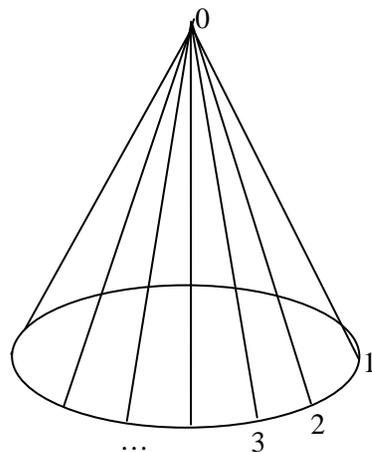


Menggambar Obyek Kerucut dan Silinder

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) $(0, h, 0)$

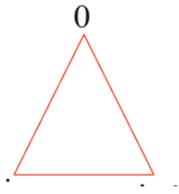
Titik 1 sampai dengan n :

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

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

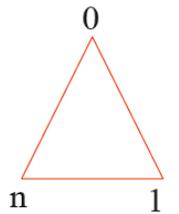
$$z_i = 0$$

Face pada kerucut didefinisikan dengan:

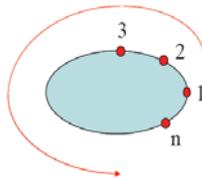


Untuk face 0 sampai dengan n-1:

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;
}

```

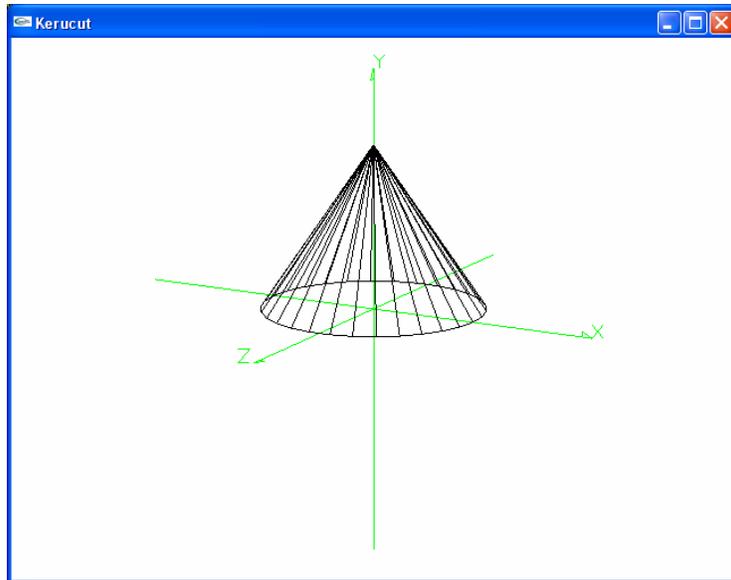
Untuk menggambar kerucut dengan tinggi 150 dan jari-jari 80, pada userdraw() dapat ditulis dengan:

```

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:

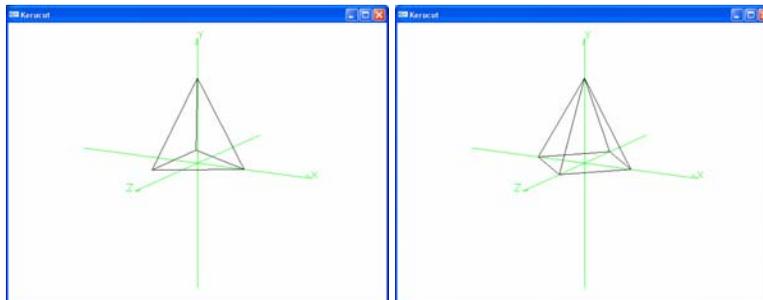
Obyek3D menyatakan nama obyek 3D

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.





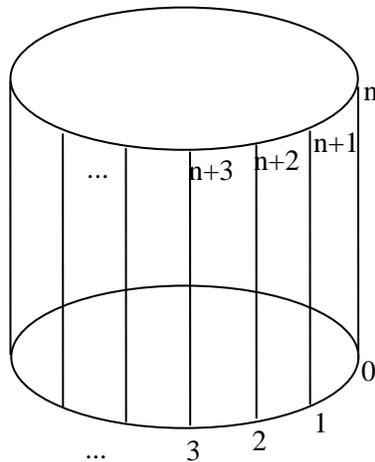
Gambar 7.10. Contoh kerucut dengan $n=3,4,6$ dan 8

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)
```

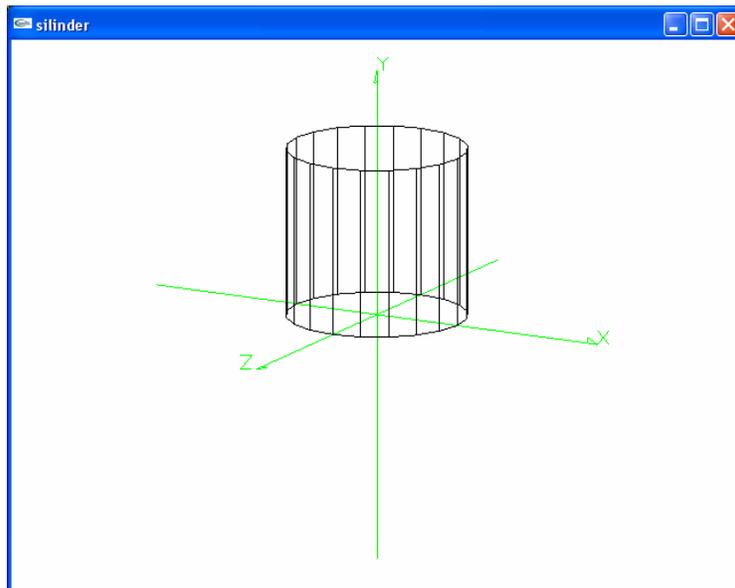
```

{
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:

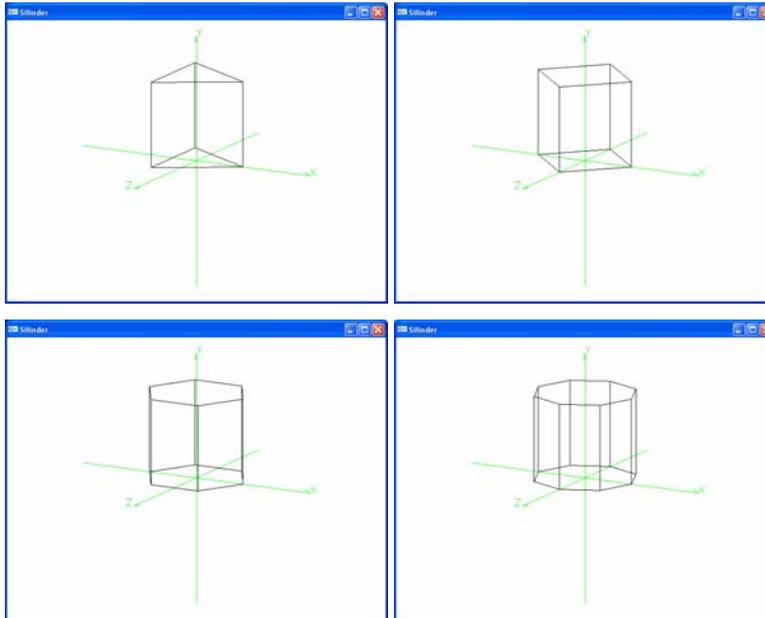
Obyek3D menyatakan nama obyek 3D

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