



# FAKULTAS TEKNIK

UNIVERSITAS NEGERI YOGYAKARTA

## LAB. SHEET PRAKTIKUM GRAFIKA KOMPUTER 1

No. : ST/EKA/PTI223/05

Revisi : 02

Senin 010409

Hal. 1 dari 8 hal.

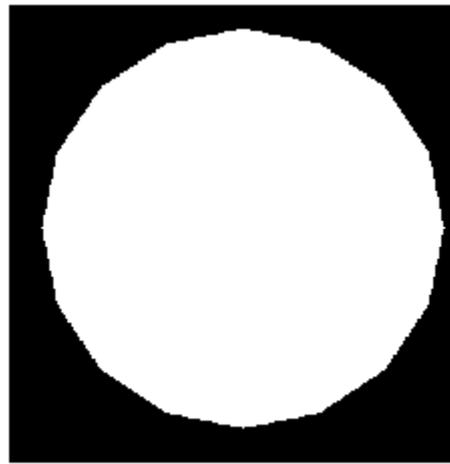
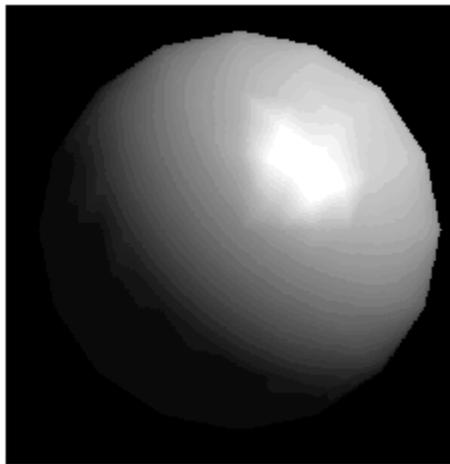
Sem.: Genap

### Pencerahan pada OpenGL

4 x 50 menit

#### A. Pendahuluan

OpenGL akan melakukan komputasi warna setiap pixel di display akhir, ditampilkan adegan itu dilakukan di frame buffer. Bagian dari komputasi ini tergantung pada pencerahan yang digunakan dalam adegan dan hal tersebut berkenaan dengan bagaimana suatu benda dalam adegan yang memantulkan atau menyerap cahaya. Sebagai contoh saja misalnya sebuah lautan memiliki warna yang berbeda pada cerah maupun pada saat hari mendung. Adanya sinar matahari atau awan menentukan apakah laut termelihat sebagai pirus terang atau abu-abu keruh kehijauan. Pada kenyataannya, sebagian besar objek bahkan yang tidak terlihat tiga dimensi sampai mereka diterangi cahaya. Gambar dibawah ini menunjukkan dua versi dari adegan yang persis sama yaitu sebuah bola, dimana satu dengan pencerahan dan satu tanpa pencerahan.



Gambar sebuah sfera dengan diterangi cahaya dan tidak

Seperti gambar diatas, wilayah gelap tampak tidak berbeda dari bulatan dua dimensi . hal Ini menunjukkan betapa pentingnya interaksi antara obyek dan cahaya adalah dalam menciptakan adegan tiga dimensi.

Agar objek yang telah dibuat terlihat lebih nyata, diperlukan tambahan efek pencerahan pada objek yang telah kita buat. Pencerahan adalah proses pemberian cahaya pada suatu objek, dikenal dengan istilah *lighting* atau iluminasi. Pada OpenGL Terdapat sebuah perbedaan penting antara Warna dan pencerahan yang perlu dipahami. Ketika menggunakan pencerahan atau tekstur pemetaan dengan efek pencerahan dihidupkan, warna dari vertex adalah efek kumulatif dari warna bahan dan cahaya yang bersinar di puncak. Ketika pencerahan dimatikan, maka warna dari vertex adalah efek pengaturan warna dan warna yang berbeda dengan bahan warna.



# FAKULTAS TEKNIK

UNIVERSITAS NEGERI YOGYAKARTA

## LAB. SHEET PRAKTIKUM GRAFIKA KOMPUTER 1

No. : ST/EKA/PTI223/05

Revisi : 02

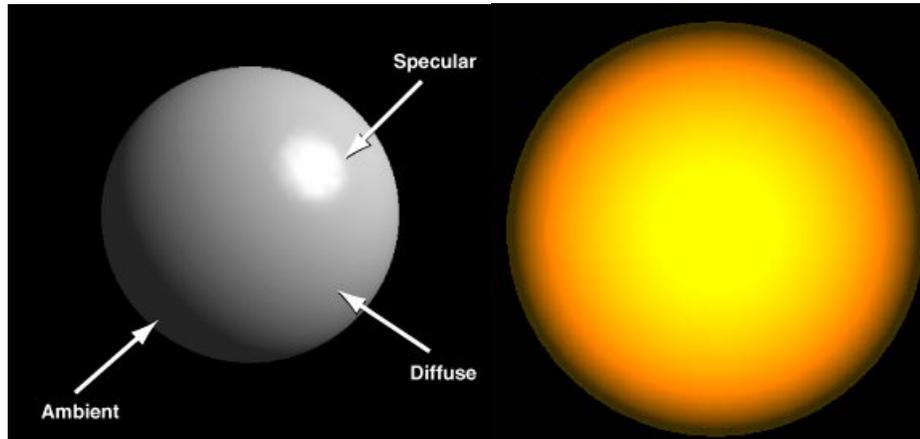
Senin 010409

Hal. 2 dari 8 hal.

Sem.: Genap

**Pencahayaan pada OpenGL**

4 x 50 menit



Gambar contoh ambient, diffuse, dan specular pada pantulan cahaya

Dengan menggunakan OpenGL, dapat memanipulasi pencahayaan dan objek dalam sebuah adegan untuk menciptakan berbagai macam efek. Dalam pokok bahasan pada bab pencahayaan akan dimulai dengan sebuah pencahayaan yang penting pada permukaan tersembunyi penghapusan. Kemudian menjelaskan bagaimana untuk mengontrol pencahayaan dalam suatu adegan, membahas model konseptual OpenGL pencahayaan, dan menggambarkan secara rinci cara mengatur pencahayaan dengan banyak parameter untuk mendapatkan efek tertentu. Pada akhir bab, perhitungan matematis yang menentukan bagaimana warna mempengaruhi pencahayaan disajikan

### **Pencahayaan pada OpenGL dan Dunia nyata**

Ketika melihat permukaan fisik, persepsi warna pada mata tergantung pada distribusi energi foton yang datang dan memicu sel-sel kerucut mata. Foton berasal dari sumber cahaya atau kombinasi dari sumber, yang sebagian diserap dan sebagian yang dipantulkan oleh permukaan. Selain itu, permukaan yang berbeda memiliki sifat yang sangat berbeda-ada, misalnya yang mengkilap dengan sempurna akan memantulkan cahaya dalam arah tertentu, sementara yang lain menyebarkan cahaya sama-sama masuk di segala penjuru.

Pencahayaan pada OpenGL hanya dengan cahaya pendekatan dan cahaya lampu seolah-olah dapat dipecah menjadi komponen merah, hijau, dan biru. Dengan demikian, warna sumber cahaya dicirikan oleh jumlah warna yang memancarkan cahaya merah, hijau, dan biru, dan materi permukaan ditandai dengan persentase komponen warna merah, hijau, dan biru yang masuk dan tercermin dalam berbagai arah. Persamaan pencahayaan pada OpenGL hanya sebuah pendekatan, tetapi satu yang bekerja cukup baik dan dapat dihitung relatif cepat. Jika menginginkan yang lebih akurat atau model pencahayaan hanya berbeda, harus melakukan perhitungan sendiri dalam perangkat lunak. Dengan perangkat lunak tersebut dapat menjadi sangat kompleks, seperti beberapa jam membaca buku teks optik pun harus meyakinkan.

Model pencahayaan Dalam OpenGL, cahaya dalam sebuah adegan berasal dari beberapa sumber cahaya yang dapat secara individual diaktifkan dan dinonaktifkan. Beberapa cahaya datang dari arah atau posisi tertentu, dan beberapa cahaya umumnya tersebar karena adanya suatu peristiwa. Sebagai contoh, ketika menghidupkan bola lampu dalam ruangan, sebagian besar berasal dari cahaya bola lampu, tetapi beberapa cahaya datang setelah ada pantulan dari dinding satu, dua, tiga, atau lebih. Cahaya yang memantul ini disebut ambient



# FAKULTAS TEKNIK

UNIVERSITAS NEGERI YOGYAKARTA

## LAB. SHEET PRAKTIKUM GRAFIKA KOMPUTER 1

No. : ST/EKA/PTI223/05

Revisi : 02

Senin 010409

Hal. 3 dari 8 hal.

Sem.: Genap

**Pencahayaan pada OpenGL**

4 x 50 menit

dan dapat diasumsikan begitu cahaya tersebar tidak ada cara untuk mengetahui arah semula, tetapi hal ini akan menghilang jika suatu sumber cahaya dimatikan.

Akhirnya, mungkin ada cahaya ambient umum dalam adegan yang tidak berasal dari sumber tertentu, seolah-olah telah tersebar beberapa kali sumber cahaya asli dan tidak mungkin untuk menentukan.

Dalam model OpenGL, sumber cahaya hanya memiliki efek ketika ada permukaan yang menyerap dan memantulkan cahaya. Setiap permukaan diasumsikan terdiri dari bahan dengan berbagai sifat. Sebuah bahan bisa memancarkan cahaya sendiri seperti lampu pada sebuah mobil atau mungkin menyebarkan beberapa cahaya yang masuk ke segala penjuru, dan mungkin juga memantulkan sebagian dari cahaya masuk dalam arah preferensial seperti cermin atau permukaan mengilap.

Model pencahayaan yang OpenGL mempertimbangkan pencahayaan yang dibagi menjadi empat komponen independen: memancarkan (emissi), ambient, diffuse, dan specular. Semua empat komponen dihitung secara independen dan kemudian ditambahkan secara bersama-sama.

### **Cahaya Ambient, Diffuse, dan specular**

Pencahayaan *Ambient* adalah cahaya yang sudah berserakan begitu banyak disebabkan oleh lingkungan dan arahnya tidak mungkin ditentukan atau tampaknya datang dari segala penjuru. *Backlighting* pada sebuah ruangan memiliki komponen *ambient* besar, karena sebagian besar cahaya yang mencapai mata yang memantul dari banyak permukaan. Sebuah lampu sorot kecil di luar rumah memiliki komponen ambient, sebagian besar cahaya dalam arah yang sama, dan karena diluar, sangat sedikit cahaya mencapai mata setelah memantul dari benda-benda lain. Ketika cahaya ambient menyerang permukaan, maka akan tersebar merata di segala penjuru.

Komponen cahaya *diffuse* adalah komponen yang berasal dari satu arah, jadi akan terang kalau hal tersebut terjadi tepat diatas sebuah permukaan dibandingkan jika hampir tidak terjadi di atas permukaan. Setelah mengenai permukaan, akan tersebar merata di segala penjuru, sehingga tampak sama-sama terang, tak peduli di mana mata berada. Setiap cahaya yang datang dari posisi atau arah tertentu mungkin memiliki komponen diffuse.

Cahaya *specular* datang dari arah tertentu, dan cenderung terpental pada permukaan dalam arah yang diinginkan. sinar laser berkualitas tinggi memantul pada cermin dan menghasilkan hampir 100 persen refleksi specular. Logam atau plastik mengilap memiliki komponen specular tinggi, dan kapur atau karpet telah hampir tidak ada. Specularity dapat juga dianggap sebagai *shininess*.

Meskipun sumber cahaya memberikan satu distribusi frekuensi, komponen ambient, diffuse, dan specular mungkin berbeda. Sebagai contoh, jika memiliki cahaya putih di sebuah ruangan dengan dinding merah, cahaya yang tersebar cenderung menjadi warna merah, meskipun cahaya secara langsung objek putih yang mencolok. OpenGL memungkinkan untuk mengatur nilai merah, hijau, dan biru untuk setiap komponen cahaya secara bebas.

## **B. Program**

### **Program 1. sebuah Lit Sphere**

Dibawah ini merupakan langkah-langkah yang diperlukan untuk menambahkan pencahayaan ke obyek.



# FAKULTAS TEKNIK

UNIVERSITAS NEGERI YOGYAKARTA

## LAB. SHEET PRAKTIKUM GRAFIKA KOMPUTER 1

No. : ST/EKA/PTI223/05

Revisi : 02

Senin 010409

Hal. 4 dari 8 hal.

Sem.: Genap

### Pencapaian pada OpenGL

4 x 50 menit

- Tentukan vektor normal untuk setiap sudut dari semua benda. Vektor normal ini menentukan orientasi objek relatif terhadap sumber cahaya.
- Buat, pilih, dan atur posisi satu atau lebih dari sebuah sumber cahaya.
- Ciptakan dan pilih model pencahayaan, yang mendefinisikan tingkat cahaya lingkungan global dan lokasi efektif dari sudut pandang (untuk keperluan perhitungan pencahayaan).
- Tentukan sifat-sifat material untuk objek-objek dalam adegan.

Contoh program dibawah akan menyelesaikan tugas ini. Program tersebut akan menampilkan sebuah bola diterangi satu sumber cahaya.

program Menggambar sebuah Lit Sphere: light.c

```
#include <GL/gl.h>
#include <GL/glu.h>
#include <GL/glut.h>
```

**void** init(**void**)

```
{
    GLfloat mat_specular[] = { 1.0, 1.0, 1.0, 1.0 };
    GLfloat mat_shininess[] = { 50.0 };
    GLfloat light_position[] = { 1.0, 1.0, 1.0, 0.0 };
    glClearColor (0.0, 0.0, 0.0, 0.0);
    glShadeModel (GL_SMOOTH);

    glMaterialfv(GL_FRONT, GL_SPECULAR, mat_specular);
    glMaterialfv(GL_FRONT, GL_SHININESS, mat_shininess);
    glLightfv(GL_LIGHT0, GL_POSITION, light_position);

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

**void** display(**void**)

```
{
    glClear (GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    glutSolidSphere (1.0, 20, 16);
    glFlush ();
}
```

**void** reshape (**int** w, **int** h)

```
{
    glViewport (0, 0, (GLsizei) w, (GLsizei) h);
    glMatrixMode (GL_PROJECTION);
    glLoadIdentity();
    if (w <= h)
        glOrtho (-1.5, 1.5, -1.5*(GLfloat)h/(GLfloat)w,
                1.5*(GLfloat)h/(GLfloat)w, -10.0, 10.0);
    else
```



# FAKULTAS TEKNIK

UNIVERSITAS NEGERI YOGYAKARTA

## LAB. SHEET PRAKTIKUM GRAFIKA KOMPUTER 1

No. : ST/EKA/PTI223/05

Revisi : 02

Senin 010409

Hal. 5 dari 8 hal.

Sem.: Genap

### Pencapaian pada OpenGL

4 x 50 menit

```
glOrtho (-1.5*(GLfloat)w/(GLfloat)h,  
         1.5*(GLfloat)w/(GLfloat)h, -1.5, 1.5, -10.0, 10.0);  
glMatrixMode(GL_MODELVIEW);  
glLoadIdentity();  
}
```

```
int main(int argc, char** argv)  
{  
    glutInit(&argc, argv);  
    glutInitDisplayMode (GLUT_SINGLE | GLUT_RGB | GLUT_DEPTH);  
    glutInitWindowSize (500, 500);  
    glutInitWindowPosition (100, 100);  
    glutCreateWindow (argv[0]);  
    init ();  
    glutDisplayFunc(display);  
    glutReshapeFunc(reshape);  
    glutMainLoop();  
    return 0;  
}
```

### Program 2 Objek sphere

```
#include <GL/glut.h>
```

```
GLfloat light_diffuse[] = {1.0, 1.0, 0.0, 1.0};  
GLfloat light_position[] = {1.0, 1.0, 1.0, 0.20};  
GLUquadricObj *qobj;
```

```
void display(void)  
{  
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);  
    glCallList(1);           /* tampilan list render sphere */  
    glutSwapBuffers();  
}
```

```
void gfxinit(void)  
{  
    qobj = gluNewQuadric();           // glut Library  
    gluQuadricDrawStyle(qobj, GLU_FILL);  
    glNewList(1, GL_COMPILE);       /* membuat tampilan sphere */  
                                     //gluSphere(radius, slices, stacks);
```

```
gluSphere(qobj, 1.0, 20, 20);
```

```
gluCylinder(qobj, 1, 1, 3, 20, 20);
```

```
gluDisk(qobj, 0.5, 1, 20, 20);
```

```
gluPartialDisk(qobj, 0.5, 1, 20, 20, 45, 270);
```

Lakukan percobaan dengan memilih salah satu dari sintak disamping kiri. Lakukan perubahan nilai

```
glEndList();  
glLightfv(GL_LIGHT0, GL_DIFFUSE, light_diffuse);  
glLightfv(GL_LIGHT0, GL_POSITION, light_position);
```



# FAKULTAS TEKNIK

UNIVERSITAS NEGERI YOGYAKARTA

## LAB. SHEET PRAKTIKUM GRAFIKA KOMPUTER 1

No. : ST/EKA/PTI223/05

Revisi : 02

Senin 010409

Hal. 6 dari 8 hal.

Sem.: Genap

**Pencapaian pada OpenGL**

4 x 50 menit

```
glEnable(GL_LIGHTING);
glEnable(GL_LIGHT0);
glEnable(GL_DEPTH_TEST);
glMatrixMode(GL_PROJECTION);
// gluPerspective(field of view in degree, aspect ratio, Z near, Z far );
gluPerspective(40.0, 1.0, 1.0, 10.0);
glMatrixMode(GL_MODELVIEW);
gluLookAt(0.0, 0.0, 5.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.);
glTranslatef(0.0, 0.0, -1.0);
}
```

```
int main(int argc, char **argv)
{
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB | GLUT_DEPTH);
    glutCreateWindow("sphere");
    glutDisplayFunc(display);
    gfxinit();
    glutCreateWindow("pencapaian");
    glutDisplayFunc(display);
    gfxinit();
    glutMainLoop();
    return 0;
}
```

### Program 3 Objek Kubus

```
#include <GL/glut.h>
```

```
GLfloat light_diffuse[] = {1.0, 0.0, 0.0, 1.0}; /* warna merah terang. */
GLfloat light_position[] = {1.0, 1.0, 1.0, 0.0};
GLfloat n[6][3] = { /* secara normal kubus ada 6 muka. */
    {-1.0, 0.0, 0.0}, {0.0, 1.0, 0.0}, {1.0, 0.0, 0.0}, {0.0, -1.0, 0.0}, {0.0, 0.0, 1.0}, {0.0, 0.0, -1.0} };
GLint faces[6][4] = {
    {0, 1, 2, 3}, {3, 2, 6, 7}, {7, 6, 5, 4}, {4, 5, 1, 0}, {5, 6, 2, 1}, {7, 4, 0, 3} };
GLfloat v[8][3];
```

```
void drawBox(void)
```

```
{
    int i;
    for (i = 0; i < 6; i++) {
        glBegin(GL_QUADS);
        glNormal3fv(&n[i][0]);
        glVertex3fv(&v[faces[i][0]][0]);
        glVertex3fv(&v[faces[i][1]][0]);
        glVertex3fv(&v[faces[i][2]][0]);
        glVertex3fv(&v[faces[i][3]][0]);
        glEnd();
    }
}
```



# FAKULTAS TEKNIK

UNIVERSITAS NEGERI YOGYAKARTA

## LAB. SHEET PRAKTIKUM GRAFIKA KOMPUTER 1

No. : ST/EKA/PTI223/05

Revisi : 02

Senin 010409

Hal. 7 dari 8 hal.

Sem.: Genap

### Pencahayaan pada OpenGL

4 x 50 menit

```
}
```

```
void display(void)
```

```
{  
  glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);  
  drawBox();  
  glutSwapBuffers();  
}
```

```
void init(void)
```

```
{  
  /* pengaturan data vertek kubus. */  
  v[0][0] = v[1][0] = v[2][0] = v[3][0] = -1;  
  v[4][0] = v[5][0] = v[6][0] = v[7][0] = 1;  
  v[0][1] = v[1][1] = v[4][1] = v[5][1] = -1;  
  v[2][1] = v[3][1] = v[6][1] = v[7][1] = 1;  
  v[0][2] = v[3][2] = v[4][2] = v[7][2] = 1;  
  v[1][2] = v[2][2] = v[5][2] = v[6][2] = -1;  
  
  /* meng-Enable-kan pencahayaan OpenGL single. */  
  glLightfv(GL_LIGHT0, GL_DIFFUSE, light_diffuse);  
  glLightfv(GL_LIGHT0, GL_POSITION, light_position);  
  glEnable(GL_LIGHT0);  
  glEnable(GL_LIGHTING);  
  glEnable(GL_DEPTH_TEST);  
  
  /* mengatur penglihatan kubus. */  
  glMatrixMode(GL_PROJECTION);  
  /* sudut pandangan, aspek ratio, kedekatan Z, jauh Z */  
  gluPerspective(40.0, 1.0, 1.0, 10.0);  
  glMatrixMode(GL_MODELVIEW);  
  gluLookAt(0.0, 0.0, 5.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.);  
  
  /* mengatur posisi kubus */  
  glTranslatef(0.0, 0.0, -1.0);  
  glRotatef(60, 1.0, 0.0, 0.0);  
  glRotatef(-20, 0.0, 0.0, 1.0);  
}
```

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

```
{  
  glutInit(&argc, argv);  
  glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB | GLUT_DEPTH);  
  glutCreateWindow("kubus warna merah 3D dengan pencahayaan");  
  glutDisplayFunc(display);  
  init();  
  glutMainLoop();  
  return 0;  
}
```



# FAKULTAS TEKNIK

UNIVERSITAS NEGERI YOGYAKARTA

## LAB. SHEET PRAKTIKUM GRAFIKA KOMPUTER 1

No. : ST/EKA/PTI223/05

Revisi : 02

Senin 010409

Hal. 8 dari 8 hal.

Sem.: Genap

**Pencerahan pada OpenGL**

4 x 50 menit

}

### C. Tugas

1. Cobalah program diatas
2. Lakukan perubahan sintak atau nilai yang ada dalam program diatas
3. Masukkan hasil pengamatan dan masukan hasilnya yang disertai gambar hasil compile dalam tabel pengamatan yang disesuaikan dengan pengamatan anda. (dengan MS Office)
4. Buatlah gambar donat dilengkapi dengan pencerahan satu sumber
5. Buatlah program gambar cone dilengkapi dengan pencerahan satu sumber