

Modul Praktikum

Pemrograman Grafik

-Objek Primitif-

Nama :

Kelas :

No. Absen :

SMK Telkom Malang

2016

Objek Primitif

A. KOMPETENSI DASAR

- Memahami jenis-jenis objek primitif.
- Memahami dan dapat menerapkan pembuatan objek primitif.

B. ALOKASI WAKTU

3 JS (3 x 45 menit)

C. PETUNJUK

- Awali setiap aktivitas dengan do'a, semoga berkah dan mendapat kemudahan.
- Simpan modul dengan nama Modul-OP-Kelas-No.Absen (contoh: Modul-OP-XIIRPL3-29).
- Kumpulkan paling lambat tanggal 30 September 2016 jam 15.00.
- Tanyakan kepada guru apabila ada hal-hal yang kurang jelas.

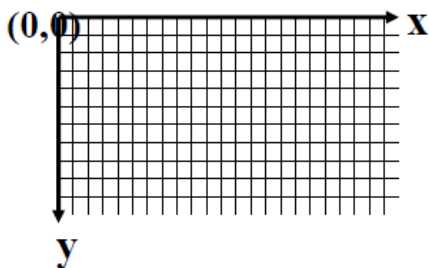
D. DASAR TEORI

Sistem Koordinat

Terdapat 2 sistem koordinat yaitu koordinat layar/monitor dan koordinat cartesian.

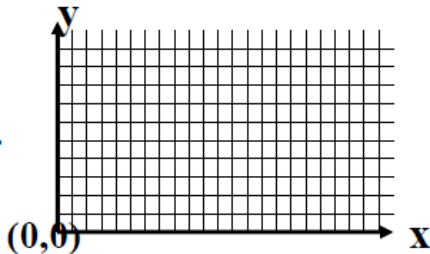
a. Koordinat layar/monitor

Yaitu koordinat yang ada pada layar komputer. Digunakan pada penerapan program komputer.

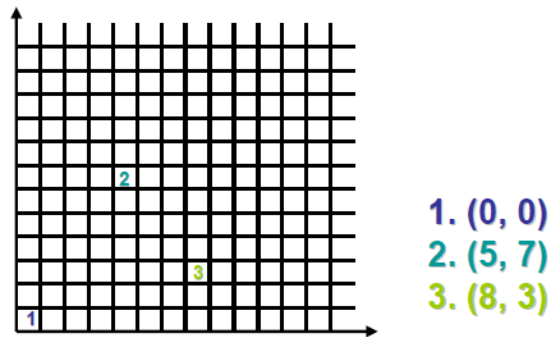


b. Koordinat Cartesian

Sistem koordinat ini digunakan di berbagai konsep matematika, statistik, maupun grafika komputer.



Contoh :



Objek Primitif

Objek primitif merupakan salah satu subbab dari grafika komputer yang terdiri dari titik, garis dan bangun dua dimensi. Contoh primitive grafika dasar antara lain:

- Titik → GL_POINTS
- Garis → GL_LINES
- Poliline → GL_LINE_STRIP
- Poligon (no fill) → GL_LINE_LOOP

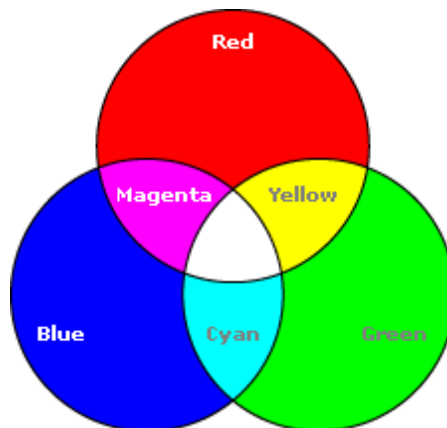
Definisi Warna Objek Primitif

RGB (Red, Green, Blue) merupakan tipe warna dasar yang banyak digunakan. Mata manusia sangat sensitif terhadap panjang gelombang cahaya. Warna merah memiliki panjang gelombang 630nm, hijau 530nm dan biru 450.

Untuk mendapatkan warna yang lain dari model RGB dapat dilakukan dengan mencampurkan warna dasar tersebut sehingga menghasilkan warna yang lain (*additive color*). Warna additive dapat dirumuskan:

$$W = R_R + G_G + B_B$$

R_R , G_G , B_B bernilai antara 0 sampai 1, dimana nilai 0 menyatakan tidak ada komponen yang digunakan dan nilai 1 menyatakan pemakaian penuh dari komponen tersebut. Untuk nilai diantaranya dapat digunakan nilai pecahan seperti 0.1 - 0.9.



Fungsi dasar menggambar titik

Terdapat 3 macam perintah untuk menggambar titik, yaitu:

- a. `glVertex2i(x,y)`
Menggambar titik pada koordinat x dan y, dengan nilai satuan berupa integer.
Contoh: `glVertex2i(10,10);`
- b. `glVertex2f(x,y)`
Menggambar titik pada koordinat x dan y, dengan nilai satuan berupa float.
Contoh: `glVertex2f(10.0,10.0);`
- c. `glVertex2d(x,y)`
Menggambar titik pada koordinat x dan y, dengan nilai satuan berupa double.
Contoh: `glVertex2d(10.0,10.0);`

Tabel contoh perintah pada OpenGL

Perintah	Arti	Keterangan
<code>glVertex2i(x,y);</code>	Lokasi titik berada (x,y)	Tipe argumennya adalah integer dan 2D (x,y)
<code>glVertex2f(x,y);</code>	Lokasi titik berada (x,y)	Tipe argumennya adalah float dan 2D (x,y)
<code>glVertex3i(x,y,z);</code>	Lokasi titik berada (x,y,z)	Tipe argumennya adalah integer dan 3D (x,y,z)
<code>glVertex3f(x,y,z);</code>	Lokasi titik berada (x,y,z)	Tipe argumennya adalah float dan 3D (x,y,z)
<code>glClearColor(R,G,B,α);</code>	Warna latar belakang	Empat komponen warna RGBA
<code>glColor3f(R,G,B)</code>	Warna latar muka	Tiga komponen warna (RGB)
<code>glColor4f(R,G,B,α)</code>	Warna latar muka	Empat komponen warna (RGBA)
<code>glBegin(GL_POINTS);</code>	Titik	Objek primitif
<code>glBegin(GL_LINES);</code>	Garis	Objek primitif
<code>glBegin(GL_LINE_STRIP);</code>	Poligaris	Objek primitif
<code>glBegin(GL_LINE_LOOP);</code>	Poligaris tertutup	Objek primitif
<code>glBegin(GL_TRIANGLES);</code>	Segitiga	Objek primitif
<code>glBegin(GL_TRIANGLE_STRIP);</code>	Segitiga	Objek primitif
<code>glBegin(GL_TRIANGLE_FAN);</code>	Segitiga	Objek primitif
<code>glBegin(GL_QUADS);</code>	Segiempat	Objek primitif
<code>glBegin(GL_QUADS_STRIP);</code>	Segiempat	Objek primitif
<code>glBegin(GL_LINE_STIPPLE);</code>	Garis putus-putus	Objek primitif
<code>glBegin(GL_POLY_STIPPLE);</code>	Poligon dengan pola tertentu	Objek primitif

E. AKTIFITAS PRAKTIKUM

1. Berikut adalah script dasar untuk kegiatan praktikum:

```
#include <Windows.h>
#include <GL\freeglut.h>
#include <iostream>

void render()
{
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    glutSwapBuffers();
}
```

```

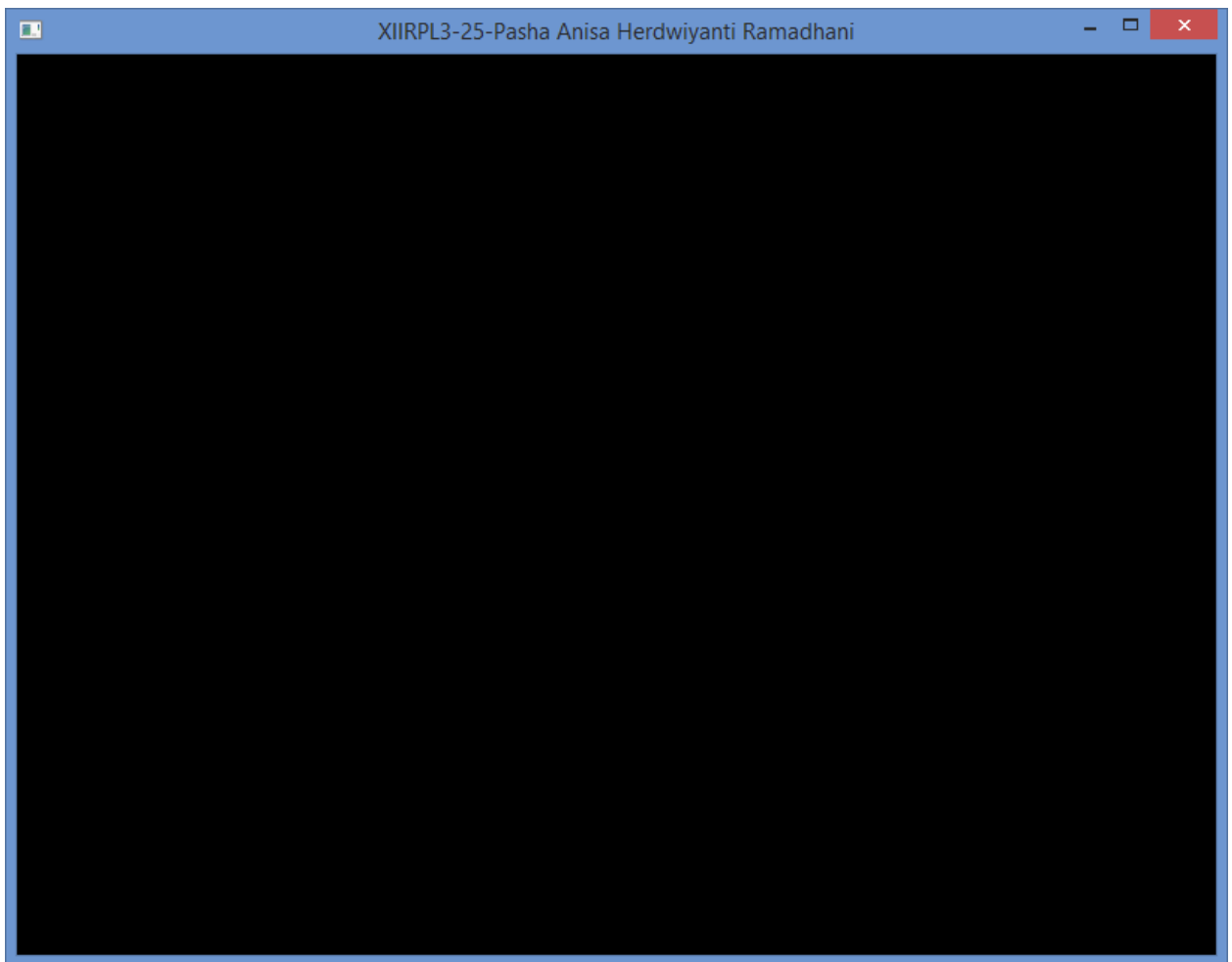
int main(int argc, char* argv[])
{
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGBA | GLUT_DEPTH);
    glutInitWindowSize(800, 600);
    glutCreateWindow("Kelas-No.Absen-Nama");
    gluOrtho2D(-400, 400, -300, 300);
    glClearColor(0, 0, 0, 0);
    glutDisplayFunc(render);

    glutMainLoop();
    return 0;
}

```

Ubahlah statement `glutCreateWindow("Kelas-No.Absen-Nama");` menjadi identitas masing-masing. Contoh : `glutCreateWindow("XIIRPL2-40-Pashatania Fitri Indah L.");`. Kemudian screenshot hasilnya.

Screenshot :



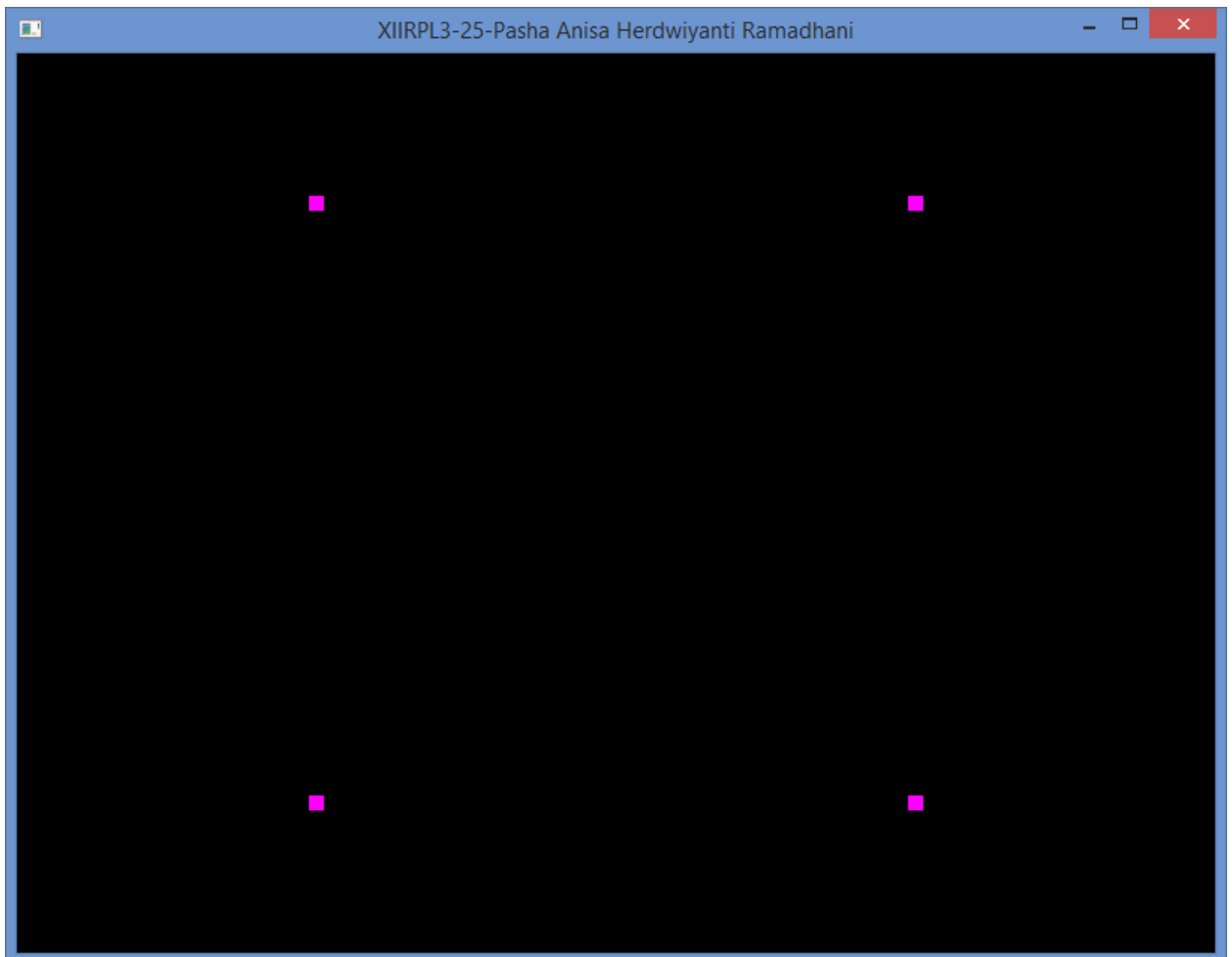
2. Buatlah project baru pada Visual Studio dan tambahkan fungsi dibawah ini pada program.

```
void gambar()  
{  
    glPointSize(10);  
    glBegin(GL_POINTS);  
    glColor3f(1, 0, 1);  
    glVertex3f(200, 200, 0);  
    glVertex3f(200, -200, 0);  
    glVertex3f(-200, -200, 0);  
    glVertex3f(-200, 200, 0);  
    glEnd();  
}
```

Panggil fungsi tersebut pada fungsi render seperti pada gambar dibawah ini. Kemudian screenshot hasilnya.

```
void render()  
{  
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);  
    gambar();  
    glutSwapBuffers();  
}
```

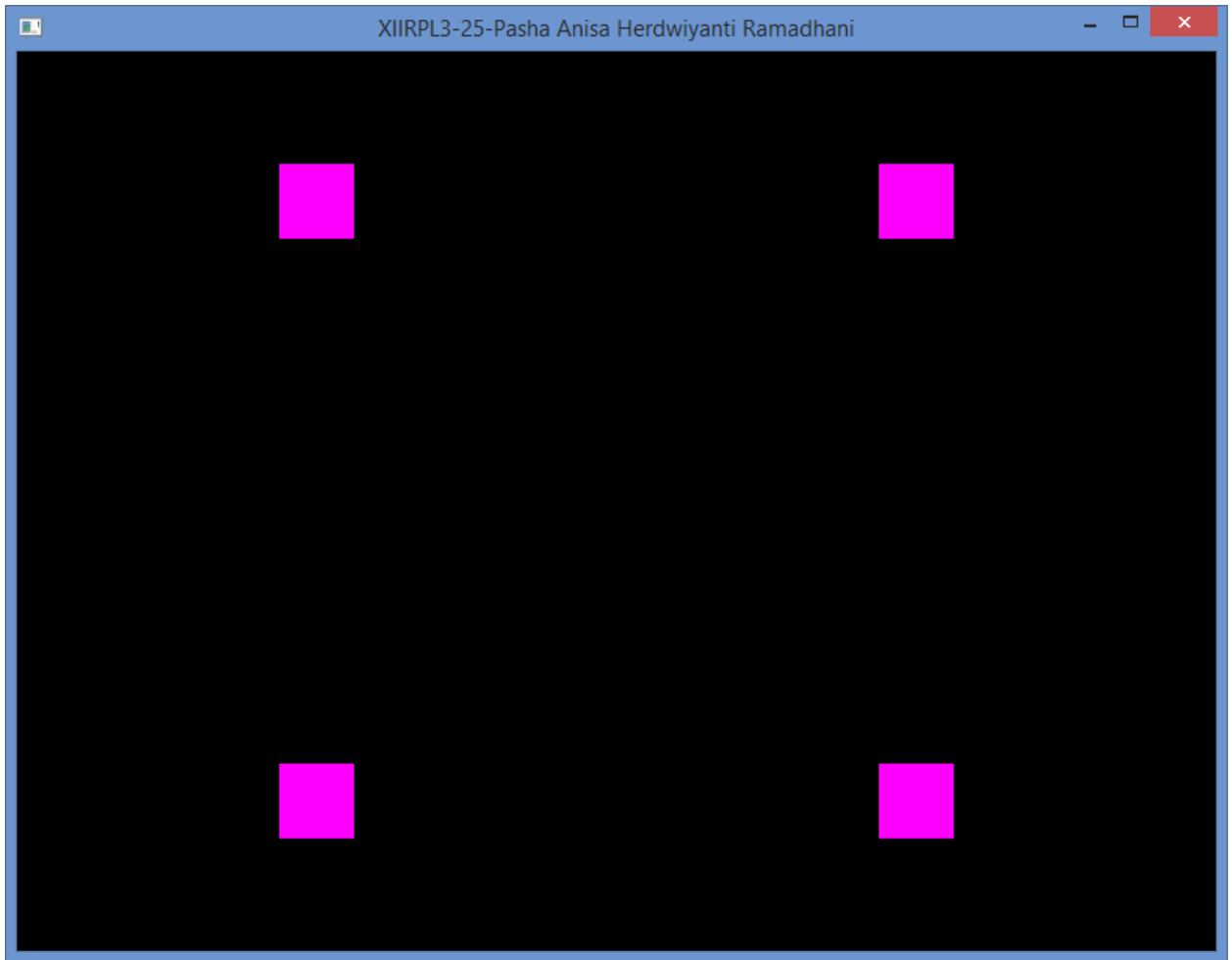
Screenshot :



3. Ubahlah statement `glPointSize(10);` pada fungsi gambar menjadi `glPointSize(50);`. Apa yang terjadi? Jelaskan fungsi `glPointSize();`

Jawab:

Point (titik) semakin besar. Fungsi `glPointSize` adalah untuk menentukan ukuran dari point (titik). Semakin besar angkanya semakin besar pula ukurannya.



4. Ubahlah nilai statement `glColor3f(1, 0, 1);` pada fungsi gambar menjadi kombinasi seperti tabel dibawah ini. Warna apa yang dihasilkan?

Tabel Warna:

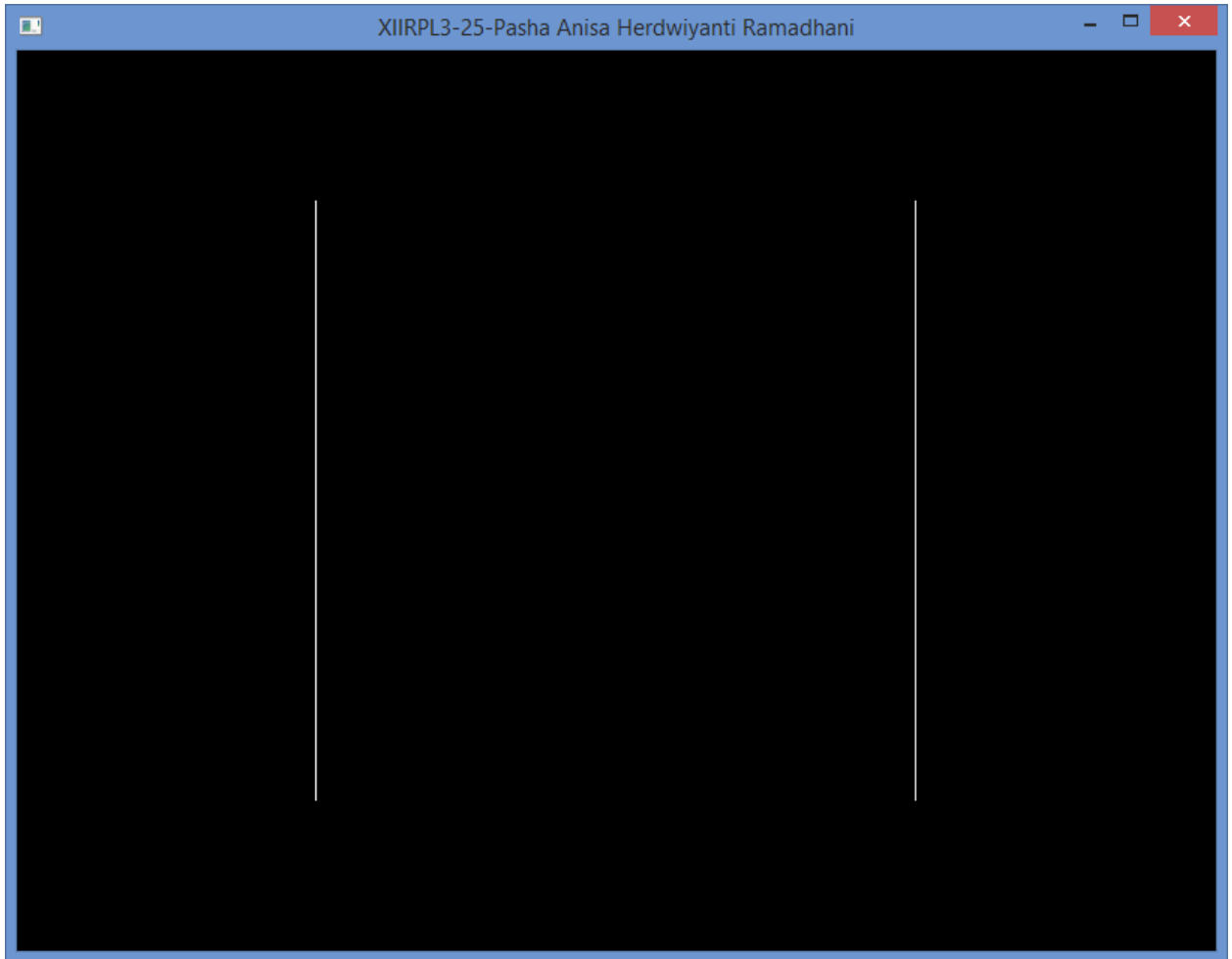
No.	R	G	B	Warna yang dihasilkan
1	0	0	0	Hitam
2	0	0	1	Biru
3	0	1	0	Hijau
4	0	1	1	Cyan

5	1	0	0	Merah
6	1	0	1	Magenta
7	1	1	0	Kuning
8	1	1	1	Putih

5. Ubahlah statement `glBegin(GL_POINTS);` pada fungsi gambar menjadi `glBegin(GL_LINES);`. Apa yang terjadi? Jelaskan fungsi `GL_LINES`.

Jawab:

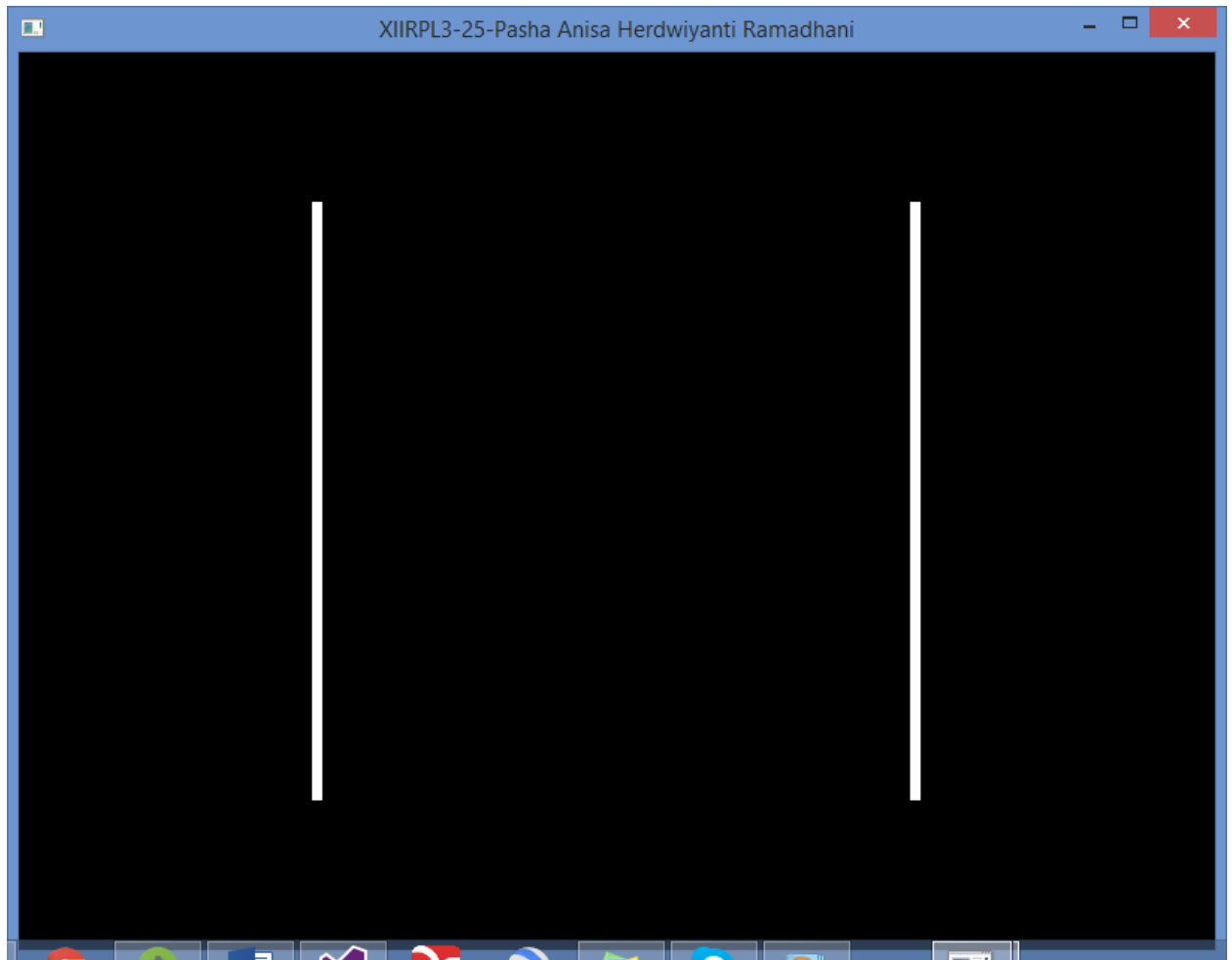
Titik berubah menjadi garis. Fungsi `GL_LINES` adalah untuk membentuk garis



6. Ubahlah statement `glPointSize(10);` pada fungsi gambar menjadi `glLineWidth(10);`. Apa yang terjadi? Jelaskan fungsi `glLineWidth()`;

Jawab:

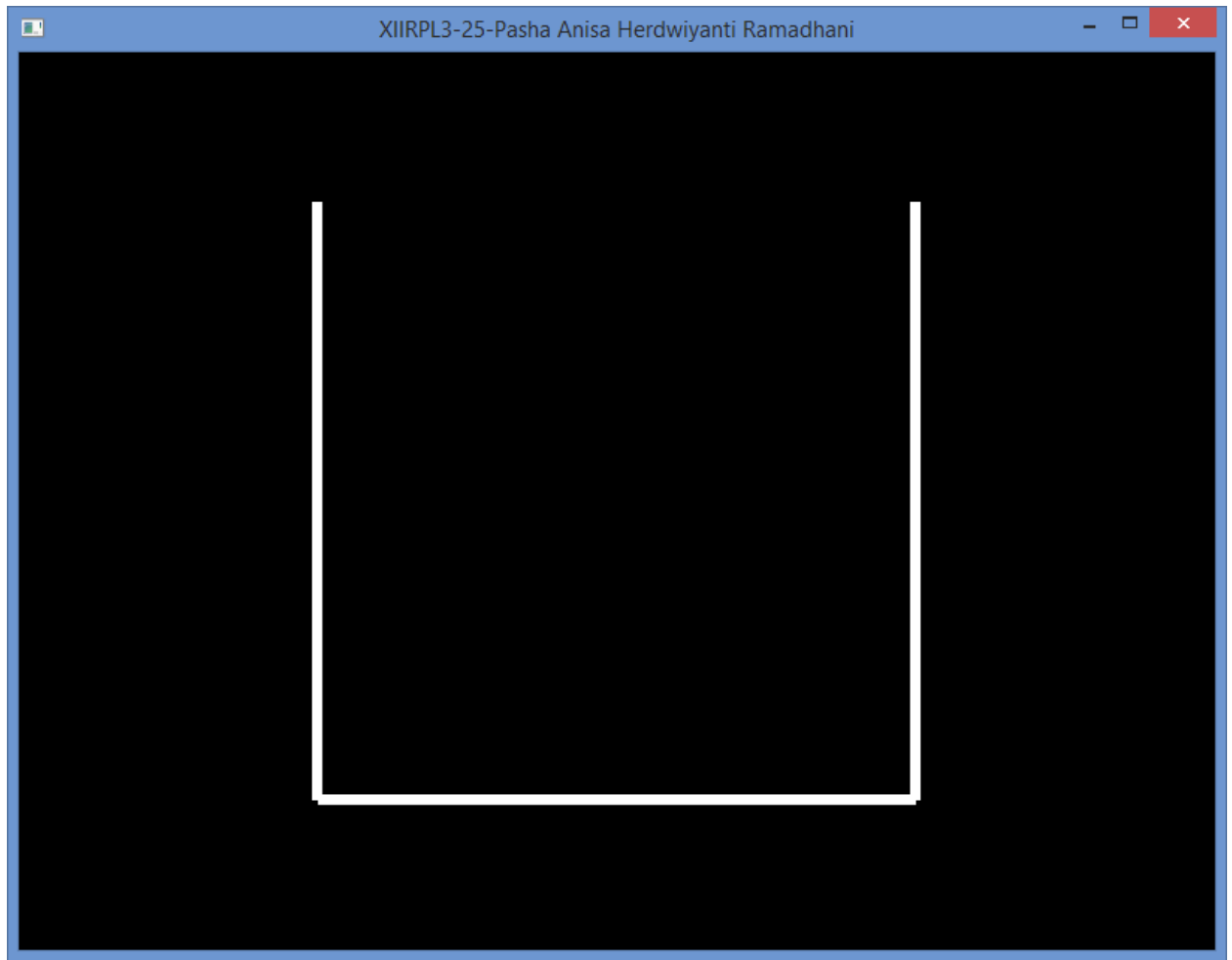
Garis menjadi lebih tebal. Fungsi `glLineWidth()` adalah untuk memperlebar garis



7. Ubahlah statement `glBegin(GL_LINES);` pada fungsi gambar menjadi `glBegin(GL_LINE_STRIP);`. Apa yang terjadi? Jelaskan fungsi `GL_LINE_STRIP`.

Jawab:

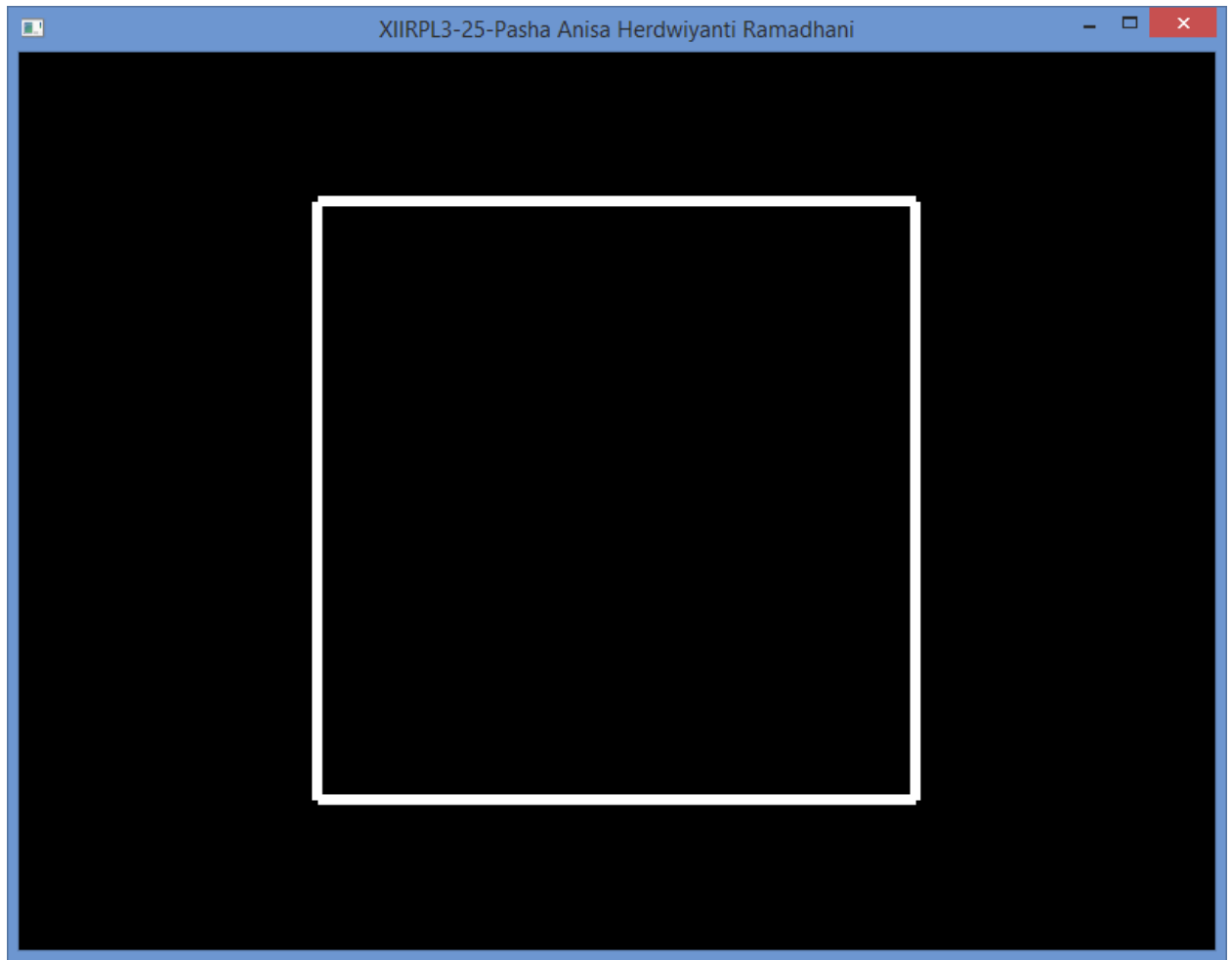
Membentuk seperti gambar. Fungsi `GL_LINE_STRIP` adalah untuk membuat bentuk garis terhubung yang terbuka



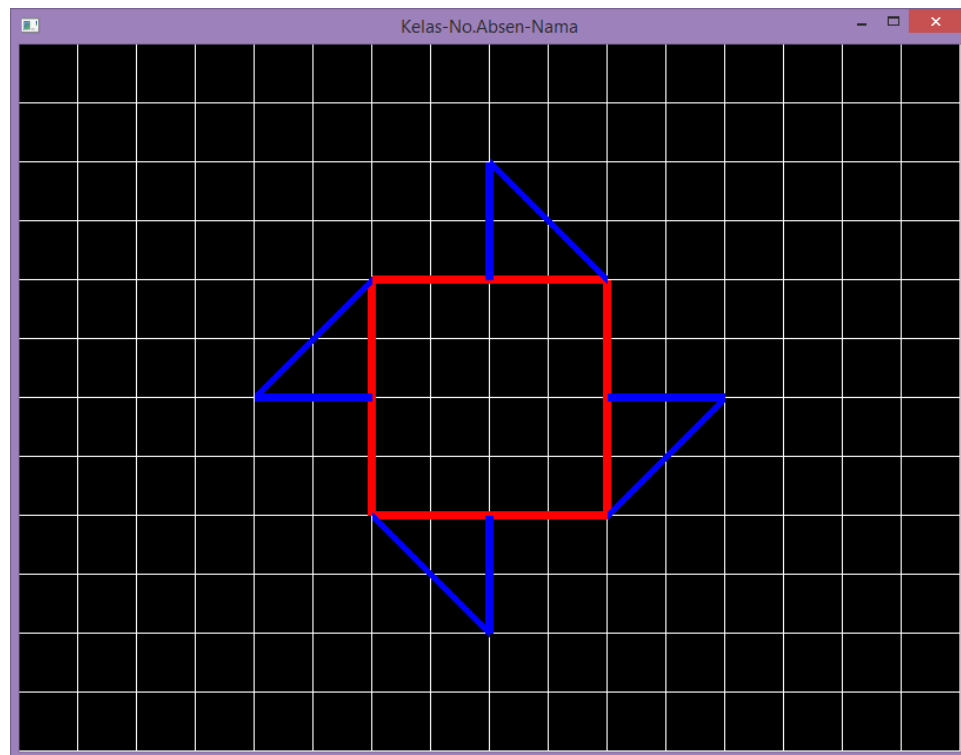
8. Ubahlah statement `glBegin(GL_LINE_STRIP);` pada fungsi gambar menjadi `glBegin(GL_LINE_LOOP);`. Apa yang terjadi? Jelaskan fungsi `GL_LINE_LOOP`.

Jawab:

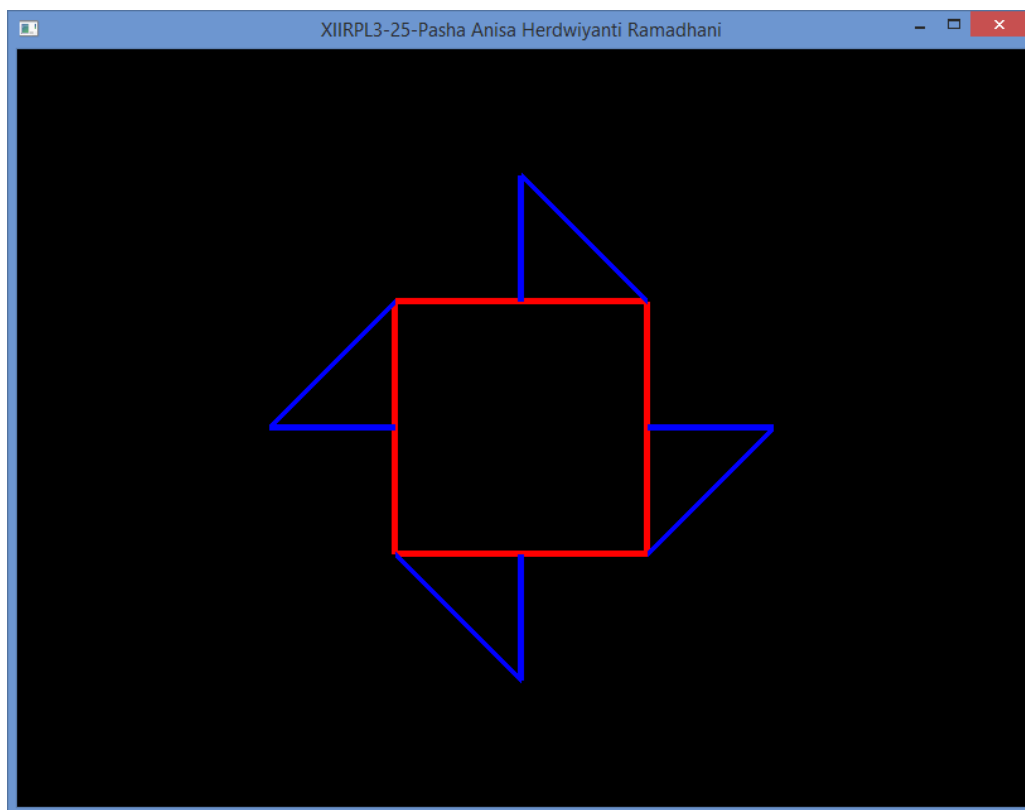
Membentuk bangun kotak (persegi). Fungsi `GL_LINE_LOOP` adalah untuk membuat bentuk di mana vertex awal dan vertex akhir saling terhubung



9. Dengan menggunakan Objek Primitif, buatlah program untuk membuat gambar baling-baling seperti berikut ini. Sajikan screenshot dan hasil program. (1 kotak = 50x50 pixel)



Screenshot :



Hasil Program :

//Script Program

```
#include <Windows.h>
#include <GL\freeglut.h>
#include <iostream>

void kotak()
{
    glLineWidth(5);
    glBegin(GL_LINE_LOOP);
    glColor3f(1, 0, 0);
    glVertex3f(100, 100, 0);
    glVertex3f(100, -100, 0);
    glVertex3f(-100, -100, 0);
    glVertex3f(-100, 100, 0);
    glEnd();
}

void segitiga()
{
    glLineWidth(5);
    glBegin(GL_LINES);
    glColor3f(0, 0, 1);
    glVertex3f(0, 200, 0);
    glVertex3f(100, 100, 0);
    glVertex3f(0, 200, 0);
    glVertex3f(0, 100, 0);

    glVertex3f(0, -200, 0);
    glVertex3f(-100, -100, 0);
    glVertex3f(0, -200, 0);
    glVertex3f(0, -100, 0);

    glVertex3f(-200, 0, 0);
    glVertex3f(-100, 100, 0);
    glVertex3f(-200, 0, 0);
    glVertex3f(-100, 0, 0);

    glVertex3f(200, 0, 0);
    glVertex3f(100, -100, 0);
    glVertex3f(200, 0, 0);
    glVertex3f(100, 0, 0);
    glEnd();
}

void render()
{
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    kotak();
    segitiga();
    glutSwapBuffers();
}

int main(int argc, char* argv[])
```

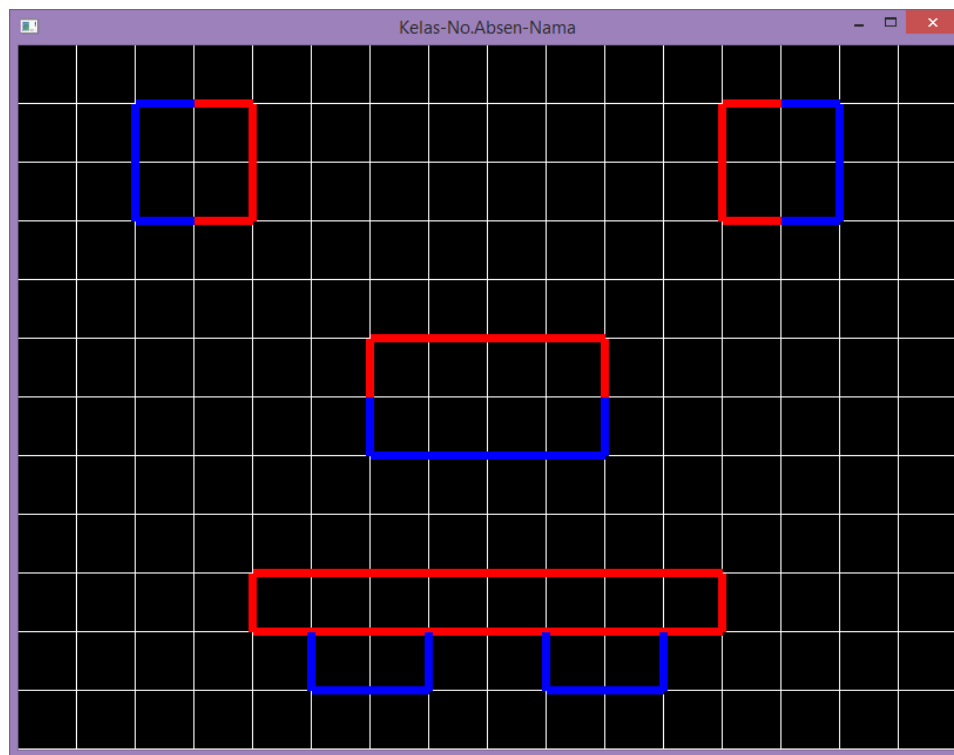
```

{
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGBA | GLUT_DEPTH);
    glutInitWindowSize(800, 600);
    glutCreateWindow("");
    gluOrtho2D(-400, 400, -300, 300);
    glClearColor(0, 0, 0, 0);
    glutDisplayFunc(render);

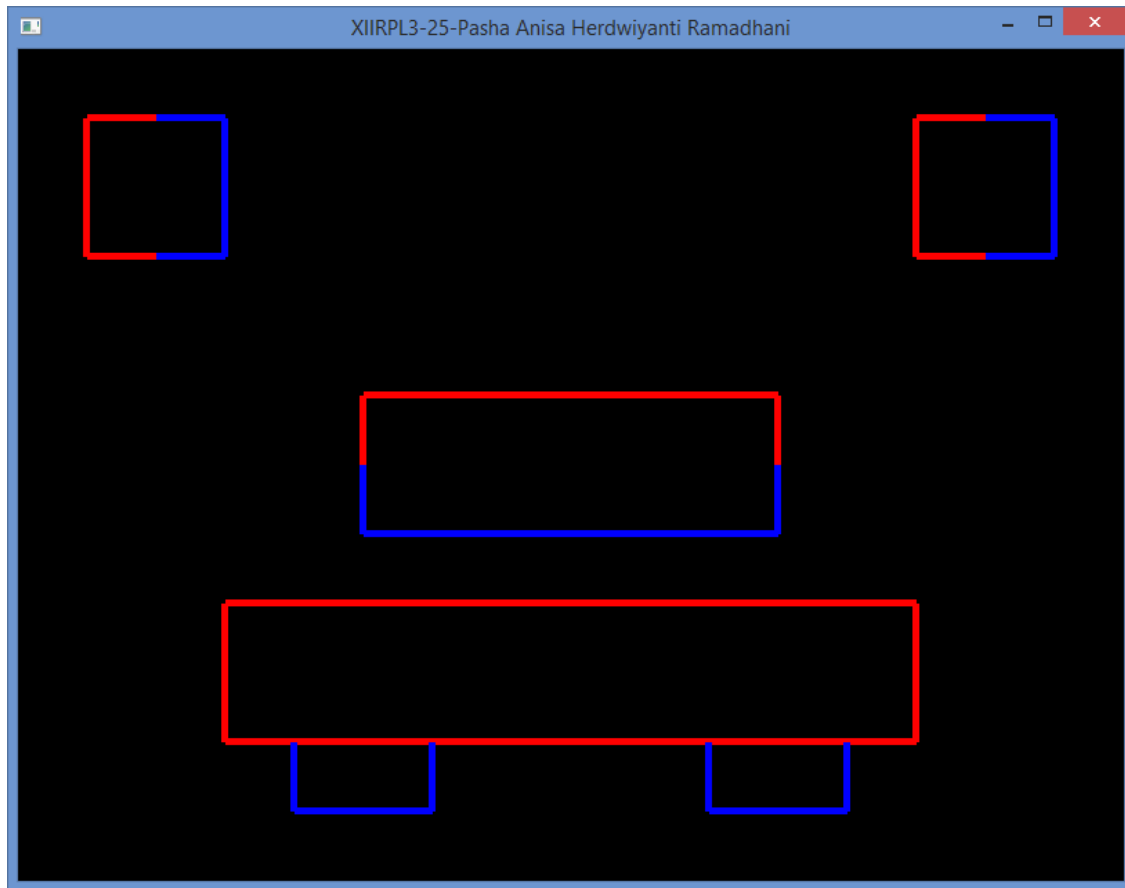
    glutMainLoop();
    return 0;
}

```

10. Dengan menggunakan Objek Primitif, buatlah program untuk membuat gambar wajah seperti berikut ini. Sajikan screenshot dan hasil program. (1 kotak = 50x50 pixel)



Screenshot :



Hasil Program :

//Script Program

```
#include <Windows.h>
#include <GL\freeglut.h>
#include <iostream>

void kotakmerahkiri()
{
    glLineWidth(5);
    glBegin(GL_LINES);
    glColor3f(1, 0, 0);
    glVertex3f(-350, 150, 0);
    glVertex3f(-300, 150, 0);
    glVertex3f(-300, 250, 0);
    glVertex3f(-350, 250, 0);

    //Garis
    glVertex3f(-350, 250, 0);
    glVertex3f(-350, 150, 0);
    glEnd();
}

void kotakbirukiri()
{
    glLineWidth(5);
    glBegin(GL_LINE_STRIP);
```

```

        glColor3f(0, 0, 1);
        glVertex3f(-300, 150, 0);
        glVertex3f(-250, 150, 0);
        glVertex3f(-250, 250, 0);
        glVertex3f(-300, 250, 0);
        glEnd();
    }

    void kotakbirukanan()
    {
        glLineWidth(5);
        glBegin(GL_LINE_STRIP);
        glColor3f(0, 0, 1);
        glVertex3f(300, 150, 0);
        glVertex3f(350, 150, 0);
        glVertex3f(350, 250, 0);
        glVertex3f(300, 250, 0);
        glEnd();
    }

    void kotakmerahkanan()
    {
        glLineWidth(5);
        glBegin(GL_LINES);
        glColor3f(1, 0, 0);
        glVertex3f(250, 150, 0);
        glVertex3f(300, 150, 0);
        glVertex3f(300, 250, 0);
        glVertex3f(250, 250, 0);

        //Garis
        glVertex3f(250, 250, 0);
        glVertex3f(250, 150, 0);
        glEnd();
    }

    void kotakmerahtengah()
    {
        glLineWidth(5);
        glBegin(GL_LINE_STRIP);
        glColor3f(1, 0, 0);
        glVertex3f(-150, 0, 0);
        glVertex3f(-150, 50, 0);
        glVertex3f(150, 50, 0);
        glVertex3f(150, 0, 0);
        glEnd();
    }

    void kotakbirutengah()
    {
        glLineWidth(5);
        glBegin(GL_LINE_STRIP);
        glColor3f(0, 0, 1);
        glVertex3f(150, 0, 0);
        glVertex3f(150, -50, 0);
        glVertex3f(-150, -50, 0);
        glVertex3f(-150, 0, 0);
        glEnd();
    }

```



```

}

void kotakmerahbawah()
{
    glLineWidth(5);
    glBegin(GL_LINE_LOOP);
    glColor3f(1, 0, 0);
    glVertex3f(250, -100, 0);
    glVertex3f(250, -200, 0);
    glVertex3f(-250, -200, 0);
    glVertex3f(-250, -100, 0);
    glEnd();
}

void bawahbirukanan()
{
    glLineWidth(5);
    glBegin(GL_LINE_STRIP);
    glColor3f(0, 0, 1);
    glVertex3f(100, -200, 0);
    glVertex3f(100, -250, 0);
    glVertex3f(200, -250, 0);
    glVertex3f(200, -200, 0);
    glEnd();
}

void bawahbirukiri()
{
    glLineWidth(5);
    glBegin(GL_LINE_STRIP);
    glColor3f(0, 0, 1);
    glVertex3f(-100, -200, 0);
    glVertex3f(-100, -250, 0);
    glVertex3f(-200, -250, 0);
    glVertex3f(-200, -200, 0);
    glEnd();
}

void render()
{
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    kotakmerahkiri();
    kotakbirukiri();
    kotakbirukanan();
    kotakmerahkanan();
    kotakmerahengah();
    kotakbirutengah();
    kotakmerahbawah();
    bawahbirukanan();
    bawahbirukiri();
    glutSwapBuffers();
}

int main(int argc, char* argv[])
{
    glutInit(&argc, argv);

```

```
glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGBA | GLUT_DEPTH);  
glutInitWindowSize(800, 600);  
glutCreateWindow("");  
gluOrtho2D(-400, 400, -300, 300);  
glClearColor(0, 0, 0, 0);  
glutDisplayFunc(render);  
  
glutMainLoop();  
return 0;  
}
```