

## MODUL 8

### Lighting dan Material

#### A. KOMPETENSI DASAR

- Memahami Inisialisasi Lighting dan Material
- Memahami Object 3D (Solid).
- Memahami dasar menampilkan susunan objek 3D dengan Lighting dan Material.

#### B. ALOKASI WAKTU

4 js (4x50 menit)

#### C. PETUNJUK

- Awali setiap aktivitas dengan do'a, semoga berkah dan mendapat kemudahan.
- Pahami Tujuan, dasar teori, dan latihan-latihan praktikum dengan baik dan benar.
- Kerjakan tugas-tugas dengan baik, sabar, dan jujur.
- Tanyakan kepada asisten/dosen apabila ada hal-hal yang kurang jelas.

#### D. DASAR TEORI

##### 1. Bentuk *Solid*

Fungsi *Solid* merupakan implementasi dari object 3D yang berpusat pada asal pemodelan sistem koordinat. Utara dan kutub selatan bola berada di Z positif dan negatif sumbu-masing-masing dan meridian utama persimpangan sumbu X positif.

Berikut adalah list untuk bangun 3D:

##### a. Kubus Solid

```
void glutSolidCube(GLdouble size);
```

##### b. Fungsi glutSolidSphere membuat bola berpusat pada asal pemodelan sistem koordinat. Utara dan kutub selatan bola berada di Z positif dan negatif sumbu-masing-masing dan meridian utama persimpangan sumbu X positif.

```
void glutSolidSphere(GLdouble radius, GLint slices, GLint stacks);
```

##### c. Kerucut Solid.

```
void glutSolidCone(GLdouble base, GLdouble height, GLint slices, GLint stacks);
```

##### d. Torus Solid.

```
void glutSolidTorus(GLdouble innerRadius, GLdouble outerRadius, GLint nsides, GLint rings);
```

## e. Dodecahedron Solid

```
void glutSolidDodecahedron(void);
```

## f. Tetra Hedron solid.

```
glutSolidTetrahedron void (void);
```

## g. Oktahedron solid.

```
void glutSolidOctahedron(void);
```

## h. Icosahedron solid.

```
void glutSolidIcosahedron(void);
```

## i. Poci teh solid

```
void glutSolidTeapot(GLdouble size);
```

## 2. Lengkapi pada fungsi init seperti di bawah ini.

```
void init(){
    GLfloat LightPosition[] = {10.0f, 10.0f, 20.0f, 0.0f};
    /* ... */

    glShadeModel(GL_SMOOTH);
    /* ... */
    glEnable(GL_DEPTH_TEST); /* ... */
    glHint(GL_PERSPECTIVE_CORRECTION_HINT, GL_NICEST);
    /* ... */
    glLightfv(GL_LIGHT0, GL_POSITION, LightPosition);

    glEnable(GL_LIGHTING);
    glEnable(GL_LIGHT0);

    return;
}
```

3. Fungsi di atas adalah fungsi inialisasi untuk pencahayaan dimana efek pencahayaan bersifat menyeluruh dan menyebabkan semua obyek memiliki warna serupa. Agar warna tiap obyek 3d berbeda dengan efek pencahayaan yang aktif maka fungsi dasarnya pada inialisasi adalah

```
glColorMaterial(GLenum face, GLenum mode);
```

Fungsi di atas dapat diaktifkan setelah **glEnable(GL\_COLOR\_MATERIAL)**. Parameter **face** adalah variabel untuk menentukan bagaimana material warna obyek 3D ditampilkan, pilihannya antara lain **GL\_FRONT** atau **GL\_BACK**, Sedangkan parameter **mode** adalah variabel untuk memilih material warna jenis apa yang diaktifkan pilihannya antara lain:

Parameter Name	Default Value	Meaning
GL_AMBIENT	(0.2, 0.2, 0.2, 1.0)	ambient color of material
GL_DIFFUSE	(0.8, 0.8, 0.8, 1.0)	diffuse color of material
GL_AMBIENT_AND_DIFFUSE		ambient and diffuse color of material
GL_SPECULAR	(0.0, 0.0, 0.0, 1.0)	specular color of material
GL_SHININESS	0.0	specular exponent
GL_EMISSION	(0.0, 0.0, 0.0, 1.0)	emissive color of material
GL_COLOR_INDEXES	(0,1,1)	ambient, diffuse, and specular color indices

### E. AKTIFITAS KELAS PRAKTIKUM

1. Buatlah tampilan sebuah solidSphere pada fungsi render. Tampilkan *screen shot*.
2. Ubahlah fungsi init menjadi sebagai berikut.

```
void init(){
    GLfloat LightPosition[] = {10.0f, 10.0f, 20.0f, 0.0f};
    GLfloat LightAmbient[] = {0.0f, 1.0f, 0.0f, 1.0f};
    GLfloat LightDiffuse[] = {0.7f, 0.7f, 0.7f, 1.0f};
    GLfloat LightSpecular[] = {0.5f, 0.5f, 0.5f, 1.0f};
    GLfloat Shine[] = { 80 };

    glShadeModel(GL_SMOOTH);
    glClearColor(0.0f, 0.0f, 0.0f, 0.5f);
    glClearDepth(1.0f);
    glEnable(GL_DEPTH_TEST); /* ... */
    glHint(GL_PERSPECTIVE_CORRECTION_HINT, GL_NICEST);

    glMaterialfv(GL_FRONT, GL_AMBIENT, LightAmbient);
    /* ... */
    glLightfv(GL_LIGHT0, GL_POSITION, LightPosition);

    glEnable(GL_LIGHTING);
    glEnable(GL_LIGHT0);
    return;
}
```

3. Tampilkan screenshot dan simpulkan penggunaan lighting – ambient untuk apa.
4. Ubahlah bagian yang ditandai kotak menjadi  

```
glMaterialfv(GL_FRONT, GL_DIFFUSE, LightDiffuse);
```

 Sajikan Screenshot dan beri kesimpulan mengenai lighting – diffuse.

5. Ubahlah bagian yang ditandai kotak menjadi  

```
glMaterialfv(GL_FRONT, GL_SPECULAR, LightSpecular);  
glMaterialfv(GL_FRONT, GL_SHININESS, Shine);
```

Sajikan Screenshot dan beri kesimpulan mengenai lighting – specular.
6. Buatlah sebuah program yang menampilkan semua **benda solid** secara mendatar di koordinat x, dengan warna yang bervariasi untuk tiap benda 3d tersebut. Sajikan screenshot dan program untuk render-nya.  
Tambahkan code di bawah ini pada fungsi rendering sebelum menampilkan obyek-obyek 3 dimensi :  

```
glEnable(GL_COLOR_MATERIAL);  
.....(obyek 3 dimensi)  
.....(obyek 3 dimensi)  
glDisable(GL_COLOR_MATERIAL);
```
7. Ubah-ubahlah posisi lighting dan parameter lighting yang lain sehingga memberikan tampilan yang berbeda.
8. Buatlah kesimpulan mengenai Lighting secara keseluruhan.

#### F. TUGAS ASISTENSI

1. Buatlah program yang menampilkan sebuah rumah dengan atap yang berbeda warna dengan temboknya dan dikelilingi oleh pagar .
2. Dari program nomor 1, ubahlah program sehingga posisi cahaya akan mengikuti posisi pointer mouse. Sajikan screenshot dari 3 posisi pointer yang berbeda.
3. Lengkapi program di atas sehingga rumah dan sekitarnya dapat diputar untuk melihat sisi lain rumah dan sekitarnya. Sajikan screenshot dari 3 sudut pandang yang berbeda. Tuliskan program lengkapnya pada fungsi init dan render.