

**TECHNICAL REPORT**  
**GRAFIKA KOMPUTER**  
**MODUL 2**



**Disusun Oleh :**

TGL. PRAKTIKUM	: Jum'at, 9 April 2021
NAMA	: Achmad Farid Alfa Waid
NRP	: 190411100073
KELOMPOK	: 2
DOSEN	: Ach. Khozaimi, S.Kom., M.Kom

**TELAH DISETUJUI TANGGAL :**

.....  
**ASISTEN PRAKTIKUM**

Rizky Alfriansyah  
(180411100068)

**JURUSAN TEKNIK INFORMATIKA**  
**FAKULTAS TEKNIK**  
**UNIVERSITAS TRUNOJOYO MADURA**

# **BAB I**

## **TUJUAN DAN DASAR TEORI**

### **A. TUJUAN**

Mahasiswa dapat mengetahui dan memahami pembuatan kurva pada openGL.

### **B. DASAR TEORI**

Kurva dalam matematika adalah garis yang tidak harus lurus. Sebuah garis lurus adalah sebuah kurva, demikian juga sebuah 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:

$$a_0 + a_1x + a_2x^2 + a_3x^3 + \dots + a_Lx^L$$

Dimana  $a_0, a_1, a_2, \dots, a_L$  adalah koefisien/konstanta.

Derajat sebuah persamaan polynomial ditentukan dari pangkat tertinggi dari variabel  $x$ .

## BAB II

### PEMBAHASAN

#### A. SOAL

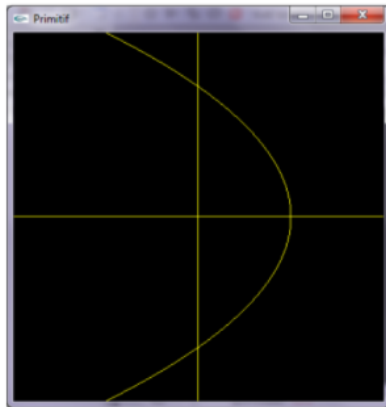
##### 1. Percobaan

Berikut adalah yang harus dilakukan selama sesi lab:

- 1) Copy contoh-contoh program diatas, jalankan, dan amati output yang ditampilkan.
- 2) Ubah program 2.1, 2.2, 2.3, dan 2.4 dari GL\_POINTS untuk menggambar kurva menjadi GL\_LINES, GL\_LINE\_STRIP, dan GL\_LINE\_LOOP! Amati perubahan tampilan yang terjadi. Mana yang menurutmu sebaiknya digunakan dan sebaiknya dihindari dalam menggambar kurva?
- 3) Modifikasi program 2.4 dari fungsi sinus menjadi fungsi tangen. Selain menggunakan fungsi tangen di header file Math.h, coba gunakan juga formula  $\text{tg}() = \sin()/\cos()$ .

##### 2. Tugas

- 1) Ubah persamaan linear pada program 2.1 menjadi  $x(t): -0.8 + 1.6t$ ;  $y(t) = -1 + 2t$
- 2) Ubah program 2.2 supaya bisa menampilkan plot seperti berikut:



Gambar yang dihasilkan tidak harus persis sama, tetapi harus dibuat semirip mungkin.

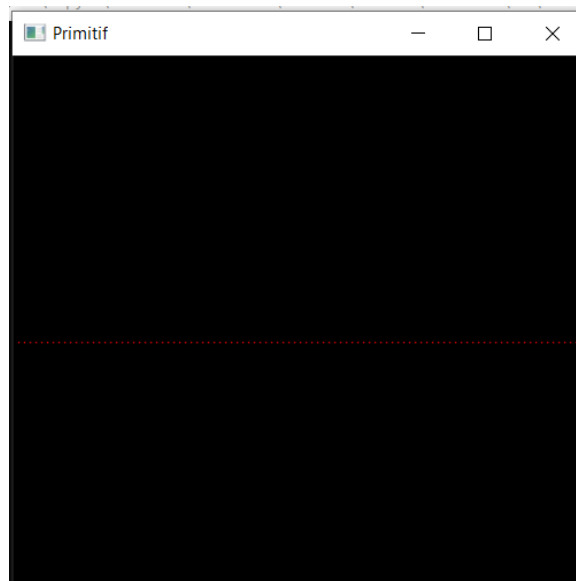
- 3) Modifikasi program 2.3 untuk menampilkan fungsi berikut:  $f(x) = (x-3)(x-2)(x-1)(x)(x+1)(x+2)(x+3)$ . Sesuaikan viewport supaya kurva dapat terlihat jelas di dalam jendela program.
- 4) Fungsi sinus memiliki bentuk baku sebagai berikut:  $y = A \sin(Bx + C) + D$  dimana: - A menentukan tinggi rendahnya 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 2.4 supaya bisa mengakomodasi bentuk baku ini. Hint: buat variabel untuk A, B, C, dan D. Program tidak perlu mempunyai fasilitas menerima inputan ketika dijalankan.

## B. JAWABAN

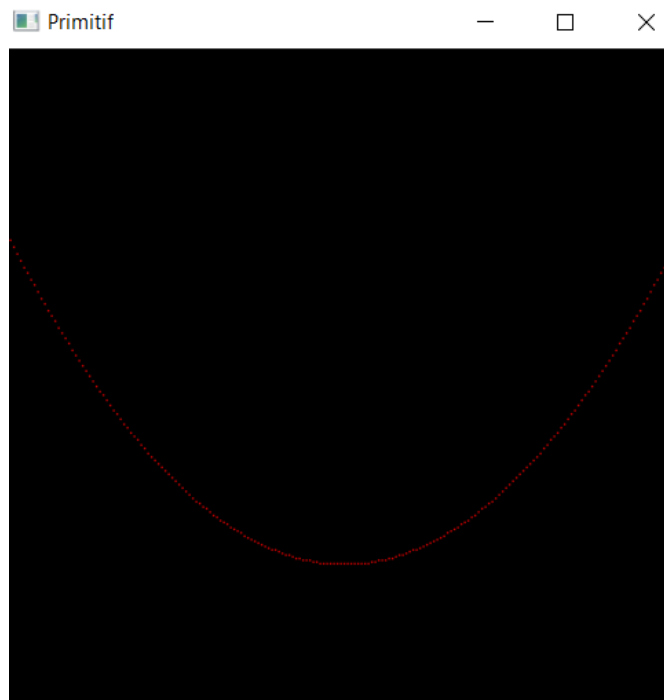
### 1. Percobaan

- 1) Hasil percobaan dan pengamatan

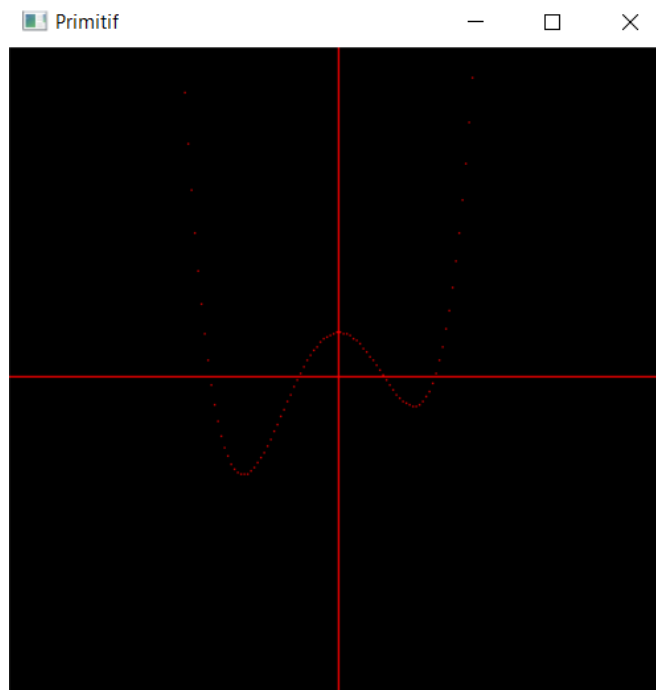
- a. Code 2.1 menghasilkan kurva berupa garis menggunakan titik



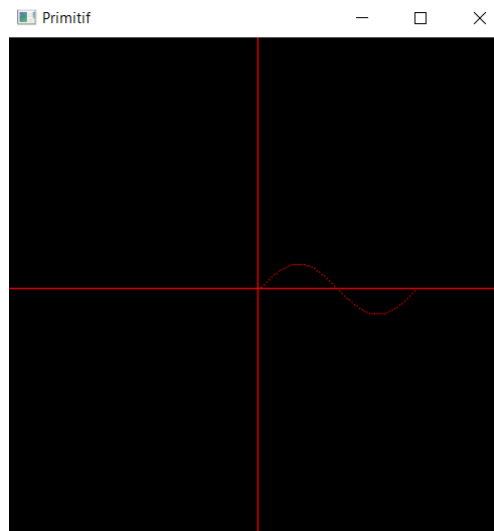
- b. Code 2.2 menghasilkan kurva melengkung



c. Code 2.3 menghasilkan kurva seperti berikut

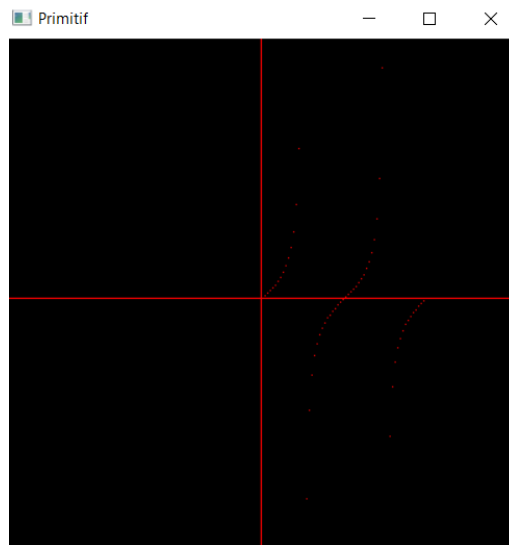


d. Code 2.4 menghasilkan kurva sinus



2) Dari semua percobaan dan perubah, `GL_LINE_STRIP` menjadi pilihan terbaik untuk kurva normal akan tetapi pada saat penggambaran kurva tangen, bentuknya menjadi aneh, jadi tidak terlalu direkomendasikan untuk kurva tangen. kemudian pada saat menggunakan `GL_LINE_LOOP`, koordinat awal dan koordinat akan otomatis terhubung, sedangkan untuk `GL_LINES` akan membuat kurva menjadi putus – putus. Jadi pilihan terbaik dari yang terbaik adalah tetap menggunakan `GL_POINTS` dengan ketentuan merapatkan jarak antar point sehingga tampilan menjadi bagus.

3) Hasilnya seperti berikut



## 2. Code Program

### 1) Nomor 1

```
#ifdef __APPLE__
#include <GLUT/glut.h>
#else
#include <GL/glut.h>
#endif
#include <stdlib.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(-10.0, 10.0, -10.0, 10.0);
    glMatrixMode(GL_MODELVIEW);
}
void display(void)
{
    /* bersihkan layar */
    glClear (GL_COLOR_BUFFER_BIT);
    glColor3f (1.0, 0.0, 0.0);
    float t = 0.0;

    //no 1
    glBegin(GL_LINE_STRIP);
    for(t = -10.0; t<=10.0; t+=0.01)
    {
        //x(t)=-0.8+1.6*t y(t)=-1+2*t
        glVertex3f (-0.8+1.6*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);
glutInit(&argc,argv);
glutInitWindowPosition(100,100);
glutInitDisplayMode(GLUT_RGB | GLUT_SINGLE);
glutCreateWindow("No 1");
glutDisplayFunc(display);
glutKeyboardFunc(kunci); myinit();
glutMainLoop();
return 0;}

```

## 2) Nomor 2

```

#ifdef __APPLE__
#include <GLUT/glut.h>
#else
#include <GL/glut.h>
#endif
#include <stdlib.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, 1.0, -1.0, 1.0);
glMatrixMode(GL_MODELVIEW);
}
void display(void)
{
/* bersihkan layar */
glClear (GL_COLOR_BUFFER_BIT);

```



```

glColor3f (1.0, 0.0, 0.0);
float t = 0.0;

//no 2
glBegin(GL_LINE_LOOP);
for(t = -10.0; t<=10.0; t+=0.01)
{
    glVertex3f (0.5-t*t, t, 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);
    glutInit(&argc,argv);
    glutInitWindowPosition(100,100);
    glutInitDisplayMode(GLUT_RGB | GLUT_SINGLE);
    glutCreateWindow("No 2");
    glutDisplayFunc(display);
    glutKeyboardFunc(kunci); myinit();
    glutMainLoop();
}

```

```
    return 0;}
```

### 3) Nomor 3

```
#ifdef __APPLE__
#include <GLUT/glut.h>
#else
#include <GL/glut.h>
#endif
#include <stdlib.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(-10.0, 10.0, -40.0, 40.0);
    glMatrixMode(GL_MODELVIEW);
}
void display(void)
{
    /* bersihkan layar */
    glClear (GL_COLOR_BUFFER_BIT);
    glColor3f (1.0, 0.0, 0.0);
    float t = 0.0;

    //no 3
    //f(x) = 1/14 (x+4) (x+1) (x-1) (x-3) + 0.5
    glBegin(GL_LINE_STRIP);
    for(t = -10.0; t<=10.0; t+=0.01)
    {
        glVertex3f(t, (t-3)*(t-2)*(t-1)*(t)*(t+1)*(t+2)*(t+3),
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);
    glutInit(&argc,argv);
    glutInitWindowPosition(100,100);
    glutInitDisplayMode(GLUT_RGB | GLUT_SINGLE);
    glutCreateWindow("No 3");
    glutDisplayFunc(display);
    glutKeyboardFunc(kunci); myinit();
    glutMainLoop();
    return 0;}

```

#### 4) Nomor 4

```

#ifdef __APPLE__
#include <GLUT/glut.h>
#else
#include <GL/glut.h>
#endif
#include <stdlib.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(-10.0, 10.0, -10.0, 10.0);
}

```

```

    glMatrixMode(GL_MODELVIEW);
}
void display(void)
{
    /* bersihkan layar */
    glClear (GL_COLOR_BUFFER_BIT);
    glColor3f (1.0, 0.0, 0.0);
    float x = 0.0;

    //no 4
    glBegin(GL_POINTS);
    //variabel baru
    float A=5;float B=2;float C=3;float D=1.2;
    for(x=0.0; x<=10; x+=0.001)
    {
        glVertex2f(x,A*sin(B*x + C) + D);
    }
    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);
    glutInit(&argc,argv);
    glutInitWindowPosition(100,100);
    glutInitDisplayMode(GLUT_RGB | GLUT_SINGLE);
    glutCreateWindow("No 4");
    glutDisplayFunc(display);
    glutKeyboardFunc(kunci); myinit();
    glutMainLoop();
    return 0;}

```

### 3. Penjelasan Code Program

#### 1) Nomor 1

Pada soal nomor 1, mahasiswa mengubah persamaan 2.1 seperti berikut:

```
"glVertex3f (-1.0 + 2.0*t, 0.0, 0.0);"
```

Menjadi:

```
"glVertex3f (-0.8+1.6*t, 1+2*t, 0.0);"
```

Nilai koordinat x dan y diubah dan disesuaikan dengan persamaan yang baru. Persamaan tersebut didapatkan dari soal, sehingga akan menghasilkan grafik kura yang baru.

#### 2) Nomor 2

Pada soal nomor 2, yaitu memodifikasi program pada code 2.2 agar sesuai dengan grafik kurva pada soal, berikut merupakan code kurva awal:

```
"glVertex3f (t, -0.5+t*t, 0.0);"
```

Kemudian akan dimodifikasi seperti berikut:

```
"glVertex3f (0.5-t*t, t, 0.0);"
```

Yaitu dengan cara menukar posisi koordinat pada kurva sebelumnya, koordinat x menjadi y, dan sebaliknya. Kemudian pada persamaannya merubah t kuadrat menjadi negatif agar kurva menghadap ke kiri dan sesuai dengan soal.

### 3) Nomor 3

Pada soal nomor 3, mahasiswa mengubah persamaan 2.3 seperti berikut:

```
"glVertex3f (t, (t+4)*(t+1)*(t-1)*(t-3)/14 + 0.5, 0.0);"
```

Menjadi:

```
"glVertex3f(t, (t-3)*(t-2)*(t-1)*(t)*(t+1)*(t+2)*(t+3), 0.0);"
```

Nilai koordinat x dan y diubah dan disesuaikan dengan persamaan yang baru. Persamaan tersebut didapatkan dari soal, sehingga akan menghasilkan grafik kurva yang baru. Selanjutnya adalah merubah code gluOrtho2D agar grafik kurva tersebut terlihat dengan jelas, gluOrtho2D yang digunakan adalah seperti berikut:

```
"gluOrtho2D(-10.0, 10.0, -40.0, 40.0);"
```

### 4) Nomor 4

praktikan membuat 4 variabel baru dengan tipe data float seperti berikut:

```
"float A=5;float B=2;float C=3;float D=1.2;"
```

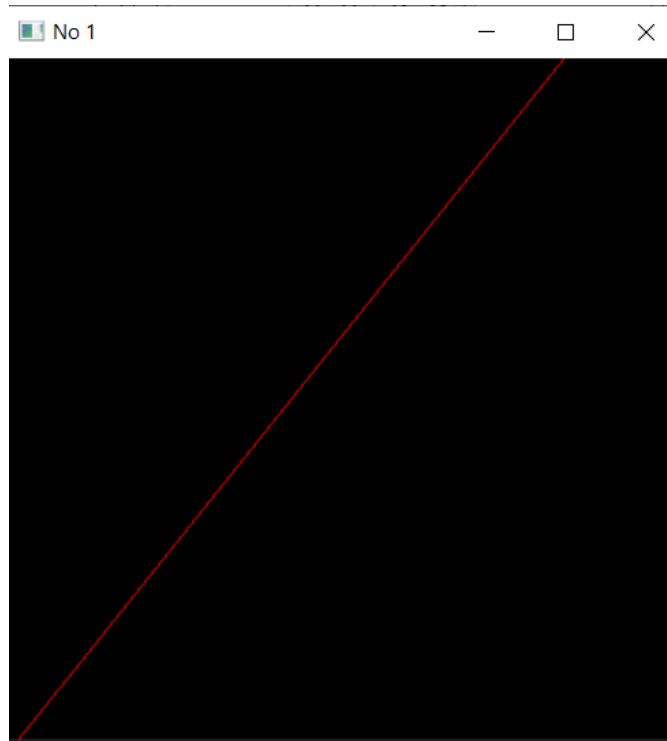
Kemudian merubah persamaan sesuai dengan instruksi di soal seperti berikut:

```
"glVertex2f(x,A*sin(B*x + C) + D);"
```

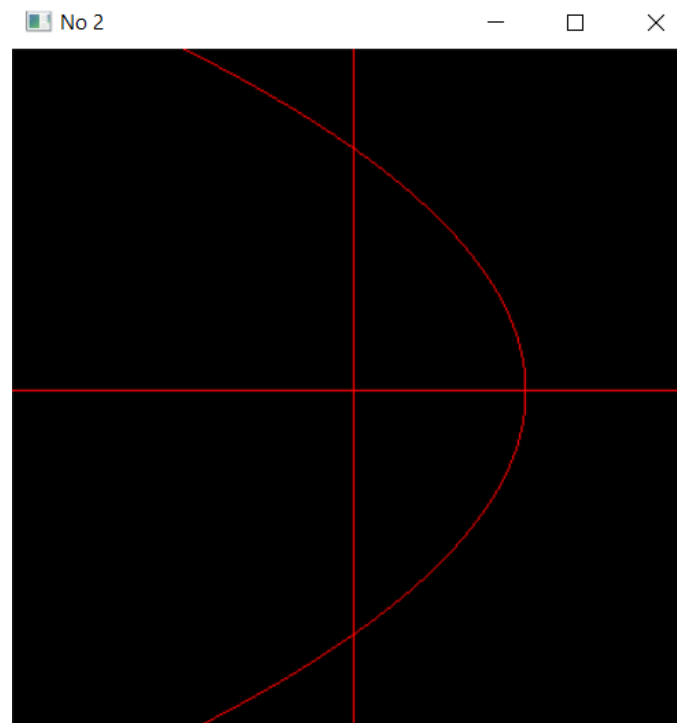
Jadi nantinya grafik kurva tersebut akan memiliki tinggi 5, dan melakukan 2 kali perulangan grafik dalam satu interval, dan memiliki pergeseran sudut sebesar 3, serta kurva tersebut bergeser sebesar 1.2 terhadap sumbu y.

#### 4. Hasil Running Program

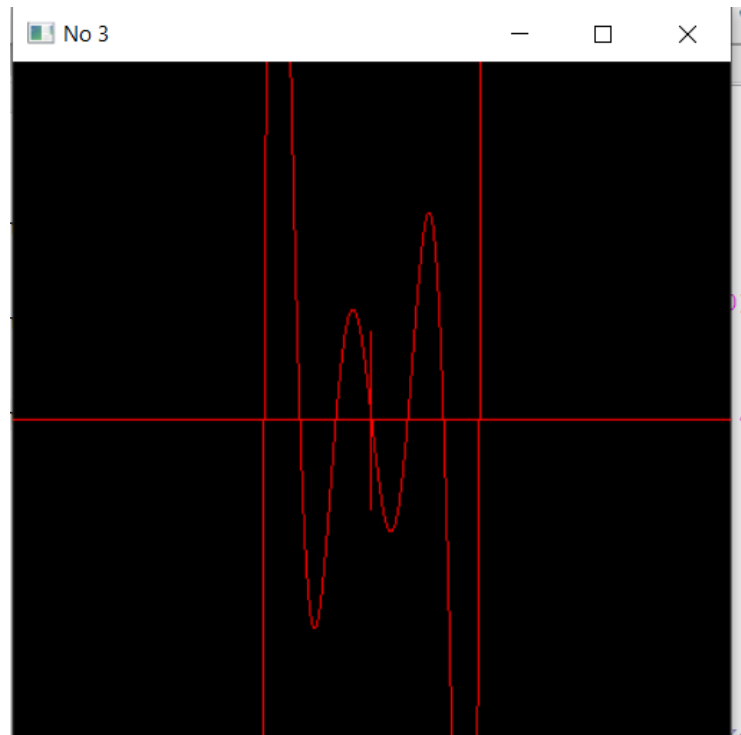
##### 1) Nomor 1



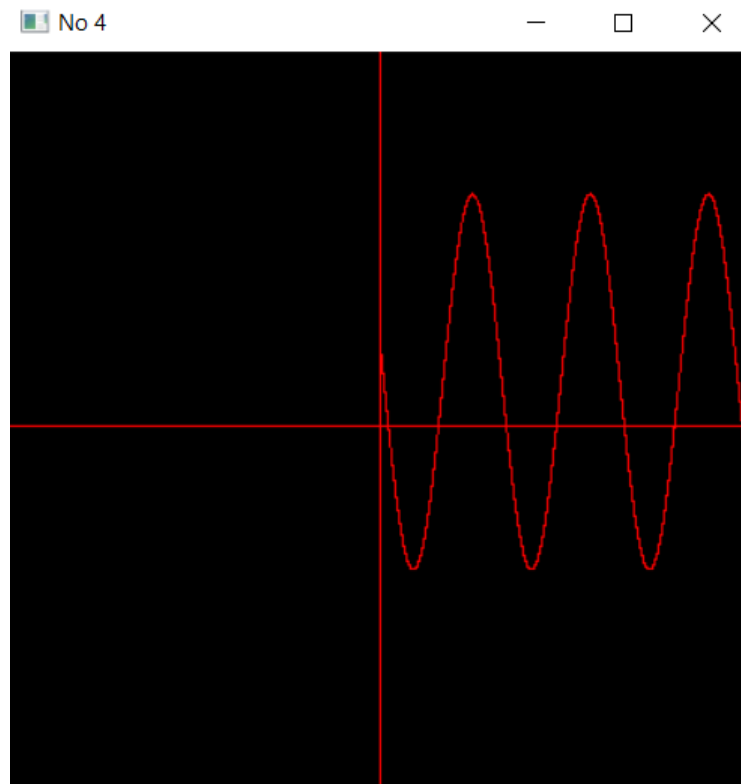
##### 2) Nomor 2



3) Nomor 3



4) Nomor 4





## **BAB II**

### **PENUTUP**

#### **A. Kesimpulan**

1. Mahasiswa mampu membuat grafik kurva dengan persamaan matematika melalui CodeBlocks.
2. GL\_POINTS menjadi pilihan terbaik untuk menggambar grafik kurva pada OpenGL.
3. Penggunaan gluOrtho2D yang harus disesuaikan agar grafik kurva terlihat dengan jelas.

#### **B. Saran**

Diperlukan banyak latihan dan mencoba megotak-atik sendiri agar mahasiswa lebih memahami dan mendalami fungsi – fungsi yang ada pada OpenGL.