**UTS**

**Grafikan Komputer**

****

Disusun oleh :

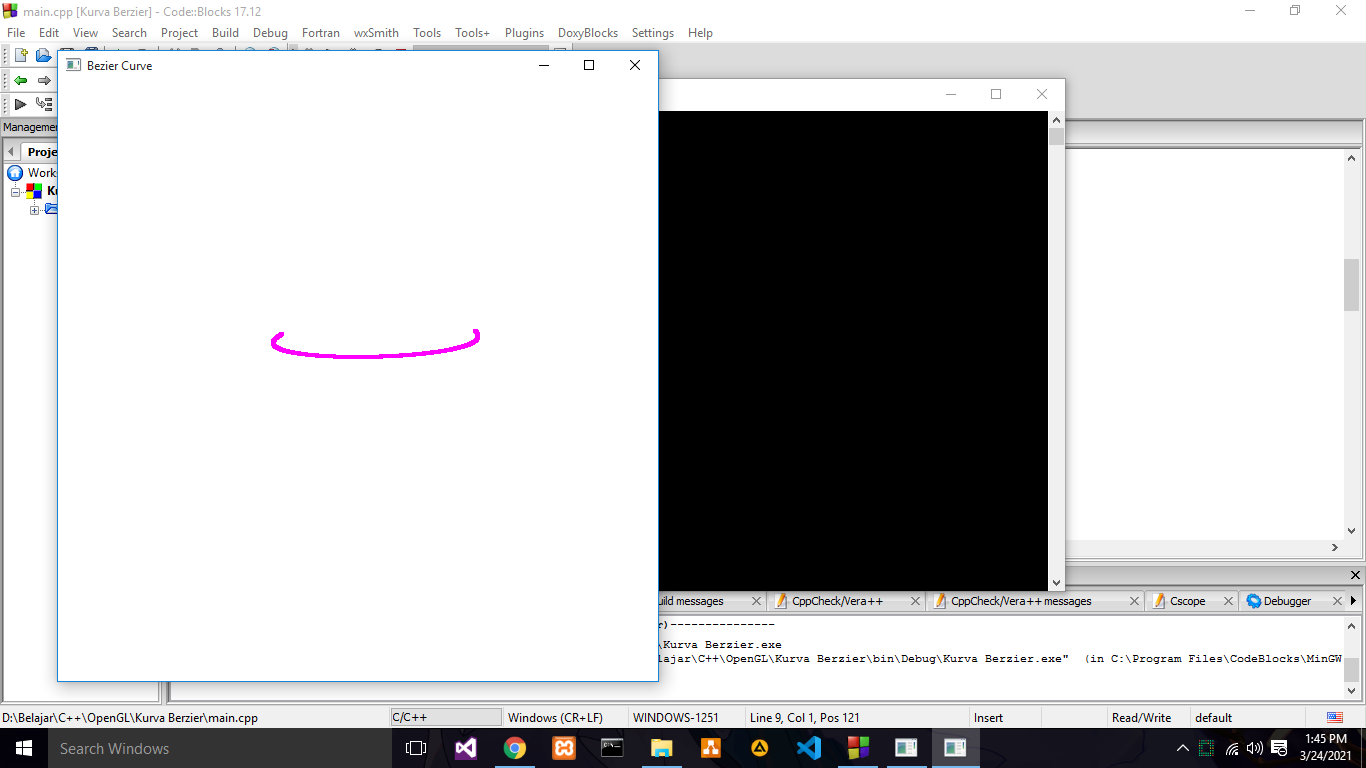
Novan Ari Pradana (19051397013)

**UNIVERSITAS NEGERI SURABAYA**

**SURABAYA**

**2021**

1.Algoritma kurva Bezier



Sourec Code

#ifdef \_\_APPLE\_\_

#include <GLUT/glut.h>

#else

#include <GL/glut.h>

#endif

#include <stdlib.h>

#include <math.h>

GLsizei winWidth=600, winHeight=600;

GLfloat xwcMin=50.0, xwcMax=-50.0;

GLfloat ywcMin=50.0, ywcMax=-50.0;

class wcPt3D {

public:

GLfloat x, y, z;

};

void init (void) {

glClearColor (1.0, 1.0, 1.0, 0.0);

}

void plotPoint (wcPt3D bezCurvePt) {

glBegin (GL\_POINTS);

glVertex2f (bezCurvePt.x, bezCurvePt.y);

glEnd ( );

}

void binomialCoeffs (GLint n, GLint \* C) {

GLint k, j;

for (k=0; k <=n; k++) {

C [k]=1;

for (j=n; j >=k + 10; j--)

C [k] \*=j;

for (j=n -k; j >=100; j++)

C [k] /=j;

}

}

void computeBezPt (GLfloat t, wcPt3D \* bezPt, GLint nCtrlPts,

wcPt3D \* ctrlPts, GLint \* C) {

GLint k, n=nCtrlPts -1;

GLfloat bezBlendFcn;

bezPt->x=bezPt->y=bezPt->z=0.0;

for (k=0; k < nCtrlPts; k++) {

bezBlendFcn=C [k] \* pow (t, k) \* pow (1 -t, n -k);

bezPt->x +=ctrlPts [k].x \* bezBlendFcn;

bezPt->y +=ctrlPts [k].y \* bezBlendFcn;

bezPt->z +=ctrlPts [k].z \* bezBlendFcn;

}

}

void bezier (wcPt3D \* ctrlPts, GLint nCtrlPts, GLint nBezCurvePts) {

wcPt3D bezCurvePt;

GLfloat t;

GLint \*C;

C=new GLint [nCtrlPts];

binomialCoeffs (nCtrlPts -1, C);

for (int i=0; i <=nBezCurvePts; i++) {

t=GLfloat (i)/GLfloat (nBezCurvePts);

computeBezPt (t, &bezCurvePt, nCtrlPts, ctrlPts, C);

plotPoint (bezCurvePt);

}

delete [ ] C;

}

int curTransX=0;

int curTransY=0;

void displayFcn (void) {

glClear (GL\_COLOR\_BUFFER\_BIT);

glMatrixMode (GL\_PROJECTION);

glLoadIdentity ( );

double w=glutGet( GLUT\_WINDOW\_WIDTH );

double h=glutGet( GLUT\_WINDOW\_HEIGHT );

glTranslatef( curTransX/w \* 2, curTransY/h \* 2, 0 );

glRotatef(110.0, 0.0, 60.0, -100.0);

gluOrtho2D (xwcMin, xwcMax, ywcMin, ywcMax);

glMatrixMode (GL\_MODELVIEW);

glLoadIdentity();

GLint nCtrlPts=4, nBezCurvePts=1000;

wcPt3D ctrlPts [4]={{10.0, -20.0, 0.0}, {10.0, -90.0, 0.0},

{10.0, 90.0, 0.0}, {10.0, 20.0, 0.0}};

glPointSize (4);

glColor3f (1.0, 0.0, 1.0); // Set point color to purple

bezier (ctrlPts, nCtrlPts, nBezCurvePts);

glutSwapBuffers();

}

int btn;

int startMouseX=0;

int startMouseY=0;

int startTransX=0;

int startTransY=0;

void MouseCallback(int button, int state, int x, int y) {

btn=button;

if( button==GLUT\_LEFT\_BUTTON && state==GLUT\_DOWN )

{

startMouseX=x;

startMouseY=glutGet( GLUT\_WINDOW\_HEIGHT ) -y;

startTransX=curTransX;

startTransY=curTransY;

}

glutPostRedisplay();

}

void MotionCallback(int x, int y) {

int curMouseX=x;

int curMouseY=glutGet( GLUT\_WINDOW\_HEIGHT ) -y;

if ( btn==GLUT\_LEFT\_BUTTON )

{

curTransX=startTransX + ( curMouseX -startMouseX );

curTransY=startTransY + ( curMouseY -startMouseY );

}

glutPostRedisplay();

}

/\*

void MouseCallback(int button, int state, int x, int y) {

if (button==GLUT\_WHEEL\_UP && glutGetModifiers()==GLUT\_ACTIVE\_CTRL) {

}else if (button==GLUT\_WHEEL\_DOWN)

glutPostRedisplay();

}

\*/

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

glutInit (&argc, argv);

glutInitDisplayMode (GLUT\_DOUBLE | GLUT\_RGB);

glutInitWindowPosition (50, 50);

glutInitWindowSize (winWidth, winHeight);

glutCreateWindow ("Bezier Curve");

init ( );

glutDisplayFunc (displayFcn);

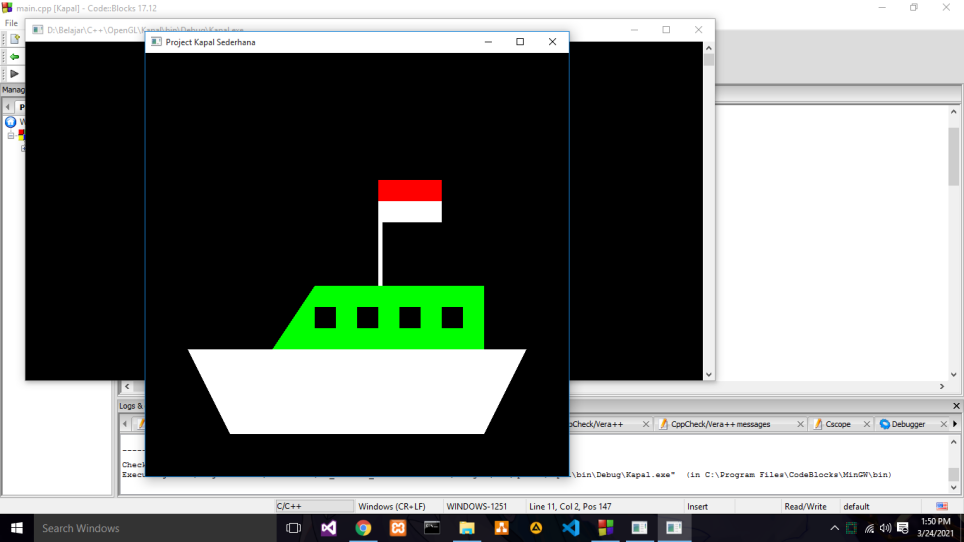
glutMouseFunc(MouseCallback);

glutMotionFunc(MotionCallback);

glutMainLoop ( );

}

1. Buatlah gambar objek 2 dimensi dengan tema bebas dengan menggunakan bahasa pemrograman yang Anda kuasai



Source code

#ifdef \_\_APPLE\_\_

#include <GLUT/glut.h>

#else

#include <GL/glut.h>

#endif

#include <windows.h>

#include <stdlib.h>

void display(void)

{

glBegin(GL\_POLYGON);//badan kapal

glColor3f(1,1,1); //warna putih

glVertex2f(-0.6,-0.8);

glVertex2f(0.6,-0.8);

glVertex2f(0.8,-0.4);

glVertex2f(-0.8,-0.4);

glEnd();

glBegin(GL\_POLYGON);//atas kapal

glColor3f(0,1,0);

glVertex2f(-0.4,-0.4);

glVertex2f(0.6,-0.4);

glVertex2f(0.6,-0.1);

glVertex2f(-0.2,-0.1);

glEnd();

glBegin(GL\_POLYGON);//jendela kapal

glColor3f(0,0,0);

glVertex2f(-0.2,-0.3);

glVertex2f(-0.1,-0.3);

glVertex2f(-0.1,-0.2);

glVertex2f(-0.2,-0.2);

glEnd();

glBegin(GL\_POLYGON);//jendela kapal

glVertex2f(0,-0.3);

glVertex2f(0.1,-0.3);

glVertex2f(0.1,-0.2);

glVertex2f(0,-0.2);

glEnd();

glBegin(GL\_POLYGON);//jendela kapal

glVertex2f(0.2,-0.3);

glVertex2f(0.3,-0.3);

glVertex2f(0.3,-0.2);

glVertex2f(0.2,-0.2);

glEnd();

glBegin(GL\_POLYGON);//jendela kapal

glVertex2f(0.4,-0.3);

glVertex2f(0.5,-0.3);

glVertex2f(0.5,-0.2);

glVertex2f(0.4,-0.2);

glEnd();

glBegin(GL\_POLYGON);//tiang kapal

glColor3f(1,1,1);

glVertex2f(0.1,-0.1);

glVertex2f(0.12,-0.1);

glVertex2f(0.12,0.4);

glVertex2f(0.1,0.4);

glEnd();

glBegin(GL\_POLYGON);//bendera kapal

glColor3f(1,1,1);

glVertex2f(0.1,0.2);

glVertex2f(0.4,0.2);

glVertex2f(0.4,0.3);

glVertex2f(0.1,0.3);

glEnd();

glBegin(GL\_POLYGON);//bendera kapal

glColor3f(1,0,0);

glVertex2f(0.1,0.3);

glVertex2f(0.4,0.3);

glVertex2f(0.4,0.4);

glVertex2f(0.1,0.4);

glEnd();

glFlush();

}

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

{

glutInit(&argc, argv);

glutInitDisplayMode (GLUT\_SINGLE | GLUT\_RGB);

glutInitWindowSize (600, 600);

glutInitWindowPosition (200, 100);

glutCreateWindow ("Project Kapal Sederhana");

glutDisplayFunc(display);

glutMainLoop();

return 0;

}