

# TRANSFORMASI 2 DIMENSI

Achmad Basuki  
Nana Ramadijanti

# Materi

- Struktur titik dan vektor
- Perubahan struktur titik ke vektor
- Perubahan struktur vektor ke titik
- Translasi
- Scalling
- Rotasi
- Perkalian Matrik
- Komposisi Transformasi

# Struktur Titik dan Vektor

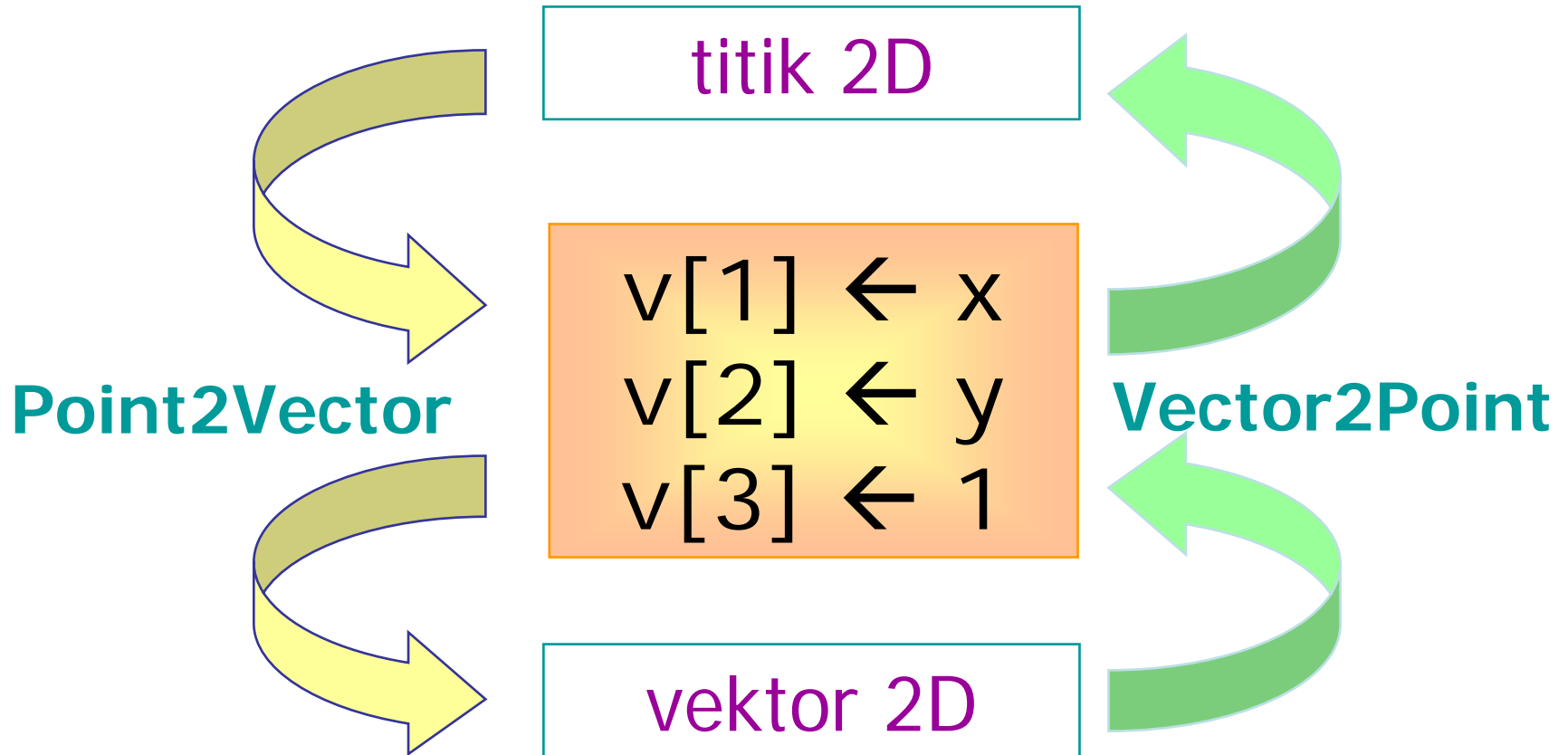
Struktur data  
dari titik 2D

```
typedef struct {  
    float x,y;  
} point2D_t;
```

Struktur data  
dari vektor 2D

```
typedef struct {  
    float v[3];  
} vector2D_t;
```

# Perubahan Titik dan Vektor



# Point2Vector

Fungsi ini digunakan untuk memindahkan tipe data titik menjadi tipe data vektor. Hal ini sangat berguna untuk operasional matrik yang digunakan dalam melakukan transformasi dan pengolahan matrik pada grafika komputer.

```
vector2D_t point2vector(point2D_t pnt)
{
    vector2D_t vec;
    vec.v[1]=pnt.x;
    vec.v[2]=pnt.y;
    vec.v[3]=1.;
}
```

# Vector2Point

Fungsi ini digunakan untuk memindahkan tipe data vektor menjadi tipe data titik. Hal ini sangat berguna untuk penyajian grafis setelah proses pengolahan matrik yang dikenakan pada obyek 2D.

```
point2D_t vector2point(vector2D_t vec)
{
    point2D_t pnt;
    pnt.x=vec.v[1];
    pnt.y=vec.v[2];
}
```

# Transformasi 2D

- **Translasi**
- **Scaling**
- **Rotasi**

# Matrik Transformasi 2D

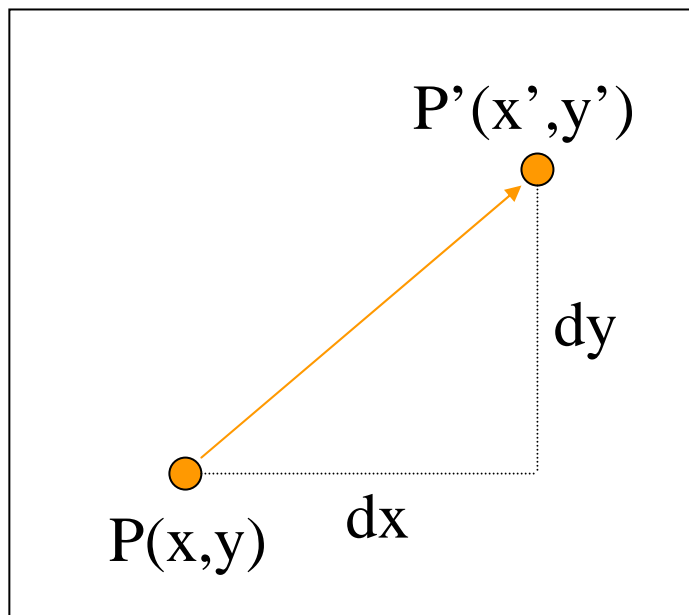
- Matrik transformasi adalah matrik yang membuat sebuah obyek mengalami perubahan baik berupa perubahan posisi, maupun perubahan ukuran.
- Matrik transformasi 2D dinyatakan dalam ukuran 3x3, dimana kolom ke-3 digunakan untuk menyediakan tempat untuk proses translasi.

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$



# Translasi

Translasi adalah perpindahan obyek dari titik P ke titik P' secara linier.



$$\begin{aligned}x' &= x + dx \\y' &= y + dy\end{aligned}$$

Model Matrik:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} dx \\ dy \end{bmatrix}$$

# Matrik Transformasi dari Translasi 2D

Proses translasi dengan menggunakan definisi vektor 2D dapat dituliskan dengan

$$\begin{bmatrix} \hat{v}_1 \\ \hat{v}_2 \\ \hat{v}_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & d_x \\ 0 & 1 & d_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & d_x \\ 0 & 1 & d_y \\ 0 & 0 & 1 \end{bmatrix}$$

Matrik  
Transformasi  
dari  
Translasi

# Implementasi Matrik Transformasi Untuk Translasi

```
matrix2D_t translationMTX(float dx, float dy)
{
    matrix2D_t trans = createIdentity();
    trans.m[0][2] = dx;
    trans.m[1][2] = dy;
    return trans;
}
```

Fungsi untuk membuat matrik identitas

# Matrik Identitas

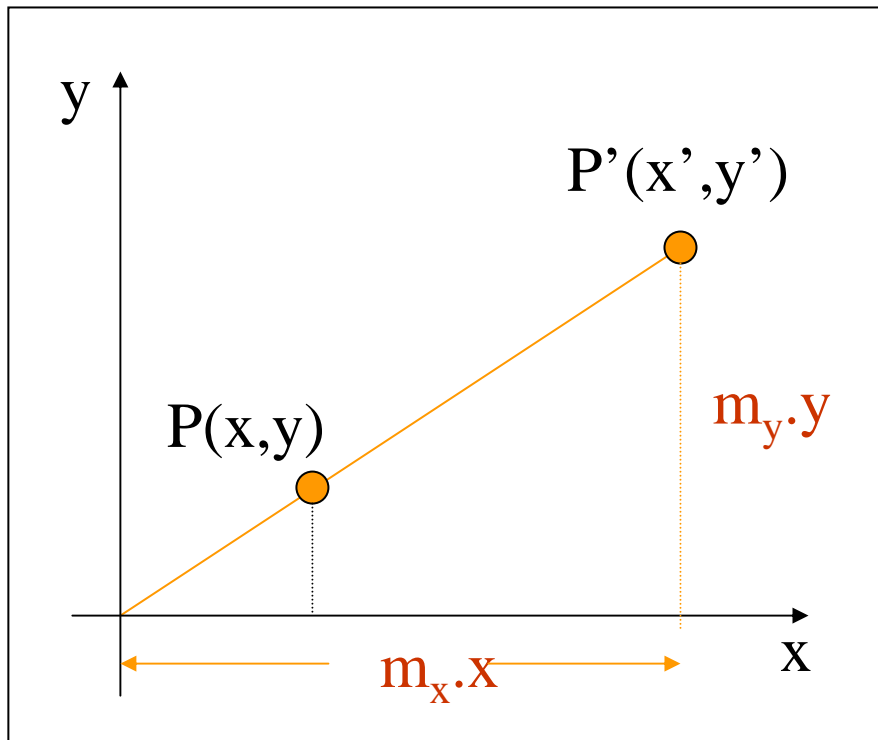
Matrik identitas adalah matrik yang nilai diagonal utamanya sama dengan satu dan lainnya nol.

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

```
matrix2D_t createIdentity(void) {  
    matrix2D_t u;  
    int i,j;  
    for (i=0;i<3;i++) {  
        for(j=0;j<3;j++) u.m[i][j]=0.;  
        u.m[i][i]=1.;  
    }  
    return u;  
}
```

# Scaling

Scaling  $m$  adalah perpindahan obyek dari titik  $P$  ke titik  $P'$ , dimana jarak titik  $P'$  adalah  $m$  kali titik  $P$



$$x' = m_x x$$
$$y' = m_y y$$

# Matrik Transformasi dari Scaling 2D

Proses scaling dengan menggunakan definisi vektor 2D dapat dituliskan dengan

$$\begin{bmatrix} \hat{v}_1 \\ \hat{v}_2 \\ \hat{v}_3 \end{bmatrix} = \begin{bmatrix} m_x & 0 & 0 \\ 0 & m_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}$$

$$\begin{bmatrix} m_x & 0 & 0 \\ 0 & m_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

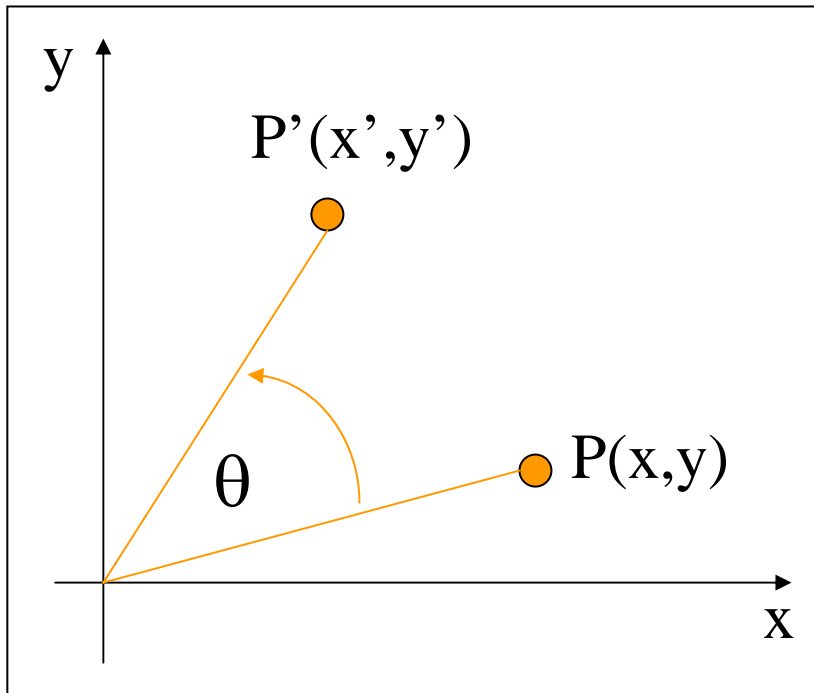
Matrik  
Transformasi  
dari Scaling

# Implementasi Matrik Tranformasi Untuk Scaling

```
matrix2D_t scalingMTX(float mx, float my)
{
    matrix2D_t scale = createIdentity();
    scale.m[0][0] = mx;
    scale.m[1][1] = my;
    return scale;
}
```

# Rotasi

Rotasi adalah perpindahan obyek dari titik P ke titik P', yang berupa pemindahan berputar sebesar sudut  $\theta$



$$\begin{aligned}x' &= x \cos(\theta) - y \sin(\theta) \\y' &= x \sin(\theta) + y \cos(\theta)\end{aligned}$$



# Matrik Transformasi dari Rotasi

Proses Rotasi  
dengan  
menggunakan  
definisi vektor2D  
dapat dituliskan  
dengan

$$\begin{bmatrix} \hat{v}_1 \\ \hat{v}_2 \\ \hat{v}_3 \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}$$

$$\begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Matrik  
Transformasi  
dari Rotasi

# Implementasi Matrik Tranformasi Untuk Rotasi

```
matrix2D_t rotationMTX(float theta)
{
    matrix2D_t rotate=createIdentity();
    float cs=cos(theta);
    float sn=sin(theta);
    rotate.m[0][0]=cs; rotate.m[0][1]=-sn;
    rotate.m[1][0]=sn; rotate.m[1][1]=cs;
    return rotate;
}
```

# Perkalian Matrik

- Perkalian matrik dengan matrik menghasilkan matrik
- Perkalian matrik dengan vektor menghasilkan vektor

Perkalian matrik ini digunakan untuk operasional transformasi dari obyek 2D dan untuk komposisi (menggabungkan) transformasi

# Perkalian Matrik dengan Matrik

Perkalian matrik a dan matrik b menghasilkan matrik c yang dirumuskan dengan

$$c_{ij} = \sum_{k=0}^2 a_{ik} b_{kj}$$

dimana i dan j bernilai 0 s/d 2

# Implementasi Perkalian Matrik dengan Matrik

```
matrix2D_t operator * (matrix2D_t a, matrix2D_t b)
{
    matrix2D_t c; //c=a*b
    int i,j,k;
    for (i=0;i<3;i++) for (j=0;j<3;j++) {
        c.m[i][j]=0;
        for (k=0;k<3;k++)
            c.m[i][j] += a.m[i][k]*b.m[k][j];
    }
    return c;
}
```

# Perkalian Matrik dengan Vektor

Perkalian matrik a dan vektor b menghasilkan vektor c yang dirumuskan dengan

$$c_i = \sum_{k=0}^2 a_{ik} b_k$$

dimana i bernilai 0 s/d 2

# Implementasi Perkalian Matrik dengan Vektor

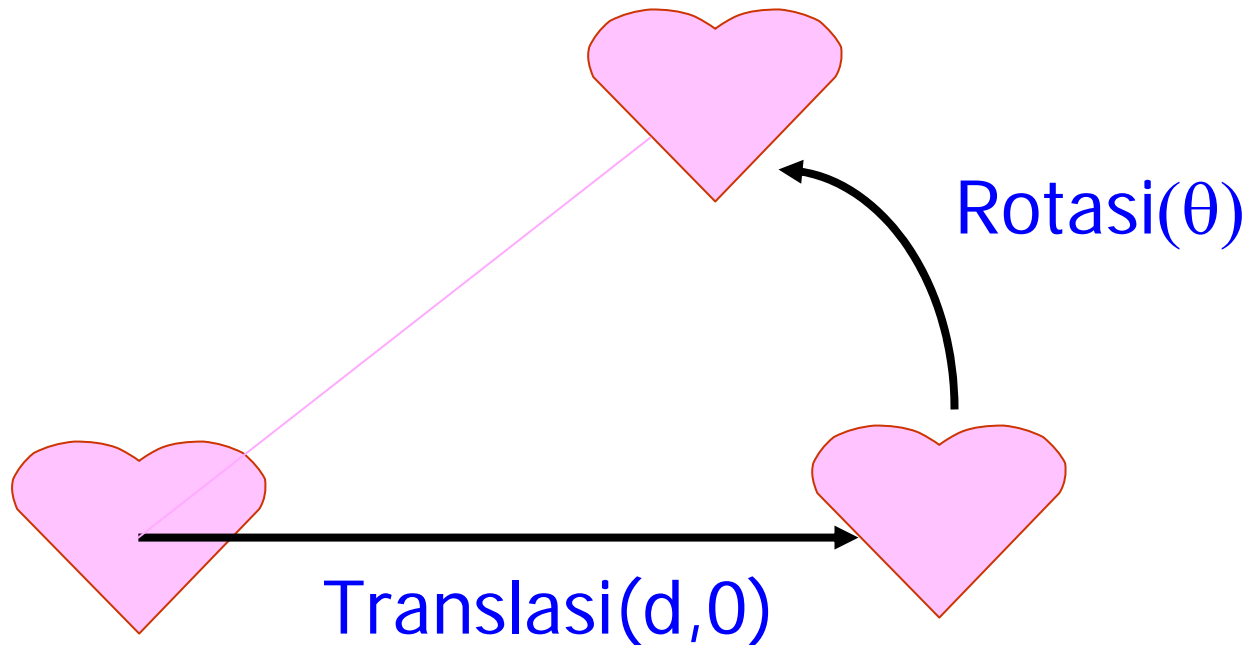
```
vector2D_t operator * (matrix2D_t a, vector2D_t b)
{
    vector2D_t c;//c=a*b
    int i,j;
    for (i=0;i<3;i++) {
        c.v[i]=0;
        for (j=0;j<3;j++)
            c.v[i] += a.m[i][j]*b.v[j];
    }
    return c;
}
```

# Komposisi Transformasi

- Komposisi transformasi adalah menggabungkan beberapa transformasi, sehingga dapat menghasilkan bentuk transformasi yang lebih kompleks
- Komposisi transformasi dapat dilakukan dengan mengalikan matrik-matrik transformasi



# Contoh Komposisi Transformasi

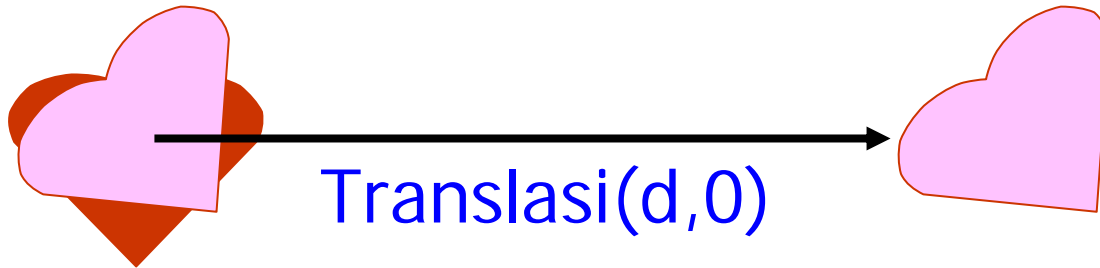


Komposisi transformasi dinyatakan dengan :

$$\text{Rotasi}(\theta) \cdot \text{Translasi}(d,0)$$

# Contoh Komposisi Transformasi

Rotasi( $\theta$ )

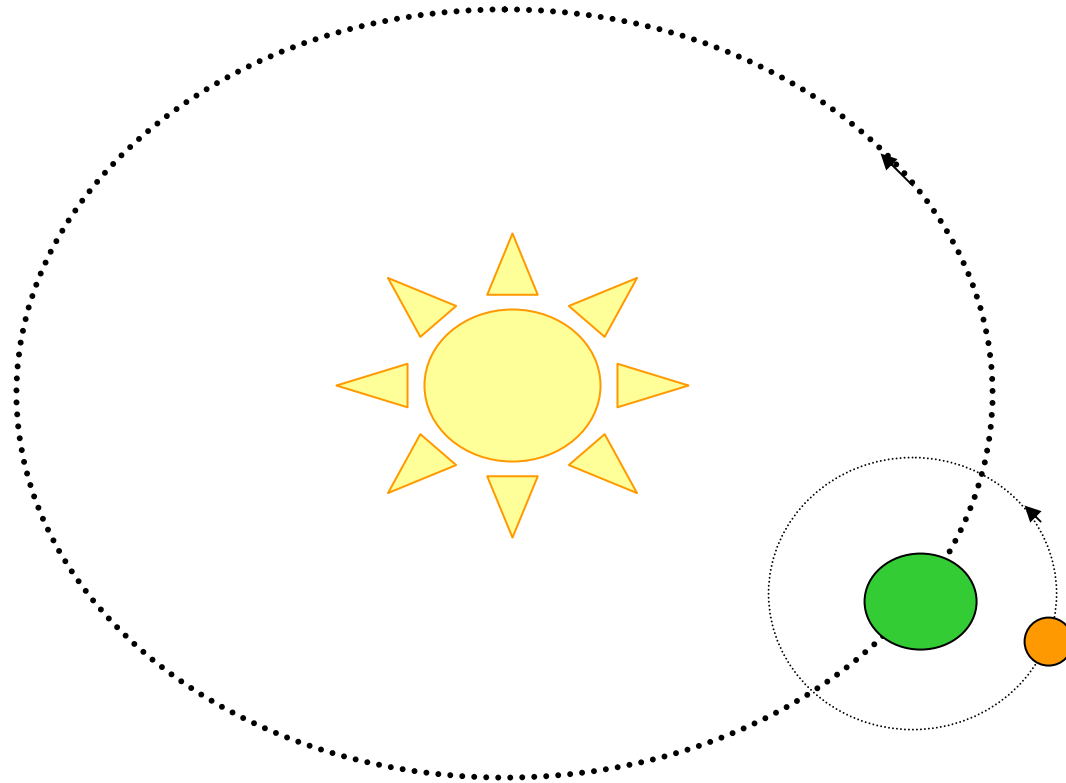


Komposisi transformasi dinyatakan dengan :

Translasi( $d,0$ ). Rotasi( $\theta$ )

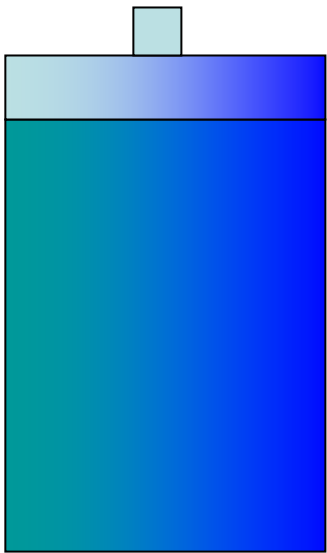
# Tugas 1 Grafika Komputer

## Menggambarkan Orbit Bumi dan Bulan

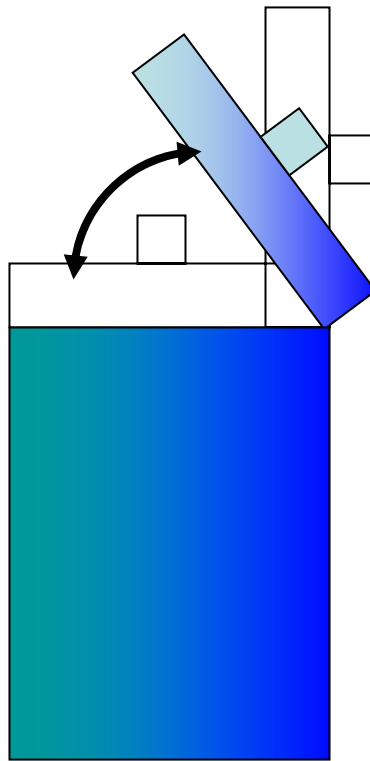


# Tugas 2 Grafika Komputer

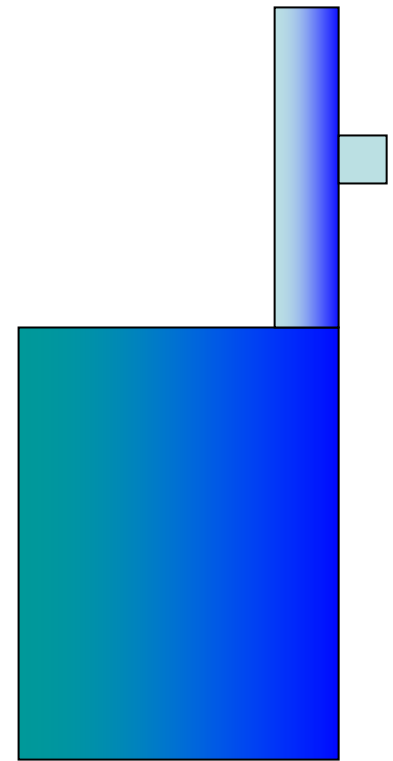
## Menggambar Gerakan Tutup Gelas Terbuka



Gelas tertutup



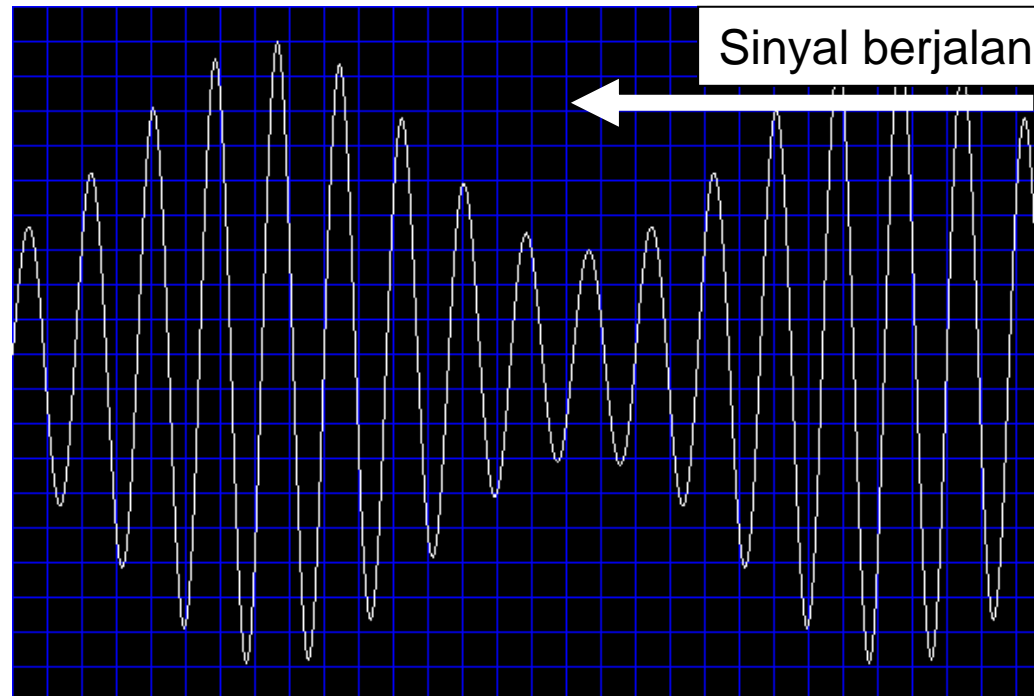
Gerakan terbuka  
dan tertutup



Gelas terbuka

# Tugas 3 Grafika Komputer

## Menggambar Osiloskop



Fungsi gelombang AM adalah:  $y = \sin(2\pi f_c t) \cdot [1 + m \cdot \sin(2\pi f_i t)]$

$f_c$  = frekwensi pembawa  
 $f_i$  = frekwensi informasi  
 $m$  = konstanta modulasi