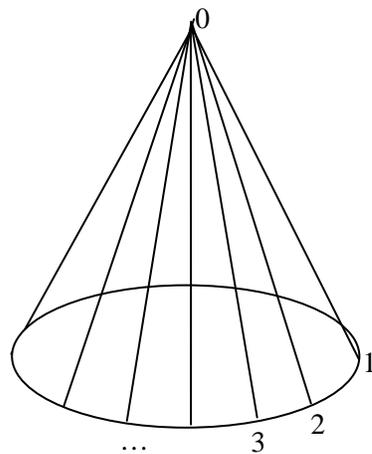


Menggambar Obyek Kerucut dan Silinder

Menggambar Kerucut

Kerucut bentuknya seperti ice-cream, yaitu menghubungkan lingkaran sebagai luasan bagian bawah dan sebuah titik di bagian atas. Bentuk kerucut dapat dilihat pada gambar 7.7 berikut ini.



Gambar 7.7. Kerucut

Untuk menggambar sebuah kerucut, maka terlebih dahulu ditentukan jumlah titik yang akan membangun lingkaran di bawah, misalkan jumlah titik pada lingkaran adalah n , maka kerucut dapat digambarkan seperti gambar 6.7 di atas. Pembentukan obyek 3 dimensi dilakukan dengan:

- mendefinisikan titik-titik pada obyek yang terdiri dari jumlah titik, dan koordinat masing-masing titik.
- Mendefinisikan face-face pada obyek yang terdiri dari jumlah face dan struktur pada masing-masing face.

Dengan demikian jumlah titik pada kerucut tersebut adalah $n+1$, dengan n untuk membentuk lingkaran dan 1 di atas. Dan jumlah face adalah $n+1$, dengan n untuk membentuk bagian sisi dan 1 bagian alas (lingkaran).

Posisi titik-titik pada kerucut dengan tinggi h dan jari-jari r dapat didefinisikan dengan:

Titik 0: (Bagian puncak) $(0,h,0)$

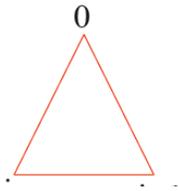
Titik 1 sampai dengan n :

$$x_i = r * \cos(i.a)$$

$$y_i = r * \sin(i.a) \text{ dimana } a \text{ adalah unit sudut: } a = \frac{\pi}{n}$$

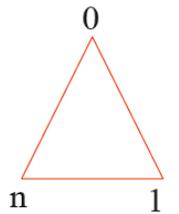
$$z_i = 0$$

Face pada kerucut didefinisikan dengan:

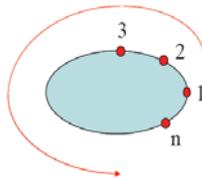


Untuk face 0 sampai dengan n-1:

Face i: terdiri dari 3 titik yaitu: 0, i, i+1



Untuk face ke n: terdiri dari 3 titik yaitu 0, n, 1



Face terakhir berupa alas lingkaran mempunyai n titik yaitu: {1, 2, 3, ..., n}

Gambar 7.8. Proses pembuatan kerucut

Sehingga fungsi untuk membuat kerucut adalah sebagai berikut:

```
void createCone(object3D_t &kerucut, int n, float r, float h){
    float a=6.28/n;
    int i;
    kerucut.pnt[0].x=0;
    kerucut.pnt[0].y=h;
    kerucut.pnt[0].z=0;
    for(i=1;i<=n;i++){
        kerucut.pnt[i].x=r*cos(i*a);
        kerucut.pnt[i].y=0;
        kerucut.pnt[i].z=r*sin(i*a);
    }
}
```

```

    for(i=0;i<n;i++){
        kerucut.fc[i].NumberOfVertices=3;
        kerucut.fc[i].pnt[0]=0;
        kerucut.fc[i].pnt[1]=i+2;
        kerucut.fc[i].pnt[2]=i+1;
        if(i==(n-1)) kerucut.fc[i].pnt[1]=1;
    }
    kerucut.fc[n].NumberOfVertices=n;
    for(i=0;i<n;i++) kerucut.fc[n].pnt[i]=i+1;
    kerucut.NumberofVertices=n+1;
    kerucut.NumberofFaces=n+1;
}

```

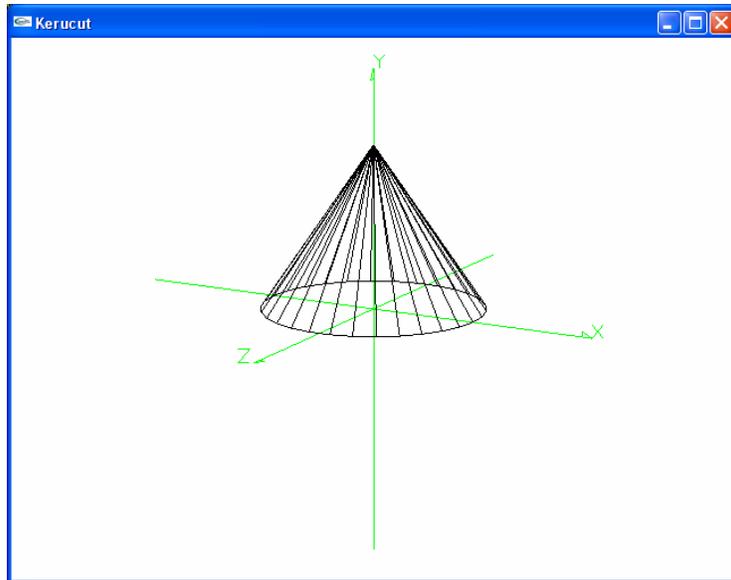
Untuk menggambar kerucut dengan tinggi 150 dan jari-jari 80, pada userdraw() dapat ditulis dengan:

```

void userdraw(void)
{
    matrix3D_t tilting=rotationXMTX(0.25)*rotationYMTX(-0.5);
    setColor(0,1,0);
    drawAxes(tilting);
    object3D_t kerucut;
    makeCone(kerucut,20,80,150);
    setColor(1,1,1);
    draw3D(kerucut,tilting);
}

```

Hasil dari program di atas adalah:



Gambar 7.9. Contoh hasil kerucut

Untuk menggambar kerucut dituliskan:

createCone(obyek3D, n, r, h)

Dimana:

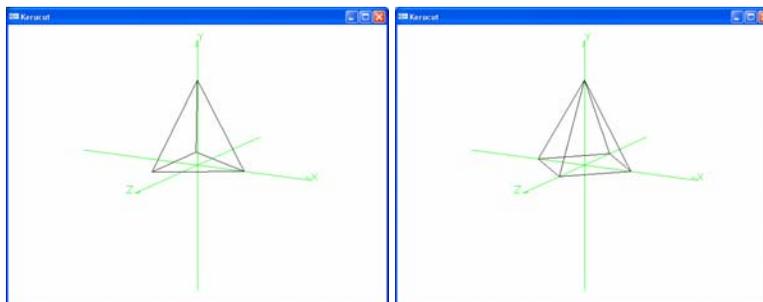
Obyek3D menyatakan nama obyek 3D

n adalah jumlah titik pembentuk lingkaran alas

r adalah jari-jari alas

h adalah tinggi kerucut

Pada gambar 7.9 kerucut dibentuk dengan $n=20$. Berikut ini contoh kerucut dengan $n=3$, $n=4$, $n=6$ dan $n=8$. Pada $n=3$ menghasilkan limas segitiga dan pada $n=4$ menghasilkan limas segi empat.





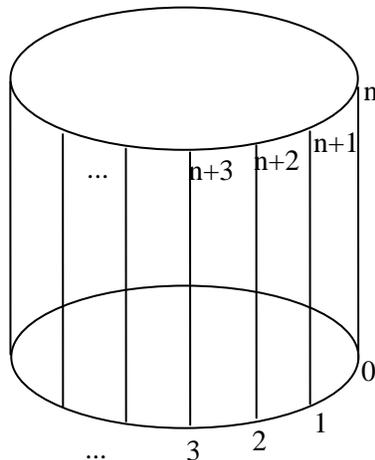
Gambar 7.10. Contoh kerucut dengan $n=3,4,6$ dan 8

Terlihat bahwa semakin banyak n akan menghasilkan lingkaran yang semakin sempurna. Hal ini juga menunjukkan bahwa limas dapat dihasilkan dari kerucut.

Menggambar Silinder

Silinder bentuknya seperti tong, yaitu mempunyai 2 lingkaran sebagai luasan bagian atas dan bawah. Bentuk silinder dapat dilihat pada gambar 7.10 berikut ini.

Teknik pembuatan silinder hampir sama dengan teknik pembuatan kerucut, hanya saja pada masing-masing face pembentuk bagian samping terdiri dari 4 titik, berbeda dengan kerucut yang hanya mempunyai 3 titik. Fungsi pembuatan silinder adalah sebagai berikut:



Gambar 7.11. Silinder

Fungsi untuk membuat silinder dengan jari-jari r dan tinggi h adalah sebagai berikut:

```

void createCylinder(object3D_t &silinder, int n, float r, float h){
    float a=6.28/n;
    int i;
    for(i=0;i<n;i++){
        silinder.pnt[i].x=r*cos(i*a);
        silinder.pnt[i].y=0;
        silinder.pnt[i].z=r*sin(i*a);
        silinder.pnt[n+i].x=r*cos(i*a);
        silinder.pnt[n+i].y=h;
        silinder.pnt[n+i].z=r*sin(i*a);
    }
    silinder.NumberofVertices=2*n;
    for(i=0;i<n;i++){
        silinder.fc[i].NumberofVertices=4;
        silinder.fc[i].pnt[0]=i;
        silinder.fc[i].pnt[1]=n+i;
        silinder.fc[i].pnt[2]=n+i+1;
        silinder.fc[i].pnt[3]=i+1;
        if(i==(n-1)){
            silinder.fc[i].pnt[2]=n;
            silinder.fc[i].pnt[3]=0;
        }
    }
    silinder.fc[n].NumberofVertices=n;
    for(i=0;i<n;i++) silinder.fc[n].pnt[i]=i;
    silinder.fc[n+1].NumberofVertices=n;
    for(i=0;i<n;i++) silinder.fc[n+1].pnt[i]=2*n-1-i;
    silinder.NumberofFaces=n+2;
}

```

Untuk menggambar silinder dengan tinggi 150 dan jari-jari 80, pada userdraw() dapat ditulis dengan:

```
void userdraw(void)
```

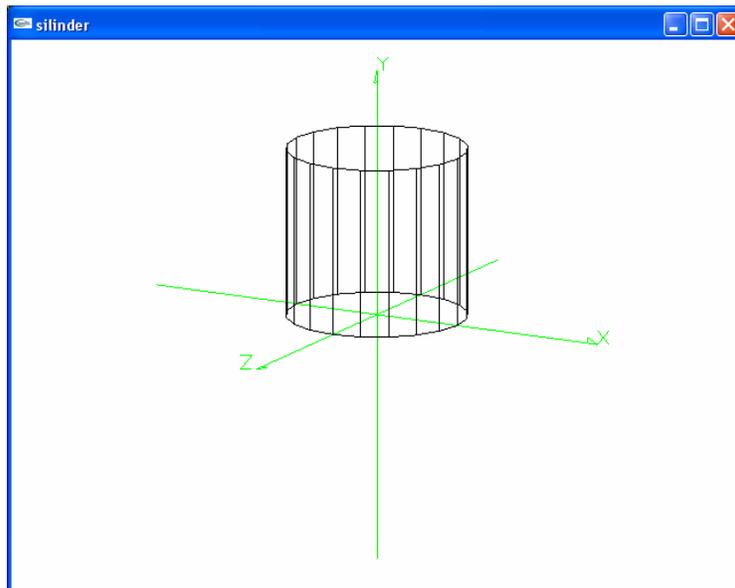
```

{
matrix3D_t tilting=rotationXMTX(0.25)*rotationYMTX(-0.5);
setColor(0,1,0);
drawAxes(tilting);
}

```

```
object3D_t silinder;  
createCylinder(silinder,20,80,150);  
setColor(1,1,1);  
draw3D(silinder,tilting);  
}
```

Hasil dari program di atas adalah:



Gambar 7.12. Contoh hasil silinder

Untuk menggambar silinder dituliskan:

createCylinder(obyek3D, n, r, h)

Dimana:

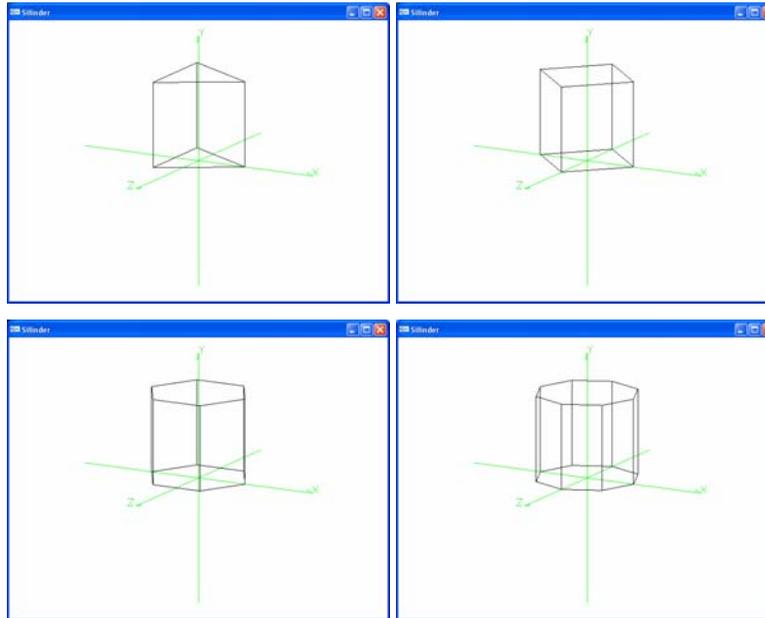
Obyek3D menyatakan nama obyek 3D

n adalah jumlah titik pembentuk lingkaran alas

r adalah jari-jari alas

h adalah tinggi silinder

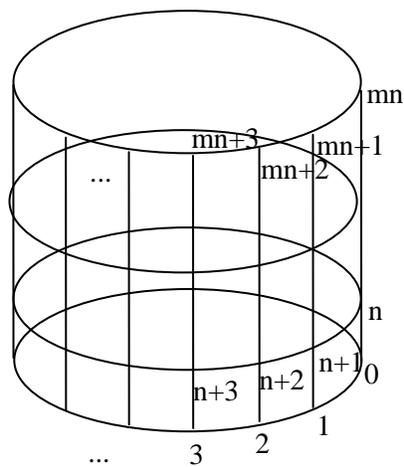
Pada gambar 7.12 silinder dibentuk dengan $n=20$. Berikut ini contoh kerucut dengan $n=3$, $n=4$, $n=6$ dan $n=8$. Pada $n=3$ menghasilkan prisma, pada $n=4$ menghasilkan balok dan pada $n=8$ akan menghasilkan hexagon.



Gambar 7.13. Contoh silinder dengan $n=3,4,6$ dan 8

Menggambar Silinder Bertumpuk

Silinder bertumpuk adalah obyek 3 dimensi yang berupa tumpukan silinder. Teknik pembuatannya hampir sama dengan teknik pembuatan silinder, tetapi obyek ini terdiri dari banyak silinder yang dijadikan satu.



Gambar 7.18. Silinder bertumpuk

Silinder bertumpuk dengan m silinder dan n titik pembentuk lingkaran mempunyai r[] sebanyak m+1 dan h[] sebanyak m. Pembuatan silinder bertumpuk dengan m silinder dan n buah titik untuk menghasilkan lingkaran adalah:

(1) Pembuatan titik

Jumlah titik = m×n

Koordinat titik-titik:

$$x_i = r_i \cdot \cos(i\theta)$$

$$y_i = b \text{ dengan } b = \sum_{j=1}^i h_j$$

$$z_i = r_i \cdot \sin(i\theta)$$

(2) Pembuatan Face

Jumlah Face = m×n+2

m×n buah face untuk bagian tepi, 2 buah face untuk lingkaran atas dan bawah.

Face bagian tepi:

Jumlah titik = 4

$$p_0 = i.n + j$$

$$p_1 = (i+1).n + j$$

$$p_2 = (i+1).n + j + 1$$

$$p_3 = i.n + j + 1$$

Face bagian alas:

Jumlah titik = n

$$p_i = i$$

Face bagian atas:

Jumlah titik = n

$$p_i = (m+1).n - i - 1$$

Sehingga fungsi untuk membuat silinder bertumpuk adalah sebagai berikut:

```
void createCylinderN (object3D_t &silinder,int m,int n,float r[],float
h[]){
    float a=6.26/n;
    float b=0;
    int i,j;
    silinder.NumberofVertices=(m+1)*n;
    for(i=0;i<=m;i++){
```

```

        if(i>0) b=b+h[i-1];
        for(j=0;j<n;j++){
            silinder.pnt[i*n+j].x=r[i]*cos(j*a);
            silinder.pnt[i*n+j].y=b;
            silinder.pnt[i*n+j].z=r[i]*sin(j*a);
        }
    }
    silinder.NumberofFaces=m*n+2;
    for(i=0;i<m;i++){
        for(j=0;j<n;j++){
            silinder.fc[i*n+j].NumberofVertices=4;
            silinder.fc[i*n+j].pnt[0]=i*n+j;
            silinder.fc[i*n+j].pnt[1]=(i+1)*n+j;
            silinder.fc[i*n+j].pnt[2]=(i+1)*n+j+1;
            silinder.fc[i*n+j].pnt[3]=i*n+j+1;
            if(j==(n-1)){
                silinder.fc[i*n+j].pnt[2]=i*n+j+1;
                silinder.fc[i*n+j].pnt[3]=(i-1)*n+j+1;
            }
        }
    }
    silinder.fc[m*n].NumberofVertices=n;
    for(i=0;i<n;i++) silinder.fc[m*n].pnt[i]=i;
    silinder.fc[m*n+1].NumberofVertices=n;
    for(i=0;i<n;i++)
        silinder.fc[m*n+1].pnt[i]=(m+1)*n-1-i;
}

```

Contoh hasil dari silinder bertumpuk dengan $m=3$, $n=20$, $r=\{60,60,60,60\}$ dan $h=\{40,40,40\}$ adalah:

```

void userdraw(void){
    matrix3D_t tilting=rotationXMTX(0.25)*rotationYMTX(-0.5);
    setColor(0,1,0);
    drawAxes(tilting);
    float r[4]={60,60,60,60};
    float h[3]={40,40,40};
    object3D_t silinder;

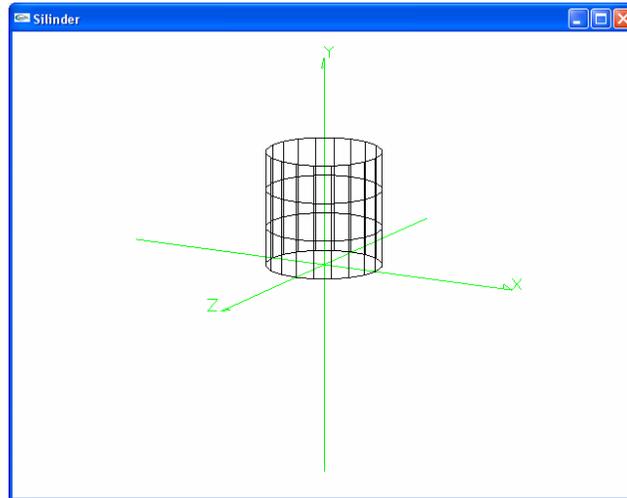
```

```

createCylinderN(silinder, 3, 20, r, h);
setColor(0, 0, 0);
draw3D(silinder, tilting);
}

```

Hasilnya adalah sebagai berikut:



Gambar 7.19. Contoh silinder bertumpuk

Untuk menggambar silinder bertumpuk dituliskan:

createCylinderN(obyek3D, m, n, r, h)

Dimana:

Obyek3D menyatakan nama obyek 3D

m adalah jumlah silinder

n adalah jumlah titik pembentuk lingkaran alas

r adalah jari-jari pada setiap lingkaran, berupa array

sejumlah m+1

h adalah tinggi pada setiap lingkaran, berupa array

sejumlah m

Sebagai contoh, silinder bertumpuk dengan 4 buah silinder, $n=20$, $r=\{50,60,60,20,20\}$, dan $h=\{10,70,50,30\}$ adalah:

```

void userdraw(void){
matrix3D_t tilting=rotationXMTX(0.25)*rotationYMTX(-0.5);
setColor(0,1,0);
drawAxes(tilting);
float r[5]={50,60,60,20,20};
float h[4]={10,70,50,30};
}

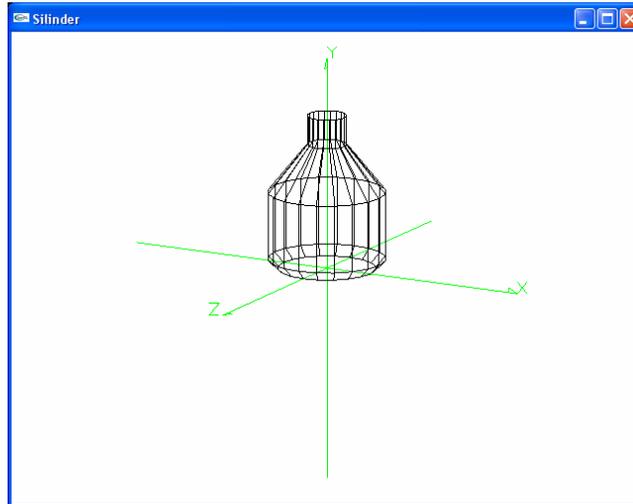
```

```

object3D_t silinder;
createCylinderN(silinder,4,20,r,h);
setColor(0,0,0);
draw3D(silinder,tilting);
}

```

Hasilnya adalah:



Gambar 7.20. Contoh hasil silinder bertumpuk

Model silinder bertumpuk ini sangat baik digunakan untuk menghasilkan obyek-obyek grafik 3 dimensi yang cukup menarik. Beberapa contoh hasil dari model silinder bertumpuk dengan mengatur nilai r dan h yang berupa array.

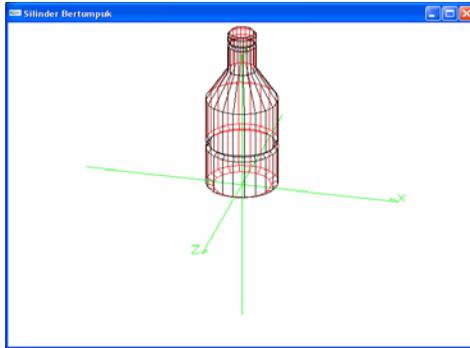
- Membuat botol

```

void userdraw(void){
matrix3D_t tilting=rotationXMTX(0.5)*rotationYMTX(-0.25);
setColor(0,1,0);
drawAxes(tilting);
object3D_t o;
float r[12]={40,50,50,45,50,50,20,20,17,20,20,17};
float h[11]={5,60,5,5,70,50,40,4,4,10,4};
createCylinderN(o,11,20,r,h);
draw3D(o,tilting);
}

```

Hasilnya adalah:

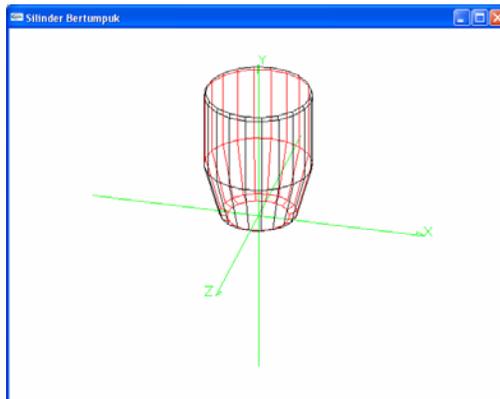


Gambar 7.21. Contoh gambar botol

- Membuat gelas

```
void userdraw(void){
matrix3D_t tilting=rotationXMTX(0.5)*rotationYMTX(-0.25);
setColor(0,1,0);
drawAxes(tilting);
object3D_t o;
float r[5]={40,50,70,70,68};
float h[4]={5,70,100,5};
createCylinderN(o,4,20,r,h);
draw3D(o,tilting);
}
```

Hasilnya adalah:



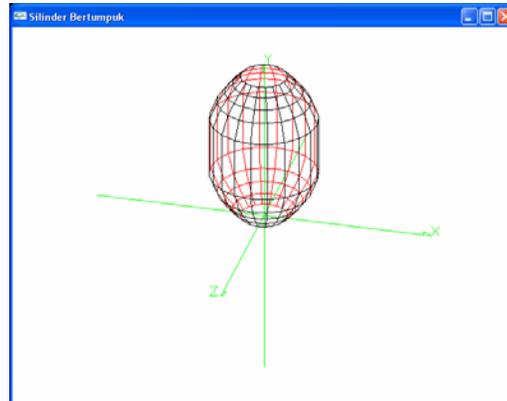
Gambar 7.22. Contoh gambar gelas

- Membuat lampu

```
void userdraw(void){
matrix3D_t tilting=rotationXMTX(0.5)*rotationYMTX(-0.25);
setColor(0,1,0);
drawAxes(tilting);
```

```
object3D_t o;  
float r[10]={30,40,50,60,70,70,60,50,40,30};  
float h[9]={10,12,14,25,80,25,14,12,10};  
createCylinderN(o,9,20,r,h);  
draw3D(o,tilting);  
}
```

Hasilnya adalah:



Gambar 7.23. Contoh gambar lampion