TECHNICAL REPORT

PRAKTIKUM GRAFIKA KOMPUTER

MODUL 3

"KURVA"

Disusun Oleh :

TGL PRAKTIKUM	: 2 APRIL 2018
NAMA	: REISKI MUHAMMA RA
NRP	: 160411100030
KELAS	: GRAFKOM E
DOSEN PENGAMPU	: FITRIYATUL QOMARIYAH, S.Kom., M.Kom.
ASISTEN	: SOFIAN EKA SANDRA

Disetujui : ...../……./……../Bangkalan
(SOFIAN EKA SANDRA) 150411100108

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS TRUNOJOYO MADURA
2017/2018

BAB I
PENDAHULUAN

1.1 Latar Belakang

Grafika Komputer telah melesat sangat jauh dibandingkan pada masa awal kemunculannya. Cakupannya telah meluas ke berbagai bidang (kedokteran, sains, teknik, bisnis, industri, seni, hiburan, iklan) dan memiliki berbagai tools untuk pengembangannya.

Salah satu tools/library pembuatan aplikasi grafik adalah OpenGL (Open Graphics Library). OpenGL adalah suatu standar grafik (API) yang menyediakan fungsi-fungsi level rendah untuk pembuatan berbagai gambar pada komputer. Sebagai API (Application Programming Interface), OpenGL bersifat platform-independent/tidak tergantung pada piranti dan platform yang digunakan. Sehingga, adalah hal yang wajar jika aplikasi OpenGL dapat dikembangkan pada macam-macam sistem operasi, seperti Windows, UNIX, Mac, Android, dll. OpenGL pada awalnya didesain untuk digunakan oleh bahasa pemrograman C/C++, namun dalam perkembangannya OpenGL dapat juga digunakan oleh bahasa pemrograman yang lain seperti Java, Tcl, Ada, Visual Basic, Delphi, maupun Fortran.

1.2 Tujuan

Tujuan pembeajaran pada modul 3 ini adalah mempelajari bagaimana cara menciptakan Kurva 2D pada OpenGL dengan menggunakan persamaan matematika , tentunya hal mendasar yang perlu diketahui adalah berhitung

BAB II
DASAR TEORI

2.1 DASAR TEORI

Kurva dalam matematika adalah garis yang tidak harus lurus. Sebuah garis lurus adalah sebuah kurva, demikian juga sebuh garis lengkung. Contoh kurva garis lengkung: lintasan parabola, grafik sinus, grafik persamaan logaritma, dll.

Mendefinisikan Kurva Menggunakan Persamaan Polynomial

Polynomial adalah persamaan matematika dalam bentuk:

$$a_0 + a_1 x + a_2 x^2 + a_3 x^3 + \ldots + a_L x^L$$

Dimana a0, a1, a2, … aL adalah koefisien/konstanta. Derajat sebuah persamaan polynomial ditentukan dari pangkat tertinggi dari variabel x.

Kurva Polynomial derajat 1

Persamaan polynomial derajat 1 disebut juga sebagai persaman linear. Jika digambar, persamaan linear menghasilkan garis lurus. Sebagai contoh, sebuah kurva yang memiliki representasi parametrik P(t) = a0 + a1t adalah sebuah garis lurus yang melewati titik a0 pada waktu t = 0, dan melewati titik a0 + a1 pada waktu t = 1. Dalam dunia 2 dimensi, P(t) terdiri dari dua persamaan: satu persamaan untuk sumbu x: x(t), dan satu persamaan untuk sumbu y: y(t). Dalam dunia 3 dimensi P(t) memiliki pula z(t).

Program di bawah ini memplot kurva dari persamaan linear P(t) dimana: x(t) = -1 + 2t; y(t) = 0.

Program 3.1

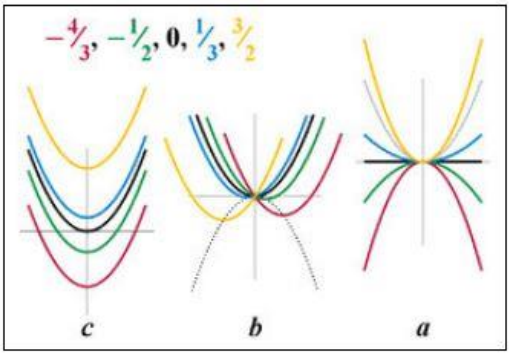
```
void display(void)
{
 /* bersihkan layar */
 glClear (GL_COLOR_BUFFER_BIT);
 glColor3f (1.0, 1.0, 0.0);
 float t = 0.0;
```

```
 glBegin(GL_POINTS);
 for(t = -1.0; t<=1.0; t+=0.01){
 /* x(t) = -1 + 2t; y(t) = 0 */
 glVertex3f (-1.0 + 2.0*t, 0.0, 0.0);
 }
glEnd();
glFlush ();
}
 void kunci(unsigned char key, int x, int y)
{
 switch (key)
 {
 /* aplikasi berhenti ketika tombol q ditekan */
 case 27 :
 case 'q':
 exit(0);
 break;
 }
 glutPostRedisplay();
}
int main(int argc, char *argv[])
{
 glutInitWindowSize(400,400);
 glutInitWindowPosition(100,100);
 glutInitDisplayMode(GLUT_RGB | GLUT_SINGLE);
 glutCreateWindow("Primitif");
 glutDisplayFunc(display);
 glutKeyboardFunc(kunci);
 glutMainLoop();
 return 0;
}
```

Kurva Polynomial derajat 2

Persamaan polynomial derajat 2 disebut juga persamaan kuadrat. Persamaan kuadrat menghasilkan grafik parabola. Bentuk umumnya adalah: y = ax2 + bx + c. Dimana a, b, dan c adalah koefisien/konstanta persamaan.

Efek dari perubahan konstanta a, b, dan c dapat dilihat pada gambar di bawah ini.



Program di bawah ini menggambar x(t) = t; y(t) = t2 – 0.5 atau y = x2 – 0.5 pada interval -1.0 sampai 1.0.

Program 3.2

```
void display(void)
{
 /* bersihkan layar */
 glClear (GL_COLOR_BUFFER_BIT);
 glColor3f (1.0, 1.0, 0.0);
 float t = 0.0;
 glBegin(GL_POINTS);
 for(t = -1.0; t<=1.0; t+=0.01){
 /* x(t) = -1 + 2t; y(t) = 0 */
 glVertex3f (t, -0.5+t*t, 0.0);
 }
 glEnd();
 glFlush ();
}
 void kunci(unsigned char key, int x, int y)
{
 switch (key)
 {
 /* aplikasi berhenti ketika tombol q ditekan */
 case 27 :
 case 'q':
 exit(0);
 break;
```

```
 }
 glutPostRedisplay();
}
int main(int argc, char *argv[])
{
 glutInitWindowSize(400,400);
 glutInitWindowPosition(100,100);
 glutInitDisplayMode(GLUT_RGB | GLUT_SINGLE);
 glutCreateWindow("Primitif");
 glutDisplayFunc(display);
 glutKeyboardFunc(kunci);
 glutMainLoop();
 return 0;
}
```

Kurva Polynomial derajat 3 atau lebih

Persamaan polynomial derajat 3 atau lebih memiliki sifat dan implementasi yang mirip seperti persamaan polynomial derajat2, hanya saja grafiknya lebih kompleks.

Program di bawah ini menggambar persamaan $y = (x+4)(x+1)(x-1)(x-3)/14 + 0.5$

Program 3.3

```
void myinit()
{
 glClearColor(0.0, 0.0, 0.0, 1.0);
 glColor3f(1.0, 0.0, 0.0);
 glMatrixMode(GL_PROJECTION);
 glLoadIdentity();
 gluOrtho2D(-10.0, 10.0, -10.0, 10.0);
 glMatrixMode(GL_MODELVIEW);
}
void display(void)
```

```
{
 /* bersihkan layar */
 glClear (GL_COLOR_BUFFER_BIT);
 glColor3f (1.0, 1.0, 0.0);
 float t = 0.0;
 //f(x) = 1/14 (x+4)(x+1)(x-1)(x-3) + 0.5
 glBegin(GL_POINTS);
 for(t = -10.0; t<=10.0; t+=0.1){
 glVertex3f (t, (t+4)*(t+1)*(t-1)*(t-3)/14 + 0.5, 0.0);
 }
 glEnd();
 glBegin(GL_LINES);
 glVertex3f(-10.0,0.0,0.0);
 glVertex3f(10.0,0.0,0.0);
 glVertex3f(0.0,-10.0,0.0);
 glVertex3f(0.0,10.0,0.0);
 glEnd();
 glFlush ();
}
 void kunci(unsigned char key, int x, int y)
{
 switch (key)
 {
 /* aplikasi berhenti ketika tombol q ditekan */
 case 27 :
 case 'q':
 exit(0);
 break;
 }
 glutPostRedisplay();
}
int main(int argc, char *argv[])
{
 glutInitWindowSize(400,400);
 glutInitWindowPosition(100,100);
 glutInitDisplayMode(GLUT_RGB | GLUT_SINGLE);
 glutCreateWindow("Primitif");
 glutDisplayFunc(display);
 glutKeyboardFunc(kunci);
```

```
 myinit();
 glutMainLoop();
 return 0;
}
```

Kurva Trigonometri

Kurva trigonometri adalah kurva yang dihasilkan dari fungsi-fungsi trigonometri: sinus, cosinus, dan tangen.

Program di bawah ini menggambar kurva berbentuk grafik fungsi sinus.

Program 3.4

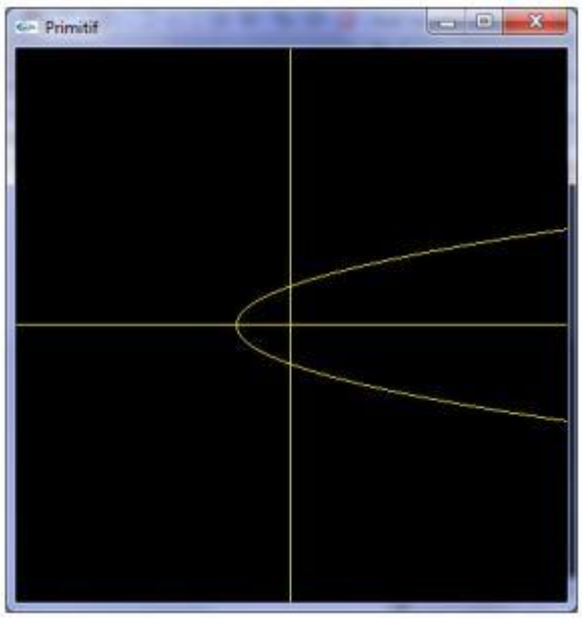
```
//Supaya bisa menggunakan fungsi sin(), program perlu
include Math.h
#include <Math.h>
void myinit()
{
 glClearColor(0.0, 0.0, 0.0, 1.0);
 glColor3f(1.0, 0.0, 0.0);
 glMatrixMode(GL_PROJECTION);
 glLoadIdentity();
gluOrtho2D(-1.0, 10.0, -2.0, 2.0);
 glMatrixMode(GL_MODELVIEW);
}
void display(void)
{
 /* bersihkan layar */
 glClear (GL_COLOR_BUFFER_BIT);
 glColor3f (1.0, 1.0, 0.0);
 float x = 0.0;
 glBegin(GL_POINTS);
 //perhitungan sudut di openGL menggunakan radian, bukan
derajat
 for(x=0.0; x<=6.28; x+=0.1)
 {
 glVertex2f(x,sin(x));
```

```
    }
    glEnd();
    glBegin(GL_LINES);
    glVertex3f(-10.0,0.0,0.0);
    glVertex3f(10.0,0.0,0.0);
    glVertex3f(0.0,-10.0,0.0);
    glVertex3f(0.0,10.0,0.0);
    glEnd();
    glFlush ();
   }
    void kunci(unsigned char key, int x, int y)
   {
    switch (key)
    {
    /* aplikasi berhenti ketika tombol q ditekan */
    case 27 :
    case 'q':
    exit(0);
    break;
    }
    glutPostRedisplay();
   }
   int main(int argc, char *argv[])
   {
    glutInitWindowSize(400,400);
    glutInitWindowPosition(100,100);
    glutInitDisplayMode(GLUT_RGB | GLUT_SINGLE);
    glutCreateWindow("Primitif");
    glutDisplayFunc(display);
    glutKeyboardFunc(kunci);
    myinit();
    glutMainLoop();
    return 0;
} □
```

BAB III
TUGAS DAN IMPLEMENTASI

3.1 TUGAS

1. Ubah persamaan linear pada program 3.1 menjadi x(t): 0.5 - t; y(t) = 1 + 2t.
2. Ubah program 3.2 supaya bisa menampilkan plot seperti berikut:



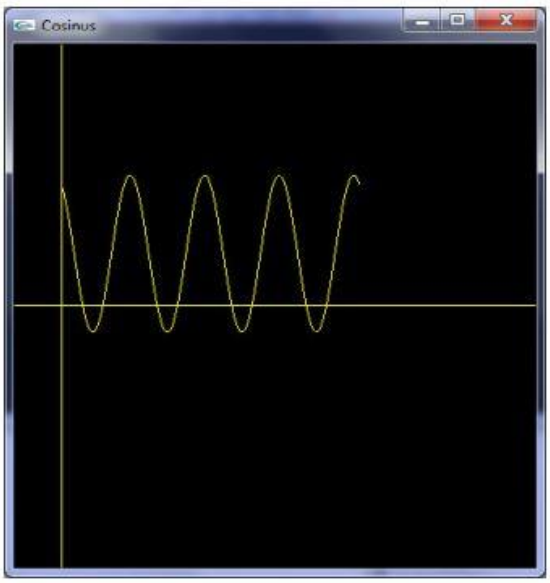
Gambar yang dihasilkan tidak harus persis sama, tetapi harus dibuat semirip mungkin. Jangan gunakan teknik transformasi (modul 4), gunakan pendekatan persamaan matematis untuk menghasilkan gambar tersebut.

3. Modifikasi program 3.3 untuk menampilkan fungsi berikut: f(x) = (x-3)(x-3)(x1)(x)(x+2)(x+2)(x+3) / 14. Sesuaikan ukuran proyeksi supaya kurva dapat terlihat jelas di dalam jendela program.
4. Fungsi Cosinus memiliki bentuk baku sebagai berikut:
   y = A Cos(Bx + C) + D
   dimana:
   - A menentukan tinggi rendahnya grafik yang dihasilkan pada sumbu y
   - B menentukan berapa kali perulangan grafik dalam satu interval
   - C menentukan pergeseran sudut inputan sinus

- D menentukan pegeseran grafik sinus pada sumbu y.

Modifikasi program 3.4 supaya bisa mengakomodasi bentuk baku ini. Hint: buat variabel untuk A, B, C, dan D. Program tidak perlu mempunyai fasilitas menerima inputan ketika dijalankan.

Sebagai contoh, berikut ini adalah gambar grafik cosinus dengan A = 3, B = 4, C = 0.5, D = 2.



3.2 PENYELESAIAN

1. Program 1

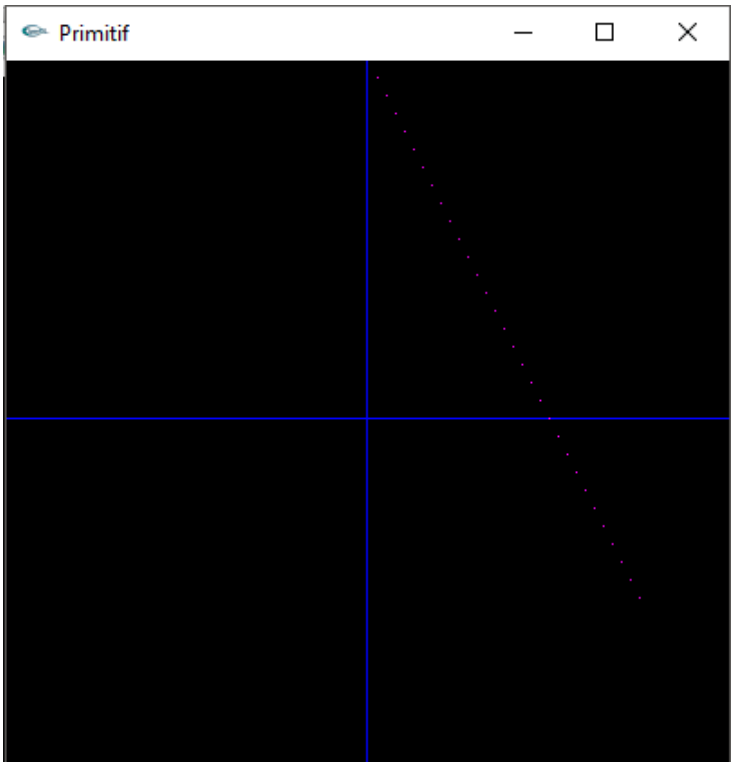
```
#include <windows.h>
#ifdef __APPLE__
#include <GLUT/glut.h>
#else
#include <GL/glut.h>
#endif
//Supaya bisa menggunakan fungsi sin(), program perlu
include Math.h
#include <Math.h>
void myinit()
{
    glClearColor(0.0, 0.0, 0.0, 1.0);
    glColor3f(0.0, 0.0, 1.0);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluOrtho2D(-2, 2, -2, 2);
    glMatrixMode(GL_MODELVIEW);
}
void display(void)
{
```

```
    /* bersihkan layar */
    glClear (GL_COLOR_BUFFER_BIT);
    glBegin(GL_LINES);
    glVertex3f(-10.0,0.0,0.0);
    glVertex3f(10.0,0.0,0.0);
    glVertex3f(0.0,-10.0,0.0);
    glVertex3f(0.0,10.0,0.0);
    glEnd();

    float t = 0.0;
    glBegin(GL_POINTS);
    glColor3f(1.0,0.0,1.0);
    for(t = -1.0; t<=1.0; t+=0.05)
    {
        /* x(t) = 0.5 - t; y(t) = 1 + 2t */
        glVertex3f (0.5 + -1*t, 1 + 2*t,0.0);
    }
    glEnd();

    glFlush ();
}
void kunci(unsigned char key, int x, int y)
{
    switch (key)
    {
    /* aplikasi berhenti ketika tombol q ditekan */
    case 27 :
    case 'q':
        exit(0);
        break;
    }
    glutPostRedisplay();
}
int main(int argc, char *argv[])
{
    glutInitWindowSize(400,400);
    glutInitWindowPosition(100,100);
    gluOrtho2D(-10,10,-10,10);
    glutInitDisplayMode(GLUT_RGB | GLUT_SINGLE);
    glutCreateWindow("Primitif");
    glutDisplayFunc(display);
    myinit();
    glutKeyboardFunc(kunci);
    glutMainLoop();
    return 0;
}
```

Hasil running :



2. Program 2

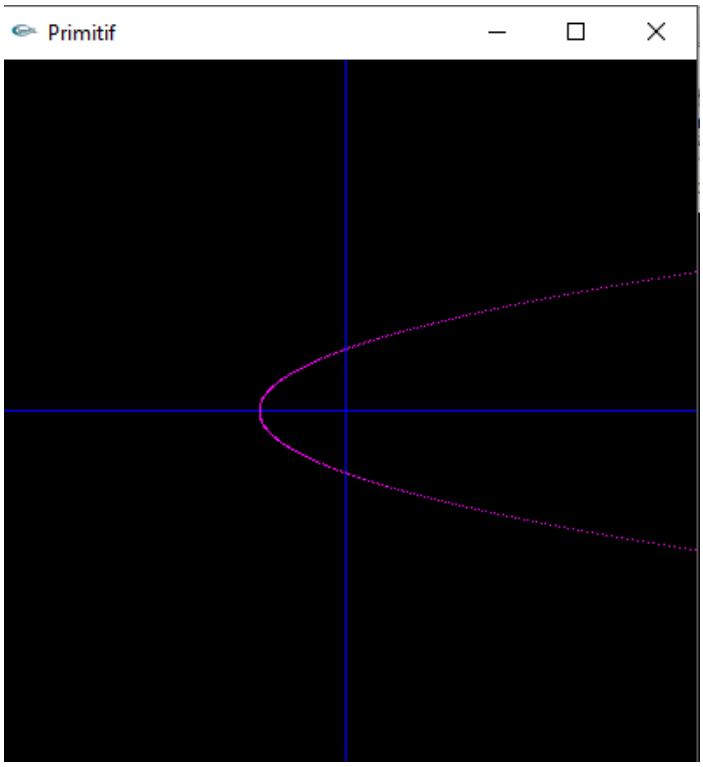
```
#include <windows.h>
#ifdef __APPLE__
#include <GLUT/glut.h>
#else
#include <GL/glut.h>
#endif
//Supaya bisa menggunakan fungsi sin(), program perlu
include Math.h
#include <Math.h>
void myinit()
{
    glClearColor(0.0, 0.0, 0.0, 1.0);
    glColor3f(0.0, 0.0, 1.0);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluOrtho2D(-2, 2, -2, 2);
    glMatrixMode(GL_MODELVIEW);
}
void display(void)
{
    /* bersihkan layar */
    glClear (GL_COLOR_BUFFER_BIT);
    glBegin(GL_LINES);
    glVertex3f(-10.0,0.0,0.0);
    glVertex3f(10.0,0.0,0.0);
    glVertex3f(0.0,-10.0,0.0);
    glVertex3f(0.0,10.0,0.0);
```

```
    glEnd();

    float t = 0.0;
    glBegin(GL_POINTS);
    glColor3f(1.0,0.0,1.0);
    for(t = -5.0; t<=5.0; t+=0.01)
    {
        /* x(t) = 0.5 - t; y(t) = 1 + 2t */
        glVertex3f (-0.5+t*t,-0.5*t, 0.0);
    }
    glEnd();

    glFlush ();
}
void kunci(unsigned char key, int x, int y)
{
    switch (key)
    {
    /* aplikasi berhenti ketika tombol q ditekan */
    case 27 :
    case 'q':
        exit(0);
        break;
    }
    glutPostRedisplay();
}
int main(int argc, char *argv[])
{
    glutInitWindowSize(400,400);
    glutInitWindowPosition(100,100);
    gluOrtho2D(-10,10,-10,10);
    glutInitDisplayMode(GLUT_RGB | GLUT_SINGLE);
    glutCreateWindow("Primitif");
    glutDisplayFunc(display);
    myinit();
    glutKeyboardFunc(kunci);
    glutMainLoop();
    return 0;
}
```

Hasil Running:



3. Program 3

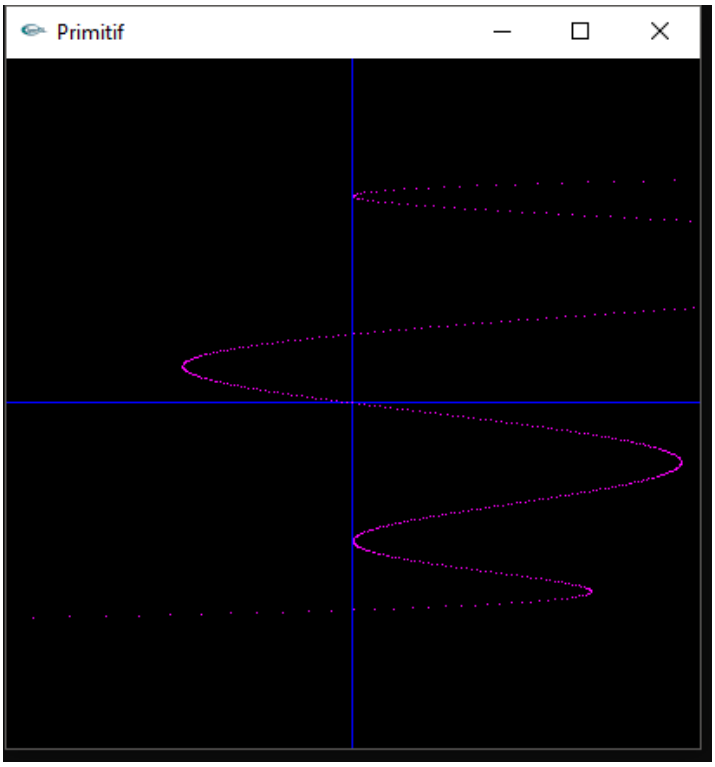
```
#include <windows.h>
#ifdef __APPLE__
#include <GLUT/glut.h>
#else
#include <GL/glut.h>
#endif
//Supaya bisa menggunakan fungsi sin(), program perlu
include Math.h
#include <Math.h>
void myinit()
{
    glClearColor(0.0, 0.0, 0.0, 1.0);
    glColor3f(0.0, 0.0, 1.0);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluOrtho2D(-5, 5, -5, 5);
    glMatrixMode(GL_MODELVIEW);
}
void display(void)
{
    /* bersihkan layar */
    glClear (GL_COLOR_BUFFER_BIT);
    glBegin(GL_LINES);
    glVertex3f(-10.0,0.0,0.0);
    glVertex3f(10.0,0.0,0.0);
```

```
    glVertex3f(0.0,-10.0,0.0);
    glVertex3f(0.0,10.0,0.0);
    glEnd();

    float t = 0.0;
    glBegin(GL_POINTS);
    glColor3f(1.0,0.0,1.0);
    for(t = -10.0; t<=10.0; t+=0.01)
    {
        /* x(t) = 0.5 - t; y(t) = 1 + 2t */
        glVertex3f ((t-3)*(t-3)*(t-
1)*(t)*(t+2)*(t+2)*(t+3)/14,t, 0.0);
    }
    glEnd();

    glFlush ();
}
void kunci(unsigned char key, int x, int y)
{
    switch (key)
    {
    /* aplikasi berhenti ketika tombol q ditekan */
    case 27 :
    case 'q':
        exit(0);
        break;
    }
    glutPostRedisplay();
}
int main(int argc, char *argv[])
{
    glutInitWindowSize(400,400);
    glutInitWindowPosition(100,100);
    gluOrtho2D(-10,10,-10,10);
    glutInitDisplayMode(GLUT_RGB | GLUT_SINGLE);
    glutCreateWindow("Primitif");
    glutDisplayFunc(display);
    myinit();
    glutKeyboardFunc(kunci);
    glutMainLoop();
    return 0;
}
```

Hasil Running :



4. Program 4

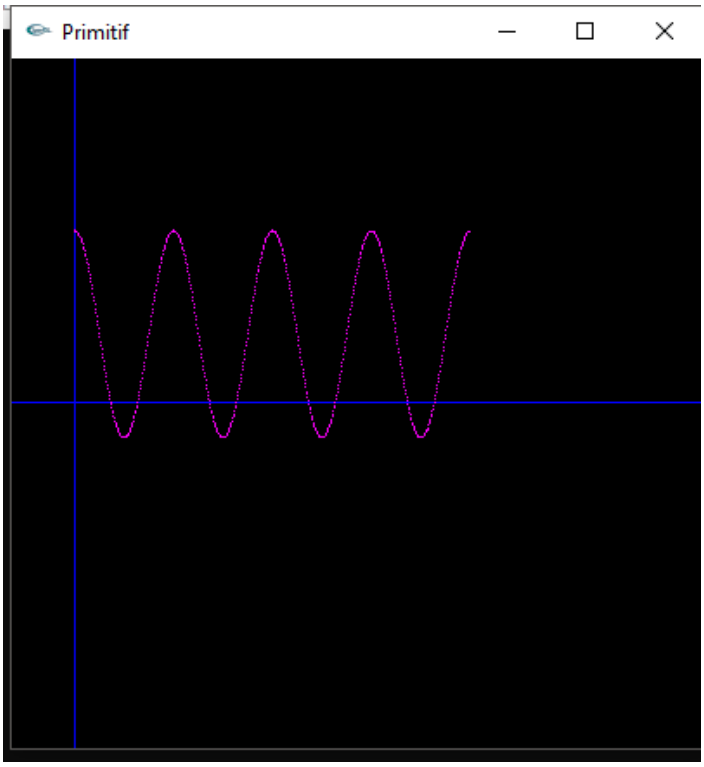
```
#include <windows.h>
#ifdef __APPLE__
#include <GLUT/glut.h>
#else
#include <GL/glut.h>
#endif
//Supaya bisa menggunakan fungsi sin(), program perlu
include Math.h
#include <Math.h>
void myinit()
{
    glClearColor(0.0, 0.0, 0.0, 1.0);
    glColor3f(0.0, 0.0, 1.0);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluOrtho2D(-1, 10, -10, 10);
    glMatrixMode(GL_MODELVIEW);
}
void display(void)
{
    /* bersihkan layar */
    glClear (GL_COLOR_BUFFER_BIT);
    glBegin(GL_LINES);
    glVertex3f(-10.0,0.0,0.0);
    glVertex3f(10.0,0.0,0.0);
    glVertex3f(0.0,-10.0,0.0);
    glVertex3f(0.0,10.0,0.0);
```

```
    glEnd();

    float t = 0.0;
    glBegin(GL_POINTS);
    glColor3f(1.0,0.0,1.0);
    for(t = 0.0; t<=6.28; t+=0.01)
    {
        /* x(t) = 0.5 - t; y(t) = 1 + 2t */
        int A=3,B=4,C=0.5,D=2;
        glVertex2f (t,A*cos(B*t+C)+D);
    }
    glEnd();

    glFlush ();
}
void kunci(unsigned char key, int x, int y)
{
    switch (key)
    {
    /* aplikasi berhenti ketika tombol q ditekan */
    case 27 :
    case 'q':
        exit(0);
        break;
    }
    glutPostRedisplay();
}
int main(int argc, char *argv[])
{
    glutInitWindowSize(400,400);
    glutInitWindowPosition(100,100);
    glutInitDisplayMode(GLUT_RGB | GLUT_SINGLE);
    glutCreateWindow("Primitif");
    glutDisplayFunc(display);
    myinit();
    glutKeyboardFunc(kunci);
    glutMainLoop();
    return 0;
}
```

Hasil running :



3.3 PENJELASAN PROGRAM

Program-program di atas menghasilkan output berupa garis-garis kurva yang dibentuk menggunakan fungsi-fungsi matematika yang telah dijelaskan pada dasar teori yang telah tersedia di atas, mulai dari kurva polynomial derajat 1, derajat 2, derajat 3 atau lebih, hingga trigonometri.

BAB IV
PENUTUP

4.1 KESIMPULAN

Pada OpenGL kita dapat membuat beberapa bentuk bangun datar 2D dengan menentukan titik-titik koordinat nya terlebih dahulu dan juga menggunakan fungsi-fungsi matematika. Dalam membuat bentuk-bentuk menggunakan perhitungan matematika tentunya selain kemampuan program juga diperlukan intelektual matematika.

4.2 SARAN

Untuk memudahkan dalam membuat bentuk yang diinginkan, hal yang perlu diperkuat adalah kemampuan matematika khususnya pada bab persamaan supaya memudahkan dalam membuat bentuk kurva yang diinginkan.