:") + info.szCSDVersion + "";
}
//-----
void __fastcall TForm1::Timer1Timer(TObject *Sender)
{
MEMORYSTATUS memory;
memory.dwLength = sizeof(memory);
GlobalMemoryStatus(&memory);
ProgressBar1->Position = (float) memory.dwMemoryLoad;

//memory terpakai secara fisik
ProgressBar2->Position = 100.0 * (memory.dwTotalPhys - memory.dwAvailPhys) /
(float) memory.dwTotalPhys;
Label9->Caption = AnsiString ((int)(memory.dwTotalPhys / 1024)) + " K Total";
Label10->Caption = AnsiString ((int)(memory.dwAvailPhys / 1024)) + " K Free";

//virtual memory terpakai
ProgressBar3->Position = 100.0 * (memory.dwTotalVirtual - memory.dwAvailVirtual) /
(float) memory.dwTotalVirtual;
Label11->Caption = AnsiString ((int)(memory.dwTotalVirtual / 1048576.0)) + " MB
Total";
```

```
Label12->Caption = AnsiString ((int)(memory.dwAvailVirtual / 1048576.0)) + " MB  
Total";  
  
//page file usage  
ProgressBar4->Position = 100.0 * (memory.dwTotalPageFile -  
memory.dwAvailPageFile) / (float) memory.dwTotalPageFile;  
Label13->Caption = AnsiString ((int)(memory.dwTotalPageFile / 1048576.0)) + " MB  
Total";  
Label14->Caption = AnsiString ((int)(memory.dwAvailPageFile / 1048576.0)) + " MB  
Total";  
}  
//-----
```

Berikut adalah output hasil RUN dari program tersebut diatas.



Gambar 17.10. Menampilkan Informasi Sistem

17.7. Multimedia

Untuk membuat sebuah program yang dapat digunakan untuk memanggil atau menjalankan file suara atau video, dapat dilakukan dengan menggunakan komponen MediaPlayer. Format yang didukung oleh Komponen Media Player ini adalah sebagai berikut:

Tabel 17.2. Format Multimedia yang didukung MediaPlayer

Device Type	Software Hardware used	Plays	Uses Tracks	Uses a display Window
dtAVIVideo	AVI Video Player for Windows	AVI Video files	No	Yes
dtCDAudio	CD Audio Player for Windows or a CD Audio Player	CD Audio Disks	Yes	No
dtDAT	Digital Audio Tape Player	Digital Audio Tapes	Yes	No
dtDigital	Video Digital Video Player for Windows	AVI, MPG, MOV files	No	Yes
dtMMMovie	MM Movie Player	MM film	No	Yes
dtOverlay	Overlay device	Analog Video	No	Yes
dtScanner	Image Scanner	N/A for Play (scans images on Record)	No	
dtSequencer	MIDI Sequencer for Windows	MIDI files	Yes	No
DtVCR	Video Cassette Recorder	Video Cassettes	No	Yes
dtWaveAudio	Wave Audio Player for Window	WAV files	No	No

17.8. Membuat Pemutar Video

Untuk membuat aplikasi yang dapat digunakan sebagai alat pemutar file-file Video atau format lainnya yang didukung oleh Tmediaplayer dapat digunakan program berikut ini:

1. Buat Form seperti berikut:
2. Tempatkan TPanel, Timage, Button, DialogBox, Label, dan EditBox.
3. Ganti masing-masing property sehingga menampilkan output seperti berikut:



Gambar 17.11. Desain Form Multimedia Player

4. Pada Tombol Buka File, masukan kode berikut:

```
void __fastcall Tfmmedia::Button1Click(TObject *Sender)
{
    if (OpenDialog1->Execute())
    {
        //Jika file avi besar, maka akan agak lama menampilkan gambar, tampilkan kursor bentuk
        gelas
        Screen->Cursor = crHourGlass;
        MediaPlayer1->FileName = OpenDialog1->FileName;
        try
        {
            MediaPlayer1->Open();
        }
        catch (...)

        {
            Application->MessageBox("Gagal Membuka File", mtWarning, MB_ICONWARNING|
            MB_OK);
        }
        //kembalikan kursor ke semula
        Screen->Cursor=crArrow;
        MediaPlayer1->DisplayRect = Rect(4,4,0,0); //tampilkan sesuai aslinya
        Edit1->Text=OpenDialog1->FileName;
    }
}
```
5. Pada Tombol Udaahan, masukan kode berikut:

```
void __fastcall Tfmmedia::Button2Click(TObject *Sender)
{
    Close();
}
```

Program Selengkapnya:

```
//-----
#include <vcl.h>
#pragma hdrstop

#include "ummedia.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
Tfmmedia *fmmedia;
//-----
__fastcall Tfmmedia::Tfmmedia(TComponent* Owner)
    : TForm(Owner)
{
}
//-----

void __fastcall Tfmmedia::Button1Click(TObject *Sender)
{
    if (OpenDialog1->Execute())
    {
        Screen->Cursor = crHourGlass;
        MediaPlayer1->FileName = OpenDialog1->FileName;
        try
        {
            MediaPlayer1->Open();
        }
    }
}
```


Pemrograman dengan C++ Builder

```
catch (...)  
{  
Application->MessageBox("Gagal Membuka File", mtWarning, MB_ICONWARNING|MB_OK);  
}  
Screen->Cursor=crArrow;  
MediaPlayer1->DisplayRect = Rect(4,4,0,0);  
Edit1->Text=OpenDialog1->FileName;  
}  
}  
//-----  
void __fastcall Tfmmedia::Button2Click(TObject *Sender)  
{  
    Close();  
}  
//-----
```

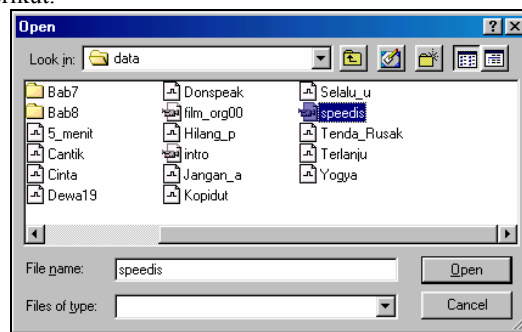
Pemakaian Program

Jika semua kode program sudah di tulis, dan Anda Run, maka akan ditampilkan output seperti berikut:



Gambar 17.12. Hasil Runing Program Multimedia Player

Untuk memilih File yang akan dijalankan, tekan tombol Buka File, kemudian akan ditampilkan kotak dialog seperti berikut:



Gambar 17.13. Kotak Dialog Pemilihan File

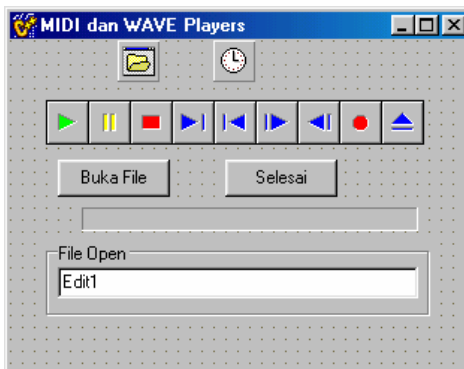


Gambar 17.14. Contoh File Video yang sedang di putar

17.9. Pemutar File Midi dan Wave

Di dalam format file suara bawaan sistem operasi Windows, ada format suara Wave dan Midi yang bisa anda nikmati. Nah bagaimanakah cara membuat program untuk memainkan file suara berformat Midi atau Wave Audio tersebut, ikuti pembahasannya berikut ini.

1. Buat Form seperti berikut:
2. Tempatkan Komponen, TMediaPlayer, TTimer, TOpenDialog, dua buah Tbutton, Progressbar, dan EditBox



Gambar 17.15. Desain Form Pemutar File MIDI dan WAVE

3. Pada Tombol Buka File, masukan kode berikut:

```
void __fastcall TForm1::Button1Click(TObject *Sender)
{
    OpenFileDialog1->FileName = "";
    OpenFileDialog1->Filter="Wave Files (*.wav)|*.wav| Midi Files(*.mid)|*.mid";
    if (OpenDialog1->Execute())
    {
        MediaPlayer1->FileName = OpenFileDialog1->FileName;
        try
        {
            MediaPlayer1->Open();
        }
        catch (...)
        {
            Application->MessageBox("Gagal Membuka File",
            mtWarning,MB_ICONWARNING|MB_OK);
        }
    }
}
```

```
    }  
    }  
    Edit1->Text=OpenDialog1->FileName;  
}
```

4. Pada Ttimer, masukan kode berikut:

```
void __fastcall TForm1::Timer1Timer(TObject *Sender)  
{  
    ProgressBar1->Position = (TProgressRange) (MediaPlayer1->Position * 100 /  
    MediaPlayer1->Length);  
    if(ProgressBar1->Position == 100)  
        Timer1->Enabled = false;  
}
```
5. Pada MediaPlayer, masukan kode berikut:

```
void __fastcall TForm1::MediaPlayer1Click(TObject *Sender,  
    TMPBtnType Button, bool &DoDefault)  
{  
    if(Button == btPlay)  
        Timer1->Enabled = true;  
    else if(Button == btStop)  
        Timer1->Enabled = false;  
    DoDefault = true;  
}
```
6. Pada Tombol Selesai, masukan kode berikut:

```
void __fastcall TForm1::Button2Click(TObject *Sender)  
{  
    Close();  
}
```

Program Selengkapnya:

```
//-----  
#include <vcl.h>  
#pragma hdrstop  
  
#include "umidiplayer.h"  
//-----  
#pragma package(smart_init)  
#pragma resource "*.dfm"  
TForm1 *Form1;  
//-----  
__fastcall TForm1::TForm1(TComponent* Owner)  
    : TForm(Owner)  
{  
}  
//-----  
  
void __fastcall TForm1::Button1Click(TObject *Sender)  
{  
    OpenFileDialog1->FileName = "";  
    OpenFileDialog1->Filter="Wave Files (*.wav)|*.wav| Midi Files(*.mid)|*.mid";  
    if (OpenDialog1->Execute())  
    {  
        MediaPlayer1->FileName = OpenFileDialog1->FileName;  
        try  
        {
```

Pemrograman dengan C++ Builder

```
MediaPlayer1->Open();
}
catch (...)
{
Application->MessageBox("Gagal Membuka File", mtWarning, MB_ICONWARNING|MB_OK);
}
}
Edit1->Text=OpenDialog1->FileName;
}
//-----

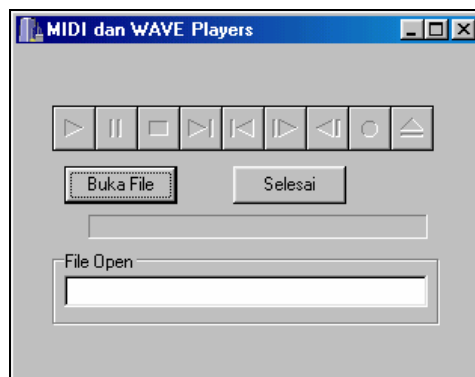
void __fastcall TForm1::Button2Click(TObject *Sender)
{
Close();
}
//-----

void __fastcall TForm1::MediaPlayer1Click(TObject *Sender,
TMPBtnType Button, bool &DoDefault)
{
if(Button == btPlay)
Timer1->Enabled = true;
else if(Button == btStop)
Timer1->Enabled= false;
DoDefault = true;
}
//-----

void __fastcall TForm1::Timer1Timer(TObject *Sender)
{
ProgressBar1->Position = (TProgressRange) (MediaPlayer1->Position * 100 / MediaPlayer1-
>Length);
if(ProgressBar1->Position == 100)
Timer1->Enabled = false;
}
//-----
```

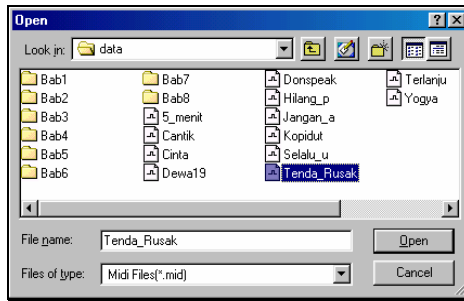
Pemakaian Program

Untuk menjalankan program Midi Player ini, dapat dilakukan dengan cara memilih menu Run, dari C++ Builder Anda, jika tidak ada kesalahan akan ditampilkan output seperti berikut:



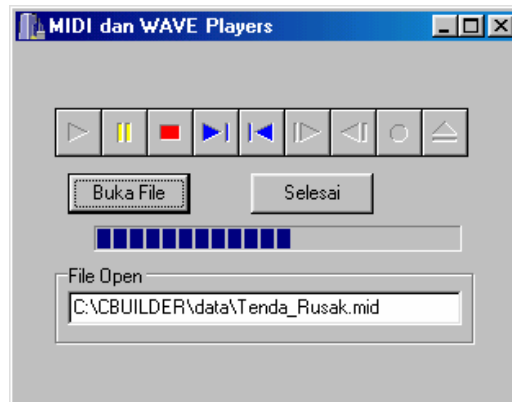
Gambar 17.16. Program Pemutar File MIDI dan WAVE

Untuk memilih nama file dan format yang diinginkan, klik tombol Buka File, setelah itu akan ditampilkan kotak dialog seperti berikut:



Gambar 17.17. Dialog Pemilihan File

Jika nama file sudah ditentukan, silahkan tekan tombol Open, atau klik dua kali pada nama file yang ingin Anda putar.



Gambar 17.18. Program Sedang Memutar Lagu