**LAPORAN PRAKTIKUM MODUL II**

**GRAFIKA KOMPUTER**

**Primitive Objects dan Objek 2 Dimensi**



**Disusun Oleh :**

**Rendra Kurniawan**

**130533608081**

**UNIVERSITAS NEGERI MALANG**

**FAKULTAS TEKNIK JURUSAN TEKNIK ELEKTRO**

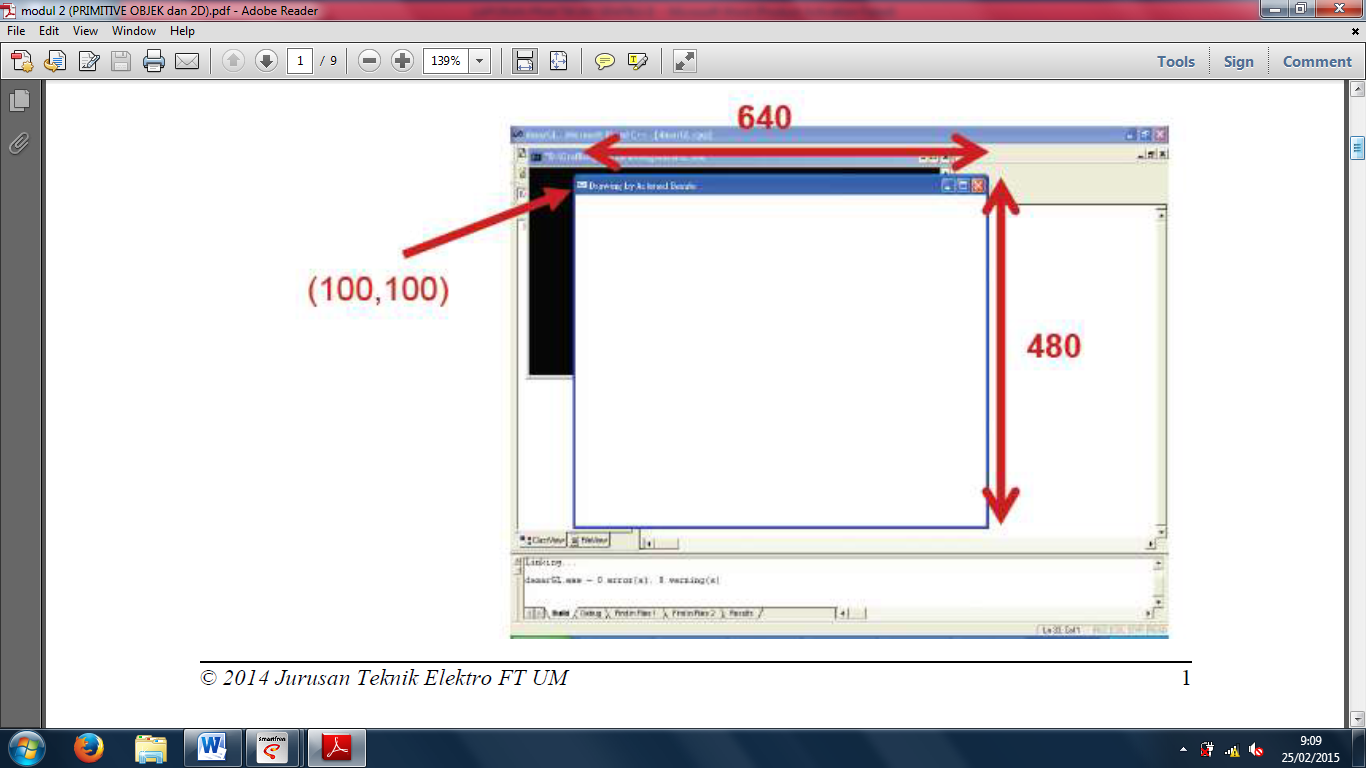
**PROGRAM STUDI S1 PENDIDIKAN TEKNIKINFORMATIKA**

**Februari – 2015**

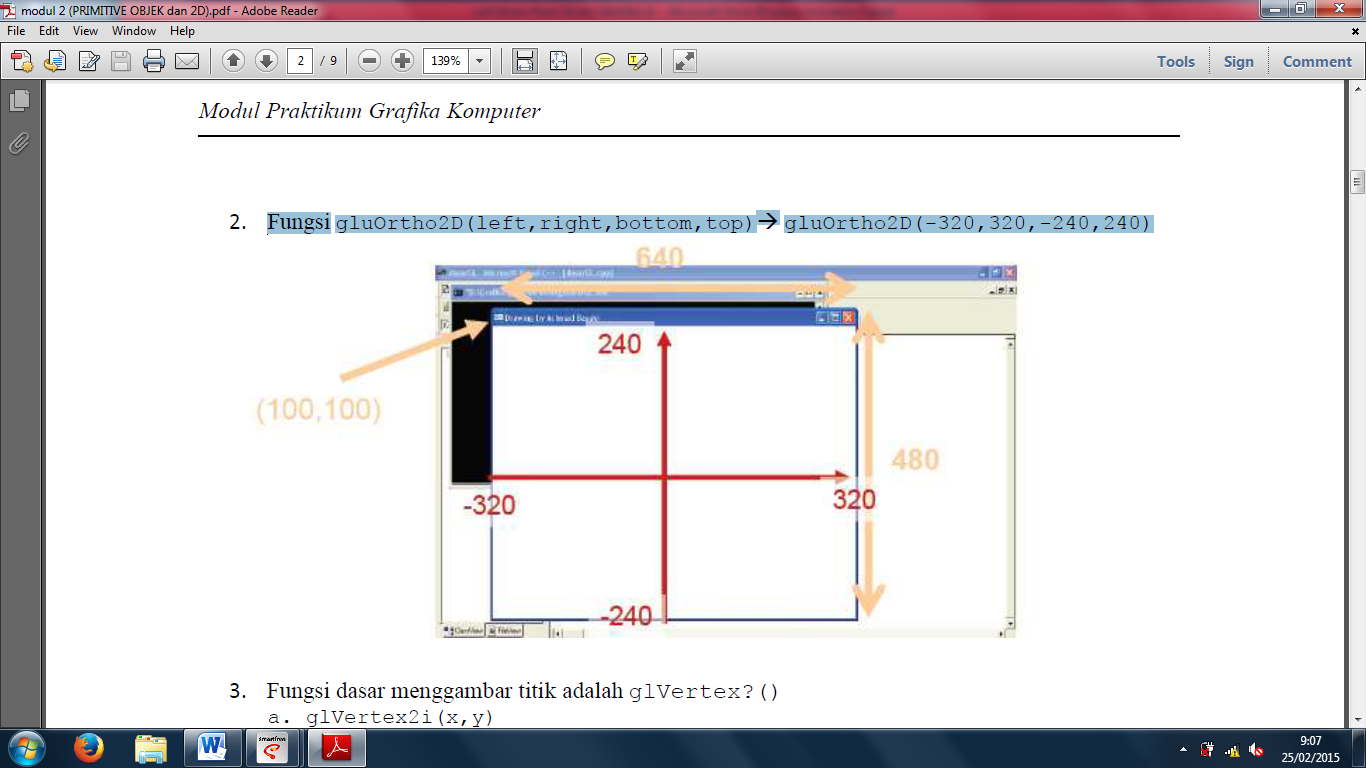
**MODUL 1**

**PENGENALAN GLUT**

1. **KOMPETENSI DASAR**
2. Memahami jenis-jenis primitive objects.
3. Memahami dan dapat membuat objek primitif.
4. Memahami penerapan Objek primitif menjadi bangun 2 dimensi dasar.
5. Memahami dan dapat membuat fungsi untuk bangun 2 dimensi dasar.
6. **DASAR TEORI**
7. Sistem Koordinat OpenGL



1. Fungsi gluOrtho2D(left,right,bottom,top) gluOrtho2D(-320,320, 240,240)



1. Fungsi dasar menggambar titik adalah glVertex?()
2. glVertex2i(x,y)

Menggambar titik pada koordinat x dan y, dengan nilai satuan berupa integer.

Contoh glVertex2i(10,10);

1. glVertex2f(x,y)

Menggambar titik pada koordinat x dan y, dengan nilai satuan berupa float.

Contoh glVertex2f(10.0,10.0);

1. glVertex2d(x,y)

Menggambar titik pada koordinat x dan y, dengan nilai satuan berupa double.

Contoh glVertex2d(10.0,10.0);

1. Objek primitif
2. Titik : GL\_POINTS
3. Garis : GL\_LINES
4. Poliline : GL\_LINE\_STRIP
5. Poligon (no fill) : GL\_LINE\_LOOP
6. Triangle : GL\_TRIANGLES

*Triangle* atau segitiga adalah tiga buah titik yang terhubung menjadi suatu segitiga dengan blok di tengahnya

1. Quads : GL\_QUADS

Quad atau segiempat adalah empat buah titik yang terhubung menjadi suatu segiempat dengan blok di tengahnya.

1. Polygon : GL\_POLYGON

Polygon merupakan suatu fungsi yang mirip dengan polyline, tetapi menghasilkan kurva tertutup dengan blok warna (*fill*).

1. Color

Untuk memberi warna pada objek, seperti titik atau garis, dapat dilakukan dengan menggunakan fungsi: glColor3f(red,green,blue);Di mana red, green, blue berada pada 0 sampai dengan 1, yang menunjukkan skala pencerahan dari masing-masing skala.

1. **LATIHAN**
   1. **Primitive Obyek**
      * 1. Berikut adalah script dasar pada fungsi Main() untuk kegiatan praktikum primitive objek

glutInitWindowPosition(100,100); //inisialisasi posisi windows program

glutInitWindowSize(640,480); //inisialisasi ukuran windows program dengan width 640 pixel dan height 480 pixel

gluOrtho2D(-320.,320.,-240.,240.); //inisialisasi ukuran sumbu x dan y sebesar 320 satuan ke kiri, 320 satuan ke kanan, 240 satuan ke bawah dan 240 satuan ke atas

* + - 1. Buatlah project baru pada Visual Studio dengan nama **prak1-Points**. Dan tambahkan fungsi callback untuk glutDisplayFunc(drawDot); . Tampilkan hasilnya berupa screenshot. Jelaskan Proses Rendering.

#include <stdlib.h>

#include <glut.h>

void drawDot (void) {

glClear(GL\_COLOR\_BUFFER\_BIT);

glBegin(GL\_POINTS); // deklarasi primitive object yang akan dibuat adalah obyek titik

glVertex2i(50,0); // posisi titik pada sumbu x sebesar 50 satuan ke kanan

glVertex2i(50,50); // posisi titik pada sumbu x sebesar 50 satuan ke kanan dan sumbu y 50 satuan ke atas

glVertex2i(0,0); // posisi titik pada sumbu x dan y adalah 0

glEnd();

glFlush();

}

void main(int argc, char \*\*argv) {

glutInit(&argc, argv);

glutInitDisplayMode (GLUT\_DEPTH | GLUT\_SINGLE | GLUT\_RGBA); // menentukan display mode awal (RGBA mode, single buffer, depth buffer)

glutInitWindowPosition(100,100); // fungsi meletakkan posisi windows

glutInitWindowSize(640,480); // fungsi membuat windows dengan ukuran sebesar 640 pixel x 480 pixel

glutCreateWindow("Pemrograman GLUT"); // fungsi membuat window dengan nama Pemrograman Glut

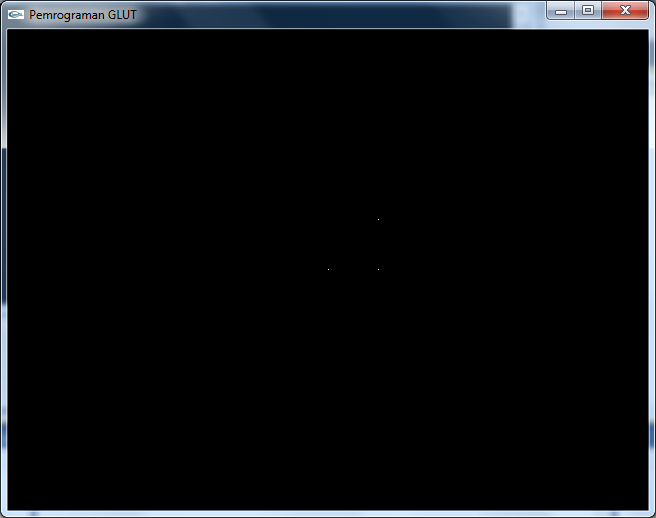
glutDisplayFunc(drawDot); // melakukan rendering fungsi drawdot yang telah dibuat

gluOrtho2D(-320.,320.,-240.,240.); // delklarasi ukuran sumbu x sebesar 320 satuan ke kiri, 320 satuan ke kanan, 240 satuan ke bawah dan 240 satuan ke atas

glutMainLoop();

}

**Output :**



**Penjelasan :**

Pembuatan titik dilakukan dengan pendeklarasian glBegin(GL\_POINTS); selanjutnya dilakukan proses rendering vertex untuk penentuan lokasi titik yang akan dibuat. Titik pertama diletakkan pada posisi 50 satuan ke kanan pada sumbu x (glVertex2i(50,0);), selanjutnya penentuan titik kedua pada lokasi 50 satuan ke arah sumbu x dan 50 satuan ke arah sumbu y (glVertex2i(50,50);), kemudian menentukan lokasi titik ketiga pada titik 0 pada sumbu x dan y (glVertex2i(0,0);),

* + - 1. Buatlah project baru pada Visual Studio dengan nama **prak1-PointsSize**. Dan tambahkan fungsi **glPointSize(int size);** untuk mengubah ukuran objek Points (**perhatian : ganti int size dengan nilai int yang diinginkan, sehingga fungsi menjadi glPointSize(5);**). Pindah-pindahkan glPointSize pada posisi a, b, dan c. Sajikan Screen shot hasil a, b, c dan simpulkan.

**Program a**

#include <stdlib.h>

#include <glut.h>

glPointSize(5);

void drawDot (void) {

glClear(GL\_COLOR\_BUFFER\_BIT);

glPointSize(5);

glBegin(GL\_POINTS);

glVertex2i(50,0);

glVertex2i(50,50);

glVertex2i(0,0);

glEnd();

glFlush();

}

void main(int argc, char \*\*argv) {

glutInit(&argc, argv);

glutInitDisplayMode (GLUT\_DEPTH | GLUT\_SINGLE | GLUT\_RGBA); // menentukan display mode awal (RGBA mode, single buffer, depth buffer)

glutInitWindowPosition(100,100); // fungsi meletakkan posisi windows

glutInitWindowSize(640,480); // fungsi membuat windows dengan ukuran sebesar 640 pixel x 480 pixel

glutCreateWindow("Pemrograman GLUT"); // fungsi membuat window dengan nama Pemrograman Glut

glutDisplayFunc(drawDot); // melakukan rendering fungsi drawdot yang telah dibuat

gluOrtho2D(-320.,320.,-240.,240.); // delklarasi ukuran sumbu x sebesar 320 satuan ke kiri, 320 satuan ke kanan, 240 satuan ke bawah dan 240 satuan ke atas

glutMainLoop();

}

**Output :** Error

**Program b**

#include <stdlib.h>

#include <glut.h>

void drawDot (void) {

glClear(GL\_COLOR\_BUFFER\_BIT);

glPointSize(5);

glBegin(GL\_POINTS);

glVertex2i(50,0);

glVertex2i(50,50);

glVertex2i(0,0);

glEnd();

glFlush();

}

void main(int argc, char \*\*argv) {

glutInit(&argc, argv);

glutInitDisplayMode (GLUT\_DEPTH | GLUT\_SINGLE | GLUT\_RGBA); // menentukan display mode awal (RGBA mode, single buffer, depth buffer)

glutInitWindowPosition(100,100); // fungsi meletakkan posisi windows

glutInitWindowSize(640,480); // fungsi membuat windows dengan ukuran sebesar 640 pixel x 480 pixel

glutCreateWindow("Pemrograman GLUT"); // fungsi membuat window dengan nama Pemrograman Glut

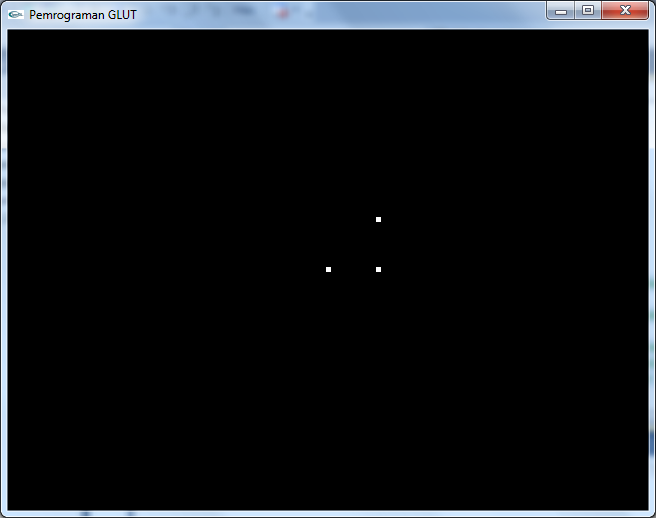
glutDisplayFunc(drawDot); // melakukan rendering fungsi drawdot yang telah dibuat

gluOrtho2D(-320.,320.,-240.,240.); // delklarasi ukuran sumbu x sebesar 320 satuan ke kiri, 320 satuan ke kanan, 240 satuan ke bawah dan 240 satuan ke atas

glutMainLoop();

}

**Output :**



**Program c**

#include <stdlib.h>

#include <glut.h>

void drawDot (void) {

glClear(GL\_COLOR\_BUFFER\_BIT);

glBegin(GL\_POINTS);

glVertex2i(50,0);

glPointSize(5);

glVertex2i(50,50);

glVertex2i(0,0);

glEnd();

glFlush();

}

void main(int argc, char \*\*argv) {

glutInit(&argc, argv);

glutInitDisplayMode (GLUT\_DEPTH | GLUT\_SINGLE | GLUT\_RGBA); // menentukan display mode awal (RGBA mode, single buffer, depth buffer)

glutInitWindowPosition(100,100); // fungsi meletakkan posisi windows

glutInitWindowSize(640,480); // fungsi membuat windows dengan ukuran sebesar 640 pixel x 480 pixel

glutCreateWindow("Pemrograman GLUT"); // fungsi membuat window dengan nama Pemrograman Glut

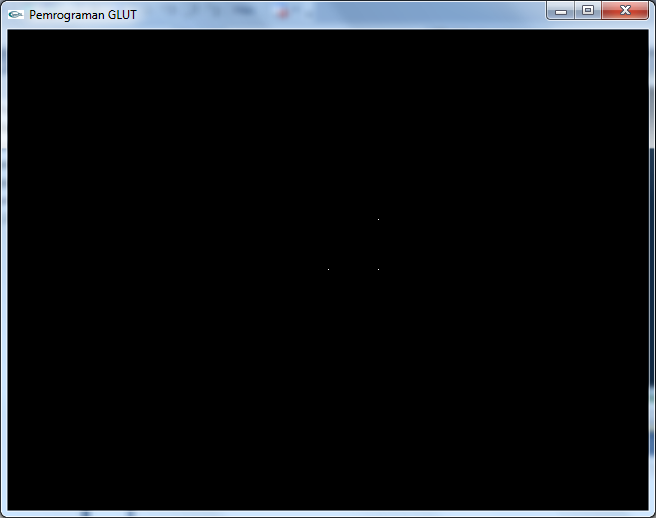
glutDisplayFunc(drawDot); // melakukan rendering fungsi drawdot yang telah dibuat

gluOrtho2D(-320.,320.,-240.,240.); // delklarasi ukuran sumbu x sebesar 320 satuan ke kiri, 320 satuan ke kanan, 240 satuan ke bawah dan 240 satuan ke atas

glutMainLoop();

}

**Output :**



**Penjelasan :**

Pembuatan titik dilakukan dengan pendeklarasian glBegin(GL\_POINTS); selanjutnya dilakukan proses rendering vertex untuk penentuan lokasi titik yang akan dibuat. Titik pertama diletakkan pada posisi 50 satuan ke kanan pada sumbu x (glVertex2i(50,0);), selanjutnya penentuan titik kedua pada lokasi 50 satuan ke arah sumbu x dan 50 satuan ke arah sumbu y (glVertex2i(50,50);), kemudian menentukan lokasi titik ketiga pada titik 0 pada sumbu x dan y (glVertex2i(0,0);), Penentuan ukuran dari titik dilakukan dengan deklarasi glPointSize(5); dimana angka 5 dapat diganti dengan angka integer lainya. Program a error karena glPointSize(5); yang dideklarasikan diletakkan diluar fungsi dan berdiri sebagai sebuah script tanpa ada hubungan dengan fungsi apapun sehingga menjadikan program error. Program b berhasil mengubah ukuran titik menjadi 5 pixel dan penempatan glPointSize(5); sudah tepat. Program c glPointSize(5); tidak memberikan pengaruh karena peletakkan glPointSize(5); harus sebelum deklarasi glBegin(GL\_POINTS);

* + - 1. Buatlah project baru pada Visual Studio dengan nama **prak1-Lines**. Dan tambahkan fungsi callback untuk glutDisplayFunc(drawLine); . Tampilkan hasilnya berupa screenshot. Dan Jelaskan Proses Rendering untuk vertexnya.

#include <stdlib.h>

#include <glut.h>

void drawLine(void) {

glClear(GL\_COLOR\_BUFFER\_BIT);

glBegin(GL\_LINES);

glVertex2d(60,50);

glVertex2d(50,100);

glVertex2d(100,0);

glEnd();

glFlush();

}

void main(int argc, char \*\*argv) {

glutInit(&argc, argv);

glutInitDisplayMode (GLUT\_DEPTH | GLUT\_SINGLE | GLUT\_RGBA); // menentukan display mode awal (RGBA mode, single buffer, depth buffer)

glutInitWindowPosition(100,100); // fungsi meletakkan posisi windows

glutInitWindowSize(640,480); // fungsi membuat windows dengan ukuran sebesar 640 pixel x 480 pixel

glutCreateWindow("Pemrograman GLUT"); // fungsi membuat window dengan nama Pemrograman Glut

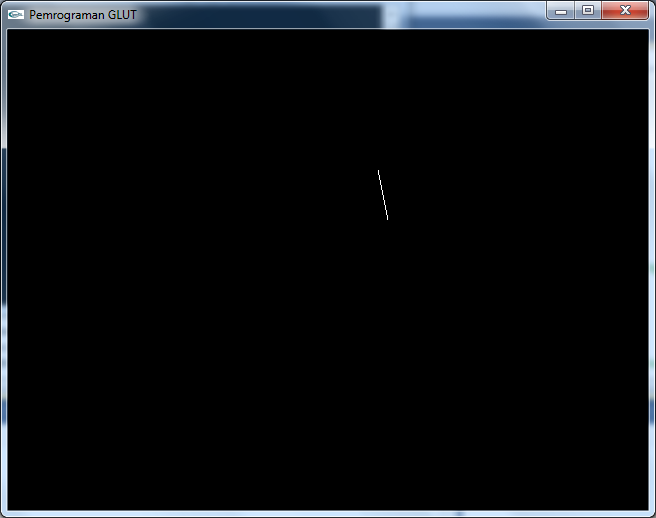
glutDisplayFunc(drawLine); // melakukan rendering fungsi drawdot yang telah dibuat

gluOrtho2D(-320.,320.,-240.,240.); // delklarasi ukuran sumbu x sebesar 320 satuan ke kiri, 320 satuan ke kanan, 240 satuan ke bawah dan 240 satuan ke atas

glutMainLoop();

}

**Output :**



**Penjelasan :**

Pembuatan garis dilakukan dengan pendeklarasian glBegin(GL\_LINES); selanjutnya dilakukan proses rendering vertex untuk penentuan lokasi titik yang akan dibuat garis. Garis dibuat dengan menghubungkan 2 titik yang telah dideklarasikan. Penggunaan glBegin(GL\_LINES); hanya akan menghasilkan sebuah garis yang dibuat dari pendeklarasian 2 titik pertama pada script. Titik pertama diletakkan pada posisi 60 satuan ke kanan pada sumbu x dan 50 satuan ke arah sumbu y (glVertex2i(60,50);), selanjutnya penentuan titik kedua pada lokasi 50 satuan ke arah sumbu x dan 100 satuan ke arah sumbu y (glVertex2i(50,100);), sehingga garis yang dibuat adalah garis yang mengubungkan 2 titik tersebut. Titik ketiga dideklarasikan tetapi tidak akan meberikan pengaruh apapun.

* + - 1. Buatlah project baru pada Visual Studio dengan nama **prak1-LineWidth**. Dan tambahkan fungsi **glLineWidth(int size);** untuk mengubah ukuran objek Points. Pindah-pindahkan **glLineWidth** pada a,b, dan c. Sajikan Screen shot hasil a,b,c dan simpulkan.

**Program a**

#include <stdlib.h>

#include <glut.h>

void drawLine(void) {

glLineWidth(10);

glClear(GL\_COLOR\_BUFFER\_BIT);

glBegin(GL\_LINES);

glVertex2d(60,50);

glVertex2d(50,100);

glVertex2d(100,0);

glEnd();

glFlush();

}

void main(int argc, char \*\*argv) {

glutInit(&argc, argv);

glutInitDisplayMode (GLUT\_DEPTH | GLUT\_SINGLE | GLUT\_RGBA); // menentukan display mode awal (RGBA mode, single buffer, depth buffer)

glutInitWindowPosition(100,100); // fungsi meletakkan posisi windows

glutInitWindowSize(640,480); // fungsi membuat windows dengan ukuran sebesar 640 pixel x 480 pixel

glutCreateWindow("Pemrograman GLUT"); // fungsi membuat window dengan nama Pemrograman Glut

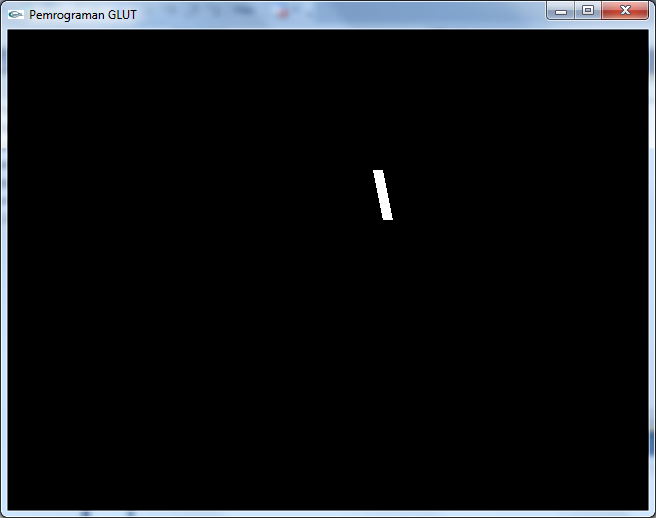
glutDisplayFunc(drawLine); // melakukan rendering fungsi drawdot yang telah dibuat

gluOrtho2D(-320.,320.,-240.,240.); // delklarasi ukuran sumbu x sebesar 320 satuan ke kiri, 320 satuan ke kanan, 240 satuan ke bawah dan 240 satuan ke atas

glutMainLoop();

}

**Output :**



**Program b dan c**

#include <stdlib.h>

#include <glut.h>

void drawLine(void) {

glClear(GL\_COLOR\_BUFFER\_BIT);

glBegin(GL\_LINES);

glVertex2d(60,50);

glVertex2d(50,100);

glLineWidth(10); // posisi b

glVertex2d(100,0);

// posisi c

glEnd();

glFlush();

}

void main(int argc, char \*\*argv) {

glutInit(&argc, argv);

glutInitDisplayMode (GLUT\_DEPTH | GLUT\_SINGLE | GLUT\_RGBA); // menentukan display mode awal (RGBA mode, single buffer, depth buffer)

glutInitWindowPosition(100,100); // fungsi meletakkan posisi windows

glutInitWindowSize(640,480); // fungsi membuat windows dengan ukuran sebesar 640 pixel x 480 pixel

glutCreateWindow("Pemrograman GLUT"); // fungsi membuat window dengan nama Pemrograman Glut

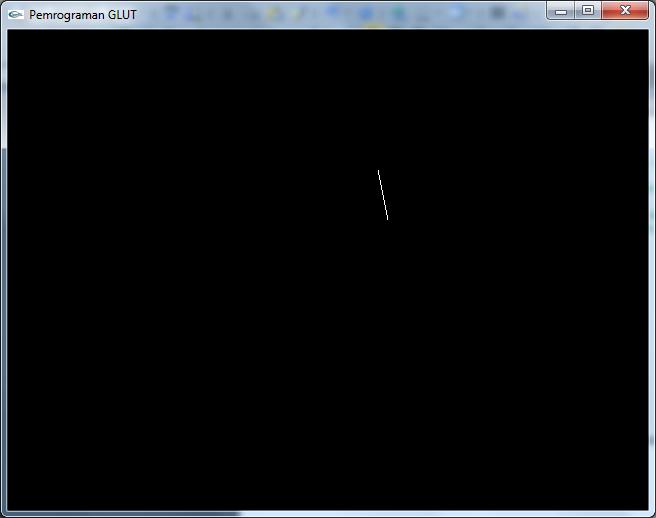
glutDisplayFunc(drawLine); // melakukan rendering fungsi drawdot yang telah dibuat

gluOrtho2D(-320.,320.,-240.,240.); // delklarasi ukuran sumbu x sebesar 320 satuan ke kiri, 320 satuan ke kanan, 240 satuan ke bawah dan 240 satuan ke atas

glutMainLoop();

}

**Output :**



**Penjelasan :**

Pembuatan garis dilakukan dengan pendeklarasian glBegin(GL\_LINES); selanjutnya dilakukan proses rendering vertex untuk penentuan lokasi titik yang akan dibuat garis. Garis dibuat dengan menghubungkan 2 titik yang telah dideklarasikan. Penggunaan glBegin(GL\_LINES); hanya akan menghasilkan sebuah garis yang dibuat dari pendeklarasian 2 titik pertama pada script. Titik pertama diletakkan pada posisi 60 satuan ke kanan pada sumbu x dan 50 satuan ke arah sumbu y (glVertex2i(60,50);), selanjutnya penentuan titik kedua pada lokasi 50 satuan ke arah sumbu x dan 100 satuan ke arah sumbu y (glVertex2i(50,100);), sehingga garis yang dibuat adalah garis yang mengubungkan 2 titik tersebut. Titik ketiga dideklarasikan tetapi tidak akan meberikan pengaruh apapun. Penentuan ukuran tebal garis dilakukan dengan deklarasi glLineWidth(10); dimana angka 10 dapat diganti dengan angka integer lainya. Program a berhasil mengubah ukuran tebal garis menjadi 10 pixel karena penempatan glLineWidth(10); sudah tepat yaitu sebelum deklarasi glBegin(GL\_LINES); Program b dan c tidak memberikan pengaruh karena peletakkan glLineWidth(10); harus sebelum deklarasi glBegin(GL\_LINES);

* + - 1. Buatlah project baru pada Visual Studio dengan nama **prak1-LineStrip**. Buatlah agar hasil yang ditampilkan membentuk **bangun simetris terbuka**. Sajikan screenshotnya. Dan Jelaskan proses rendering vertexnya.

#include <stdlib.h>

#include <glut.h>

void drawPoliline(void) {

glClear(GL\_COLOR\_BUFFER\_BIT);

glBegin(GL\_LINE\_STRIP);

glVertex2d(-100,0);

glVertex2d(0,100);

glVertex2d(100,0);

glEnd();

glFlush();

}

void main(int argc, char \*\*argv) {

glutInit(&argc, argv);

glutInitDisplayMode (GLUT\_DEPTH | GLUT\_SINGLE | GLUT\_RGBA); // menentukan display mode awal (RGBA mode, single buffer, depth buffer)

glutInitWindowPosition(100,100); // fungsi meletakkan posisi windows

glutInitWindowSize(640,480); // fungsi membuat windows dengan ukuran sebesar 640 pixel x 480 pixel

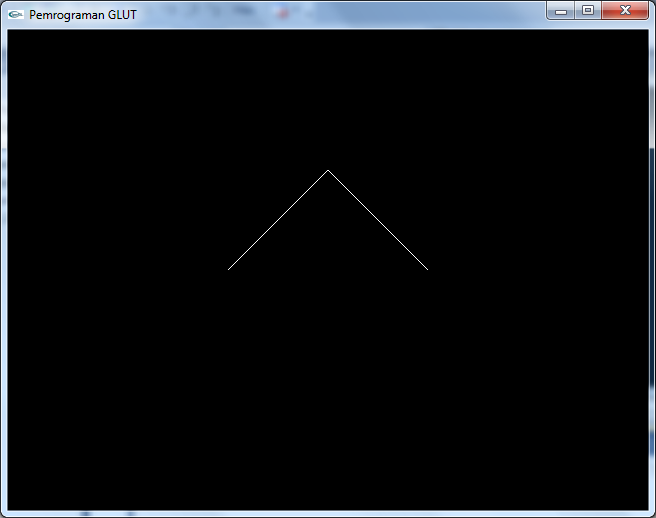
glutCreateWindow("Pemrograman GLUT"); // fungsi membuat window dengan nama Pemrograman Glut

glutDisplayFunc(drawPoliline); // melakukan rendering fungsi drawdot yang telah dibuat

gluOrtho2D(-320.,320.,-240.,240.); // delklarasi ukuran sumbu x sebesar 320 satuan ke kiri, 320 satuan ke kanan, 240 satuan ke bawah dan 240 satuan ke atas

glutMainLoop();}

**Output :**



**Penjelasan :**

Pembuatan garis dilakukan dengan pendeklarasian glBegin(GL\_LINE\_STRIP); dimana terdapat perbedaan pada glBegin(GL\_LINES); dan glBegin(GL\_LINE\_STRIP);. glBegin(GL\_LINE\_STRIP); dapat memproses banyak titik untuk dibuat garis sehingga bisa membuat bidang terbuka. Namun kelemahannya adalah bahwa glBegin(GL\_LINE\_STRIP); hanya akan membuat garis sampai deklarasi titik terakhir dan titik terakhir tidak akan dihubungkan dengan itik pertama. Selanjutnya dilakukan proses rendering vertex untuk penentuan lokasi titik yang akan dibuat garis. Garis dibuat dengan menghubungkan titik yang telah dideklarasikan. Titik pertama diletakkan pada posisi 50 satuan ke kanan pada sumbu x (glVertex2i(50,0);), selanjutnya penentuan titik kedua pada lokasi 100 satuan ke arah sumbu y atas (glVertex2i(0,100);), selanjutnya penentuan titik ketiga pada lokasi 50 satuan ke arah sumbu x kiri (glVertex2i(-50,0);), sehingga garis yang dibuat adalah garis yang mengubungkan 3 titik tersebut tetapi titik pertama dan ketiga tidak terbentuk garis

* + - 1. Buatlah project baru pada Visual Studio dengan nama **prak1-Poligon**. Buatlah agar hasil yang ditampilkan membentuk **bangun simetris**. Sajikan screenshotnya. Dan jelaskan proses rendering vertexnya.

#include <stdlib.h>

#include <glut.h>

void drawPoliline(void) {

glClear(GL\_COLOR\_BUFFER\_BIT);

glBegin(GL\_LINE\_LOOP);

glVertex2d(50,0);

glVertex2d(0,50);

glVertex2d(-50,0);

glVertex2d(-25,-50);

glVertex2d(25,-50);

glEnd();

glFlush();

}

void main(int argc, char \*\*argv) {

glutInit(&argc, argv);

glutInitDisplayMode (GLUT\_DEPTH | GLUT\_SINGLE | GLUT\_RGBA); // menentukan display mode awal (RGBA mode, single buffer, depth buffer)

glutInitWindowPosition(100,100); // fungsi meletakkan posisi windows

glutInitWindowSize(640,480); // fungsi membuat windows dengan ukuran sebesar 640 pixel x 480 pixel

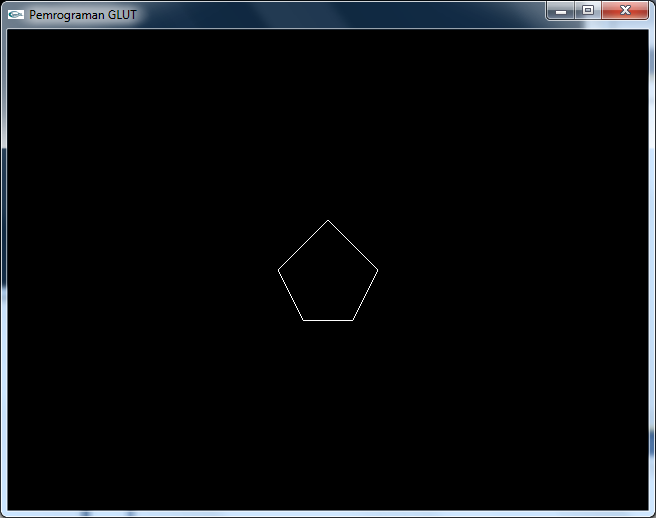
glutCreateWindow("Pemrograman GLUT"); // fungsi membuat window dengan nama Pemrograman Glut

glutDisplayFunc(drawPoliline); // melakukan rendering fungsi drawdot yang telah dibuat

gluOrtho2D(-320.,320.,-240.,240.); // delklarasi ukuran sumbu x sebesar 320 satuan ke kiri, 320 satuan ke kanan, 240 satuan ke bawah dan 240 satuan ke atas

glutMainLoop();

}



**Penjelasan :**

Pembuatan garis dilakukan dengan pendeklarasian glBegin(GL\_LINE\_LOOP); yang dapat memproses banyak titik untuk dibuat garis sehingga bisa membuat bidang dari garis. Kelebihan dibanding glBegin(GL\_LINE\_STRIP); adalah deklarasi titik titik terakhir akan dihubungkan dengan titik pertama yang dideklarasikan. Selanjutnya dilakukan proses rendering vertex untuk penentuan lokasi titik yang akan dibuat garis. Garis dibuat dengan menghubungkan titik yang telah dideklarasikan. Titik pertama diletakkan pada posisi 50 satuan ke kanan pada sumbu x (glVertex2i(50,0);), selanjutnya penentuan titik kedua pada lokasi 100 satuan ke arah sumbu y atas (glVertex2i(0,100);), selanjutnya penentuan titik ketiga pada lokasi 50 satuan ke arah sumbu x kiri (glVertex2i(-50,0);), selanjutnya penentuan titik kempat pada lokasi 25 satuan ke arah sumbu x kiri dan 50 satuan kearah bawah sumbu y (glVertex2i(-25,-50);), selanjutnya penentuan titik kelima pada lokasi 25 satuan ke arah sumbu x kanan dan 50 satuan ke arah sumbu y bawah (glVertex2i(25,-50);), sehingga garis yang dibuat adalah garis yang mengubungkan 5 titik tersebut.

* + - 1. Buatlah project baru pada Visual Studio dengan nama **prak1-Triangle**. Buatlah agar hasil yang ditampilkan membentuk **2 segitiga yang terpisah**. Sajikan screenshotnya. Dan jelaskan proses rendering vertexnya

#include <stdlib.h>

#include <glut.h>

void drawPoliline(void) {

glClear(GL\_COLOR\_BUFFER\_BIT);

glBegin(GL\_TRIANGLES);

glVertex2d(10,0);

glVertex2d(35,50);

glVertex2d(60,0);

glVertex2d(-10,0);

glVertex2d(-35,50);

glVertex2d(-60,0);

glEnd();

glFlush();

}

void main(int argc, char \*\*argv) {

glutInit(&argc, argv);

glutInitDisplayMode (GLUT\_DEPTH | GLUT\_SINGLE | GLUT\_RGBA); // menentukan display mode awal (RGBA mode, single buffer, depth buffer)

glutInitWindowPosition(100,100); // fungsi meletakkan posisi windows

glutInitWindowSize(640,480); // fungsi membuat windows dengan ukuran sebesar 640 pixel x 480 pixel

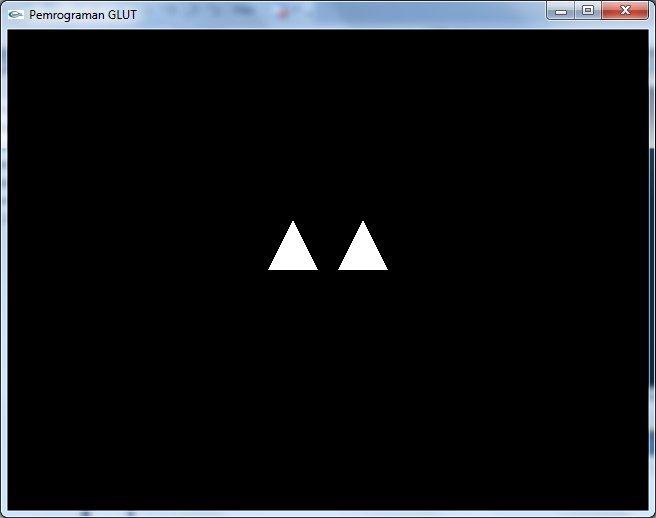
glutCreateWindow("Pemrograman GLUT"); // fungsi membuat window dengan nama Pemrograman Glut

glutDisplayFunc(drawPoliline); // melakukan rendering fungsi drawdot yang telah dibuat

gluOrtho2D(-320.,320.,-240.,240.); // delklarasi ukuran sumbu x sebesar 320 satuan ke kiri, 320 satuan ke kanan, 240 satuan ke bawah dan 240 satuan ke atas

glutMainLoop();

}



**Penjelasan :**

Pembuatan bidang segitiga dilakukan dengan pendeklarasian glBegin(GL\_TRIANGLES); yang dapat memproses titik untuk dibuat bidang segitiga. Titik yang akan dibuat segitiga adalah 3 titik pertama yang dideklarasikan. Jika terdapat 3 titik lagi maka 3 titik tersebut akan diproses lagi untuk dibuat bidang segitiga. Titik pertama diletakkan pada posisi 10 satuan ke kanan pada sumbu x (glVertex2i(10,0);), selanjutnya penentuan titik kedua pada lokasi 35 satuan ke arah sumbu x kanan 50 satuan ke arah atas sumbu y (glVertex2i(35,50);), selanjutnya penentuan titik ketiga pada lokasi 60 satuan ke arah sumbu x kanan (glVertex2i(60,0);) titik – titik tersebut diproses menjadi bidang segitiga pertama.

Selanjutnya untuk membuat segitiga kedua penentuan titik kempat pada lokasi 10 satuan ke arah sumbu x kiri (glVertex2i(-10,0);), selanjutnya penentuan titik kelima pada lokasi 35 satuan ke arah sumbu x kiri dan 50 satuan ke arah sumbu y atas (glVertex2i(35,50);), selanjutnya penentuan titik keenam pada lokasi 60 satuan ke arah sumbu x kiri (glVertex2i(35,50);), sehingga terbentuk 2 segitiga.

* + - 1. Buatlah project baru pada Visual Studio dengan nama **prak1-TriangleStrip**. Buatlah agar hasil yang ditampilkan membentuk **Bangun Bebas**. Sajikan screenshotnya. Dan jelaskan proses rendering vertexnya.

#include <stdlib.h>

#include <glut.h>

void drawPoliline(void) {

glClear(GL\_COLOR\_BUFFER\_BIT);

glBegin(GL\_TRIANGLE\_STRIP);

glVertex2d(10,0);

glVertex2d(35,50);

glVertex2d(60,0);

glVertex2d(-10,0);

glVertex2d(-35,50);

glVertex2d(-60,0);

glEnd();

glFlush();

}

void main(int argc, char \*\*argv) {

glutInit(&argc, argv);

glutInitDisplayMode (GLUT\_DEPTH | GLUT\_SINGLE | GLUT\_RGBA); // menentukan display mode awal (RGBA mode, single buffer, depth buffer)

glutInitWindowPosition(100,100); // fungsi meletakkan posisi windows

glutInitWindowSize(640,480); // fungsi membuat windows dengan ukuran sebesar 640 pixel x 480 pixel

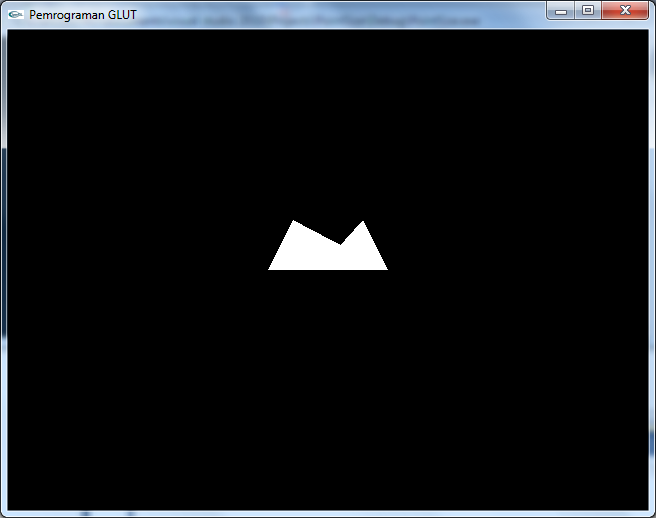
glutCreateWindow("Pemrograman GLUT"); // fungsi membuat window dengan nama Pemrograman Glut

glutDisplayFunc(drawPoliline); // melakukan rendering fungsi drawdot yang telah dibuat

gluOrtho2D(-320.,320.,-240.,240.); // delklarasi ukuran sumbu x sebesar 320 satuan ke kiri, 320 satuan ke kanan, 240 satuan ke bawah dan 240 satuan ke atas

glutMainLoop();

}



**Penjelasan :**

Pembuatan bidang bebas dilakukan dengan pendeklarasian glBegin(GL\_TRIANGLE\_STRIP); yang dapat memproses titik untuk dibuat bidang bebas. Titik pertama diletakkan pada posisi 10 satuan ke kanan pada sumbu x (glVertex2i(10,0);), selanjutnya penentuan titik kedua pada lokasi 35 satuan ke arah sumbu x kanan 50 satuan ke arah atas sumbu y (glVertex2i(35,50);), selanjutnya penentuan titik ketiga pada lokasi 60 satuan ke arah sumbu x kanan (glVertex2i(60,0);) titik – titik tersebut diproses menjadi bidang segitiga pertama.

Selanjutnya untuk membuat segitiga kedua penentuan titik kempat pada lokasi 10 satuan ke arah sumbu x kiri (glVertex2i(-10,0);), selanjutnya penentuan titik kelima pada lokasi 35 satuan ke arah sumbu x kiri dan 50 satuan ke arah sumbu y atas (glVertex2i(35,50);), selanjutnya penentuan titik keenam pada lokasi 60 satuan ke arah sumbu x kiri (glVertex2i(35,50);), sehingga terbentuk bidang yang tidak beraturan.

* + - 1. Buatlah project baru pada Visual Studio dengan nama **prak1-TriangleFan**. Buatlah agar hasil yang ditampilkan membentuk **Bangun Bebas**. Sajikan screenshotnya. Dan jelaskan proses rendering vertexnya.

#include <stdlib.h>

#include <glut.h>

void drawPoliline(void) {

glClear(GL\_COLOR\_BUFFER\_BIT);

glBegin(GL\_TRIANGLE\_FAN);

glVertex2d(10,0);

glVertex2d(35,50);

glVertex2d(60,0);

glVertex2d(-10,0);

glVertex2d(-35,50);

glVertex2d(-60,0);

glEnd();

glFlush();

}

void main(int argc, char \*\*argv) {

glutInit(&argc, argv);

glutInitDisplayMode (GLUT\_DEPTH | GLUT\_SINGLE | GLUT\_RGBA); // menentukan display mode awal (RGBA mode, single buffer, depth buffer)

glutInitWindowPosition(100,100); // fungsi meletakkan posisi windows

glutInitWindowSize(640,480); // fungsi membuat windows dengan ukuran sebesar 640 pixel x 480 pixel

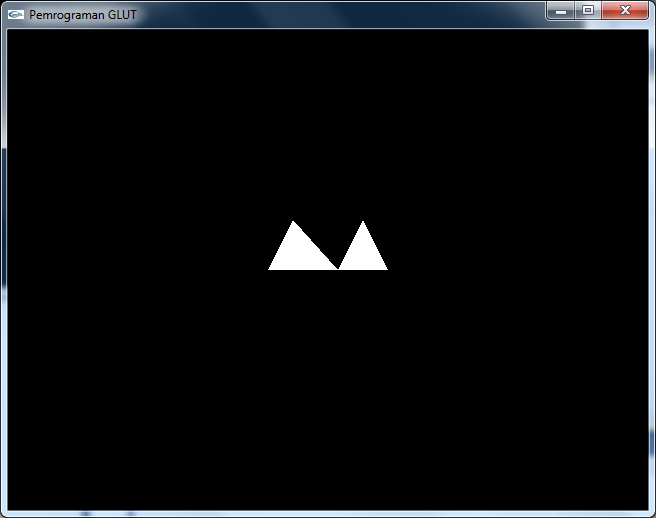
glutCreateWindow("Pemrograman GLUT"); // fungsi membuat window dengan nama Pemrograman Glut

glutDisplayFunc(drawPoliline); // melakukan rendering fungsi drawdot yang telah dibuat

gluOrtho2D(-320.,320.,-240.,240.); // delklarasi ukuran sumbu x sebesar 320 satuan ke kiri, 320 satuan ke kanan, 240 satuan ke bawah dan 240 satuan ke atas

glutMainLoop();

}



**Penjelasan :**

Pembuatan bidang dilakukan dengan pendeklarasian glBegin(GL\_TRIANGLE\_FAN); yang dapat memproses titik untuk dibuat bidang. Titik yang akan diproses adalah 3 titik pertama yang dideklarasikan. Jika terdapat titik tambahan maka titik tersebut akan langsung dihubungkan dengan bidang segitiga yang telah dibuat sebelumnya. Titik pertama diletakkan pada posisi 10 satuan ke kanan pada sumbu x (glVertex2i(10,0);), selanjutnya penentuan titik kedua pada lokasi 35 satuan ke arah sumbu x kanan 50 satuan ke arah atas sumbu y (glVertex2i(35,50);), selanjutnya penentuan titik ketiga pada lokasi 60 satuan ke arah sumbu x kanan (glVertex2i(60,0);) titik – titik tersebut diproses menjadi bidang segitiga pertama.

Selanjutnya untuk membuat segitiga kedua penentuan titik kempat pada lokasi 10 satuan ke arah sumbu x kiri (glVertex2i(-10,0);), selanjutnya penentuan titik kelima pada lokasi 35 satuan ke arah sumbu x kiri dan 50 satuan ke arah sumbu y atas (glVertex2i(35,50);), selanjutnya penentuan titik keenam pada lokasi 60 satuan ke arah sumbu x kiri (glVertex2i(35,50);), sehingga terbentuk 2 segitiga yang saling terhubung.

* + - 1. Buatlah project baru pada Visual Studio dengan nama **prak1-Quads**. Buatlah agar hasil yang ditampilkan membentuk **2 Bangun Segi Empat Terpisah**. Sajikan screenshotnya. Dan jelaskan proses rendering vertexnya.

#include <stdlib.h>

#include <glut.h>

void drawPoliline(){

glBegin(GL\_QUADS);

glVertex2i(100,0);

glVertex2i(100,50);

glVertex2i(50,100);

glVertex2i(50,0);

glVertex2i(-100,0);

glVertex2i(-100,50);

glVertex2i(-50,100);

glVertex2i(-50,0);

glEnd();

glFlush();

}

void main(int argc, char \*\*argv) {

glutInit(&argc, argv);

glutInitDisplayMode (GLUT\_DEPTH | GLUT\_SINGLE | GLUT\_RGBA); // menentukan display mode awal (RGBA mode, single buffer, depth buffer)

glutInitWindowPosition(100,100); // fungsi meletakkan posisi windows

glutInitWindowSize(640,480); // fungsi membuat windows dengan ukuran sebesar 640 pixel x 480 pixel

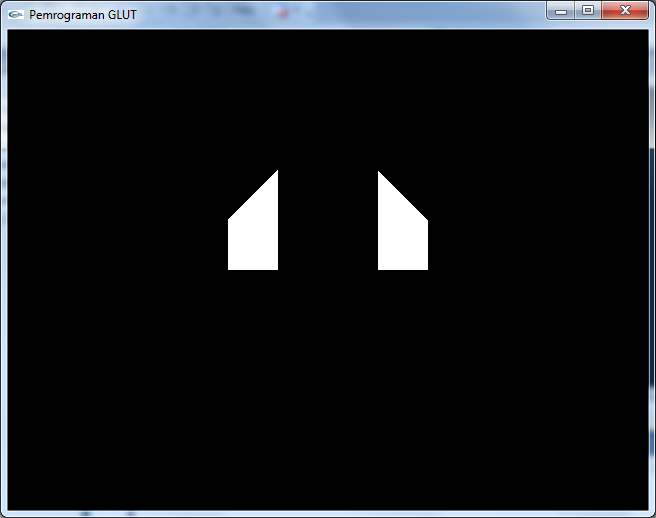
glutCreateWindow("Pemrograman GLUT"); // fungsi membuat window dengan nama Pemrograman Glut

glutDisplayFunc(drawPoliline); // melakukan rendering fungsi drawdot yang telah dibuat

gluOrtho2D(-320.,320.,-240.,240.); // delklarasi ukuran sumbu x sebesar 320 satuan ke kiri, 320 satuan ke kanan, 240 satuan ke bawah dan 240 satuan ke atas

glutMainLoop();

}



**Penjelasan :**

Pembuatan bidang dilakukan dengan pendeklarasian glBegin(GL\_QUADS); yang dapat memproses 4 titik untuk dibuat bidang segiempat. Titik yang akan diproses adalah 4 titik pertama yang dideklarasikan. Jika terdapat 4 titik tambahan maka titik tersebut akan dibuat bidang segiempat baru. Titik pertama diletakkan pada posisi 100 satuan ke kanan pada sumbu x (glVertex2i(100,0);), selanjutnya penentuan titik kedua pada lokasi 100 satuan ke arah sumbu x kanan dan 50 satuan ke arah atas sumbu y (glVertex2i(100,50);), selanjutnya penentuan titik ketiga pada lokasi 50 satuan ke arah sumbu x kanan dan 100 satuan ke arah sumbu y (glVertex2i(50,100);) dan selanjutnya penentuan titik keempat pada lokasi 50 satuan ke arah sumbu x kanan (glVertex2i(50,0);) titik – titik tersebut diproses menjadi bidang segiempat. Selanjutnya untuk membuat segiempat kedua penentuan titik kelima pada posisi 100 satuan ke kiri pada sumbu x (glVertex2i(-100,0);), selanjutnya penentuan titik keenam pada lokasi 100 satuan ke arah sumbu x kiri dan 50 satuan ke arah atas sumbu y (glVertex2i(-100,50);), selanjutnya penentuan titik ketujuh pada lokasi 50 satuan ke arah sumbu x kiri dan 100 satuan ke arah sumbu y (glVertex2i(-50,100);) dan selanjutnya penentuan titik kedelapan pada lokasi 50 satuan ke arah sumbu x kiri (glVertex2i(-50,0);) titik – titik tersebut diproses menjadi bidang segiempat kedua sehingga terbentuk 2 segiempat yang saling terpisah.

* + - 1. Buatlah project baru pada Visual Studio dengan nama **prak1-QuadStrip**. Buatlah agar hasil yang ditampilkan membentuk **Bangun Bebas**. Sajikan screenshotnya. Dan jelaskan proses rendering vertexnya.

#include <stdlib.h>

#include <glut.h>

void drawPoliline(){

glBegin(GL\_QUAD\_STRIP);

glVertex2i(100,0);

glVertex2i(100,50);

glVertex2i(50,100);

glVertex2i(50,0);

glVertex2i(-100,0);

glVertex2i(-100,50);

glVertex2i(-50,100);

glVertex2i(-50,0);

glEnd();

glFlush();

}

void main(int argc, char \*\*argv) {

glutInit(&argc, argv);

glutInitDisplayMode (GLUT\_DEPTH | GLUT\_SINGLE | GLUT\_RGBA); // menentukan display mode awal (RGBA mode, single buffer, depth buffer)

glutInitWindowPosition(100,100); // fungsi meletakkan posisi windows

glutInitWindowSize(640,480); // fungsi membuat windows dengan ukuran sebesar 640 pixel x 480 pixel

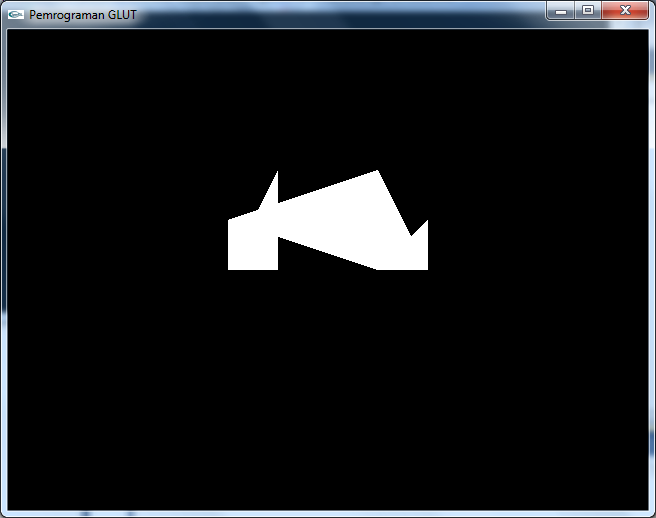
glutCreateWindow("Pemrograman GLUT"); // fungsi membuat window dengan nama Pemrograman Glut

glutDisplayFunc(drawPoliline); // melakukan rendering fungsi drawdot yang telah dibuat

gluOrtho2D(-320.,320.,-240.,240.); // delklarasi ukuran sumbu x sebesar 320 satuan ke kiri, 320 satuan ke kanan, 240 satuan ke bawah dan 240 satuan ke atas

glutMainLoop();

}



**Penjelasan :**

Pembuatan bidang dilakukan dengan pendeklarasian glBegin(GL\_QUAD\_STRIP); yang dapat memproses titik untuk dibuat bidang tak beraturan. Titik pertama diletakkan pada posisi 100 satuan ke kanan pada sumbu x (glVertex2i(100,0);), selanjutnya penentuan titik kedua pada lokasi 100 satuan ke arah sumbu x kanan dan 50 satuan ke arah atas sumbu y (glVertex2i(100,50);), selanjutnya penentuan titik ketiga pada lokasi 50 satuan ke arah sumbu x kanan dan 100 satuan ke arah sumbu y (glVertex2i(50,100);) dan selanjutnya penentuan titik keempat pada lokasi 50 satuan ke arah sumbu x kanan (glVertex2i(50,0);) Selanjutnya untuk membuat segiempat kedua penentuan titik kelima pada posisi 100 satuan ke kiri pada sumbu x (glVertex2i(-100,0);), selanjutnya penentuan titik keenam pada lokasi 100 satuan ke arah sumbu x kiri dan 50 satuan ke arah atas sumbu y (glVertex2i(-100,50);), selanjutnya penentuan titik ketujuh pada lokasi 50 satuan ke arah sumbu x kiri dan 100 satuan ke arah sumbu y (glVertex2i(-50,100);) dan selanjutnya penentuan titik kedelapan pada lokasi 50 satuan ke arah sumbu x kiri (glVertex2i(-50,0);) titik – titik tersebut diproses menjadi bidang tak beraturan

* + - 1. Buatlah project baru pada Visual Studio dengan nama **prak1-Poligon**. Buatlah agar hasil yang ditampilkan membentuk **Bangun Bebas**. Sajikan screenshotnya. Dan jelaskan proses rendering vertexnya

#include <stdlib.h>

#include <glut.h>

void drawPoliline(){

glBegin(GL\_POLYGON);

glVertex2i(100,0);

glVertex2i(100,50);

glVertex2i(50,100);

glVertex2i(50,0);

glVertex2i(-100,0);

glVertex2i(-100,50);

glVertex2i(-50,100);

glVertex2i(-50,0);

glEnd();

glFlush();

}

void main(int argc, char \*\*argv) {

glutInit(&argc, argv);

glutInitDisplayMode (GLUT\_DEPTH | GLUT\_SINGLE | GLUT\_RGBA); // menentukan display mode awal (RGBA mode, single buffer, depth buffer)

glutInitWindowPosition(100,100); // fungsi meletakkan posisi windows

glutInitWindowSize(640,480); // fungsi membuat windows dengan ukuran sebesar 640 pixel x 480 pixel

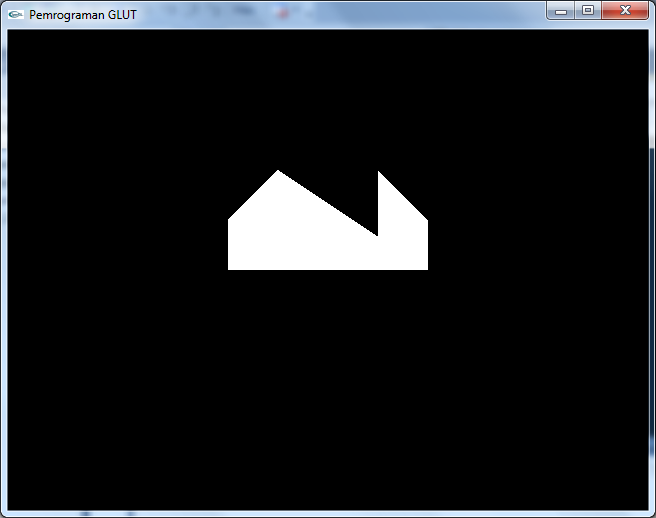
glutCreateWindow("Pemrograman GLUT"); // fungsi membuat window dengan nama Pemrograman Glut

glutDisplayFunc(drawPoliline); // melakukan rendering fungsi drawdot yang telah dibuat

gluOrtho2D(-320.,320.,-240.,240.); // delklarasi ukuran sumbu x sebesar 320 satuan ke kiri, 320 satuan ke kanan, 240 satuan ke bawah dan 240 satuan ke atas

glutMainLoop();

}



**Penjelasan :**

Pembuatan bidang dilakukan dengan pendeklarasian glBegin(GL\_POLYGON); yang dapat memproses banyak titik untuk dibuat bidang. Titik yang akan diproses adalah titik pertama yang dideklarasikan dan belanjut sampat pendeklarasian titik selanjutnya. Titik pertama diletakkan pada posisi 100 satuan ke kanan pada sumbu x (glVertex2i(100,0);), selanjutnya penentuan titik kedua pada lokasi 100 satuan ke arah sumbu x kanan dan 50 satuan ke arah atas sumbu y (glVertex2i(100,50);), selanjutnya penentuan titik ketiga pada lokasi 50 satuan ke arah sumbu x kanan dan 100 satuan ke arah sumbu y (glVertex2i(50,100);) dan selanjutnya penentuan titik keempat pada lokasi 50 satuan ke arah sumbu x kanan (glVertex2i(50,0);) Selanjutnya untuk membuat segiempat kedua penentuan titik kelima pada posisi 100 satuan ke kiri pada sumbu x (glVertex2i(-100,0);), selanjutnya penentuan titik keenam pada lokasi 100 satuan ke arah sumbu x kiri dan 50 satuan ke arah atas sumbu y (glVertex2i(-100,50);), selanjutnya penentuan titik ketujuh pada lokasi 50 satuan ke arah sumbu x kiri dan 100 satuan ke arah sumbu y (glVertex2i(-50,100);) dan selanjutnya penentuan titik kedelapan pada lokasi 50 satuan ke arah sumbu x kiri (glVertex2i(-50,0);) titik – titik tersebut diproses menjadi bidang polygon dengan pemrosesan titik dilakukan sesuai dengan urutan pendeklarasian titik sehingga terbentuk bidang polygon.

* 1. **Obyek 2D**
     + 1. Buatlah project baru pada Visual Studio. Buatlah agar tampilan obyek 2 dimensi berbentuk persegi panjang seperti di bawah ini, gunakan primitif dan struktur vertex yang paling efisien. Tulis program dan Tampilkan hasilnya berupa screenshot yang berupa tampilan 3 bangun dengan posisi, dimensi, dan warna yang berbeda.

#include <stdlib.h>

#include <glut.h>

void SegiEmpat (int posx, int posy, int w, int h) {

glBegin (GL\_QUADS);

glVertex2i (posx,posy);

glVertex2i (w+posx,posy);

glVertex2i (w+posx,posy-h);

glVertex2i (posx,posy-h);

glEnd();

}

void renderScene(void) {

glClear(GL\_COLOR\_BUFFER\_BIT);

glColor3f(0,1,0);

SegiEmpat (-150,150,50,50);

glColor3f(1,0,0);

SegiEmpat (0,150,100,100);

glColor3f(1,0,0);

SegiEmpat (0,-0,50,50);

glFlush();

}

void main(int argc, char \*\*argv) {

glutInit(&argc, argv);

glutInitDisplayMode (GLUT\_DEPTH | GLUT\_SINGLE | GLUT\_RGBA); // menentukan display mode awal (RGBA mode, single buffer, depth buffer)

glutInitWindowPosition(100,100); // fungsi meletakkan posisi windows

glutInitWindowSize(640,480); // fungsi membuat windows dengan ukuran sebesar 640 pixel x 480 pixel

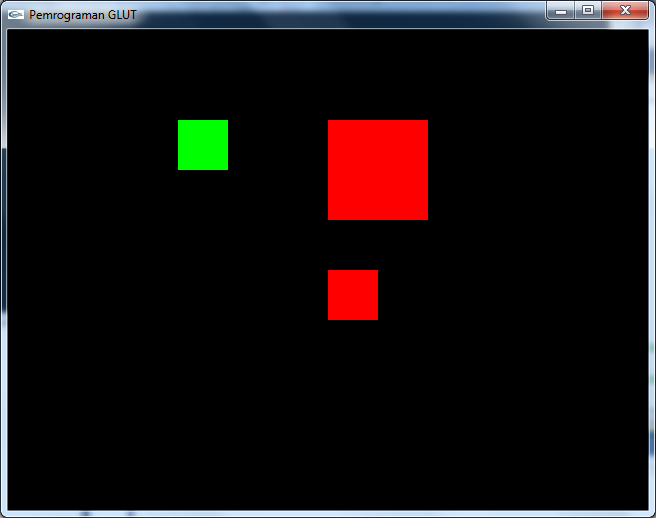
glutCreateWindow("Pemrograman GLUT"); // fungsi membuat window dengan nama Pemrograman Glut

glutDisplayFunc(renderScene); // melakukan rendering fungsi renderScene yang telah dibuat

gluOrtho2D(-320.,320.,-240.,240.); // delklarasi ukuran sumbu x sebesar 320 satuan ke kiri, 320 satuan ke kanan, 240 satuan ke bawah dan 240 satuan ke atas

glutMainLoop();

}



**Penjelasan :**

Program yang dibuat adalah bangun segiempat sebanyak 3 buah dengan ukuran,warna,dan posisi yang berbeda. Pembuatan bidang dilakukan dengan perintah gl\_quads. Fungsi quads diikuti 4 parameter bidang segiempat yaitu posx, posy , w dan h.

* + - 1. Buatlah project baru pada Visual Studio. Buatlah agar tampilan obyek 2 dimensi berbentuk Trapesium siku seperti di bawah ini gunakan primitif dan struktur vertex yang paling efisien. Tulis program dan Tampilkan hasilnya berupa screenshot yang berupa tampilan 3 bangun dengan posisi, dimensi, dan warna yang berbeda.

#include <stdlib.h>

#include <glut.h>

void trapesium (int posx,int posy,int s,int i,int h){

glBegin(GL\_TRIANGLE\_FAN);

glVertex2i(posx,posy);

glVertex2i(posx+s,posy);

glVertex2i(posx+i,posy-h);

glVertex2i(posx,posy-h);

glEnd();

}

void renderScene(void){

glClear(GL\_COLOR\_BUFFER\_BIT);

glColor3f(1,1,0);

trapesium(-140,140,75,150,150);

glColor3f(0,1,0);

trapesium(50,50,50,200,200);

glColor3f(1,0,0);

trapesium(-100,-50,80,150,150);

glFlush();

}

void main(int argc, char \*\*argv) {

glutInit(&argc, argv);

glutInitDisplayMode (GLUT\_DEPTH | GLUT\_SINGLE | GLUT\_RGBA); // menentukan display mode awal (RGBA mode, single buffer, depth buffer)

glutInitWindowPosition(100,100); // fungsi meletakkan posisi windows

glutInitWindowSize(640,480); // fungsi membuat windows dengan ukuran sebesar 640 pixel x 480 pixel

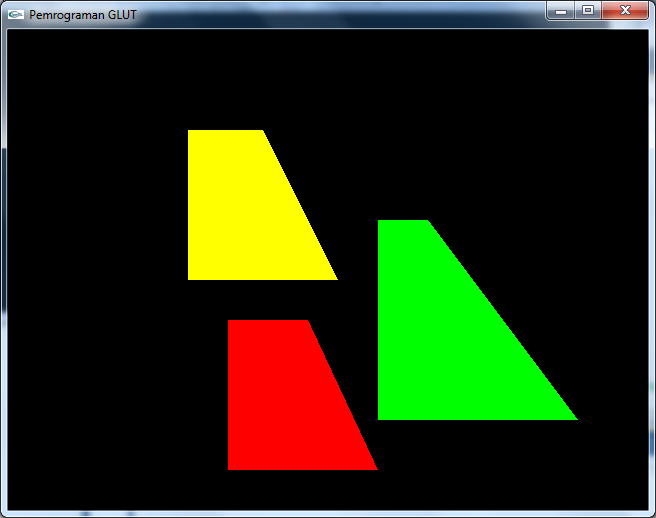
glutCreateWindow("Pemrograman GLUT"); // fungsi membuat window dengan nama Pemrograman Glut

glutDisplayFunc(renderScene); // melakukan rendering fungsi renderScene yang telah dibuat

gluOrtho2D(-320.,320.,-240.,240.); // delklarasi ukuran sumbu x sebesar 320 satuan ke kiri, 320 satuan ke kanan, 240 satuan ke bawah dan 240 satuan ke atas

glutMainLoop();

}



**Penjelasan :**

Program yang dibuat adalah bangun segiempat sebanyak 3 buah dengan ukuran,warna,dan posisi yang berbeda. Pembuatan bidang dilakukan dengan perintah gl\_quads. Fungsi quads diikuti 4 parameter bidang segiempat yaitu posx, posy , w dan h.

* + - 1. Buatlah project baru pada Visual Studio. Buatlah agar tampilan obyek 2 dimensi berbentuk Trapesium sama kaki seperti di bawah ini, gunakan primitif dan struktur vertex yang paling efisien. Tulis program dan Tampilkan hasilnya berupa screenshot yang berupa tampilan 3 bangun dengan posisi, dimensi, dan warna yang berbeda.

#include <stdlib.h>

#include <glut.h>

void trapesium (int posx,int posy,int s,int i,int h){

glBegin(GL\_TRIANGLES);

glVertex2i(posx,posy);

glVertex2i(posx+s,posy);

glVertex2i(posx+s+(i-s/2),posy-h);

glVertex2i(posx+s+(i-s/2),posy-h);

glVertex2i(posx,posy);

glVertex2i(posx-(i-s/2),posy-h);

glEnd();

}

void renderScene(void){

glClear(GL\_COLOR\_BUFFER\_BIT);

glColor3f(0,1,1);

trapesium(-100,-150,50,80,90);

glColor3f(0,1,1);

trapesium(-100,-50,70,90,90);

glColor3f(0,0,1);

trapesium(0,80,100,100,100);

glFlush();

}

void main(int argc, char \*\*argv) {

glutInit(&argc, argv);

glutInitDisplayMode (GLUT\_DEPTH | GLUT\_SINGLE | GLUT\_RGBA); // menentukan display mode awal (RGBA mode, single buffer, depth buffer)

glutInitWindowPosition(100,100); // fungsi meletakkan posisi windows

glutInitWindowSize(640,480); // fungsi membuat windows dengan ukuran sebesar 640 pixel x 480 pixel

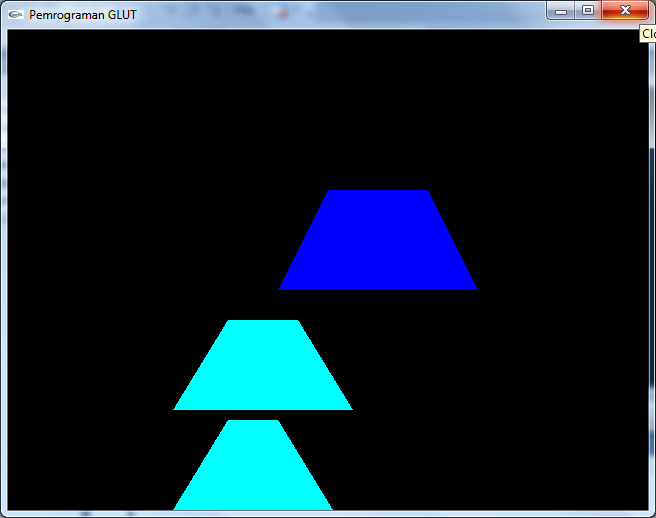
glutCreateWindow("Pemrograman GLUT"); // fungsi membuat window dengan nama Pemrograman Glut

glutDisplayFunc(renderScene); // melakukan rendering fungsi renderScene yang telah dibuat

gluOrtho2D(-320.,320.,-240.,240.); // delklarasi ukuran sumbu x sebesar 320 satuan ke kiri, 320 satuan ke kanan, 240 satuan ke bawah dan 240 satuan ke atas

glutMainLoop();

}



**Penjelasan :**

Program yang dibuat adalah bangun segiempat sebanyak 3 buah dengan ukuran,warna,dan posisi yang berbeda. Pembuatan bidang dilakukan dengan perintah gl\_quads. Fungsi quads diikuti 4 parameter bidang segiempat yaitu posx, posy , w dan h.

* + - 1. Buatlah project baru pada Visual Studio. Buatlah agar tampilan obyek 2 dimensi berbentuk jajaran genjang seperti di bawah ini, gunakan primitif dan struktur vertex **yang paling efisien**. Tulis program dan Tampilkan hasilnya berupa screenshot yang berupa tampilan 3 bangun dengan posisi, dimensi, dan warna yang berbeda.

#include <stdlib.h>

#include <glut.h>

void jajargenjang (int posx, int posy, int w, int h) {

glBegin (GL\_QUADS);

glVertex2i (posx,posy);

glVertex2i (w+posx+w,posy);

glVertex2i (w+posx,posy-h);

glVertex2i (posx-w,posy-h);

glEnd();

}

void renderScene(void) {

glClear(GL\_COLOR\_BUFFER\_BIT);

glColor3f(0,1,0);

jajargenjang(-200,200,50,50);

glColor3f(1,1,0);

jajargenjang(-100,100,-50,50);

glColor3f(0,0,1);

jajargenjang(20,20,75,50);

glFlush();

}

void main(int argc, char \*\*argv) {

glutInit(&argc, argv);

glutInitDisplayMode (GLUT\_DEPTH | GLUT\_SINGLE | GLUT\_RGBA); // menentukan display mode awal (RGBA mode, single buffer, depth buffer)

glutInitWindowPosition(100,100); // fungsi meletakkan posisi windows

glutInitWindowSize(640,480); // fungsi membuat windows dengan ukuran sebesar 640 pixel x 480 pixel

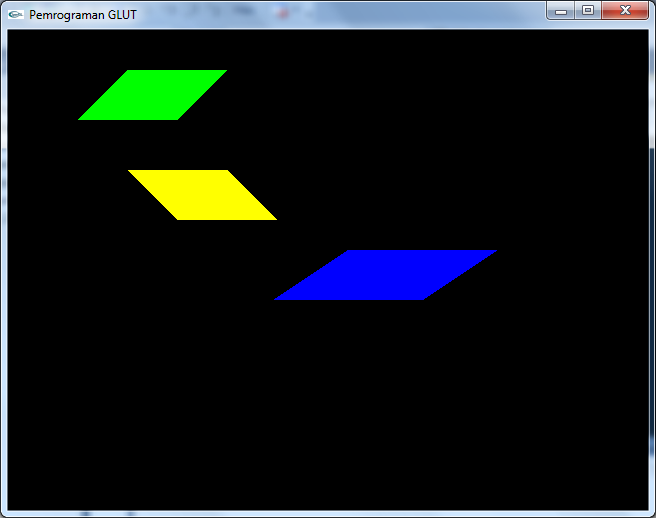
glutCreateWindow("Pemrograman GLUT"); // fungsi membuat window dengan nama Pemrograman Glut

glutDisplayFunc(renderScene); // melakukan rendering fungsi renderScene yang telah dibuat

gluOrtho2D(-320.,320.,-240.,240.); // delklarasi ukuran sumbu x sebesar 320 satuan ke kiri, 320 satuan ke kanan, 240 satuan ke bawah dan 240 satuan ke atas

glutMainLoop();

}



**Penjelasan :**

Program yang dibuat adalah bangun segiempat sebanyak 3 buah dengan ukuran,warna,dan posisi yang berbeda. Pembuatan bidang dilakukan dengan perintah gl\_quads. Fungsi quads diikuti 4 parameter bidang segiempat yaitu posx, posy , w dan h.

1. **TUGAS ASISTENSI**
   * + 1. Buatlah sebuah fungsi untuk membuat tampilan objek 2 dimensi berbentuk belah ketupat seperti di bawah ini, gunakan primitif dan struktur vertex yang paling efisien. Tulis program dan Tampilkan hasilnya berupa screenshot yang berupa tampilan 3 bangun dengan posisi, dimensi, dan warna yang berbeda.

#include <stdlib.h>

#include <glut.h>

void belahketupat (int posx,int posy,int w,int h){

glBegin(GL\_QUAD\_STRIP);

glVertex2i(posx,posy);

glVertex2i(posx+(w/2),posy-(h/2));

glVertex2i(posx-(w/2),posy-(h/2));

glVertex2i(posx,posy-h);

glEnd();

}

void renderScene(void){

glClear(GL\_COLOR\_BUFFER\_BIT);

glColor3f(1,1,0);

belahketupat (70,-70,70,70);

glColor3f(0,1,1);

belahketupat (150,-30,70,-100);

glColor3f(1,0,1);

belahketupat (-50,-50,150,-150);

glFlush();

}

void main(int argc, char \*\*argv) {

glutInit(&argc, argv);

glutInitDisplayMode (GLUT\_DEPTH | GLUT\_SINGLE | GLUT\_RGBA); // menentukan display mode awal (RGBA mode, single buffer, depth buffer)

glutInitWindowPosition(100,100); // fungsi meletakkan posisi windows

glutInitWindowSize(640,480); // fungsi membuat windows dengan ukuran sebesar 640 pixel x 480 pixel

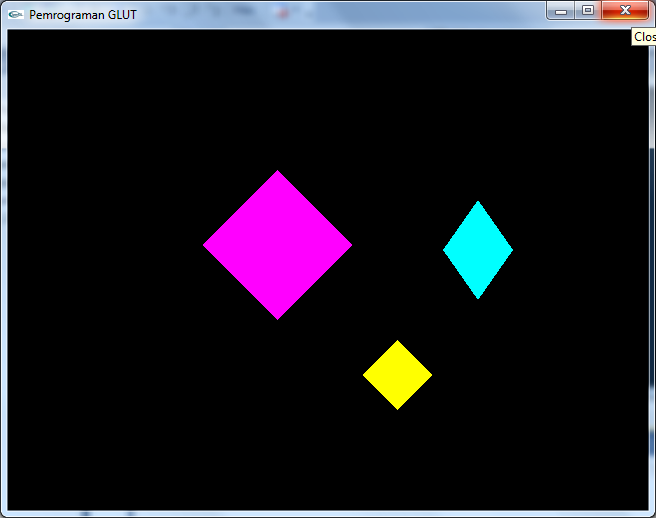
glutCreateWindow("Pemrograman GLUT"); // fungsi membuat window dengan nama Pemrograman Glut

glutDisplayFunc(renderScene); // melakukan rendering fungsi renderScene yang telah dibuat

gluOrtho2D(-320.,320.,-240.,240.); // delklarasi ukuran sumbu x sebesar 320 satuan ke kiri, 320 satuan ke kanan, 240 satuan ke bawah dan 240 satuan ke atas

glutMainLoop();

}



**Penjelasan :**

Program yang dibuat adalah bangun segiempat sebanyak 3 buah dengan ukuran,warna,dan posisi yang berbeda. Pembuatan bidang dilakukan dengan perintah gl\_quads. Fungsi quads diikuti 4 parameter bidang segiempat yaitu posx, posy , w dan h.

* + - 1. Buatlah sebuah fungsi untuk membuat tampilan objek 2 dimensi berbentuk layang –layang seperti di bawah ini, gunakan primitif dan struktur vertex yang paling efisien. Tulis program dan Tampilkan hasilnya berupa screenshot yang berupa tampilan 3 bangun dengan posisi, dimensi, dan warna yang berbeda.

#include <stdlib.h>

#include <glut.h>

void segiEmpat(int posx,int posy,int w, int h){

glBegin(GL\_QUADS);

glVertex2i(posx,posy);// posisi A

glVertex2i(posx+w,posy-h);// posisi B

glVertex2i(posx,posy-3\*h);// posisi C

glVertex2i(posx-w,posy-h);// posisi D

glEnd();

}

void renderScene(void){

glClear(GL\_COLOR\_BUFFER\_BIT);

glColor3f(0,1,1); //hijau

segiEmpat(-200,130,90,90);

glColor3f(1,1,0); //merah

segiEmpat(-30,80,90,90);

glColor3f(0,0,1); //kuning

segiEmpat(200,130,90,90);

glFlush();

}

void main(int argc, char \*\*argv) {

glutInit(&argc, argv);

glutInitDisplayMode (GLUT\_DEPTH | GLUT\_SINGLE | GLUT\_RGBA); // menentukan display mode awal (RGBA mode, single buffer, depth buffer)

glutInitWindowPosition(100,100); // fungsi meletakkan posisi windows

glutInitWindowSize(640,480); // fungsi membuat windows dengan ukuran sebesar 640 pixel x 480 pixel

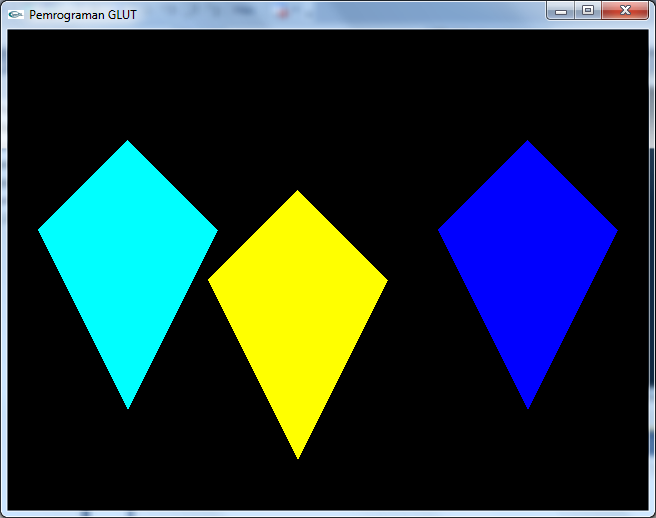
glutCreateWindow("Pemrograman GLUT"); // fungsi membuat window dengan nama Pemrograman Glut

glutDisplayFunc(renderScene); // melakukan rendering fungsi renderScene yang telah dibuat

gluOrtho2D(-320.,320.,-240.,240.); // delklarasi ukuran sumbu x sebesar 320 satuan ke kiri, 320 satuan ke kanan, 240 satuan ke bawah dan 240 satuan ke atas

glutMainLoop();

}



**Penjelasan :**

Program yang dibuat adalah bangun segiempat sebanyak 3 buah dengan ukuran,warna,dan posisi yang berbeda. Pembuatan bidang dilakukan dengan perintah gl\_quads. Fungsi quads diikuti 4 parameter bidang segiempat yaitu posx, posy , w dan h.

* + - 1. Buatlah Fungsi untuk lingkaran dengan menentukan struktur vertex dan paramater yang digunakan dalam fungsinya.

#include <stdlib.h>

#include <glut.h>

#include <math.h>

const double PI = 3.14;

int i;

//fungsi untuk membuat lingkaran

void lingkaran(int radius, int jumlah\_titik, int x\_tengah, int y\_tengah) {

glBegin(GL\_POLYGON);

//rumus untuk menempatkan vertex agar membentuk sebuah lingkaran

for (i=0;i<=360;i++){

float sudut=i\*(2\*PI/jumlah\_titik);

float x=x\_tengah+radius\*cos(sudut);

float y=y\_tengah+radius\*sin(sudut);

glVertex2f(x,y);

}

glEnd();

}

void renderScene() {

glColor3f(0, 1, 0);

lingkaran(100,90,0,0);

glFlush();

}

void main(int argc, char \*\*argv){

glutInit(&argc, argv);

glutInitWindowPosition(100,100);

glutInitWindowSize(500,500);

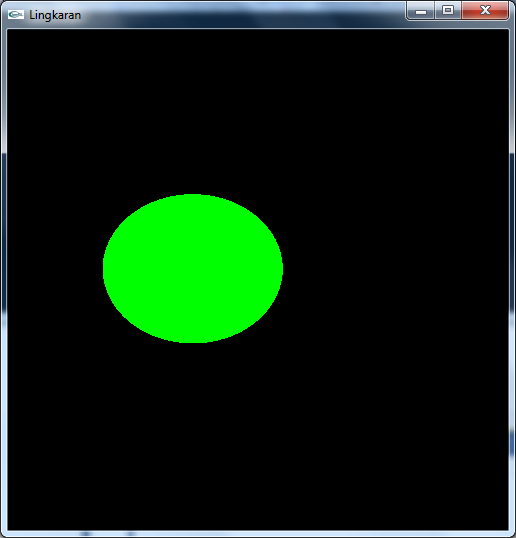
glutCreateWindow("Lingkaran");

gluOrtho2D(-205,350,-350,320);

glutDisplayFunc(renderScene);

glutMainLoop();

}



1. **KESIMPULAN**

Dalam Grafika Komputer Terdapat Objek primitif diantaranya :

**Titik** : GL\_POINTS**,**

**Garis** : GL\_LINES,

**Poliline** : GL\_LINE\_STRIP,

**Poligon (no fill)** : GL\_LINE\_LOOP,

**Triangle** : GL\_TRIANGLES,

**Quads** : GL\_QUADS,

**Polygon** : GL\_POLYGON

1. **DAFTAR PUSTAKA**

Modul 2 Praktikum Grafika Komputer Universitas Negeri Malang