

## TEXTURE MAPPING pada PERMUKAAN BOLA

Berikut ini salah satu contoh program untuk mengimplementasikan texture mapping pada permukaan bola. Caranya dengan menggunakan fungsi yang sudah tersedia pada library GLU yaitu: `gluQuadricTexture()`, dan untuk membuat sebuah objek bola dengan menggunakan fungsi `gluSphere()`.

Source code:

```
//  
// main.c  
// textureMappingBola  
//  
// Created by alfa suni on 6/4/13.  
// Copyright (c) 2013 alfa suni. All rights reserved.  
//  
  
#include <stdio.h>  
#include <stdlib.h>  
#include <GLUT/GLUT.h>  
#include "SOIL.h"  
  
GLUquadricObj *pBumi;  
GLuint texture;  
GLuint tex_2d;  
GLfloat z_pos = -3.0f;  
GLfloat rot = 0.0f;  
  
GLuint LoadGLTexture(const char *filename)  
{  
  
    // load an image file directly as a new OpenGL texture  
    tex_2d = SOIL_load_OGL_texture(filename, SOIL_LOAD_AUTO, SOIL_CREATE_NEW_ID,  
SOIL_FLAG_INVERT_Y);  
  
    // check for an error during the load process  
    if(tex_2d == 0) {  
        printf("SOIL loading error: '%s'\n", SOIL_last_result());  
    }  
  
    return tex_2d;  
}
```

```

void init(void)
{
    texture = LoadGLTexture("bumi.bmp");

    glEnable(GL_TEXTURE_2D);
    glShadeModel(GL_SMOOTH);

    glClearColor(0.0f, 0.0f, 0.0f, 0.5f);

    pBumi = gluNewQuadric();
    gluQuadricNormals(pBumi, GLU_SMOOTH);
    gluQuadricTexture(pBumi, GL_TRUE);
}

```

```

void renderScene(void)
{
    glClear(GL_COLOR_BUFFER_BIT);

    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();

    glTranslatef(0.0, 0.0, z_pos);
    glRotatef(-115.0, 1.0, 0.0, 0.0);

    glPushMatrix();
        glRotatef(rot, 0.0, 0.0, 1.0);
        glBindTexture(GL_TEXTURE_2D, texture);
        gluSphere(pBumi, 1.0, 360, 180);
    glPopMatrix();

    glutSwapBuffers();
}

```

```

void myKeyboard(unsigned char key, int x, int y)
{
    switch (key) {
        case '<':
        case ',':
            z_pos -= 0.1f;
            glutPostRedisplay();
            break;

        case '>':
        case '.':
            z_pos += 0.1f;
            glutPostRedisplay();
            break;
    }
}

```

```

        case 27:
            exit(0);
            break;

        default:
            break;
    }
}

void myTimeOut(int id)
{
    rot += 10.0f;
    glutPostRedisplay();
    glutTimerFunc(100, myTimeOut, 0);
}

void resize(int width, int height)
{
    glViewport(0, 0, width, height);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluPerspective(45.0, (GLdouble)width/(GLdouble)height, 1.0, 300.0);

    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();
}

int main(int argc, char * argv[])
{
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGBA | GLUT_DEPTH);
    glutInitWindowSize(500, 500);
    glutInitWindowPosition(0, 0);
    glutCreateWindow("Texture Mapping Bola");

    init();

    glutDisplayFunc(renderScene);
    glutKeyboardFunc(myKeyboard);
    glutReshapeFunc(resize);
    glutTimerFunc(100, myTimeOut, 0);

    glutMainLoop();

    return 0;
}

```