

TUGAS
PENGANTAR GRAFIK KOMPUTER DAN OLAH CITRA

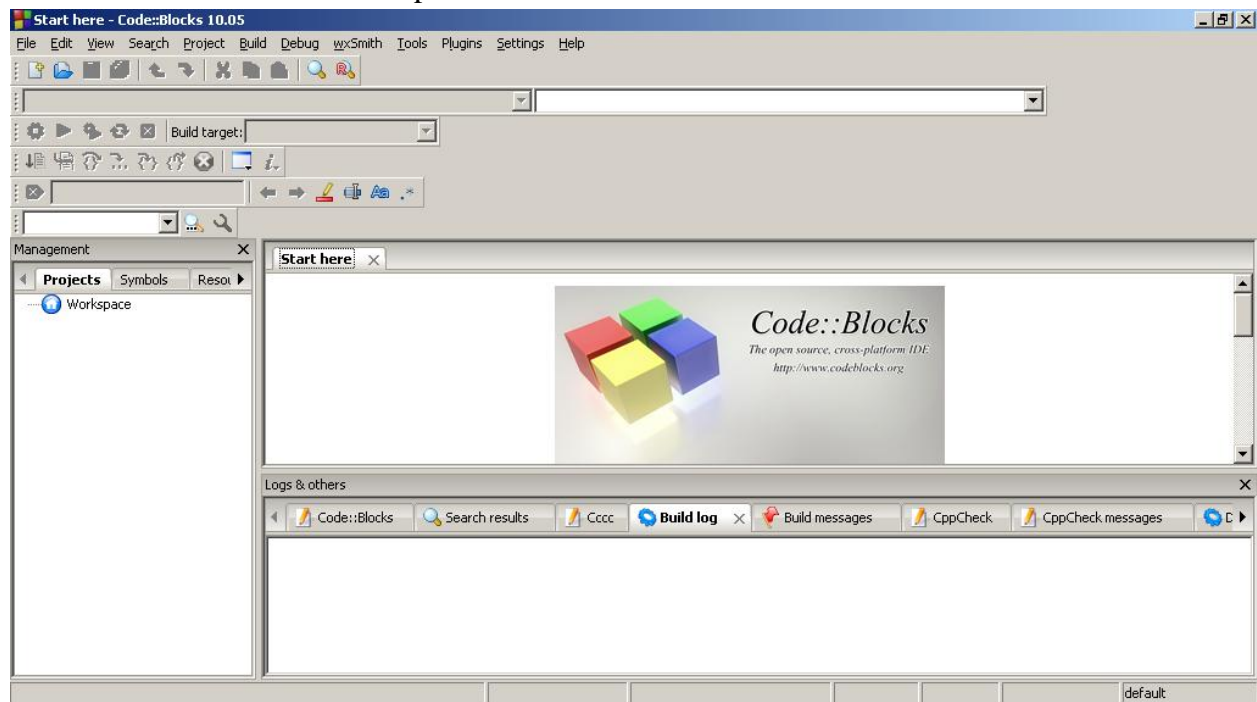


NAMA : HERU SANJAYA

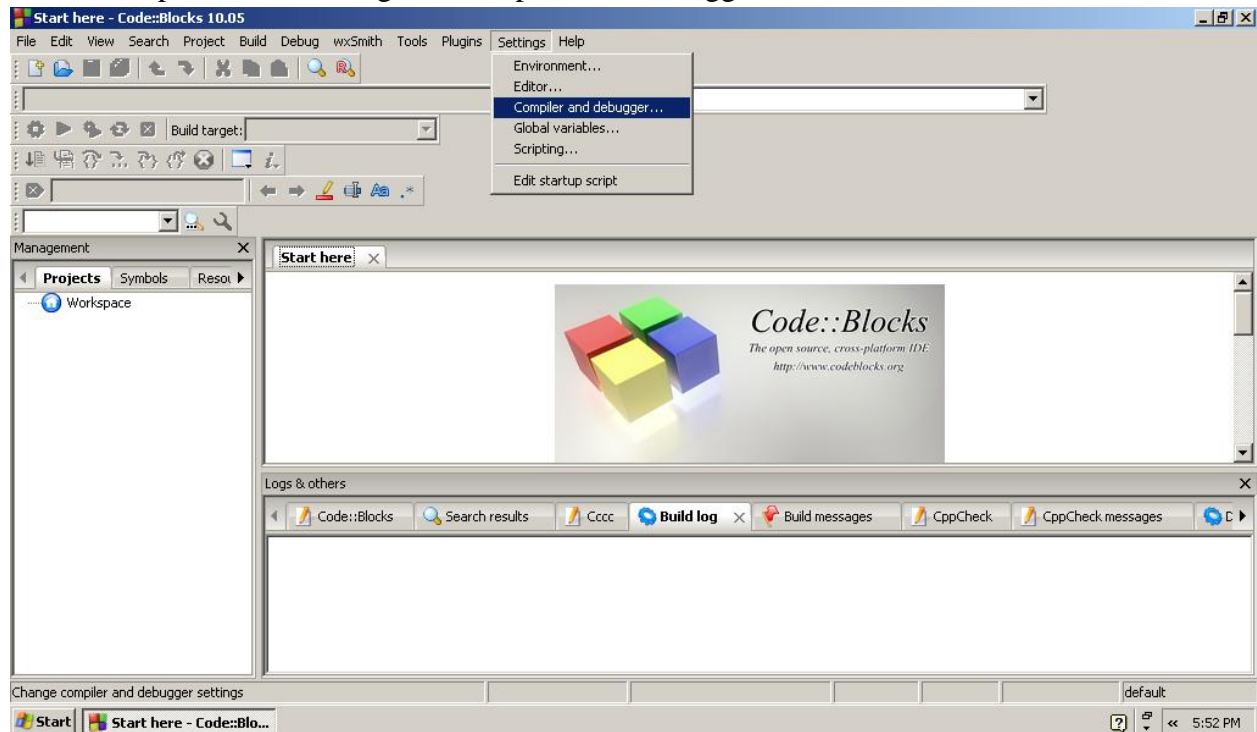
NPM : 39110387

KELAS : 2DB23

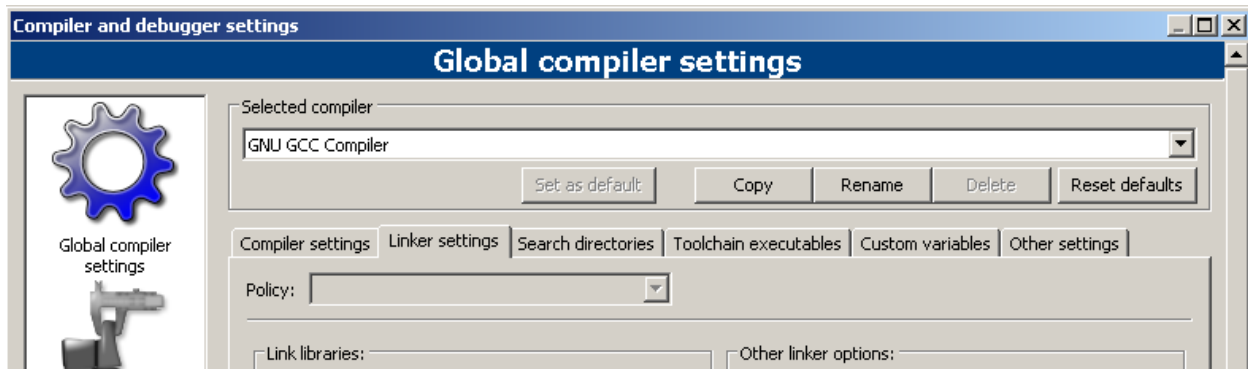
Gambar dibawah ini adalah tampilan awal Code Blocks.



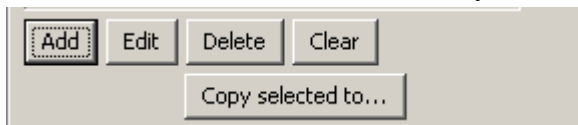
Setelah itu, pilih menu Settings → Compiler and Debugger.



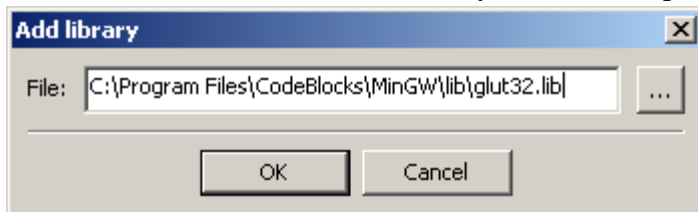
Maka akan muncul tampilan berikutnya, yaitu Global Compiler Settings. Pilih Linker Settings, contohnya seperti gambar dibawah ini.



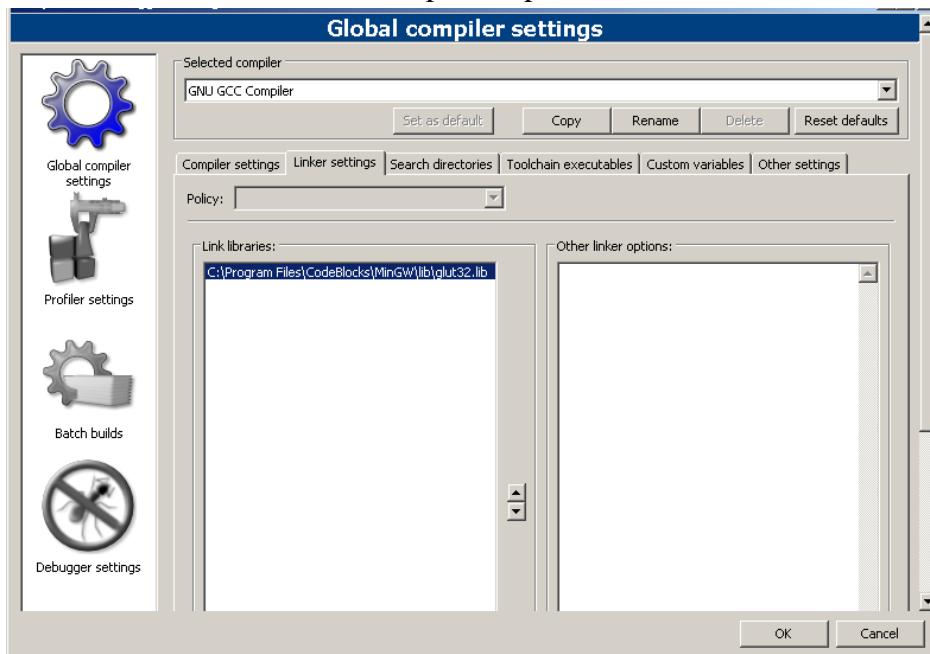
Klik Add untuk menambahkan library, contohnya seperti gambar dibawah ini.



Maka akan muncul kotak Add Library, lalu ketik seperti gambar dibawah ini.

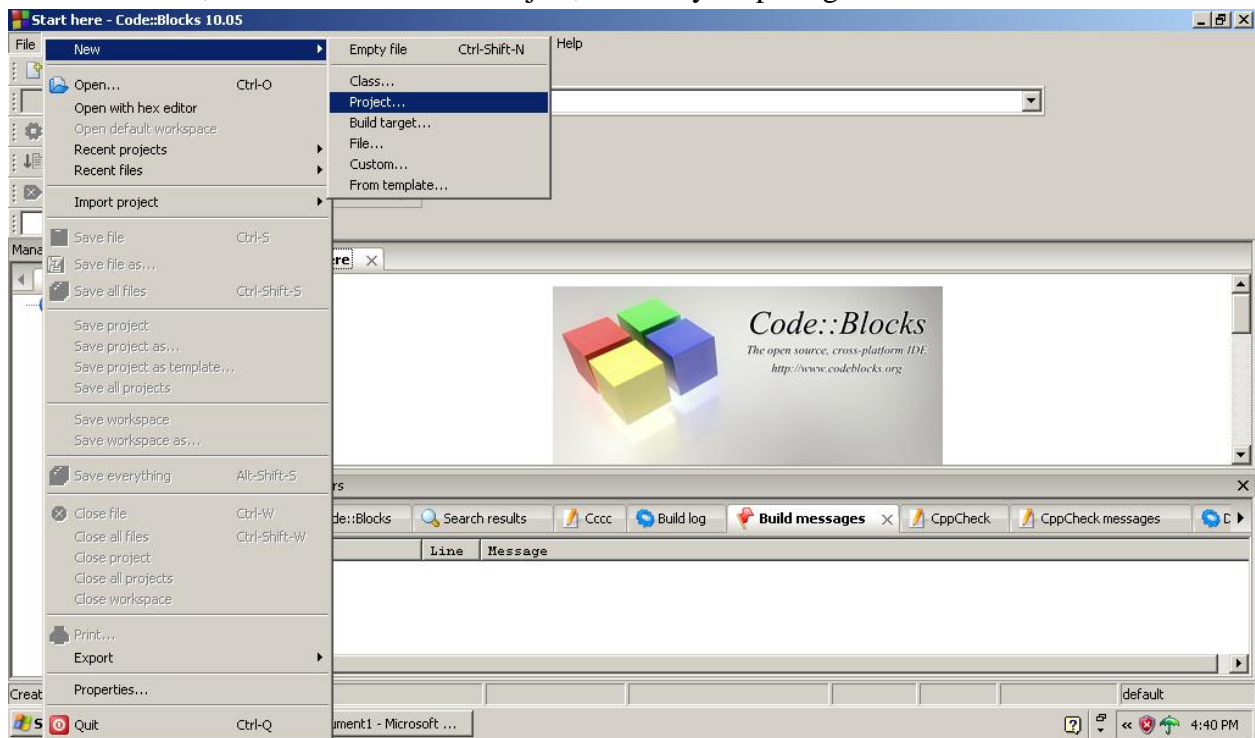


Klik OK, maka akan muncul tampilan seperti berikut ini, lalu klik Ok.

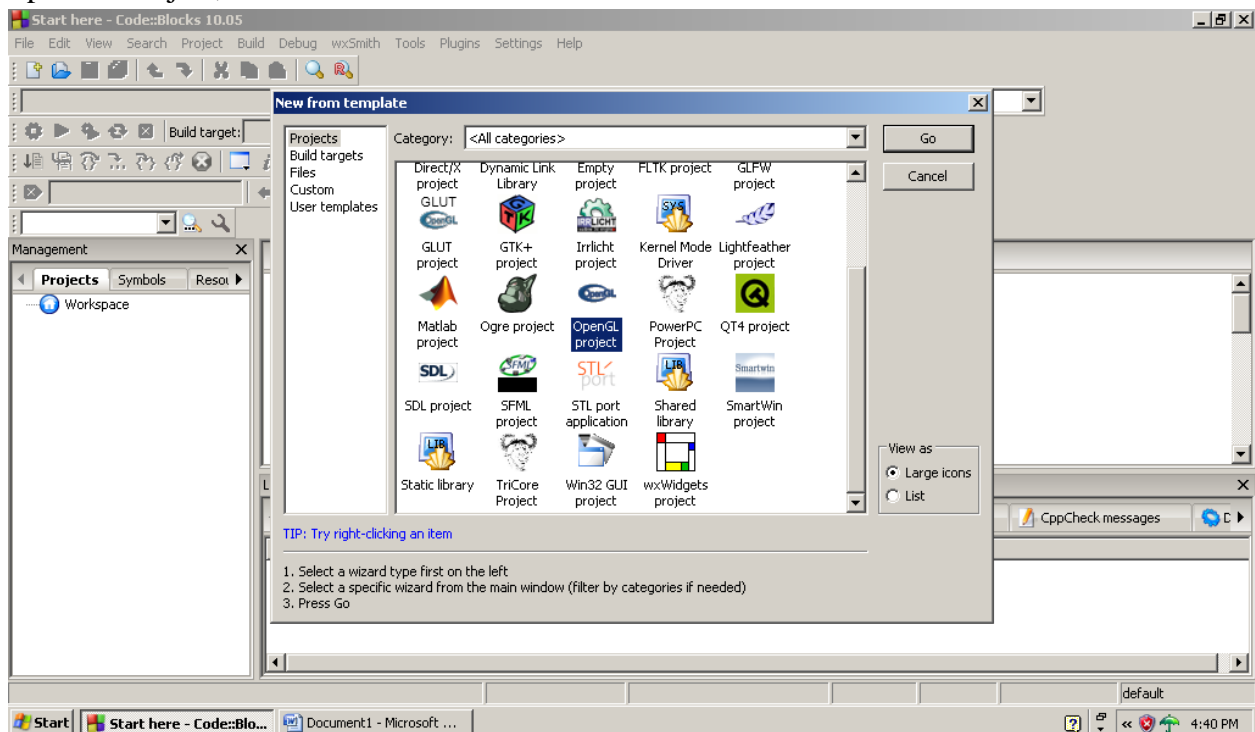


Latihan 1.

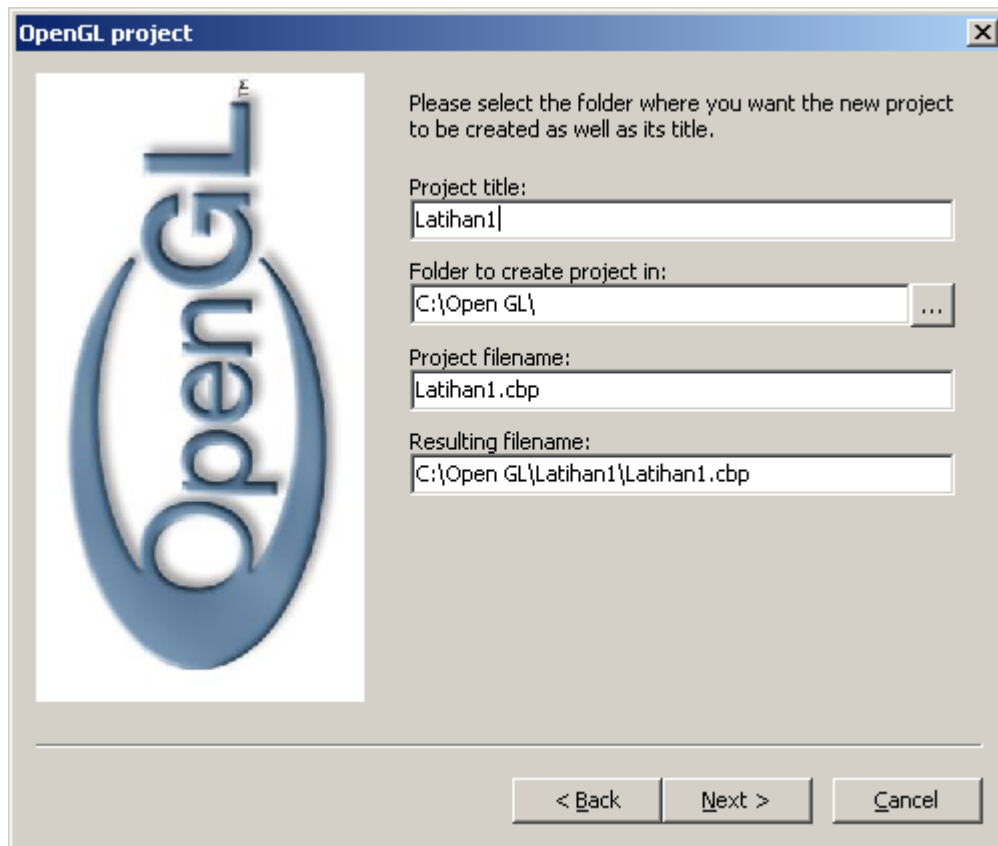
Untuk memulai, klik File → New → Project, contohnya seperti gambar dibawah ini.



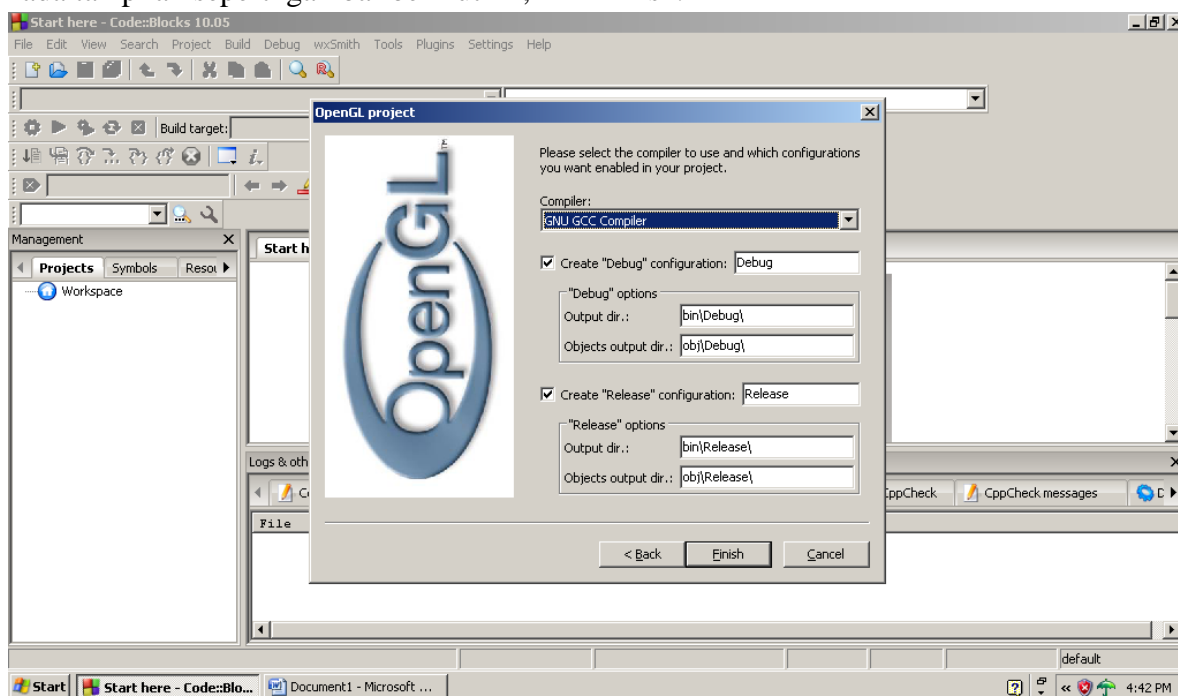
Setelah itu akan muncul tampilan New From Template seperti gambar dibawah ini, pilih OpenGL Project, lalu klik Go.



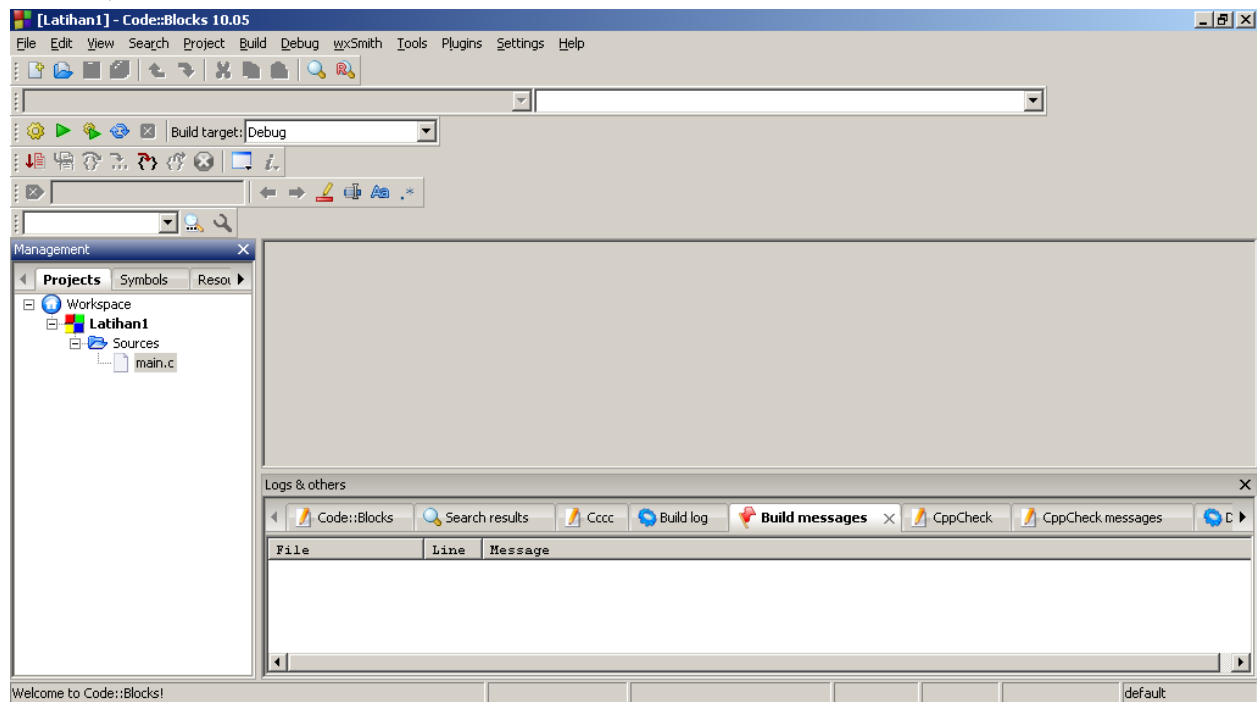
Maka akan muncul tampilan OpenGL Project seperti gambar dibawah ini. Pada Project Title, isi Latihan1, lalu klik Next.



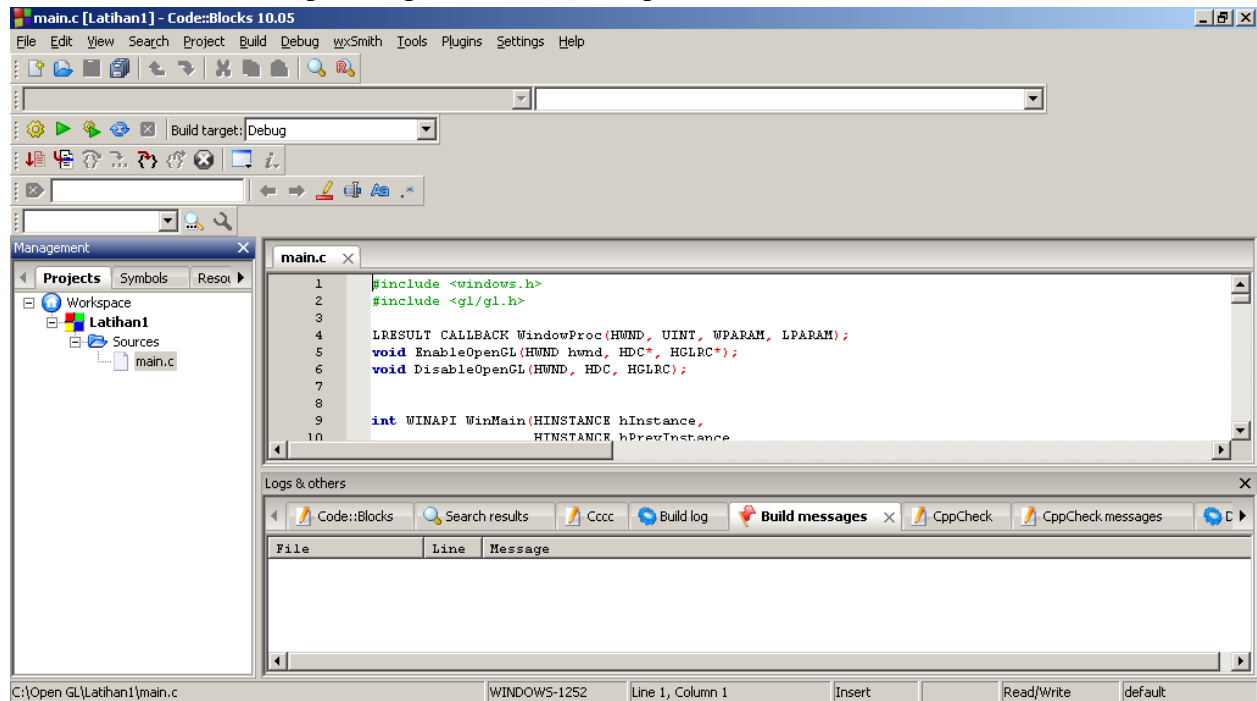
Pada tampilan seperti gambar berikut ini, klik Finish.



Maka, pada kotak Management, akan muncul tampilan seperti berikut ini. Klik dua kali pada Sources, lalu klik dua kali main.c.



Maka akan muncul tampilan seperti berikut, isi seperti berikut ini.



```

#include <windows.h>
#include <gl/gl.h>

LRESULT CALLBACK WindowProc(HWND, UINT, WPARAM, LPARAM);
void EnableOpenGL(HWND hwnd, HDC*, HGLRC*);
void DisableOpenGL(HWND, HDC, HGLRC);

int WINAPI WinMain(HINSTANCE hInstance,
                  HINSTANCE hPrevInstance,
                  LPSTR lpCmdLine,
                  int nCmdShow)
{
    WNDCLASSEX wcex;
    HWND hwnd;
    HDC hDC;
    HGLRC hRC;
    MSG msg;
    BOOL bQuit = FALSE;
    float theta = 0.0f;

    /* register window class */
    wcex.cbSize = sizeof(WNDCLASSEX);
    wcex.style = CS_OWNDL;
    wcex.lpfnWndProc = WindowProc;
    wcex.cbClsExtra = 0;
    wcex.cbWndExtra = 0;
    wcex.hInstance = hInstance;
    wcex.hIcon = LoadIcon(NULL, IDI_APPLICATION);
    wcex.hCursor = LoadCursor(NULL, IDC_ARROW);
    wcex.hbrBackground = (HBRUSH)GetStockObject(BLACK_BRUSH);
    wcex.lpszMenuName = NULL;
    wcex.lpszClassName = "GLSample";
    wcex.hIconSm = LoadIcon(NULL, IDI_APPLICATION);

    if (!RegisterClassEx(&wcex))
        return 0;

    /* create main window */

```

```

hwnd = CreateWindowEx(0,
    "GLSample",
    "OpenGL Sample",
    WS_OVERLAPPEDWINDOW,
    CW_USEDEFAULT,
    CW_USEDEFAULT,
    256,
    256,
    NULL,
    NULL,
    hInstance,
    NULL);

ShowWindow(hwnd, nCmdShow);

/* enable OpenGL for the window */
EnableOpenGL(hwnd, &hDC, &hRC);

/* program main loop */
while (!bQuit)
{
    /* check for messages */
    if (PeekMessage(&msg, NULL, 0, 0, PM_REMOVE))
    {
        /* handle or dispatch messages */
        if (msg.message == WM_QUIT)
        {
            bQuit = TRUE;
        }
        else
        {
            TranslateMessage(&msg);
            DispatchMessage(&msg);
        }
    }
    else
    {
        /* OpenGL animation code goes here */

        glClearColor(0.0f, 0.0f, 0.0f, 0.0f);
    }
}

```



```

    glClear(GL_COLOR_BUFFER_BIT);

    glPushMatrix();
    glRotatef(theta, 0.0f, 0.0f, 1.0f);

    glBegin(GL_TRIANGLES);

        glColor3f(1.0f, 0.0f, 0.0f); glVertex2f(0.0f, 1.0f);
        glColor3f(0.0f, 1.0f, 0.0f); glVertex2f(0.87f, -0.5f);
        glColor3f(0.0f, 0.0f, 1.0f); glVertex2f(-0.87f, -0.5f);

    glEnd();

    glPopMatrix();

    SwapBuffers(hDC);

    theta += 1.0f;
    Sleep (1);
}
}

/* shutdown OpenGL */
DisableOpenGL(hwnd, hDC, hRC);

/* destroy the window explicitly */
DestroyWindow(hwnd);

return msg.wParam;
}

LRESULT CALLBACK WindowProc(HWND hwnd, UINT uMsg, WPARAM wParam,
LPARAM lParam)
{
    switch (uMsg)
    {
        case WM_CLOSE:
            PostQuitMessage(0);
            break;
    }
}

```

```

    case WM_DESTROY:
        return 0;

    case WM_KEYDOWN:
    {
        switch (wParam)
        {
            case VK_ESCAPE:
                PostQuitMessage(0);
                break;
        }
    }
    break;

    default:
        return DefWindowProc(hwnd, uMsg, wParam, lParam);
}

return 0;
}

void EnableOpenGL(HWND hwnd, HDC* hDC, HGLRC* hRC)
{
    PIXELFORMATDESCRIPTOR pfd;

    int iFormat;

    /* get the device context (DC) */
    *hDC = GetDC(hwnd);

    /* set the pixel format for the DC */
    ZeroMemory(&pfd, sizeof(pfd));

    pfd.nSize = sizeof(pfd);
    pfd.nVersion = 1;
    pfd.dwFlags = PFD_DRAW_TO_WINDOW |
        PFD_SUPPORT_OPENGL | PFD_DOUBLEBUFFER;
    pfd.iPixelFormat = PFD_TYPE_RGBA;
    pfd.cColorBits = 24;
    pfd.cDepthBits = 16;
}

```

```
pfd.iLayerType = PFD_MAIN_PLANE;
```

```
iFormat = ChoosePixelFormat(*hDC, &pfd);
```

```
SetPixelFormat(*hDC, iFormat, &pfd);
```

```
/* create and enable the render context (RC) */
```

```
*hRC = wglCreateContext(*hDC);
```

```
wglMakeCurrent(*hDC, *hRC);
```

```
}
```

```
void DisableOpenGL (HWND hwnd, HDC hDC, HGLRC hRC)
```

```
{
```

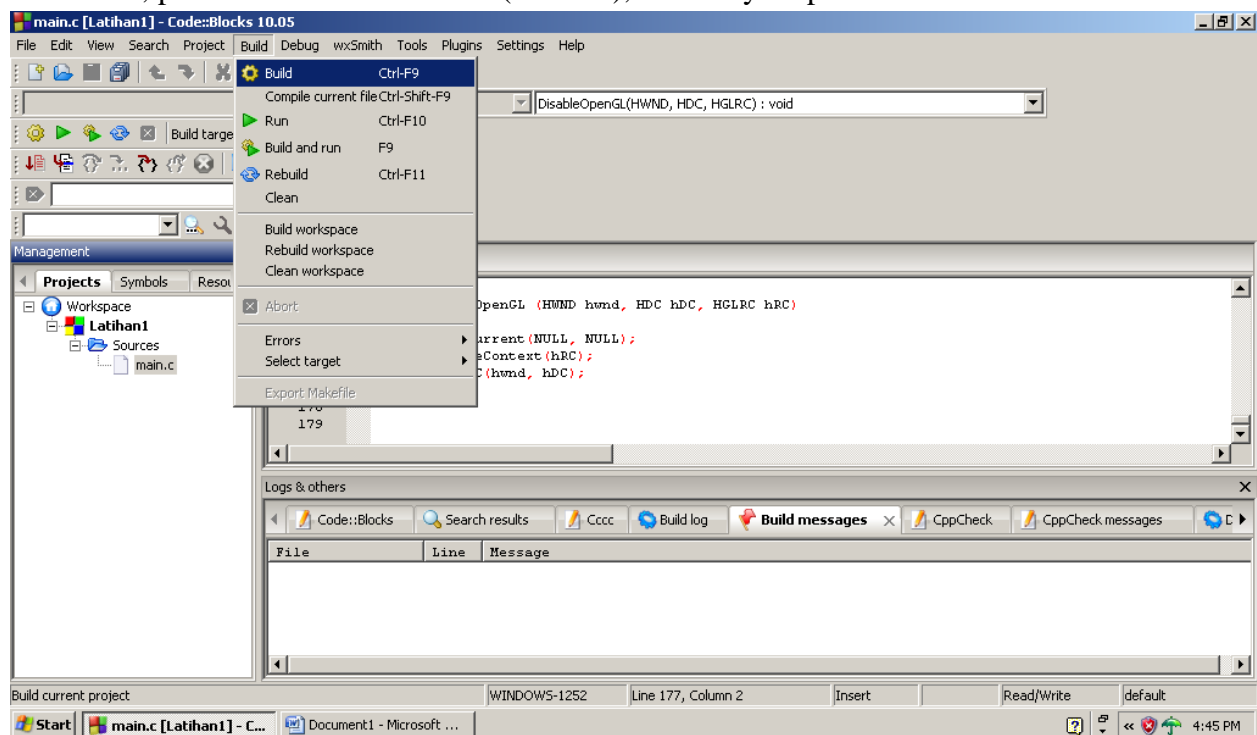
```
    wglMakeCurrent(NULL, NULL);
```

```
    wglDeleteContext(hRC);
```

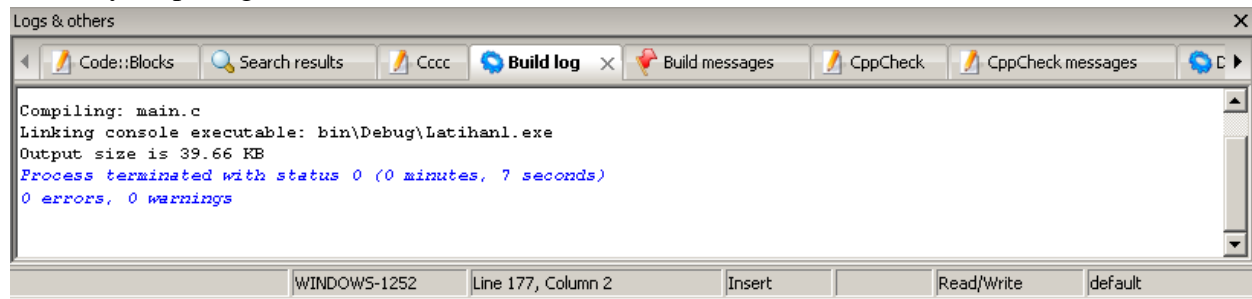
```
    ReleaseDC(hwnd, hDC);
```

```
}
```

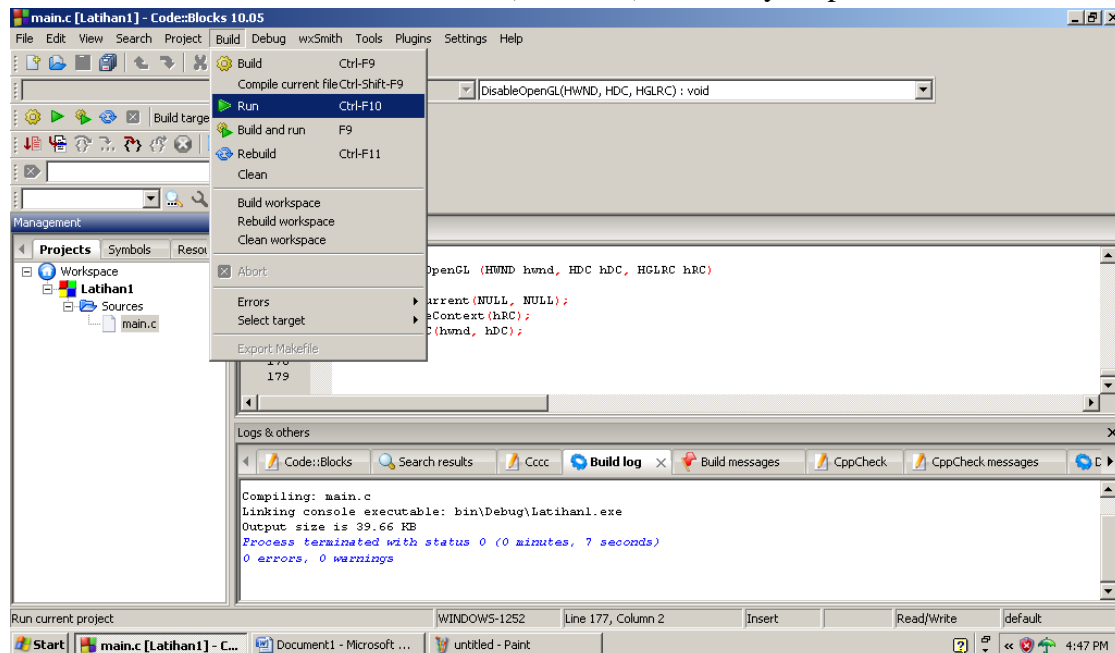
Setelah itu, pilih menu Build → Build (Ctrl+F9), contohnya seperti berikut ini.



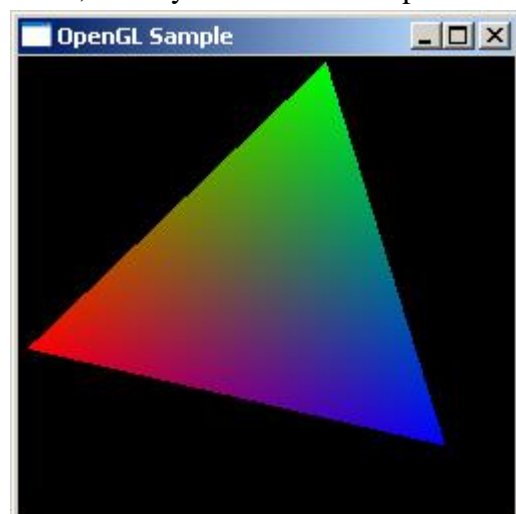
Maka, pada Logs & Others akan memberikan laporan, apakah ada yang error atau tidak, contohnya seperti gambar dibawah ini.



Jika tidak ada error, klik Build → Run (Ctrl+F10), contohnya seperti berikut ini.



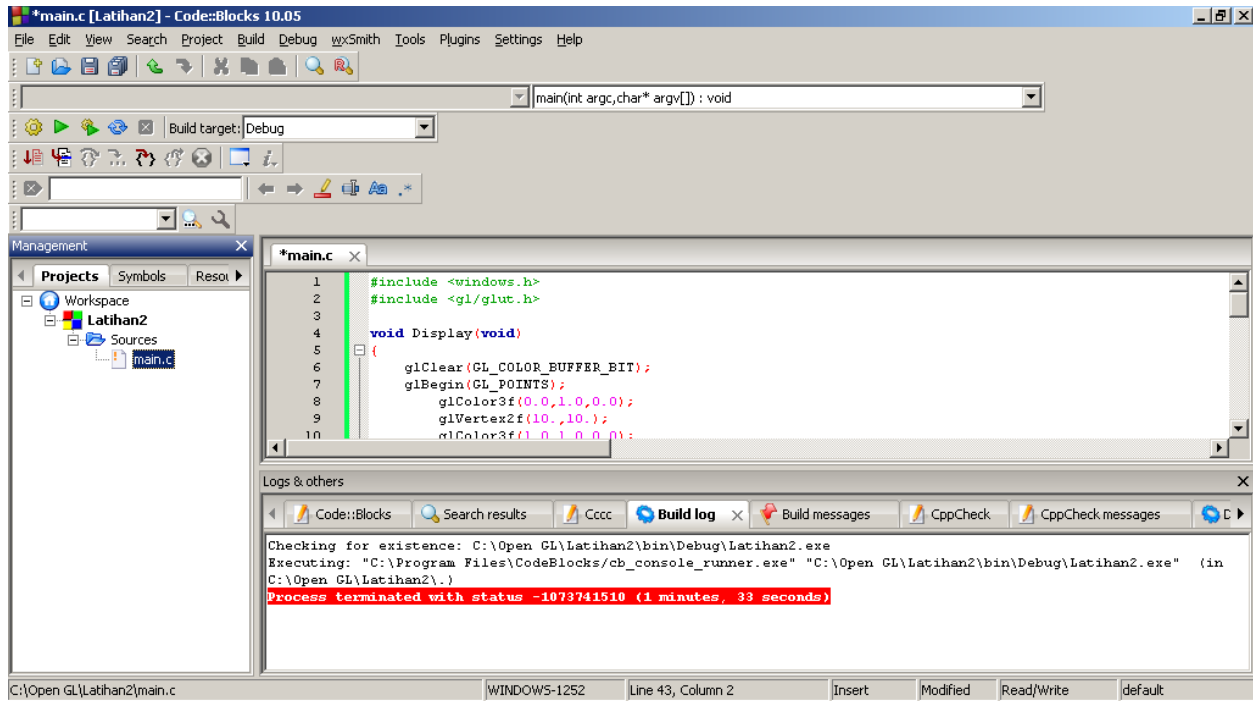
Maka, hasilnya akan terlihat seperti berikut ini.



Latihan 2.

Untuk Latihan 2, langkah-langkahnya sama seperti Latihan 1. Hanya saja, berbeda pada bahasa pemrogramannya.

Isi main.c seperti berikut ini.



The screenshot shows the Code::Blocks IDE interface. The main editor window displays the following C code in main.c:

```
1 #include <windows.h>
2 #include <gl/glut.h>
3
4 void Display(void)
5 {
6     glClear(GL_COLOR_BUFFER_BIT);
7     glBegin(GL_POINTS);
8     glColor3f(0.0,1.0,0.0);
9     glVertex2f(10.,10.);
10    glColor3f(1.0,1.0,0.0);
```

The build log window at the bottom shows the following output:

```
Checking for existence: C:\Open GL\Latihan2\bin\Debug\Latihan2.exe
Executing: "C:\Program Files\CodeBlocks\cb_console_runner.exe" "C:\Open GL\Latihan2\bin\Debug\Latihan2.exe" (in
C:\Open GL\Latihan2\.)
Process terminated with status -1073741510 (1 minutes, 33 seconds)
```

```
#include <windows.h>
```

```
#include <gl/glut.h>
```

```
void Display(void)
```

```
{
```

```
    glClear(GL_COLOR_BUFFER_BIT);
```

```
    glBegin(GL_POINTS);
```

```
        glColor3f(0.0,1.0,0.0);
```

```
        glVertex2f(10.,10.);
```

```
        glColor3f(1.0,1.0,0.0);
```

```
        glVertex2f(10.,110.);
```

```
        glColor3f(0.0,0.0,1.0);
```

```
        glVertex2f(150.,110.0);
```

```
        glColor3f(1.0,1.0,1.0);
```

```
        glVertex2f(150.,10);
```

```
    glEnd();
```

```
    glFlush();
```

```
}

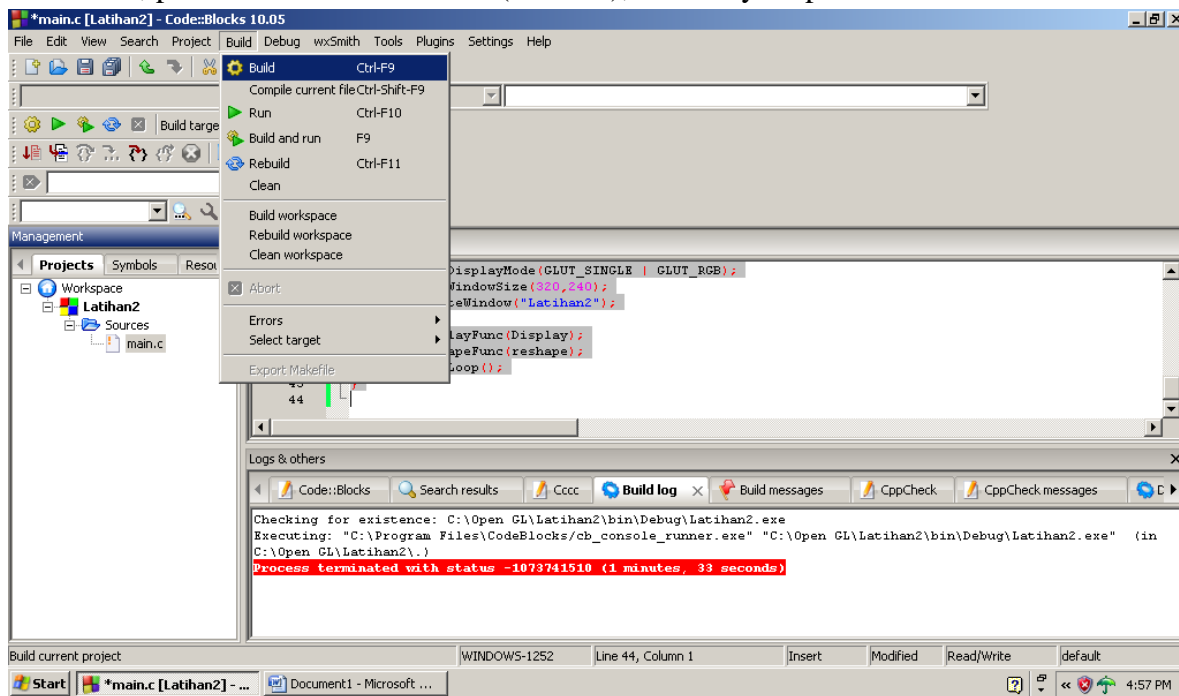
void reshape(int w,int h)
{
    glViewport(0,0,(GLsizei)w,(GLsizei)h);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluOrtho2D(0.0,160.0,0.0,120);

}

void init(void)
{
    glClearColor(1.0,0.0,0.0,1.0);
    glPointSize(20.0);
}

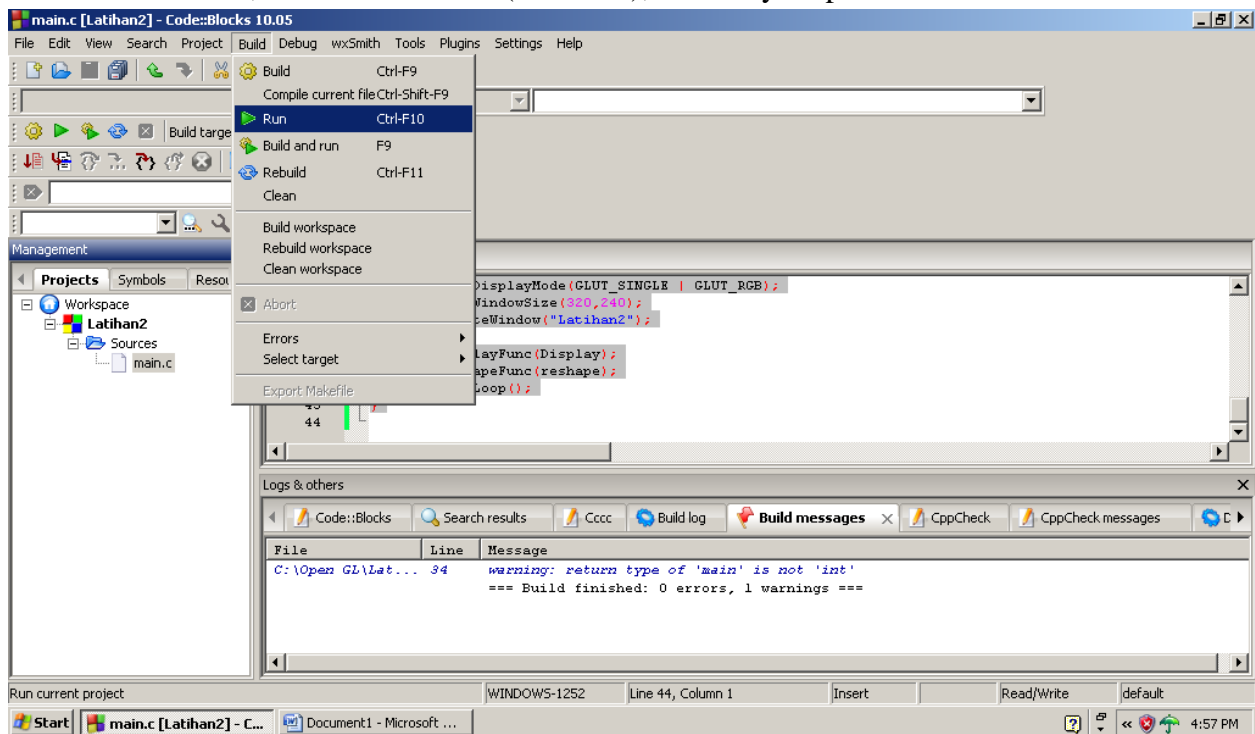
void main(int argc,char* argv[])
{
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
    glutInitWindowSize(320,240);
    glutCreateWindow("Latihan2");
    init();
    glutDisplayFunc(Display);
    glutReshapeFunc(reshape);
    glutMainLoop();
}
```

Setelah itu, pilih menu Build → Build (Ctrl+F9), contohnya seperti berikut ini.



Maka, pada Logs & Others akan memberikan laporan, apakah ada yang error atau tidak.

Jika tidak ada error, klik Build → Run (Ctrl+F10), contohnya seperti berikut ini.



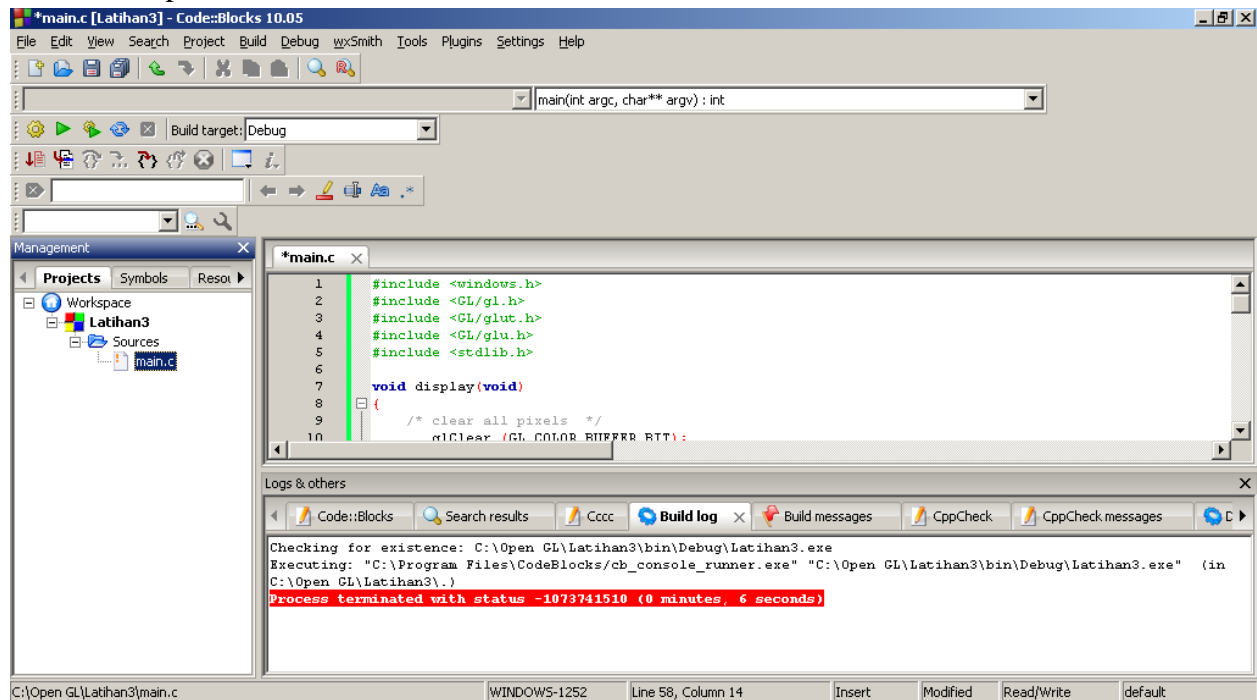
Maka, hasilnya akan terlihat seperti berikut ini.



Latihan 3.

Untuk Latihan 3, langkah-langkahnya sama seperti Latihan 1. Hanya saja, berbeda pada bahasa pemrogramannya.

Isi main.c seperti berikut ini.




```

#include <windows.h>
#include <GL/gl.h>
#include <GL/glut.h>
#include <GL/glu.h>
#include <stdlib.h>

void display(void)
{
    /* clear all pixels */
    glClear (GL_COLOR_BUFFER_BIT);

    /* draw white polygon (rectangle) with corners at
    * (0.25, 0.25, 0.0) and (0.75, 0.75, 0.0)
    */
    glColor3f (1.0, 1.0, 0.0);
    glBegin(GL_POLYGON);
        glVertex3f (0.25, 0.25, 0.0);
        glVertex3f (0.75, 0.25, 0.0);
        glVertex3f (0.75, 0.75, 0.0);
        glVertex3f (0.25, 0.75, 0.0);
    glEnd();

    /* don't wait!
    * start processing buffered OpenGL routines
    */
    glFlush ();
}

void init (void)
{
    /* select clearing color */
    glClearColor (0.0, 0.0, 0.0, 0.0);

    /* initialize viewing values */
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    glOrtho(0.0, 1.0, 0.0, 1.0, -1.0, 1.0);
}

/*

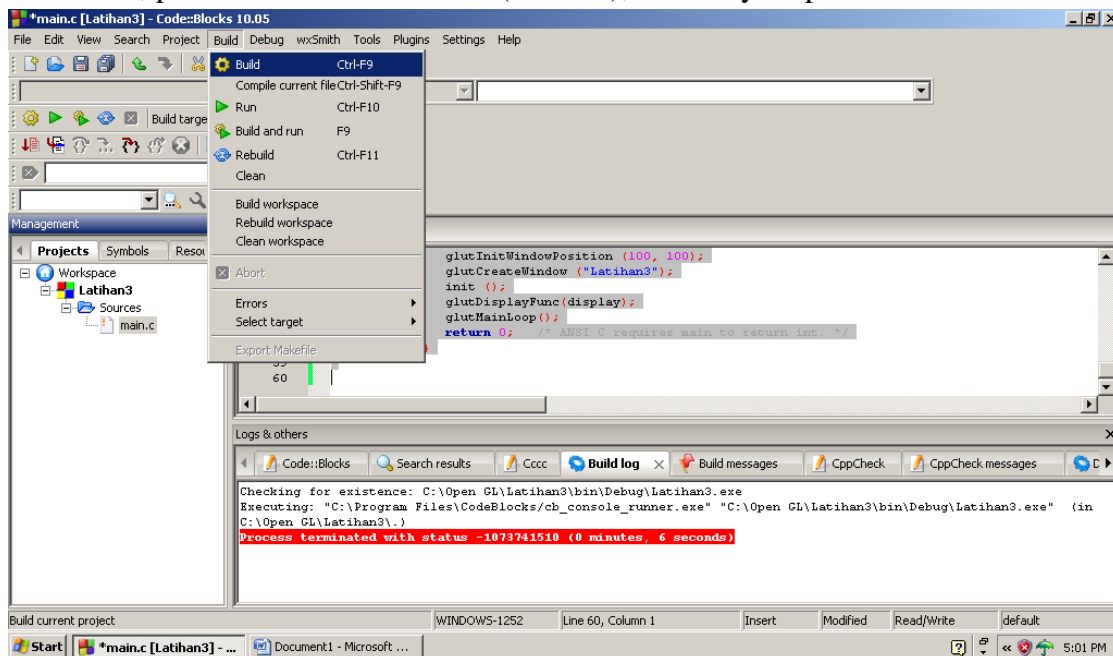
```

```

* Declare initial window size, position, and display mode
* (single buffer and RGABA). Open window with "hello"
* in its title bar. Call initialization routines.
* Register callback function to display graphics.
* Enter main loop and process events.
*/
int main(int argc, char** argv)
{
    glutInit(&argc, argv);
    glutInitDisplayMode (GLUT_SINGLE | GLUT_RGB);
    glutInitWindowSize (250, 250);
    glutInitWindowPosition (100, 100);
    glutCreateWindow ("Latihan3");
    init ();
    glutDisplayFunc(display);
    glutMainLoop();
    return 0; /* ANSI C requires main to return int. */
}

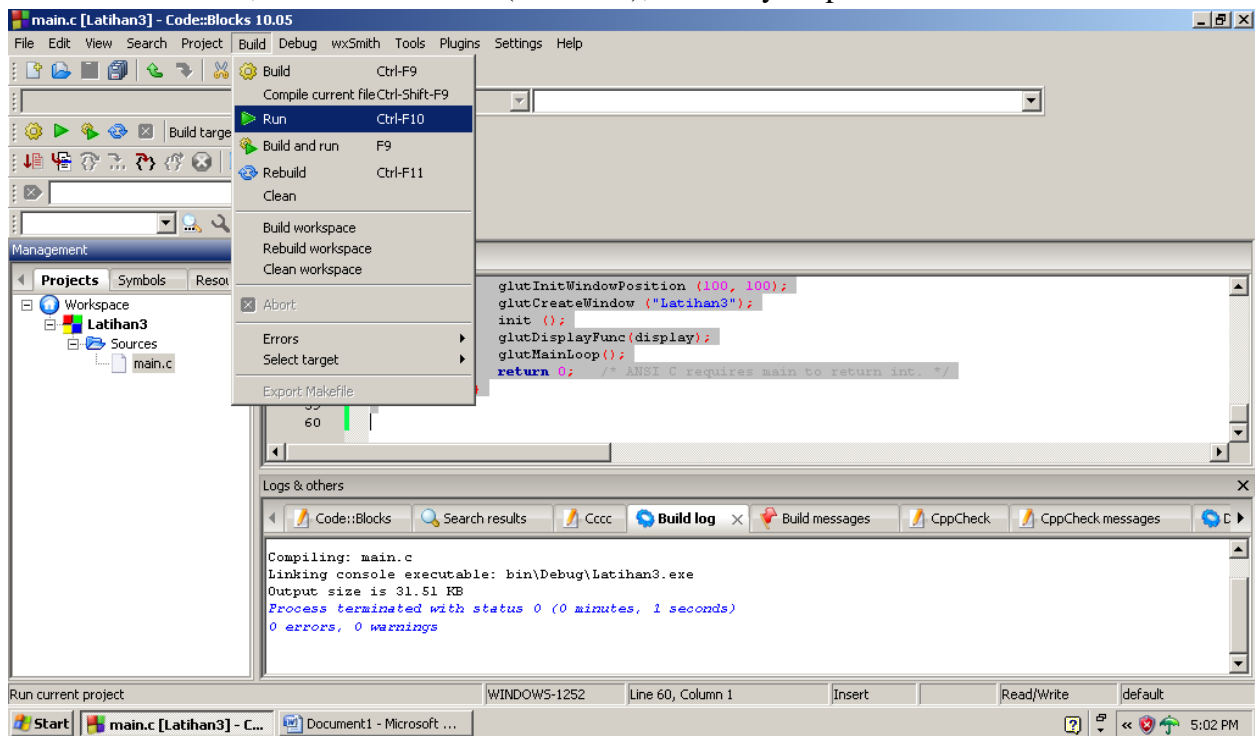
```

Setelah itu, pilih menu Build → Build (Ctrl+F9), contohnya seperti berikut ini.

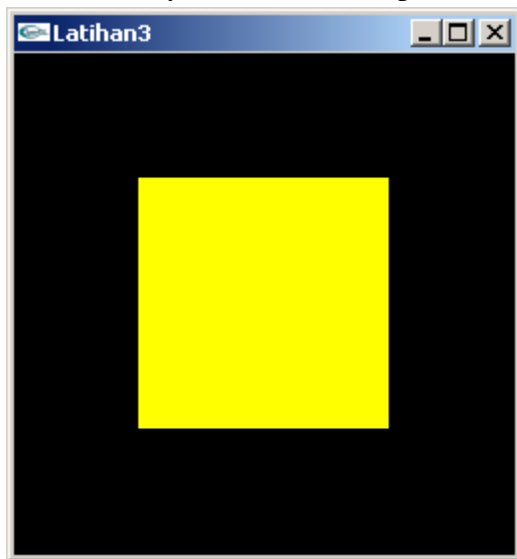


Maka, pada Logs & Others akan memberikan laporan, apakah ada yang error atau tidak.

Jika tidak ada error, klik Build → Run (Ctrl+F10), contohnya seperti berikut ini.



Maka, hasilnya akan terlihat seperti berikut ini.



Mohon maaf apabila banyak kesalahan dalam tulisan ini, terimakasih.