Lampiran 1 : Listing Program WinMote.h

```
// WinMote.h : main header file for the PROJECT_NAME application
//

#pragma once

#ifndef __AFXWIN_H__
	#error "include 'stdafx.h' before including this file for
PCH"
#endif

#include "resource.h"		// main symbols


// CWinMoteApp:
// See WinMote.cpp for the implementation of this class
//

class CWinMoteApp : public CWinApp
{
public:
	CWinMoteApp();

// Overrides
	public:
	virtual BOOL InitInstance();

// Implementation

	DECLARE_MESSAGE_MAP()
};

extern CWinMoteApp theApp;
```

Lampiran 2 : Listing Program WinMote.cpp

```
// WinMote.cpp : Defines the class behaviors for the application.
//

#include "stdafx.h"
#include "WinMote.h"
#include "WinMoteDlg.h"

#ifdef _DEBUG
#define new DEBUG_NEW
#endif


// CWinMoteApp

BEGIN_MESSAGE_MAP(CWinMoteApp, CWinApp)
      ON_COMMAND(ID_HELP, &CWinApp::OnHelp)
END_MESSAGE_MAP()


// CWinMoteApp construction

CWinMoteApp::CWinMoteApp()
{
      // TODO: add construction code here,
      // Place all significant initialization in InitInstance
}


// The one and only CWinMoteApp object

CWinMoteApp theApp;


// CWinMoteApp initialization

BOOL CWinMoteApp::InitInstance()
{
      // InitCommonControlsEx() is required on Windows XP if an
      application
      // manifest specifies use of ComCtl32.dll version 6 or later
      to enable
      // visual styles.  Otherwise, any window creation will fail.


      INITCOMMONCONTROLSEX InitCtrls;
      InitCtrls.dwSize = sizeof(InitCtrls);
      // Set this to include all the common control classes you
      want to use
      // in your application.
      InitCtrls.dwICC = ICC_WIN95_CLASSES;
      InitCommonControlsEx(&InitCtrls);

      CWinApp::InitInstance();

      if (!AfxSocketInit())
```

Lampiran 2 : Listing Program WinMote.cpp (lanjutan)

```
	{
		AfxMessageBox(IDP_SOCKETS_INIT_FAILED);
		return FALSE;
	}

	AfxEnableControlContainer();

	// Standard initialization
	// If you are not using these features and wish to reduce
	the size
	// of your final executable, you should remove from the
	following
	// the specific initialization routines you do not need
	// Change the registry key under which our settings are
	stored
	// TODO: You should modify this string to be something
	appropriate
	// such as the name of your company or organization
	SetRegistryKey(_T("Local AppWizard-Generated
	Applications"));

	CWinMoteDlg dlg;
	m_pMainWnd = &dlg;
	INT_PTR nResponse = dlg.DoModal();
	if (nResponse == IDOK)
	{
		// TODO: Place code here to handle when the dialog is
		//  dismissed with OK
	}
	else if (nResponse == IDCANCEL)
	{
		// TODO: Place code here to handle when the dialog is
		//  dismissed with Cancel
	}

	// Since the dialog has been closed, return FALSE so that we
	exit the
	//  application, rather than start the application's message
	pump.
	return FALSE;
}
```

Lampiran 3 : Listing Program WinMoteDlg.h

```
// WinMoteDlg.h : header file
//

#pragma once
#include "afxwin.h"
#include "afxcmn.h"


// CWinMoteDlg dialog
class CWinMoteDlg : public CDialog
{
// Construction
public:
	CWinMoteDlg(CWnd* pParent = NULL);  // standard constructor

// Dialog Data
	enum { IDD = IDD_WINMOTE_DIALOG };

	protected:
	virtual void DoDataExchange(CDataExchange* pDX);	//
DDX/DDV support


// Implementation
protected:
	HICON m_hIcon;

	// Generated message map functions
	virtual BOOL OnInitDialog();
	afx_msg void OnSysCommand(UINT nID, LPARAM lParam);
	afx_msg void OnPaint();
	afx_msg HCURSOR OnQueryDragIcon();
	DECLARE_MESSAGE_MAP()
public:
	afx_msg void OnBnClickedAdd();
public:
	CListBox m_main_appname1;
public:
	CEdit m_main_apppath1;
public:
	afx_msg void OnLbnSelchangeAppname();
public:
	afx_msg void OnBnClickedRemove();
public:
	afx_msg void OnBnClickedLaunch();
public:
	CButton m_main_remove;
public:
	CListCtrl m_main_remote;
public:
	SOCKET s;
public:
	SOCKADDR_IN sender;
public:
	SOCKADDR_IN local;
```

Lampiran 3 : Listing Program WinMoteDlg.h (lanjutan)

```
public:
	afx_msg void OnBnClickedAddremote();
public:
	afx_msg void OnBnClickedOk();
public:
	afx_msg void OnBnClickedAddhkey();
public:
	CButton m_addremotekey;
public:
	CEdit m_remotenamectrl;
public:
	afx_msg void OnBnClickedCancel();
public:
	afx_msg void OnLvnItemchangedRemotekey(NMHDR *pNMHDR,
LRESULT *pResult);
public:
	afx_msg void CWinMoteDlg::KeyPress();
	afx_msg void CWinMoteDlg::MouseMotion();
	afx_msg void CWinMoteDlg::MouseClick();
public:
	CButton m_start_remote;
	CButton m_stop_remote;
	BYTE vkey;
	BYTE modifier;
	DWORD item;
	DWORD item2;
	DWORD dx;
	DWORD dy;
	CListBox m_hwnd;
	CButton m_cancel;
	HWND wnd;
	bool startenable;
	int resolusi;
	int jmlapp;
	int jmlapp2;
	int jmlkey;
	int jmlkey2;
	CString remotekey;
	CString emukey;
	DWORD emudata;
	CString app;
	CString applipath;
private:
	static BOOL CALLBACK EnumWindowsProc (HWND hwnd, LPARAM
lParam);
	afx_msg void OnBnClickedStart();
	afx_msg void OnBnClickedStop();
	afx_msg void OnBnClickedAddspecial();
	afx_msg void OnBnClickedClear();
	afx_msg void OnBtnClickedOk();
	afx_msg void OnBtnClickedCancel();
	afx_msg void OnBnClickedClearall();
	afx_msg void OnFileExit();
	afx_msg void OnHelpAboutwinmote();
	afx_msg void OnBnClickedSet();
```

Lampiran 3 : Listing Program WinMoteDlg.h (lanjutan)

```
      afx_msg void OnBnClickedSaveapp();
      afx_msg void OnBnClickedLoadapp();
      afx_msg void OnBnClickedSavekey();
      afx_msg void OnBnClickedLoadkey();
};

UINT thread(LPVOID p);
UINT thread2(LPVOID p);
```

Lampiran 4 : Listing Program WinMoteDlg.cpp

```
// WinMoteDlg.cpp : implementation file
//

#include "stdafx.h"
#include "WinMote.h"
#include "WinMoteDlg.h"
#include "AddDlg.h"
#include "AddHot.h"
#include "AddSpecial.h"


#ifdef _DEBUG
#define new DEBUG_NEW
#endif

CStringArray path; //buat array string dengan nama variabel path
CString nama;
// CAboutDlg dialog used for App About

class CAboutDlg : public CDialog
{
public:
	CAboutDlg();

// Dialog Data
	enum { IDD = IDD_ABOUTBOX };

	protected:
	virtual void DoDataExchange(CDataExchange* pDX);    //
DDX/DDV support

// Implementation
protected:
	DECLARE_MESSAGE_MAP()
};

CAboutDlg::CAboutDlg() : CDialog(CAboutDlg::IDD)
{
}

void CAboutDlg::DoDataExchange(CDataExchange* pDX)
{
	CDialog::DoDataExchange(pDX);
}

BEGIN_MESSAGE_MAP(CAboutDlg, CDialog)
END_MESSAGE_MAP()


// CWinMoteDlg dialog



CWinMoteDlg::CWinMoteDlg(CWnd* pParent /*=NULL*/)
```

Lampiran 4 : Listing Program WinMoteDlg.cpp (lanjutan)

```
	: CDialog(CWinMoteDlg::IDD, pParent)
	, vkey(0)
	, modifier(0)
	, item(0)
	, dx(0)
	, dy(0)
	, item2(0)
	, startenable(false)
{
	m_hIcon = AfxGetApp()->LoadIcon(IDI_ICON1);
}

void CWinMoteDlg::DoDataExchange(CDataExchange* pDX)
{
	CDialog::DoDataExchange(pDX);
	DDX_Control(pDX, IDC_APPNAME, m_main_appname1);
	DDX_Control(pDX, IDC_APPPATH, m_main_apppath1);
	DDX_Control(pDX, IDC_REMOVE, m_main_remove);
	DDX_Control(pDX, IDC_REMOTEKEY, m_main_remote);
	DDX_Control(pDX, IDC_ADDREMOTE, m_addremotekey);
	DDX_Control(pDX, IDC_EDITREMOTE, m_remotenamectrl);
	DDX_Control(pDX, IDC_START, m_start_remote);
	DDX_Control(pDX, IDC_STOP, m_stop_remote);
	DDX_Control(pDX, IDC_HWND, m_hwnd);
	DDX_Control(pDX, IDC_CANCEL, m_cancel);
}

BEGIN_MESSAGE_MAP(CWinMoteDlg, CDialog)
	ON_WM_SYSCOMMAND()
	ON_WM_PAINT()
	ON_WM_QUERYDRAGICON()
	//}}AFX_MSG_MAP
	ON_BN_CLICKED(IDC_ADD, &CWinMoteDlg::OnBnClickedAdd)
	ON_LBN_SELCHANGE(IDC_APPNAME,&CWinMoteDlg::OnLbnSelchangeApp
	name)
	ON_BN_CLICKED(IDC_REMOVE, &CWinMoteDlg::OnBnClickedRemove)
	ON_BN_CLICKED(IDC_LAUNCH, &CWinMoteDlg::OnBnClickedLaunch)
	ON_BN_CLICKED(IDC_ADDREMOTE,&CWinMoteDlg::OnBnClickedAddremo
	te)
	ON_BN_CLICKED(IDOK, &CWinMoteDlg::OnBnClickedOk)
	ON_BN_CLICKED(IDC_ADDHKEY, &CWinMoteDlg::OnBnClickedAddhkey)
	ON_BN_CLICKED(IDCANCEL, &CWinMoteDlg::OnBnClickedCancel)
	ON_NOTIFY(LVN_ITEMCHANGED, IDC_REMOTEKEY,
	&CWinMoteDlg::OnLvnItemchangedRemotekey)
	ON_BN_CLICKED(IDC_START, &CWinMoteDlg::OnBnClickedStart)
	ON_BN_CLICKED(IDC_STOP, &CWinMoteDlg::OnBnClickedStop)
	ON_BN_CLICKED(IDC_ADDSPECIAL,&CWinMoteDlg::OnBnClickedAddspe
	cial)
	ON_BN_CLICKED(IDC_CLEAR, &CWinMoteDlg::OnBnClickedClear)
	ON_BN_CLICKED(IDC_OK, &CWinMoteDlg::OnBtnClickedOk)
	ON_BN_CLICKED(IDC_CANCEL, &CWinMoteDlg::OnBtnClickedCancel)
	ON_BN_CLICKED(IDC_CLEARALL,&CWinMoteDlg::OnBnClickedClearall
	)
	ON_COMMAND(ID_FILE_EXIT, &CWinMoteDlg::OnFileExit)
	ON_COMMAND(ID_HELP_ABOUTWINMOTE,&CWinMoteDlg::OnHelpAboutwin
	mote)
```

Lampiran 4 : Listing Program WinMoteDlg.cpp (lanjutan)

```
      ON_BN_CLICKED(IDC_SET, &CWinMoteDlg::OnBnClickedSet)
      ON_BN_CLICKED(IDC_SAVEAPP, &CWinMoteDlg::OnBnClickedSaveapp)
      ON_BN_CLICKED(IDC_LOADAPP, &CWinMoteDlg::OnBnClickedLoadapp)
      ON_BN_CLICKED(IDC_SAVEKEY, &CWinMoteDlg::OnBnClickedSavekey)
                  ON_BN_CLICKED(IDC_LOADKEY,
            &CWinMoteDlg::OnBnClickedLoadkey)
END_MESSAGE_MAP()


// CWinMoteDlg message handlers

BOOL CWinMoteDlg::OnInitDialog()
{
      CDialog::OnInitDialog();

      // Add "About..." menu item to system menu.

      // IDM_ABOUTBOX must be in the system command range.
      ASSERT((IDM_ABOUTBOX & 0xFFF0) == IDM_ABOUTBOX);
      ASSERT(IDM_ABOUTBOX < 0xF000);

      CMenu* pSysMenu = GetSystemMenu(FALSE);
      if (pSysMenu != NULL)
      {
            CString strAboutMenu;
            strAboutMenu.LoadString(IDS_ABOUTBOX);
            if (!strAboutMenu.IsEmpty())
            {
                  pSysMenu->AppendMenu(MF_SEPARATOR);
                  pSysMenu->AppendMenu(MF_STRING, IDM_ABOUTBOX,
                  strAboutMenu);
            }
      }

      // Set the icon for this dialog.  The framework does this
      automatically
      //  when the application's main window is not a dialog
      SetIcon(m_hIcon, TRUE);             // Set big icon
      SetIcon(m_hIcon, FALSE);            // Set small icon

      // TODO: Add extra initialization here
      //set ukuran array string sejumlah 10 elemen
      path.SetSize(11, 1);
      //masukkan string kosong ke array string index ke 11
      sebanyak 1 kali
      path.InsertAt(11, "", 1);

      CRect rect;
      CListCtrl *pMainRemote =
      (CListCtrl*)GetDlgItem(IDC_REMOTEKEY);

      pMainRemote->GetClientRect(rect);

      //masukkan kolom index 0 ke listctrl (kolom ini adl item
      utama)
```

Lampiran 4 : Listing Program WinMoteDlg.cpp (lanjutan)

```
	pMainRemote->InsertColumn(0, "REMOTE KEY", LVCFMT_CENTER,
	100, 0);

	//masukkan kolom index 1 ke listctrl (subitem index 1)
	pMainRemote->InsertColumn(1, "HOTKEY", LVCFMT_LEFT, 175, 1);


	//set style listctrl
	pMainRemote->SetExtendedStyle(LVS_EX_GRIDLINES |
	LVS_EX_FULLROWSELECT | LVS_EX_SUBITEMIMAGES |
	LVS_EX_ONECLICKACTIVATE);

	return TRUE;  // return TRUE  unless you set the focus to a
control
}

UINT thread2(LPVOID p)
{
	int ret=1;
	int i=0;
	//inisialisasi buffer sebesar 50 karakter
	char buff[50];
	char *result;
	char *result3;
	char *result5;
	CString data;
	CString key;
	CWinMoteDlg *dlg2=(CWinMoteDlg*)AfxGetApp()->GetMainWnd();

	//disable button add remote key
	dlg2->m_addremotekey.EnableWindow(FALSE);
	//disable button start remote
	dlg2->m_start_remote.EnableWindow(FALSE);
	//disable button cancel
	dlg2->m_cancel.EnableWindow(FALSE);

	//apabila tidak ada error yg terjadi lakukan looping
	while (ret!=SOCKET_ERROR)
	{
		//menerima data dari socket s, datanya ditampung ke
		buff
		ret = recv(dlg2->s, buff, 50,0);
		//apabila koneksi ditutup maka keluar
		if (ret == 0) break;

		//apabila tidak ada error, lakukan perintah berikut
		if (ret != SOCKET_ERROR)
		{
			//masukkan null terminated string ke "pointer ke
			ret"
			buff[ret] = '\0';
			//temukan spasi pertama di dalam data yang
			diterima
			result=strpbrk(buff, " ");
			result++;   //majukan pointer 1 karakter
```

Lampiran 4 : Listing Program WinMoteDlg.cpp (lanjutan)

```
int result2= (int) (result-buff);   //dapatkan
panjang string dari awal string sampai pointer
terakhir
result3=strpbrk(result, " ");       //temukan
spasi berikutnya
result3++;
int result4= (int) (result3-buff);
result5=strpbrk(result3, " ");
result5++;
int result6= (int) (result5-buff);
data = buff;

key = data.Mid(result4,result6-result4-1);
//ambil string tombol remote

int count = dlg2->m_main_remote.GetItemCount();
//dapatkan jumlah item di listctrl

while (count != 0)
//jika item tidak 0 jalankan looping
{
      count--;
      //kurangi count
      //dapatkan item yang ada di list ctrl
      remote
      CString item = dlg2->
      m_main_remote.GetItemText(count,0);

      //bandingkan key remote yang masuk dengan
      item yang ada di list ctrl remote
      int cmp = strcmp(item,key);
      //kalau sama
      if (cmp == 0)
      {
      //isi dword item dengan data item yang
      diset untuk key remote di list
      dlg2->item = dlg2->
      m_main_remote.GetItemData(count);

            //kalau dword ada isinya
            if (dlg2->item != 0)
            {
                  //kalau dword item berisi
                  event mouse
                  if (dlg2->item ==
                  MOUSEEVENTF_LEFTDOWN)
                  {

                        dlg2->item2 =
                        MOUSEEVENTF_LEFTUP;
                        dlg2->
                  CWinMoteDlg::MouseClick();
                  } else
                  if (dlg2->item ==
                  MOUSEEVENTF_RIGHTDOWN)
```

Lampiran 4 : Listing Program WinMoteDlg.cpp (lanjutan)

```
{
      dlg2->item2 =
      MOUSEEVENTF_RIGHTUP;
dlg2->
CWinMoteDlg::MouseClick();
} else
if (dlg2->item ==
MOUSEEVENTF_MOVE)
{
CString key =
dlg2->
m_main_remote.GetItemText(coun
t, 1);
      if (key == "Mouse Move
      Up")
      {

            dlg2->dx = 0;
            dlg2->dy = -*dlg2
             ->resolusi;
            dlg2->
      CWinMoteDlg::MouseMotion
      ();
      } else
      if (key == "Mouse Move
      Down")
      {

            dlg2->dx = 0;
            dlg2->dy = dlg2
            ->resolusi;
            dlg2->
            CWinMoteDlg::Mouse
      Motion();
      } else
      if (key == "Mouse Move
      Left")
      {

            dlg2->dx =
            -1*dlg2->resolusi;
            dlg2->dy = 0;
            dlg2->
            CWinMoteDlg::Mouse
      Motion();
      } else
      if (key == "Mouse Move
      Right")
      {

            dlg2->dx = dlg2
            ->resolusi;
            dlg2->dy = 0;
            dlg2->
            CWinMoteDlg::Mouse
      Motion();
```

Lampiran 4 : Listing Program WinMoteDlg.cpp (lanjutan)

```
                                        }
                                    }
                                    //kalau dword item berisi
                                    tombol keyboard
                                    else
                                    {
                                        //dapatkan loword dari
                                        dword item
                                        WORD itemvkey =
                                        LOWORD(dlg2->item);
                                        if (itemvkey != 0)
                                        {
                                            //dapatkan vkey
                                            dlg2->vkey =
                                            LOBYTE(itemvkey);
                                            //dapatkan
                                            modifier flag
                                            (penekanan shift,
                                            alt, ctrl)
                                            dlg2->modifier =
                                            HIBYTE(itemvkey);
                                            //emulasi tombol
                                            keyboard
                                            dlg2->
                                            CWinMoteDlg::KeyPr
                                        ess();
                                        }
                                    }
                                }
                            }
                        }
                    }

        }

        //tutup koneksi soket
        closesocket(dlg2->s);
        //akhiri penggunaan WS2_32.dll
        WSACleanup();
        //akhiri thread
        AfxEndThread(0);
        return 0;
}

void CWinMoteDlg::KeyPress()
{
        //aktifkan window yang dipilih
        ::SetForegroundWindow(wnd);
        if ( (modifier & (HOTKEYF_SHIFT)) == (HOTKEYF_SHIFT))
        {
            ::keybd_event(0x10,0,0,0);   //shift
            ::keybd_event(vkey,0,0,0);   //vkey
            ::Sleep(20);
            ::keybd_event(0x10,0,KEYEVENTF_KEYUP,0);
            ::keybd_event(vkey,0,KEYEVENTF_KEYUP,0);
```

Lampiran 4 : Listing Program WinMoteDlg.cpp (lanjutan)

```
} else
if ( (modifier & (HOTKEYF_CONTROL)) == (HOTKEYF_CONTROL) )
{
      ::keybd_event(0x11,0,0,0);   //ctrl
      ::keybd_event(vkey,0,0,0);     //vkey
      ::Sleep(20);
      ::keybd_event(0x11,0,KEYEVENTF_KEYUP,0);
      ::keybd_event(vkey,0,KEYEVENTF_KEYUP,0);
} else
if ( (modifier & (HOTKEYF_ALT)) == (HOTKEYF_ALT))
{
      ::keybd_event(0x12,0,0,0);   //alt
      ::keybd_event(vkey,0,0,0);   //vkey
      ::Sleep(20);
      ::keybd_event(0x12,0,KEYEVENTF_KEYUP,0);
      ::keybd_event(vkey,0,KEYEVENTF_KEYUP,0);
} else
if ( (modifier & (HOTKEYF_SHIFT | HOTKEYF_CONTROL)) ==
(HOTKEYF_SHIFT | HOTKEYF_CONTROL))
{
      ::keybd_event(0x10,0,0,0);   //shift
      ::keybd_event(0x11,0,0,0);   //ctrl
      ::keybd_event(vkey,0,0,0);   //vkey
      ::Sleep(20);
      ::keybd_event(0x10,0,KEYEVENTF_KEYUP,0);
      ::keybd_event(0x11,0,KEYEVENTF_KEYUP,0);
      ::keybd_event(vkey,0,KEYEVENTF_KEYUP,0);
} else
if ( (modifier & (HOTKEYF_CONTROL | HOTKEYF_ALT)) ==
(HOTKEYF_CONTROL | HOTKEYF_ALT))
{
      ::keybd_event(0x11,0,0,0);   //ctrl
      ::keybd_event(0x12,0,0,0);   //alt
      ::keybd_event(vkey,0,0,0);   //vkey
      ::Sleep(20);
      ::keybd_event(0x11,0,KEYEVENTF_KEYUP,0);
      ::keybd_event(0x12,0,KEYEVENTF_KEYUP,0);
      ::keybd_event(vkey,0,KEYEVENTF_KEYUP,0);
} else
if ( (modifier & (HOTKEYF_SHIFT | HOTKEYF_ALT)) ==
(HOTKEYF_SHIFT | HOTKEYF_ALT))
{
      ::keybd_event(0x10,0,0,0);   //shift
      ::keybd_event(0x12,0,0,0);   //alt
      ::keybd_event(vkey,0,0,0);   //vkey
      ::Sleep(20);
      ::keybd_event(0x10,0,KEYEVENTF_KEYUP,0);
      ::keybd_event(0x12,0,KEYEVENTF_KEYUP,0);
      ::keybd_event(vkey,0,KEYEVENTF_KEYUP,0);
} else
if( (modifier & (HOTKEYF_SHIFT | HOTKEYF_CONTROL |
HOTKEYF_ALT)) == (HOTKEYF_SHIFT | HOTKEYF_CONTROL |
HOTKEYF_ALT))
{
      ::keybd_event(0x10,0,0,0);   //shift
```

Lampiran 4 : Listing Program WinMoteDlg.cpp (lanjutan)

```
            ::keybd_event(0x11,0,0,0);   //ctrl
            ::keybd_event(0x12,0,0,0);   //alt
            ::keybd_event(vkey,0,0,0);   //vkey
            ::Sleep(20);
            ::keybd_event(0x10,0,KEYEVENTF_KEYUP,0);
            ::keybd_event(0x11,0,KEYEVENTF_KEYUP,0);
            ::keybd_event(0x12,0,KEYEVENTF_KEYUP,0);
            ::keybd_event(vkey,0,KEYEVENTF_KEYUP,0);
      }
      else
      {
            ::keybd_event(vkey,0,0,0);
            ::Sleep(20);
            ::keybd_event(vkey,0,KEYEVENTF_KEYUP,0);
      }
}

void CWinMoteDlg::MouseMotion()
{
      //setforeground application pakai window handle
      ::SetForegroundWindow(wnd);
      //item mouse move, gerakkan pointer sebesar dx & dy
      ::mouse_event(item,dx,dy,0,0); //dflags,dx,dy,0,0
      ::Sleep(1);
}

void CWinMoteDlg::MouseClick()
{

      ::SetForegroundWindow(wnd);
      //item click down
      ::mouse_event(item,0,0,0,0); //dflags,dx,dy,0,0
      //delay
      ::Sleep(30);
      //item click up
      ::mouse_event(item2,0,0,0,0);
      ::Sleep(30);
}

void CWinMoteDlg::OnSysCommand(UINT nID, LPARAM lParam)
{
      if ((nID & 0xFFF0) == IDM_ABOUTBOX)
      {
            CAboutDlg dlgAbout;
            dlgAbout.DoModal();
      }
      else
      {
            CDialog::OnSysCommand(nID, lParam);
      }
}

// If you add a minimize button to your dialog, you will need the
code below
```

Lampiran 4 : Listing Program WinMoteDlg.cpp (lanjutan)

```
//  to draw the icon.  For MFC applications using the
document/view model,
//  this is automatically done for you by the framework.

void CWinMoteDlg::OnPaint()
{
      if (IsIconic())
      {
            CPaintDC dc(this); // device context for painting
            SendMessage(WM_ICONERASEBKGND,
reinterpret_cast<WPARAM>(dc.GetSafeHdc()), 0);

            // Center icon in client rectangle
            int cxIcon = GetSystemMetrics(SM_CXICON);
            int cyIcon = GetSystemMetrics(SM_CYICON);
            CRect rect;
            GetClientRect(&rect);
            int x = (rect.Width() - cxIcon + 1) / 2;
            int y = (rect.Height() - cyIcon + 1) / 2;

            // Draw the icon
            dc.DrawIcon(x, y, m_hIcon);
      }
      else
      {
            CDialog::OnPaint();
      }
}

// The system calls this function to obtain the cursor to display
while the user drags
//  the minimized window.
HCURSOR CWinMoteDlg::OnQueryDragIcon()
{
      return static_cast<HCURSOR>(m_hIcon);
}


void CWinMoteDlg::OnBnClickedAdd()
{
      CListBox *pMainAppname = (CListBox*) GetDlgItem
      (IDC_APPNAME);
      //dapatkan jumlah item di list box application name
      int count = pMainAppname->GetCount();

      //cek apakah jumlah item sama dengan 10 atau tidak, jika
      tidak lanjut
      if (count != 10)
      {
      //buat dialog penambahan application
      CAddDlg adddlg;
      INT_PTR nRet = -1;
      nRet = adddlg.DoModal();

      switch ( nRet )
```

Lampiran 4 : Listing Program WinMoteDlg.cpp (lanjutan)

```
	{
		case -1:
			AfxMessageBox("Dialog box could not be
			created!");
			break;

		case IDOK:
			{
				//adddlg.app_name -> variable application
				name dari adddlg
				//tambahkan string ke list application
				name
				int no =
				pMainAppname->AddString(adddlg.app_name);

				int id = no; //masukkan nilai id untuk id
				string array

				//adddlg.app_path -> variable application
				path yang didapat dari adddlg

				//masukkan string ke array path, index ke
				id, sebanyak 1 kali
				path.InsertAt(id, adddlg.app_path, 1);

			}
			break;

		case IDCANCEL:
			break;

		default:
			break;
	};
	}
	//kalau jumlah application sudah 10 tidak bisa ditambah lagi
	else
	{
		AfxMessageBox("Maximum Application Reached, Please
		Remove One Item To Proceed!");
	}
}

void CWinMoteDlg::OnLbnSelchangeAppname()
{
	//dapatkan curent selection dari list box application name

	CListBox *pAppName = (CListBox*) GetDlgItem (IDC_APPNAME);
	int indexsel = pAppName->GetCurSel();


	//dapatkan string application path dari array string index
	ke indexsel
	nama = path.GetAt(indexsel);
```

Lampiran 4 : Listing Program WinMoteDlg.cpp (lanjutan)

```
//masukkan application path ke edit box application path
SetDlgItemText(IDC_APPPATH, nama);

//reset/refresh isi dari list hwnd

CListBox *pHwnd = (CListBox*) GetDlgItem (IDC_HWND);
pHwnd->ResetContent();
//jalankan fungsi enumwindow utk mendapatkan handle dari top
window yg aktif
EnumWindows (EnumWindowsProc, (LPARAM)this/*pointer to the
dialog box*/);

//dapatkan banyaknya jumlah item di list hwnd

int count = pHwnd->GetCount();

//kalau ada item di list box hwnd
if (count != LB_ERR)
{
      for (int i=0; i<count; i++)
      {
            //dapatkan panjang text dari item ke i list hwnd

            int len = pHwnd->GetTextLen(i);

            CString handle;
            //dapatkan text dari item ke i list hwnd

            pHwnd->GetText(i, handle.GetBuffer(len));


            //dapatkan panjang text dari item ke indexsel
            dari list appname
            int len2 = pAppName->GetTextLen(indexsel);

            CString title;
            //dapatkan text dari item ke indexsel dari list
            appname
            pAppName->GetText(indexsel,
            title.GetBuffer(len2));

            //jika string handle mengandung string title
            if (strstr(handle,title))
            {
                  if (startenable == TRUE)
                  {
                  //dapatkan handle window
                  wnd = ::FindWindowA(NULL, handle);
                  //tampilkan window
                  ::ShowWindow(wnd, SW_SHOWNORMAL);
                  //aktifkan window
                  ::SetForegroundWindow(wnd);
                  //berikan fokus keyboard & mouse ke window
                  ::SetFocus(wnd);
                  i = count;
```

Lampiran 4 : Listing Program WinMoteDlg.cpp (lanjutan)

```
                        }
                        i = count;
                    }
                }
        }
}
void CWinMoteDlg::OnBnClickedRemove()
{
        //dapatkan index dari item yang sedang dipilih
        CListBox *pAppName = (CListBox*) GetDlgItem (IDC_APPNAME);

        int indexsel = pAppName->GetCurSel();

        //kalau ada item yang dipilih di list box application name
        if (indexsel != LB_ERR)
        {
                //hapus string yang ada pada index yang ditunjuk

                pAppName->DeleteString(indexsel);

                //hapus application path dari array string pada index
                yg sama
                path.RemoveAt(indexsel);
                //isi nama dengan ""
                nama = path.GetAt(11);

                //hapus string application path saat application name
                dihapus dari list
                SetDlgItemText(IDC_APPPATH, nama);

        }
}

void CWinMoteDlg::OnBnClickedLaunch()
{
        //kalau application path ada isinya
        if (nama != "")
        {
                //jalankan proram dengan path nama
                ShellExecute(NULL, "open", nama, NULL, NULL,
SW_SHOWNORMAL);
        }
        //kalau application path tidak ada isinya
        else
        {
                AfxMessageBox("Nothing To Be Launch!");
        }
}

UINT thread(LPVOID p)
{
        int ret=1;
        char buff[50];
        char *result;
        char *result3;
```

Lampiran 4 : Listing Program WinMoteDlg.cpp (lanjutan)

```
char *result5;
CString data;
CString key;
CWinMoteDlg *dlg=(CWinMoteDlg*)AfxGetApp()->GetMainWnd();
//disable button add remote key
dlg->m_addremotekey.EnableWindow(FALSE);
//disable button start remote
dlg->m_start_remote.EnableWindow(FALSE);
//disable button stop remote
dlg->m_stop_remote.EnableWindow(FALSE);

//kalau koneksi winsock tidak terjadi error, lakukan looping
while(ret!=SOCKET_ERROR)
{
      //set pesan pada edit box notify
      dlg->SetDlgItemTextA(IDC_NOTIFY, "Press Remote Button
      n Click OK");
      ret = recv(dlg->s, buff, 50,0);

      //kalau koneksi winsock diputus, keluar
      if (ret == 0) break;

      //aktifkan dialog utama
      dlg->SetForegroundWindow();

      if (ret != SOCKET_ERROR)
      {
            buff[ret] = '\0';
            result=strpbrk(buff, " ");
            result++;
            int result2= (int) (result-buff);
            result3=strpbrk(result, " ");
            result3++;
            int result4= (int) (result3-buff);
            result5=strpbrk(result3, " ");
            result5++;
            int result6= (int) (result5-buff);
            data = buff;

            //dapatkan key remote

            key.GetBuffer(result6-result4-1);
            key.Format(data.Mid(result4,result6-result4-1));

            //set text remote key
            dlg->SetDlgItemText(IDC_EDITREMOTE,key);

      }
}

//tutup soket
closesocket(dlg->s);
//akhiri penggunaan WS2_32.dll
WSACleanup();
//akhiri thread
```

Lampiran 4 : Listing Program WinMoteDlg.cpp (lanjutan)

```
	AfxEndThread(0);
	return 0;
}


void CWinMoteDlg::OnBnClickedAddremote()
{
	WSADATA wsd1;
	CString m_sError;

	UpdateData(TRUE);

	if (WSAStartup(MAKEWORD(2,2), &wsd1) != 0)
	{
		AfxMessageBox("WSAStartup failed!");
		return;
	}

	s = socket(AF_INET, SOCK_STREAM, 0);
	if (s == INVALID_SOCKET)
	{
		m_sError.Format("socket() failed; %d",
			WSAGetLastError());
		AfxMessageBox(m_sError);
		return;
	}

	local.sin_family = AF_INET;
	local.sin_port = htons(8765);
	local.sin_addr.s_addr = inet_addr("127.0.0.1");

	if ( connect( s, (SOCKADDR*) &local, sizeof(local) ) ==
	SOCKET_ERROR)
	{
		m_sError.Format("Cannot Connect To WinLirc Server",
		WSAGetLastError());
		AfxMessageBox(m_sError);
		return;
	}

	AfxBeginThread(&thread,0);

}

void CWinMoteDlg::OnBnClickedOk()
{
	// TODO: Add your control notification handler code here
}

void CWinMoteDlg::OnBnClickedAddhkey()
{

	//dapatkan index baris yang dipilih
	CListCtrl *pMainRemote = (CListCtrl*) GetDlgItem
	(IDC_REMOTEKEY);
```

Lampiran 4 : Listing Program WinMoteDlg.cpp (lanjutan)

```
	int selected = pMainRemote->GetSelectionMark();

	//kalau ada baris yang dipilih
	if (selected != -1)
	{
		//jalankan dialogbox addhotkey
		CAddHot addhotkey;
		INT_PTR nRet = -1;
		nRet = addhotkey.DoModal();

		switch ( nRet )
		{
			case -1:
				AfxMessageBox("Dialog box could not be
				created!");
				break;

			case IDOK:
				{
					CString hotkey =
					addhotkey.hotkeyString;

					//isi kolom hotkey dengan string
					hotkey
					pMainRemote->SetItemText(selected,
					1, hotkey);

					//set data item dengan word wHotkey
					pMainRemote->SetItemData(selected,
					addhotkey.wHotkey);
				}
			break;

			case IDCANCEL:
				break;

			default:
				// Do something
				break;
		};
	}
	else
	{
		AfxMessageBox("Please Select Remote Key In The
		List!");
	}
}

void CWinMoteDlg::OnBnClickedCancel()
{
	// TODO: Add your control notification handler code here
	closesocket(s);
	WSACleanup();
	OnCancel();
}
```

Lampiran 4 : Listing Program WinMoteDlg.cpp (lanjutan)

```
void CWinMoteDlg::OnLvnItemchangedRemotekey(NMHDR *pNMHDR, LRESULT
*pResult)
{

}

BOOL CALLBACK CWinMoteDlg::EnumWindowsProc (HWND hWnd, LPARAM
lParam)
{
      CWinMoteDlg *pDlg = (CWinMoteDlg *)lParam;

      int nItem = 0;

      //Make sure that the window is visible
      TCHAR szWindowText [MAX_PATH];
      if (!::IsWindowVisible (hWnd))
            return TRUE;

      //Get the text on the title bar
      ::GetWindowText (hWnd, szWindowText, MAX_PATH);

      //If the window is Process Manager than don't display it
      if (_tcsstr (_T("Program Manager"), szWindowText))
            return TRUE;

      //Add the info to the list control
      nItem = pDlg->m_hwnd.InsertString (0, szWindowText);

      return TRUE;
}
void CWinMoteDlg::OnBnClickedStart()
{
      WSADATA wsd2;
      CString m_sError;
      startenable = TRUE;

      UpdateData(TRUE);

      if (WSAStartup(MAKEWORD(2,2), &wsd2) != 0)
      {
            AfxMessageBox("WSAStartup failed!");
            return;
      }

      s = socket(AF_INET, SOCK_STREAM, 0);
      if (s == INVALID_SOCKET)
      {
            m_sError.Format("socket() failed; %d",
                  WSAGetLastError());
            AfxMessageBox(m_sError);
            return;
      }

      local.sin_family = AF_INET;
      local.sin_port = htons(8765);
```

Lampiran 4 : Listing Program WinMoteDlg.cpp (lanjutan)

```
	local.sin_addr.s_addr = inet_addr("127.0.0.1");

	if ( connect( s, (SOCKADDR*) &local, sizeof(local) ) ==
	SOCKET_ERROR)
	{
		m_sError.Format("Cannot Connect To WinLirc Server",
		WSAGetLastError());
		AfxMessageBox(m_sError);
		return;
	}

	AfxBeginThread(&thread2,0);
}

void CWinMoteDlg::OnBnClickedStop()
{
	startenable = FALSE;
	//enable button add remote key
	m_addremotekey.EnableWindow(TRUE);
	//enable button start remote
	m_start_remote.EnableWindow(TRUE);
	//enable button cancel
	m_cancel.EnableWindow(TRUE);
	closesocket(s);
	WSACleanup();
}

void CWinMoteDlg::OnBnClickedAddspecial()
{
	//dapatkan index item remote key yang dipilih
	CListCtrl *pMainRemote = (CListCtrl*) GetDlgItem
	(IDC_REMOTEKEY);
	int selected = pMainRemote->GetSelectionMark();

	//kalau ada item yang dipilih
	if (selected != -1)
	{
		//tampilkan dialog addspecial
		CAddSpecial addspecial;
		INT_PTR nRet = -1;
		nRet = addspecial.DoModal();

		switch ( nRet )
		{
			case -1:
				AfxMessageBox("Dialog box could not be
				created!");
				break;

			case IDOK:
				{
					CString hotkey = addspecial.keyname;
					if (hotkey == "Mouse Left Click")
					{
```

Lampiran 4 : Listing Program WinMoteDlg.cpp (lanjutan)

```
pMainRemote->
SetItemText(selected, 1, hotkey);

pMainRemote->
SetItemData(selected,
      addspecial.dflag);
} else
if (hotkey == "Mouse Right Click")
{

pMainRemote->
SetItemText(selected, 1, hotkey);

pMainRemote->
SetItemData(selected,
      addspecial.dflag);
} else
if (hotkey == "Mouse Move Up")
{

pMainRemote->
SetItemText(selected, 1, hotkey);

pMainRemote->
SetItemData(selected,
      addspecial.dflag);
} else
if (hotkey == "Mouse Move Down")
{
      pMainRemote->
      SetItemText(selected, 1,
             hotkey);

      pMainRemote->
      SetItemData(selected,
            addspecial.dflag);
} else
if (hotkey == "Mouse Move Left")
{

pMainRemote->
SetItemText(selected, 1, hotkey);

pMainRemote->
SetItemData(selected,
      addspecial.dflag);
} else
if (hotkey == "Mouse Move Right")
{

pMainRemote->
SetItemText(selected, 1, hotkey);

pMainRemote->
SetItemData(selected,
```

Lampiran 4 : Listing Program WinMoteDlg.cpp (lanjutan)

```
                                    addspecial.dflag);
                              }
                              else
                              {

                              pMainRemote->
                              SetItemText(selected, 1, hotkey);

                              pMainRemote->
                              SetItemData(selected,
                                    addspecial.vkey);
                              }
                        }
                  break;

                  case IDCANCEL:
                        break;

                  default:
                        // Do something
                        break;
            };
      }
      else
      {
            AfxMessageBox("Please Select Remote Key In The
            List!");
      }
}

void CWinMoteDlg::OnBnClickedClear()
{
      CListCtrl *pMainRemote = (CListCtrl*) GetDlgItem
      (IDC_REMOTEKEY);
      int selected = pMainRemote->GetSelectionMark();

      //kalau ada item yang dipilih
      if (selected != -1)
      {
            //hapus item yang dipilih
            pMainRemote->DeleteItem(selected);
      }
      else
      {
            AfxMessageBox("Please Select Remote Key In The
            List!");
      }

}

void CWinMoteDlg::OnBtnClickedOk()
{
      CString remote;
      CEdit *pRemoteKey = (CEdit*)GetDlgItem(IDC_EDITREMOTE);
      int textlen = pRemoteKey->LineLength(0);
```

Lampiran 4 : Listing Program WinMoteDlg.cpp (lanjutan)

```
      if (textlen != 0)
      {
            //ambil remote key dari edit control
            pRemoteKey->GetLine(0, remote.GetBuffer(textlen),
            textlen);
            m_addremotekey.EnableWindow(TRUE);
            m_start_remote.EnableWindow(TRUE);
            m_stop_remote.EnableWindow(TRUE);

            CEdit *pNotify = (CEdit*)GetDlgItem(IDC_NOTIFY);
            SetDlgItemTextA(IDC_NOTIFY, "");        //bersihkan
            tulisan notifikasi "please add remote key"

            SetDlgItemTextA(IDC_EDITREMOTE, "");    //bersihkan
            tulisan remote key

            //masukkan remote key ke item list ctrl
            CListCtrl *pMainRemote =
            (CListCtrl*)GetDlgItem(IDC_REMOTEKEY);
            int count = pMainRemote->GetItemCount();
            pMainRemote->InsertItem(count, remote);

            closesocket(s);
            WSACleanup();
      }
}

void CWinMoteDlg::OnBtnClickedCancel()
{
      // TODO: Add your control notification handler code here
      closesocket(s);
      WSACleanup();
      m_addremotekey.EnableWindow(TRUE);
      m_start_remote.EnableWindow(TRUE);
      m_stop_remote.EnableWindow(TRUE);
      SetDlgItemTextA(IDC_NOTIFY, "");
      SetDlgItemTextA(IDC_EDITREMOTE, "");
}

void CWinMoteDlg::OnBnClickedClearall()
{
      CListCtrl *pMainRemote = (CListCtrl*) GetDlgItem
      (IDC_REMOTEKEY);
      pMainRemote->DeleteAllItems();
}

void CWinMoteDlg::OnFileExit()
{
      closesocket(s);
      WSACleanup();
      CWinMoteDlg::EndDialog(1);
}

void CWinMoteDlg::OnHelpAboutwinmote()
{
      CAboutDlg().DoModal();    //keluarkan aboutbox}
```

Lampiran 5 : Listing Program AddDlg.h

```
#pragma once
#include "afxwin.h"


// CAddDlg dialog

class CAddDlg : public CDialog
{
	DECLARE_DYNAMIC(CAddDlg)

public:
	CAddDlg(CWnd* pParent = NULL);   // standard constructor
	virtual ~CAddDlg();

// Dialog Data
	enum { IDD = IDD_ADD };

protected:
	virtual void DoDataExchange(CDataExchange* pDX);    //
DDX/DDV support

	DECLARE_MESSAGE_MAP()
public:
	CEdit m_add_appname1;
public:
	CEdit m_add_apppath1;
public:
	afx_msg void OnBnClickedOk();
public:
	afx_msg void OnBnClickedBrowse();
public:
	CString filetitle;
public:
	CString app_path;
public:
	CString app_name;
};
```

Lampiran 6 : Listing Program AddDlg.cpp

```
// AddDlg.cpp : implementation file
//

#include "stdafx.h"
#include "WinMote.h"
#include "AddDlg.h"


// CAddDlg dialog

IMPLEMENT_DYNAMIC(CAddDlg, CDialog)

CAddDlg::CAddDlg(CWnd* pParent /*=NULL*/)
	: CDialog(CAddDlg::IDD, pParent)
	, filetitle(_T(""))
	, app_path(_T(""))
	, app_name(_T(""))
{

}

CAddDlg::~CAddDlg()
{
}

void CAddDlg::DoDataExchange(CDataExchange* pDX)
{
	CDialog::DoDataExchange(pDX);
	DDX_Control(pDX, IDC_APPNAME, m_add_appname1);
	DDX_Control(pDX, IDC_APPPATH, m_add_apppath1);
}


BEGIN_MESSAGE_MAP(CAddDlg, CDialog)
	ON_BN_CLICKED(IDOK, &CAddDlg::OnBnClickedOk)
	ON_BN_CLICKED(IDC_BROWSE, &CAddDlg::OnBnClickedBrowse)
END_MESSAGE_MAP()


// CAddDlg message handlers

void CAddDlg::OnBnClickedOk()
{
	CString text;
	CString text2;
	int nRet = IDOK;


	CEdit *pAppName = (CEdit*) GetDlgItem (IDC_APPNAME);
	int textlen = pAppName->LineLength();


	pAppName->GetLine(0, text.GetBuffer(textlen), textlen);
```

Lampiran 6 : Listing Program AddDlg.cpp (lanjutan)

```
	CEdit *pAppPath = (CEdit*) GetDlgItem (IDC_APPPATH);
	int textlen2 = pAppPath->LineLength();


	pAppPath->GetLine(0, text2.GetBuffer(textlen2), textlen2);

	if (text == "" || text2 == "")
	{
		AfxMessageBox("Application Name & Path musn't be
		empty!");
	}
	else
	{
		app_name.GetBuffer(textlen);
		app_name.Format(text);
		EndDialog(nRet);
	}
}

void CAddDlg::OnBnClickedBrowse()
{
	CFileDialog fileDlg (TRUE, "exe", "*.exe",
		OFN_FILEMUSTEXIST| OFN_HIDEREADONLY,
		"Executable(*.exe)|*.exe|All Files (*.*)|*.*||",
		this);

	//fileDlg.DoModal();
	if( fileDlg.DoModal ()==IDOK )
	{
		CString path = fileDlg.GetPathName();
		filetitle = fileDlg.GetFileTitle();
		app_path = path;
		CEdit *pAppPath = (CEdit*) GetDlgItem (IDC_APPPATH);
		SetDlgItemTextA(IDC_APPPATH, path);
	}

}
```

Lampiran 7 : Listing Program AddHot.h

```
#pragma once
#include "afxcmn.h"


// CAddHot dialog

class CAddHot : public CDialog
{
	DECLARE_DYNAMIC(CAddHot)

public:
	CAddHot(CWnd* pParent = NULL);   // standard constructor
	virtual ~CAddHot();

// Dialog Data
	enum { IDD = IDD_ADD_HOT };

protected:
	virtual void DoDataExchange(CDataExchange* pDX);    //
DDX/DDV support

	DECLARE_MESSAGE_MAP()
public:
	afx_msg void OnBnClickedOk();
public:
	CHotKeyCtrl m_hotkey;
public:
	CString hotkeyString;
public:
	WORD wHotkey;
};
```

Lampiran 8 : Listing Program AddHot.cpp

```
// AddHot.cpp : implementation file
//

#include "stdafx.h"
#include "WinMote.h"
#include "AddHot.h"


// CAddHot dialog

IMPLEMENT_DYNAMIC(CAddHot, CDialog)

CAddHot::CAddHot(CWnd* pParent /*=NULL*/)
	: CDialog(CAddHot::IDD, pParent)
	, hotkeyString(_T(""))
{

}

CAddHot::~CAddHot()
{
}

void CAddHot::DoDataExchange(CDataExchange* pDX)
{
	CDialog::DoDataExchange(pDX);
	DDX_Control(pDX, IDC_HOTKEY1, m_hotkey);
}


BEGIN_MESSAGE_MAP(CAddHot, CDialog)
	ON_BN_CLICKED(IDOK, &CAddHot::OnBnClickedOk)
END_MESSAGE_MAP()


// CAddHot message handlers

void CAddHot::OnBnClickedOk()
{
	int nRet = IDOK;

	/* WORD */ wHotkey = (WORD)::SendMessage(m_hotkey,
	HKM_GETHOTKEY, 0, 0);
//	LOBYTE(wHotkey) = virtual key codes
//	HIBYTE(wHotkey) = modifier flag (alt, shift, ctrl, extended)
	hotkeyString = m_hotkey.GetHotKeyName();
	EndDialog(nRet);
}
```

Lampiran 9 : Listing Program AddSpecial.h

```
#pragma once
#include "afxwin.h"


// CAddSpecial dialog

class CAddSpecial : public CDialog
{
	DECLARE_DYNAMIC(CAddSpecial)

public:
	CAddSpecial(CWnd* pParent = NULL);   // standard constructor
	virtual ~CAddSpecial();

// Dialog Data
	enum { IDD = IDD_ADDSPECIAL };

protected:
	virtual void DoDataExchange(CDataExchange* pDX);    //
DDX/DDV support
	virtual BOOL OnInitDialog();

	DECLARE_MESSAGE_MAP()
public:
	CComboBox m_combo;
public:
	afx_msg void OnCbnSelchangeCombo();
public:
	afx_msg void OnBnClickedOk();
public:
	BYTE vkey;
public:
	CString keyname;
public:
	DWORD dflag;
};
```

Lampiran 10 : Listing Program AddSpecial.cpp

```
// AddSpecial.cpp : implementation file
//

#include "stdafx.h"
#include "WinMote.h"
#include "AddSpecial.h"


// CAddSpecial dialog

IMPLEMENT_DYNAMIC(CAddSpecial, CDialog)

CAddSpecial::CAddSpecial(CWnd* pParent /*=NULL*/)
	: CDialog(CAddSpecial::IDD, pParent)
	, vkey(0)
	, keyname(_T(""))
	, dflag(0)
{

}

CAddSpecial::~CAddSpecial()
{
}

void CAddSpecial::DoDataExchange(CDataExchange* pDX)
{
	CDialog::DoDataExchange(pDX);
	DDX_Control(pDX, IDC_COMBO, m_combo);
}

BOOL CAddSpecial::OnInitDialog()
{
	CDialog::OnInitDialog();
	m_combo.InsertString(0,"Tab");
	m_combo.InsertString(1,"Esc");
	m_combo.InsertString(2,"Backspace");
	m_combo.InsertString(3,"Enter");
	m_combo.InsertString(4,"Windows Key");
	m_combo.InsertString(5,"Space");
	m_combo.InsertString(6,"Del");
	m_combo.InsertString(7,"Mouse Left Click");
	m_combo.InsertString(8,"Mouse Right Click");
	m_combo.InsertString(9,"Mouse Move Up");
	m_combo.InsertString(10,"Mouse Move Down");
	m_combo.InsertString(11,"Mouse Move Left");
	m_combo.InsertString(12,"Mouse Move Right");

	return TRUE;
}
BEGIN_MESSAGE_MAP(CAddSpecial, CDialog)
	ON_CBN_SELCHANGE(IDC_COMBO,
	&CAddSpecial::OnCbnSelchangeCombo)
	ON_BN_CLICKED(IDOK, &CAddSpecial::OnBnClickedOk)
END_MESSAGE_MAP()
```

Lampiran 10 : Listing Program AddSpecial.cpp (lanjutan)

```
// CAddSpecial message handlers

void CAddSpecial::OnCbnSelchangeCombo()
{
	int sel = m_combo.GetCurSel();
	switch (sel)
	{
		case 0:
			{
				int len = m_combo.GetLBTextLen(0);
				m_combo.GetLBText(0,
				keyname.GetBuffer(len));
				vkey = 0x09;
			}
			break;

		case 1:
			{
				int len = m_combo.GetLBTextLen(1);
				m_combo.GetLBText(1,
				keyname.GetBuffer(len));
				vkey = 0x1B;
			}
			break;

		case 2:
			{
				int len = m_combo.GetLBTextLen(2);
				m_combo.GetLBText(2,
				keyname.GetBuffer(len));
				vkey = 0x08;
			}
			break;

		case 3:
			{
				int len = m_combo.GetLBTextLen(3);
				m_combo.GetLBText(3,
				keyname.GetBuffer(len));
				vkey = 0x0D;
			}
			break;

		case 4:
			{
				int len = m_combo.GetLBTextLen(4);
				m_combo.GetLBText(4,
				keyname.GetBuffer(len));
				vkey = 0x5B;
			}
			break;

		case 5:
			{
				int len = m_combo.GetLBTextLen(5);
```

Lampiran 10 : Listing Program AddSpecial.cpp (lanjutan)

```
                m_combo.GetLBText(5,
                keyname.GetBuffer(len));
                vkey = 0x20;
        }
        break;

case 6:
        {
                int len = m_combo.GetLBTextLen(6);
                m_combo.GetLBText(6,
                keyname.GetBuffer(len));
                vkey = 0x2E;
        }
        break;

case 7:
        {
                int len = m_combo.GetLBTextLen(7);
                m_combo.GetLBText(7,
                keyname.GetBuffer(len));
                dflag = MOUSEEVENTF_LEFTDOWN;
        }
        break;

case 8:
        {
                int len = m_combo.GetLBTextLen(8);
                m_combo.GetLBText(8,
                keyname.GetBuffer(len));
                dflag = MOUSEEVENTF_RIGHTDOWN;
        }
        break;

case 9:
        {
                int len = m_combo.GetLBTextLen(9);
                m_combo.GetLBText(9,
                keyname.GetBuffer(len));
                dflag = MOUSEEVENTF_MOVE;
        }
        break;

case 10:
        {
                int len = m_combo.GetLBTextLen(10);
                m_combo.GetLBText(10,
                keyname.GetBuffer(len));
                dflag = MOUSEEVENTF_MOVE;
        }
        break;

case 11:
        {
                int len = m_combo.GetLBTextLen(11);
```

Lampiran 10 : Listing Program AddSpecial.cpp (lanjutan)

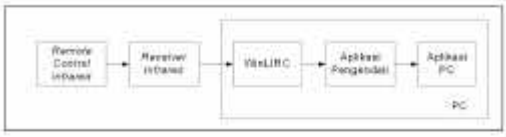
```
                        m_combo.GetLBText(11,
                        keyname.GetBuffer(len));
                        dflag = MOUSEEVENTF_MOVE;
                  }
                  break;

            case 12:
                  {
                        int len = m_combo.GetLBTextLen(12);
                        m_combo.GetLBText(12,
                        keyname.GetBuffer(len));
                        dflag = MOUSEEVENTF_MOVE;
                  }
                  break;
      };
}

void CAddSpecial::OnBnClickedOk()
{
      int nRet = IDOK;
      EndDialog(nRet);
}
```

Lampiran 11 : Draft Poster Tugas Akhir

PENGENDALIAN PC MELALUI INFRARED MEMANFAATKAN SOFTWARE OPEN SOURCE "WinLIRC"

Nama / NRP : Donny Nugroho / 23401019
E-mail : sunstrider_d@yahoo.com
Pembimbing 1 : Petrus Santoso, S.T, M.Sc.
Pembimbing 2 : Ir.Resmana Lim, M.Eng.

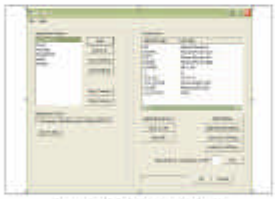
Dengan teknologi infrared, pengendalian perangkat elektronik dapat dilakukan melalui jarak yang cukup jauh. Perangkat elektronik yang dapat dikontrol melalui infrared tidak hanya player VCD atau TV tetapi juga komputer.



Gambar Blok Diagram Sistem

Gambar Receiver Infrared

Pada Tugas Akhir ini dibuat sistem untuk mengendalikan aplikasi komputer berbasiskan remote kontrol infrared. Perancangan sistem melibatkan receiver infrared dan sebuah software client untuk mengendalikan aplikasi komputer. Software server open source WinLIRC digunakan untuk menerjemahkan sinyal infrared dari receiver ke dalam bentuk teks dan mengirimnya ke software client. Client kemudian menerjemahkannya menjadi perintah penekanan tombol keyboard atau mouse untuk mengendalikan aplikasi komputer yang diinginkan.



Gambar Software Client

Tabel Pengujian Sensitivitas Remote Sony Menggunakan RM-33E+

Jarak	Tingkat Keberhasilan		
	Sudut 0°	Sudut 45°	Sudut 90°
1 m	100 %	100 %	100 %
2 m	100 %	100 %	100 %
3 m	100 %	100 %	100 %
4 m	100 %	100 %	100 %
5 m	100 %	100 %	100 %
6 m	100 %	100 %	100 %
7 m	100 %	100 %	100 %
8 m	100 %	100 %	80 %
9 m	100 %	100 %	80 %
10 m	100 %	90 %	80 %
11 m	100 %	70 %	20 %
12 m	100 %	70 %	0 %
13 m	100 %	70 %	0 %
14 m	100 %	60 %	0 %
15 m	100 %	60 %	0 %
16 m	90 %	40 %	0 %
17 m	90 %	0 %	0 %
18 m	90 %	0 %	0 %
19 m	80 %	0 %	0 %
20 m	80 %	0 %	0 %
21 m	80 %	0 %	0 %
22 m	60 %	0 %	0 %
23 m	70 %	0 %	0 %
24 m	60 %	0 %	0 %
25 m	50 %	0 %	0 %
26 m	30 %	0 %	0 %
27 m	20 %	0 %	0 %
28 m	10 %	0 %	0 %
29 m	10 %	0 %	0 %
30 m	0 %	0 %	0 %

Tabel Pengujian Sensitivitas Remote Gadmei RM009A

Jarak	Tingkat Keberhasilan		
	Sudut 0°	Sudut 45°	Sudut 90°
1 m	100 %	100 %	100 %
2 m	100 %	100 %	100 %
3 m	100 %	100 %	100 %
4 m	100 %	100 %	100 %
5 m	100 %	100 %	100 %
6 m	100 %	100 %	100 %
7 m	100 %	100 %	100 %
8 m	100 %	100 %	100 %
9 m	100 %	100 %	80 %
10 m	100 %	80 %	70 %
11 m	100 %	80 %	20 %
12 m	100 %	80 %	0 %
13 m	100 %	80 %	0 %
14 m	100 %	70 %	0 %
15 m	90 %	40 %	0 %
16 m	90 %	20 %	0 %
17 m	80 %	0 %	0 %
18 m	80 %	0 %	0 %
19 m	60 %	0 %	0 %
20 m	50 %	0 %	0 %
21 m	30 %	0 %	0 %
22 m	30 %	0 %	0 %
23 m	30 %	0 %	0 %
24 m	30 %	0 %	0 %
25 m	20 %	0 %	0 %
26 m	20 %	0 %	0 %
27 m	20 %	0 %	0 %
28 m	10 %	0 %	0 %
29 m	0 %	0 %	0 %
30 m	0 %	0 %	0 %

Berdasarkan hasil pengujian, receiver yang dibuat mampu menerima sinyal infrared sampai dengan jarak 10 meter, sehingga layak untuk dipakai. Sedangkan software client yang dibuat sudah diuji di atas dua platform, yaitu Windows 98 dan Windows XP Pro SP2. Pengujian membuktikan bahwa software client dapat mengendalikan aplikasi komputer walaupun masih terdapat beberapa bug, seperti tampilan yang tidak dikehendaki, dan terjadinya crash, terutama saat dijalankan di Windows 98.

Jurusan Teknik Elektro
Fakultas Teknologi Industri
Universitas Kristen Petra