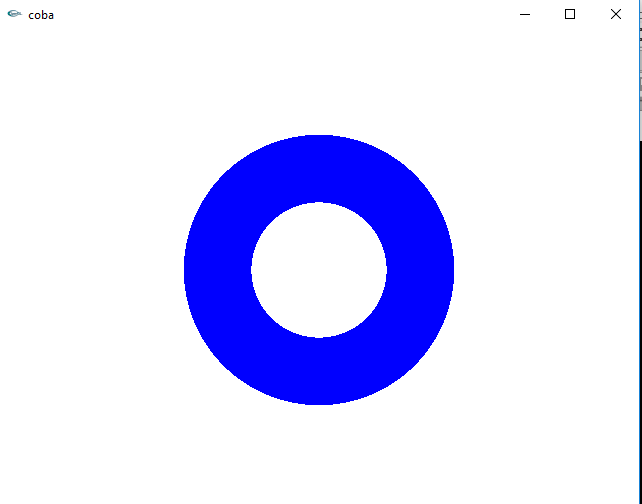
Tugas Modul 2 : Ovaldy | 161111062

1. #include <windows.h>
2. #include "GL/glut.h"
3. #include "math.h"
4. #define PI 3.14159265
6. **void** initGL()
7. {
8. glClearColor(1.0f, 1.0f, 1.0f, 1.0f);
9. glClearDepth(1.0f);
10. glEnable(GL\_DEPTH\_TEST);
11. glDepthFunc(GL\_LEQUAL);
12. glShadeModel(GL\_SMOOTH);
13. glHint(GL\_PERSPECTIVE\_CORRECTION\_HINT, GL\_NICEST);
14. }
16. **void** timer(**int** value)
17. {
18. glutPostRedisplay();
19. glutTimerFunc(15, timer, 0);
20. }

23. **void** reshape(GLsizei width, GLsizei height)
24. {
25. **if** (height == 0)
26. height = 1;
27. GLfloat aspect = (GLfloat)width / (GLfloat)height;
28. glViewport(0, 0, width, height);
29. glMatrixMode(GL\_PROJECTION);
30. glLoadIdentity();
31. gluPerspective(45.0f, aspect, 0.1f, 100.0f);
32. }
34. **float** degreeToRadian(**float** degree) {
35. // 360 degree = 2 pi radian
36. //   1 degree = 2 pi / 360 radian
37. //   1 degree = pi / 180 radian
38. **return** (degree \* PI) / 180.0;
39. }
41. **void** segiBeraturan(**float** xPusat, **float** yPusat, **float** r, **int** segi, **float** sudutAwal, **float** sudutAkhir) {
42. **float** besarSudutRad = degreeToRadian(360.0/segi);
43. **float** sudutAwalRad = degreeToRadian(sudutAwal);
44. **float** sudutAkhirRad = degreeToRadian(sudutAkhir);
45. glBegin(GL\_LINES);
46. **for** (**int** titikKe = 0; titikKe < segi; titikKe ++) {
47. **float** sudut1 = besarSudutRad \* titikKe + sudutAwalRad;
48. **float** sudut2 = besarSudutRad \* (titikKe + 1) + sudutAwalRad;
49. **float** x1 = cos(sudut1) \* r + xPusat;
50. **float** x2 = cos(sudut2) \* r + xPusat;
51. **float** y1 = sin(sudut1) \* r + yPusat;
52. **float** y2 = sin(sudut2) \* r + yPusat;
53. glVertex3f(x1, y1, 0);
54. glVertex3f(x2, y2, 0);
55. **if** (sudut2 >= sudutAkhirRad) {
56. **break**;
57. }
58. }
59. glEnd();
60. }
61. **void** linesXY(){
62. glBegin(GL\_LINES);
63. glVertex3f(-1.414, 0, 0);
64. glVertex3f(0, -1.414, 0);
65. glVertex3f(0, -1.414, 0);
66. glVertex3f(1.414, 0, 0);
67. glEnd();
68. }
69. **void** segi3(){
70. glBegin(GL\_TRIANGLES);
71. glColor3f(1.0f, 0.0f, 0.0f);
72. glVertex3f(0.0f, 0.5f, 0.0f);
73. glColor3f(1.0f, 0.0f, 1.0f);
74. glVertex3f(-0.5f, -0.5f, 0.0f);
75. glColor3f(1.0f, 1.0f, 0.0f);
76. glVertex3f(0.5f, -0.5f, 0.0f);
77. glVertex3f(1.0f, -0.5f, 0.0f);
78. glVertex3f(0.5f, -0.5f, 1.0f);
79. glVertex3f(0.5f, -1.5f, 0.0f);
81. glEnd();
82. }
84. **void** strip\_segi3(){
85. glBegin(GL\_TRIANGLE\_STRIP);
86. glVertex3f(0, 0.25f,0);
87. glVertex3f(-0.25f, -0.375f, 0);
88. glVertex3f(0.14f, 0.03f, 0);
89. glColor3f(1,0,0);
90. glVertex3f(0.35f, 0.25f, 0);
91. glVertex3f(0.85f, -0.35f, 0);
92. glEnd();
93. }
94. **void** quads(){
95. glBegin(GL\_QUADS);
96. glVertex3f(0,0.75f,0);
97. glVertex3f(-0.75f,0,0);
98. glVertex3f(0,-0.75f,0);
99. glVertex3f(0.75f,0,0);
100. glEnd();
101. }
102. **void** poligami(){
103. glBegin(GL\_POLYGON);
104. glVertex3f(0,0.5f,0);
105. glVertex3f(-0.5f,0.2f,0);
106. glVertex3f(-0.5f,-0.2f,0);
107. glVertex3f(0,-0.5f,0);
108. glVertex3f(0,0.5f,0);
109. glVertex3f(0.5f,0.2f,0);
110. glVertex3f(0.5f,-0.2f,0);
111. glVertex3f(0,-0.5f,0);
112. glEnd();
114. }
115. **void** hurufT(){
116. glBegin(GL\_POLYGON);
117. glVertex3f(0.1f,0.2f,0);
118. glVertex3f(0.1f,-0.2f,0);
119. glVertex3f(-0.1f,-0.2f,0);
120. glVertex3f(-0.1f,0.2f,0);
121. glVertex3f(-0.3f,0.2f,0);
122. glVertex3f(-0.3f,0.4f,0);
123. glVertex3f(0.3f,0.4f,0);
124. glVertex3f(0.3f,0.2f,0);
125. glEnd();
127. }
128. **void** bintang(){
129. glBegin(GL\_POLYGON);
130. glColor3f(1.0f, 0.0f, 0.0f);
131. glVertex3f(0.2f,-0.1f,0);
132. glVertex3f(0.3f,-0.5f,0);;
133. glVertex3f(0.0f,-0.2f,0);
134. glColor3f(1.0f, 0.0f, 0.0f);
135. glVertex3f(-0.2f,-0.5f,0);
136. glColor3f(1.0f, 1.0f, 0.0f);
137. glVertex3f(-0.1f,-0.1f,0);;
138. glVertex3f(-0.5f,0.0f,0);;
139. glVertex3f(-0.05f,0.1f,0);
140. glVertex3f(0.05f,0.5f,0);
141. glVertex3f(0.1f,0.1f,0);
142. glVertex3f(0.5f,0.0f,0);
144. glEnd();
146. }
147. **void** gambarHati(){
148. glBegin(GL\_LINES);
149. glVertex3f(-1.414, 0, 0);
150. glVertex3f(0, -1.414, 0);
151. glVertex3f(0, -1.414, 0);
152. glVertex3f(1.414, 0, 0);
153. glEnd();
154. segiBeraturan( 0.707, 0.707, 1, 30, -45, 135);
155. segiBeraturan(-0.707, 0.707, 1, 30, 45, 225);
156. }
157. **float** sudut = 0;
158. **void** lingkaran1(){
159. **int** i,radius,jumlah\_titik,x\_tengah,y\_tengah;
160. radius = 70;
161. jumlah\_titik = 360;
162. x\_tengah = 50;
163. y\_tengah = 50;
164. glBegin(GL\_POLYGON);
165. **for** (i=0;i<jumlah\_titik; i++){
166. //float sudut;
167. sudut = (**float**) (i\*(2\*PI/jumlah\_titik));
168. **float** x = (**float**) (x\_tengah+radius \* cos(sudut));
169. **float** y = (**float**) (y\_tengah+radius \* sin(sudut));
170. glVertex2f(x/100, y/100);
171. }
172. glEnd();
173. }
174. **void** lingkaran2(){
175. **int** i,radius,jumlah\_titik,x\_tengah,y\_tengah;
176. radius = 35;
177. jumlah\_titik = 360;
178. x\_tengah = 50;
179. y\_tengah = 50;
180. glBegin(GL\_POLYGON);
181. **for** (i=0;i<jumlah\_titik; i++){
182. //float sudut;
183. sudut = (**float**) (i\*(2\*PI/jumlah\_titik));
184. **float** x = (**float**) (x\_tengah+radius \* cos(sudut));
185. **float** y = (**float**) (y\_tengah+radius \* sin(sudut));
186. glVertex2f(x/100, y/100);
187. }
188. glEnd();
189. }
191. **void** display()
192. {
193. glClear(GL\_COLOR\_BUFFER\_BIT | GL\_DEPTH\_BUFFER\_BIT);
194. glMatrixMode(GL\_MODELVIEW);
195. glLoadIdentity();
197. glTranslatef(-0.5, -0.5, -3.0);
198. //glRotatef(sudut, 0,0,1);
199. glColor3f(0,0,1);
200. lingkaran1();
201. glTranslatef(0, 0, 0);
202. glColor3f(1,1,1);
203. lingkaran2();
205. sudut++;
207. glFlush();
208. glutSwapBuffers();
209. }
211. **int** main(**int** argc, **char** \*\*argv)
212. {
213. glutInit(&argc, argv);
214. glutInitDisplayMode(GLUT\_DOUBLE | GLUT\_DEPTH);
215. glutInitWindowSize(640, 480);
216. glutInitWindowPosition(50, 50);
217. glutCreateWindow("coba");
218. glutDisplayFunc(display);
219. glutReshapeFunc(reshape);
220. initGL();
221. glutTimerFunc(0, timer, 0);
222. glutMainLoop();
223. **return** 0;
224. }



1. #include <windows.h>
2. #include "GL/glut.h"
3. #include "math.h"
4. #define PI 3.14159265
6. **void** initGL()
7. {
8. glClearColor(1.0f, 1.0f, 1.0f, 1.0f);
9. glClearDepth(1.0f);
10. glEnable(GL\_DEPTH\_TEST);
11. glDepthFunc(GL\_LEQUAL);
12. glShadeModel(GL\_SMOOTH);
13. glHint(GL\_PERSPECTIVE\_CORRECTION\_HINT, GL\_NICEST);
14. }
16. **void** timer(**int** value)
17. {
18. glutPostRedisplay();
19. glutTimerFunc(15, timer, 0);
20. }

23. **void** reshape(GLsizei width, GLsizei height)
24. {
25. **if** (height == 0)
26. height = 1;
27. GLfloat aspect = (GLfloat)width / (GLfloat)height;
28. glViewport(0, 0, width, height);
29. glMatrixMode(GL\_PROJECTION);
30. glLoadIdentity();
31. gluPerspective(45.0f, aspect, 0.1f, 100.0f);
32. }
34. **float** degreeToRadian(**float** degree) {
35. // 360 degree = 2 pi radian
36. //   1 degree = 2 pi / 360 radian
37. //   1 degree = pi / 180 radian
38. **return** (degree \* PI) / 180.0;
39. }
41. **void** segiBeraturan(**float** xPusat, **float** yPusat, **float** r, **int** segi, **float** sudutAwal, **float** sudutAkhir) {
42. **float** besarSudutRad = degreeToRadian(360.0/segi);
43. **float** sudutAwalRad = degreeToRadian(sudutAwal);
44. **float** sudutAkhirRad = degreeToRadian(sudutAkhir);
45. glBegin(GL\_LINES);
46. **for** (**int** titikKe = 0; titikKe < segi; titikKe ++) {
47. **float** sudut1 = besarSudutRad \* titikKe + sudutAwalRad;
48. **float** sudut2 = besarSudutRad \* (titikKe + 1) + sudutAwalRad;
49. **float** x1 = cos(sudut1) \* r + xPusat;
50. **float** x2 = cos(sudut2) \* r + xPusat;
51. **float** y1 = sin(sudut1) \* r + yPusat;
52. **float** y2 = sin(sudut2) \* r + yPusat;
53. glVertex3f(x1, y1, 0);
54. glVertex3f(x2, y2, 0);
55. **if** (sudut2 >= sudutAkhirRad) {
56. **break**;
57. }
58. }
59. glEnd();
60. }
61. **void** linesXY(){
62. glBegin(GL\_LINES);
63. glVertex3f(-1.414, 0, 0);
64. glVertex3f(0, -1.414, 0);
65. glVertex3f(0, -1.414, 0);
66. glVertex3f(1.414, 0, 0);
67. glEnd();
68. }
69. **void** segi3(){
70. glBegin(GL\_TRIANGLES);
71. glColor3f(1.0f, 0.0f, 0.0f);
72. glVertex3f(0.0f, 0.5f, 0.0f);
73. glColor3f(1.0f, 0.0f, 1.0f);
74. glVertex3f(-0.5f, -0.5f, 0.0f);
75. glColor3f(1.0f, 1.0f, 0.0f);
76. glVertex3f(0.5f, -0.5f, 0.0f);
77. glVertex3f(1.0f, -0.5f, 0.0f);
78. glVertex3f(0.5f, -0.5f, 1.0f);
79. glVertex3f(0.5f, -1.5f, 0.0f);
81. glEnd();
82. }
84. **void** strip\_segi3(){
85. glBegin(GL\_TRIANGLE\_STRIP);
86. glVertex3f(0, 0.25f,0);
87. glVertex3f(-0.25f, -0.375f, 0);
88. glVertex3f(0.14f, 0.03f, 0);
89. glColor3f(1,0,0);
90. glVertex3f(0.35f, 0.25f, 0);
91. glVertex3f(0.85f, -0.35f, 0);
92. glEnd();
93. }
94. **void** quads(){
95. glBegin(GL\_QUADS);
96. glVertex3f(0,0.75f,0);
97. glVertex3f(-0.75f,0,0);
98. glVertex3f(0,-0.75f,0);
99. glVertex3f(0.75f,0,0);
100. glEnd();
101. }
102. **void** poligami(){
103. glBegin(GL\_POLYGON);
104. glVertex3f(0,0.5f,0);
105. glVertex3f(-0.5f,0.2f,0);
106. glVertex3f(-0.5f,-0.2f,0);
107. glVertex3f(0,-0.5f,0);
108. glVertex3f(0,0.5f,0);
109. glVertex3f(0.5f,0.2f,0);
110. glVertex3f(0.5f,-0.2f,0);
111. glVertex3f(0,-0.5f,0);
112. glEnd();
114. }
115. **void** hurufT(){
116. glBegin(GL\_POLYGON);
117. glVertex3f(0.1f,0.2f,0);
118. glVertex3f(0.1f,-0.2f,0);
119. glVertex3f(-0.1f,-0.2f,0);
120. glVertex3f(-0.1f,0.2f,0);
121. glVertex3f(-0.3f,0.2f,0);
122. glVertex3f(-0.3f,0.4f,0);
123. glVertex3f(0.3f,0.4f,0);
124. glVertex3f(0.3f,0.2f,0);
125. glEnd();
127. }
128. **void** bintang(){
129. glBegin(GL\_POLYGON);
130. glColor3f(1.0f, 0.0f, 0.0f);
131. glVertex3f(0.2f,-0.1f,0);
132. glVertex3f(0.3f,-0.5f,0);;
133. glVertex3f(0.0f,-0.2f,0);
134. glColor3f(1.0f, 0.0f, 0.0f);
135. glVertex3f(-0.2f,-0.5f,0);
136. glColor3f(1.0f, 1.0f, 0.0f);
137. glVertex3f(-0.1f,-0.1f,0);;
138. glVertex3f(-0.5f,0.0f,0);;
139. glVertex3f(-0.05f,0.1f,0);
140. glVertex3f(0.05f,0.5f,0);
141. glVertex3f(0.1f,0.1f,0);
142. glVertex3f(0.5f,0.0f,0);
144. glEnd();
146. }
147. **void** gambarHati(){
148. glBegin(GL\_LINES);
149. glVertex3f(-1.414, 0, 0);
150. glVertex3f(0, -1.414, 0);
151. glVertex3f(0, -1.414, 0);
152. glVertex3f(1.414, 0, 0);
153. glEnd();
154. segiBeraturan( 0.707, 0.707, 1, 30, -45, 135);
155. segiBeraturan(-0.707, 0.707, 1, 30, 45, 225);
156. }
157. **float** sudut = 0;
158. **void** lingkaran1(){
159. **int** i,radius,jumlah\_titik,x\_tengah,y\_tengah;
160. radius = 70;
161. jumlah\_titik = 360;
162. x\_tengah = 50;
163. y\_tengah = 50;
164. glBegin(GL\_POLYGON);
165. **for** (i=0;i<jumlah\_titik; i++){
166. //float sudut;
167. **if**(i>(jumlah\_titik/3\*2)){
168. glColor3f(0,1,0);
169. }**else** **if**(i>(jumlah\_titik/3)){
170. glColor3f(0,0,0);
171. }**else**{
172. glColor3f(1,0,0);
173. }
174. sudut = (**float**) (i\*(2\*PI/jumlah\_titik));
175. **float** x = (**float**) (x\_tengah+radius \* cos(sudut));
176. **float** y = (**float**) (y\_tengah+radius \* sin(sudut));
177. glVertex2f(x/100, y/100);
178. }
179. glEnd();
180. }
181. **void** lingkaran2(){
182. **int** i,radius,jumlah\_titik,x\_tengah,y\_tengah;
183. radius = 35;
184. jumlah\_titik = 360;
185. x\_tengah = 50;
186. y\_tengah = 50;
187. glBegin(GL\_POLYGON);
188. **for** (i=0;i<jumlah\_titik; i++){
189. //float sudut;
190. sudut = (**float**) (i\*(2\*PI/jumlah\_titik));
191. **float** x = (**float**) (x\_tengah+radius \* cos(sudut));
192. **float** y = (**float**) (y\_tengah+radius \* sin(sudut));
193. glVertex2f(x/100, y/100);
194. }
195. glEnd();
196. }
198. **void** display()
199. {
200. glClear(GL\_COLOR\_BUFFER\_BIT | GL\_DEPTH\_BUFFER\_BIT);
201. glMatrixMode(GL\_MODELVIEW);
202. glLoadIdentity();
204. glTranslatef(-0.5, -0.5, -3.0);
205. //glRotatef(sudut, 0,0,1);
206. glColor3f(0,1,0);
207. lingkaran1();
208. glTranslatef(0, 0, 0);
209. glColor3f(1,1,1);
210. lingkaran2();
212. sudut++;
214. glFlush();
215. glutSwapBuffers();
216. }
218. **int** main(**int** argc, **char** \*\*argv)
219. {
220. glutInit(&argc, argv);
221. glutInitDisplayMode(GLUT\_DOUBLE | GLUT\_DEPTH);
222. glutInitWindowSize(640, 480);
223. glutInitWindowPosition(50, 50);
224. glutCreateWindow("coba");
225. glutDisplayFunc(display);
226. glutReshapeFunc(reshape);
227. initGL();
228. glutTimerFunc(0, timer, 0);
229. glutMainLoop();
230. **return** 0;
231. }

