



Beberapa Trik Pembuatan Grafik 2D Menggunakan OpenGL

Achmad Basuki, Nana R, Setiawardhana
Surabaya, April 2008



Materi

- Membuat Langit Bergradiasi
- Membuat Gunung
- Membuat Bayangan Gunung Di Air
- Membuat Bintang Berkelip
- Membuat Efek Hujan
- Membuat Grafik Berjalan
- Membuat Bola-Bola Air

Membuat Langit Bergradiasi





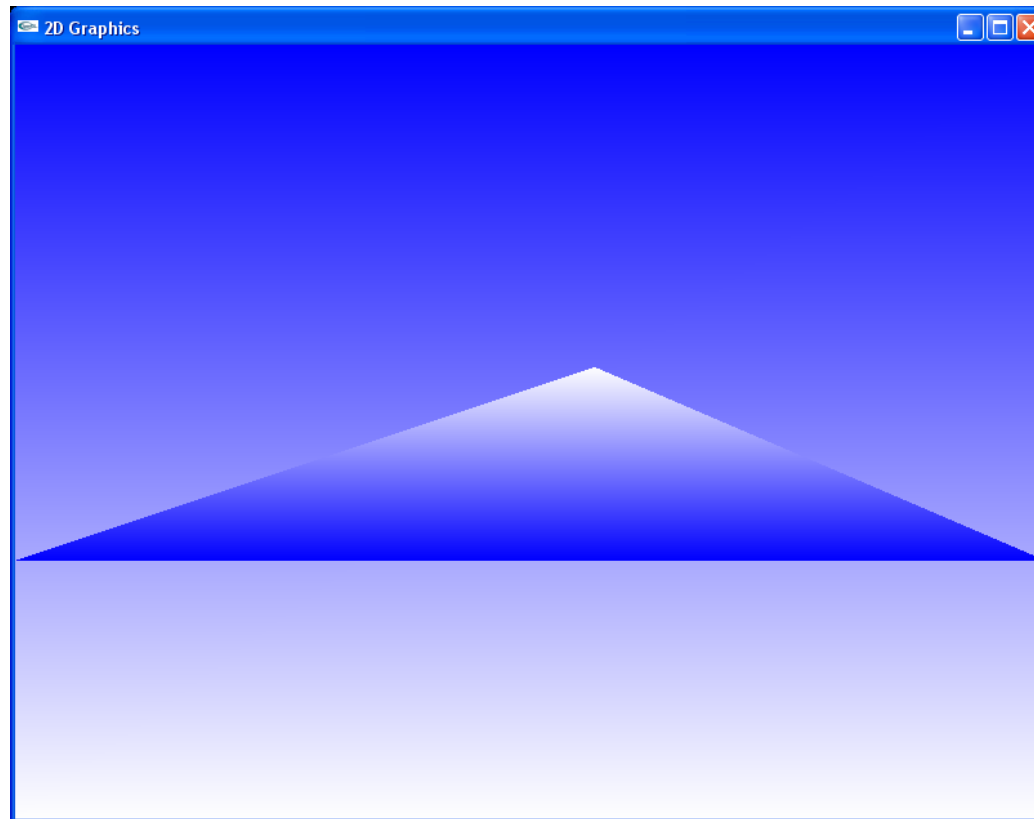
Membuat Langit Bergradiasi

```
// Membuat kotak yang memenuhi ruang koordinat
// Koordinat yang digunakan (-400,400,-300,300)
point2D_t kotak[4]={{-400,-300},{400,-300},{400,300},{-400,300}};

// Membuat warna yang sesuai dengan posisi titik di kotak
color_t col[4]={{1,1,1},{1,1,1},{0,0,1},{0,0,1}};

// Gambar dengan gradatePolygon
gradatePolygon(kotak,col,4);
```

Membuat Gunung

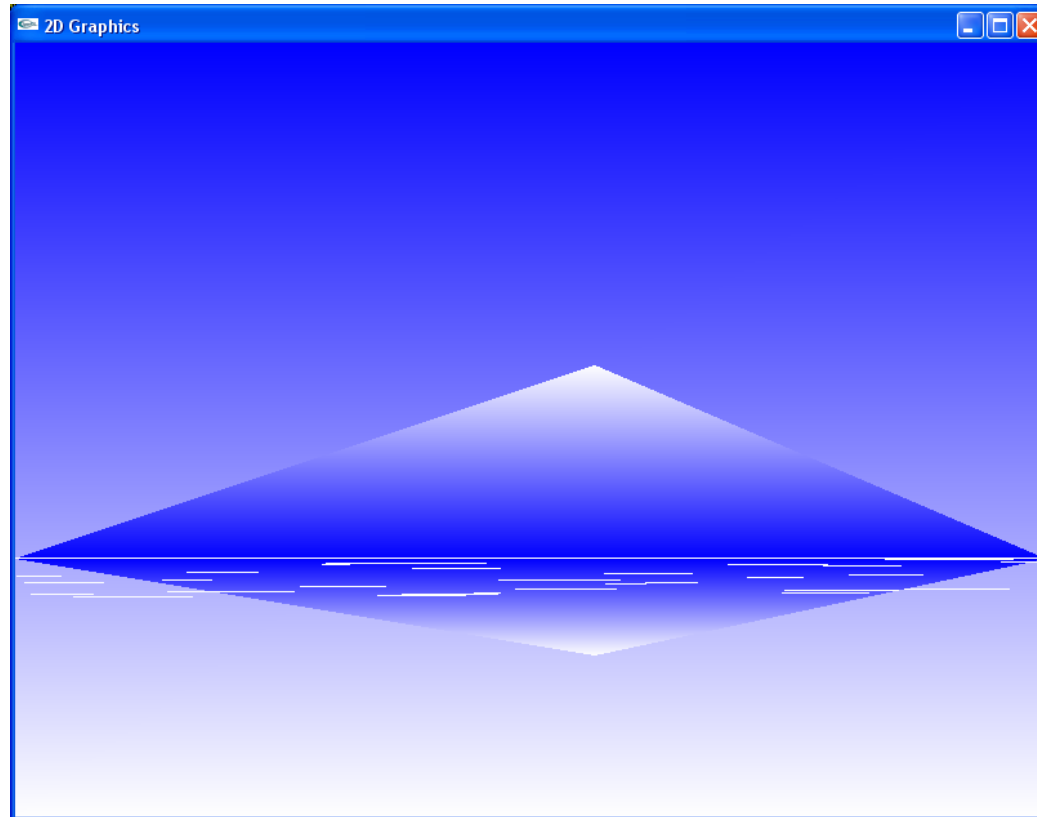




Membuat Gunung

```
point2D_t gunung[3]={{-400,-100},{400,-100},{50,50}};  
color_t colGunung[3]={{0,0,1},{0,0,1},{1,1,1}};  
grayscalePolygon(gunung,colGunung,3);
```

Membuat Bayangan Gunung Di Air





Membuat Bayangan Gunung Di Air

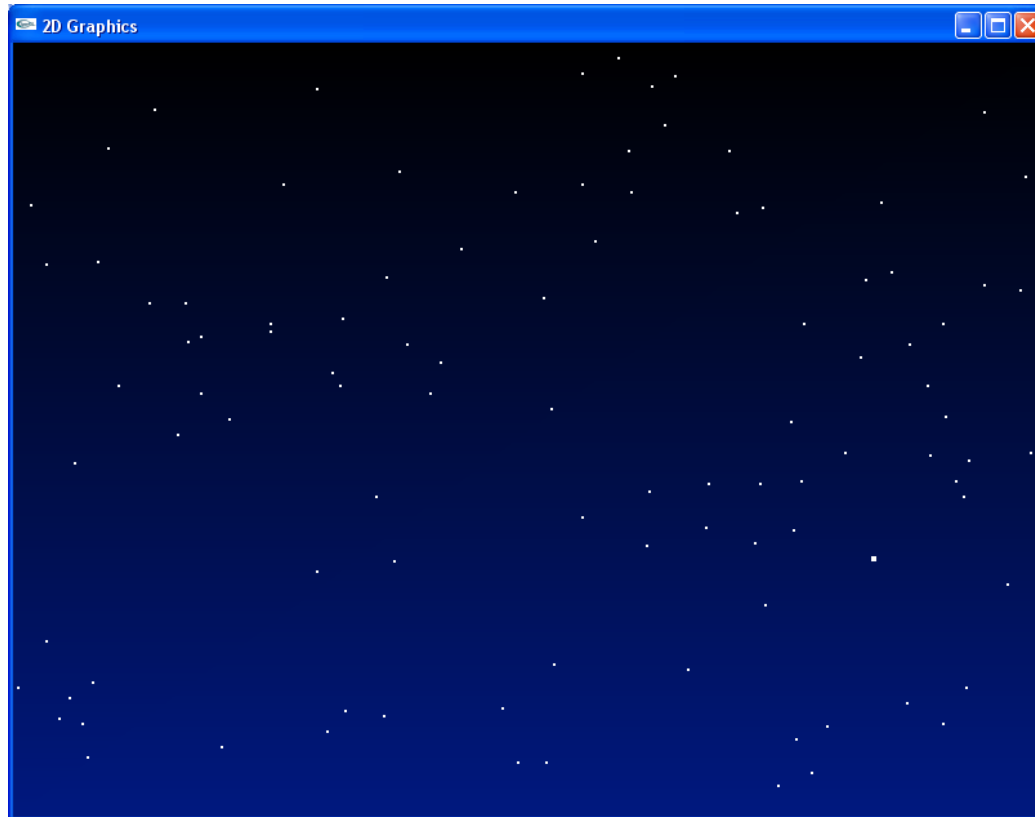
```
point2D_t bgunung[3]={{-400,-101},{400,-101},{50,-175}};  
color_t colbGunung[3]={{0,0,1},{0,0,1},{1,1,1}};  
gradatePolygon(bgunung,colbGunung,3);
```




Membuat Efek Air

```
float xp,yp,dx;  
setColor(1,1,1);  
for(int i=1;i<30;i++){  
    xp=800*(float)rand()/RAND_MAX-400;  
    yp=30*(float)rand()/RAND_MAX-130;  
    dx=20+30*(float)rand()/RAND_MAX;  
    drawLine(xp,yp,xp+dx,yp);  
}
```

Membuat Bintang Berkelip





Membuat Bintang Berkedip

- Membuat langit bergradiasi dari hitam ke biru
- Mengacak posisi bintang satu kali saja ketika program pertama dijalankan
- Menggambar bintang
- Membuat bintang berkedip



Membuat Bintang Berkedip:

Membuat langit bergradiasi hitam ke biru

```
point2D_t langit[4]={{0,0},{400,0},{400,300},{0,300}};  
color_t wLangit[4]={{0,0.1,0.5},{0,0.1,0.5},{0,0,0},{0,0,0}};  
gradatePolygon(langit,wLangit,4);
```

Membuat Bintang Berkedip:

Mengacak posisi bintang saat program pertama kali dijalankan (tick=0)

```
static int tick=0;
static point2D_t bintang[100];
int i;
if(tick==0){
    for(i=0;i<100;i++){
        bintang[i].x=rand()%400;
        bintang[i].y=rand()%300;
    }
}
```



Membuat Bintang Berkedip: Menggambar Bintang

```
setColor(1,1,1);  
glPointSize(2);  
for(i=0;i<100;i++)  
    drawDot(bintang[i].x,bintang[i].y);
```



Membuat Bintang Berkedip: Membuat Bintang Berkedip

```
int k=rand()%100;  
glPointSize(4);  
drawDot(bintang[k].x,bintang[k].y);
```

Membuat Efek Hujan





Membuat Efek Hujan

- Membuat Langit
- Membuat Tanah
- Mengacak posisi awal garis-garis hujan
- Menggerakkan garis-garis hujan
- Menggambar hujan

Membuat Efek Hujan

Definisi sistem koordinat: gluOrtho2D(0,400,0,300);

// Membuat langit

```
point2D_t langit[4]={{0,0},{400,0},{400,300},{0,300}};  
color_t wLangit[4]={{1,1,1},{1,1,1},{0,0.1,0.5},{0,0.1,0.5}};  
grayscalePolygon(langit,wLangit,4);
```

// Membuat tanah

```
point2D_t  
tanah[5]={{0,0},{400,0},{400,50},{300,40},{0,35}};  
color_t wTanah[5]={{0.5,0.2,0},{0.5,0.2,0},{1,0.9,0.5},  
    {1,0.9,0.5}, {0.7,0.5,0.1}};  
grayscalePolygon(tanah,wTanah,5);
```

Membuat Efek Hujan

Membuat Langit dan Tanah

// Membuat langit

```
point2D_t langit[4]={{0,0},{400,0},{400,300},{0,300}};  
color_t wLangit[4]={{1,1,1},{1,1,1},{0,0.1,0.5},{0,0.1,0.5}};  
grayscalePolygon(langit,wLangit,4);
```

// Membuat tanah

```
point2D_t  
tanah[5]={{0,0},{400,0},{400,50},{300,40},{0,35}};  
color_t wTanah[5]={{0.5,0.2,0},{0.5,0.2,0},{1,0.9,0.5},  
    {1,0.9,0.5}, {0.7,0.5,0.1}};  
grayscalePolygon(tanah,wTanah,5);
```



Membuat Efek Hujan

Mengacak Posisi Awal Garis-Garis Hujan

```
// Mengacak posisi awal garis-garis hujan
static int tick=0;
static point2D_t hujan[200];
int i;
if(tick==0){
    for(i=0;i<200;i++){
        hujan[i].x=rand()%400;
        hujan[i].y=rand()%300;
    }
}
```

Membuat Efek Hujan

Menggerakkan Garis-Garis Hujan

```
// Menggerakkan posisi hujan
for(i=0;i<200;i++){
    hujan[i].x++;
    hujan[i].y--;
    // Jika garis hujan di luar window
    // diganti dengan yang baru
    if((hujan[i].x>400) || (hujan[i].y<0)){
        hujan[i].x=rand()%400;
        hujan[i].y=rand()%300;
    }
}
```



Membuat Efek Hujan

Menggambar Garis-Garis Hujan

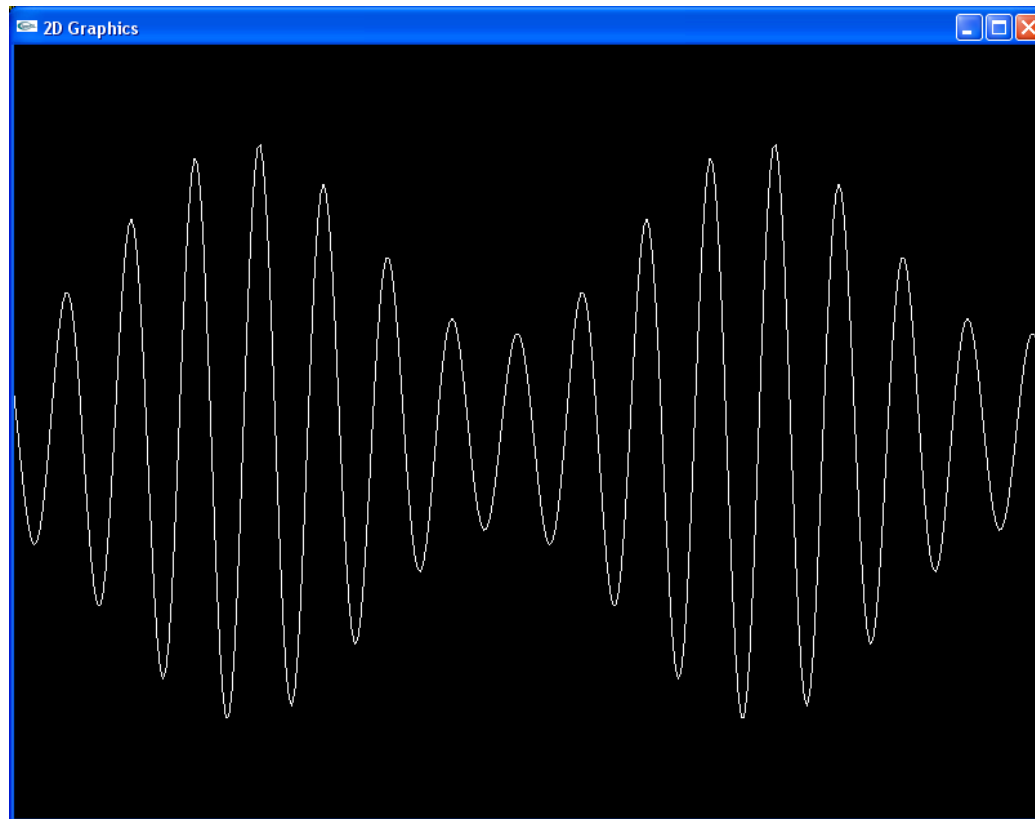
```
// Menggambar hujan
```

```
setColor(1,1,1);
```

```
for(i=0;i<200;i++)
```

```
    drawLine(hujan[i].x,hujan[i].y,hujan[i].x+5,hujan[i].y-5);
```

Membuat Grafik Berjalan



$$F(x) = \sin(16*x)*(1+0.5*\sin(2*x))$$

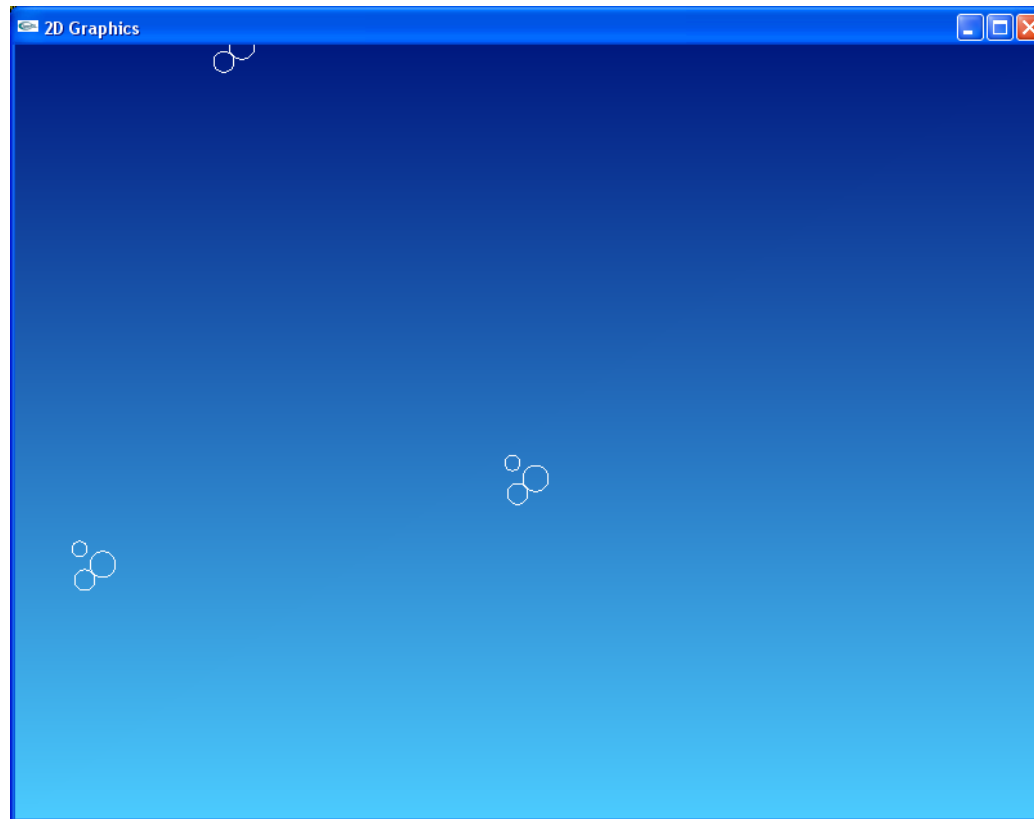


Membuat Grafik Berjalan

Sistem Koordinat: `gluOrtho2D(0,360,-2,2);`

```
static int a=0;
point2D_t p[360];
for(int i=0;i<360;i++){
    float s=(i+a)/57.3;
    p[i].x=i;
    p[i].y=sin(16*s)*(1+0.5*sin(2*s));
}
setColor(1,1,1);
drawPolyline(p,360);
a++;
```


Membuat Gelembung Air





Membuat Gelembung Air

Sistem Koordinat: gluOrtho2D(0,400,0,300);

```
// Membuat Latar Belakang
```

```
point2D_t latar[4]={{0,0},{400,0},{400,300},{0,300}};
```

```
color_t wLatar[4]={{0.3,0.8,1},{0.3,0.8,1},{0,0.1,0.5},{0,0.1,0.5}};
```

```
gradatePolygon(latar,wLatar,4);
```

Membuat Gelembung Air

Mengacak Posisi Awal Gelembung Air

```
// Mengacak posisi Gelembung Air
static point2D_t gelAir[3];
static int tick=0;
static float d[3];
int i;
if(tick==0){
    for(i=0;i<3;i++){
        gelAir[i].x=rand()%400;
        gelAir[i].y=10;
        d[i]=(float)(rand()%100)/100+0.5;
    }
}
tick++;
```

Membuat Gelembung Air

Menjalankan Gelembung Air

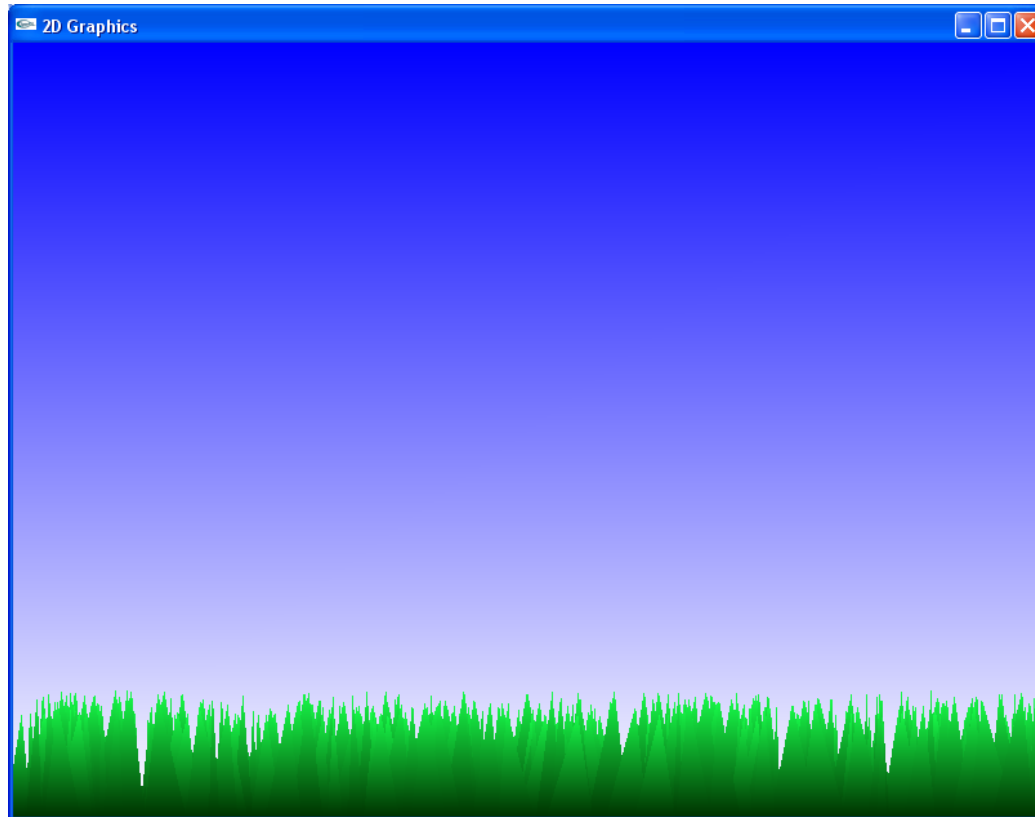
```
// Menjalankan Gelembung air
for(i=0;i<3;i++){
    gelAir[i].y+=d[i];
    if(gelAir[i].y>300){
        gelAir[i].x=rand()%400;
        gelAir[i].y=10;
        d[i]=(float)(rand()%100)/100+0.5;
    }
}
```

Membuat Gelembung Air

Menggambar Gelembung Air

```
// Menggambar gelembung air dengan 3 buah lingkaran
point2D_t p[90],p0;
setColor(1,1,1);
for(i=0;i<3;i++){
    createCircle(p,gelAir[i],4,90);
    drawPolygon(p,90);
    p0.x=gelAir[i].x+7; p0.y=gelAir[i].y+6;
    createCircle(p,p0,5,90);
    drawPolygon(p,90);
    p0.x=gelAir[i].x-2; p0.y=gelAir[i].y+12;
    createCircle(p,p0,3,90);
    drawPolygon(p,90);
}
```

Membuat Rumput Bergoyang





Membuat Rumput Bergoyang

Definisi Obyek 2D untuk Rumput

```
typedef struct {  
    point2D_t p,p0;  
    float d,s;  
} obyek_t;
```



Membuat Rumput Bergoyang

Sistem Koordinat: gluOrtho2D(0,400,0,300);

```
// Membuat langit
```

```
point2D_t langit[4]={{0,0},{400,0},{400,300},{0,300}};  
color_t wLangit[4]={{1,1,1},{1,1,1},{0,0,1},{0,0,1}};  
gradatePolygon(langit,wLangit,4);
```


Membuat Rumput Bergoyang

Mengacak Posisi Awal Rumput

```
// Mengacak Posisi awal rumput
static obyek_t rumput[300];
static int tick=0;
int i;
if(tick==0){
    for(i=0;i<300;i++){
        rumput[i].p.x=rand()%400;
        rumput[i].p.y=40+rand()%10;
        rumput[i].d=rand()%8+2;
        rumput[i].s=float(rand()%100+1)/1000;
        rumput[i].p0=rumput[i].p;
    }
}
tick++;
```

Membuat Rumput Bergoyang

Menggoyangkan Rumput

```
// Menggoyangkan rumput
float xa,xb;
for(i=0;i<300;i++){
    rumput[i].p.x+=rumput[i].s;
    xa=rumput[i].p0.x-0.5*rumput[i].d;
    xb=rumput[i].p0.x+0.5*rumput[i].d;
    if((rumput[i].p.x<xa) || (rumput[i].p.x>xb))
        rumput[i].s=-rumput[i].s;
}
```

Membuat Rumput Bergoyang

Menggambar Rumput

```
// Menggambar rumput
point2D_t p[3];
color_t cp[3]={{0.1,1,0.3},{0,0.2,0},{0,0.2,0}};
for(i=0;i<300;i++){
    p[0].x=rumput[i].p.x;
    p[0].y=rumput[i].p.y;
    p[1].x=rumput[i].p0.x-rumput[i].d;
    p[1].y=0;
    p[2].x=rumput[i].p0.x+rumput[i].d;
    p[2].y=0;
    gradatePolygon(p,cp,3);
}
```