BAB 7
SKALA, ROTASI DAN TRANSLASI

7.1 Pendahuluan

Pada sebuah game skala rotasi dan translasi sangat dibutuhkan ketika membuat sebuah game yang menggunakan grafik vektor dan grafik bitmap. Penggunaannya seperti pemindahan sebuah object, memper-besar atau memper kecil ukuran peta ataupun merotasi arah dari object seperti pesawat, dll.

7.2 Tujuan

Setelah mempelajari modul ini peserta diharapkan dapat:

- Memahami fungsi-fungsi dari Skala,Rotasi dan Translasi
- Mengimplementasikan Skala pada sebuah Object
- Menerapkan Rotasi dan Translasi pada Object

7.3 Translasi

Translasi adalah bentuk yang paling sederhana dari transformasi, yaitu metode pemindahan koordinat asal ke koordinat baru. Matriks transformasi untuk translasi dengan vektor translasi adalah :

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

Maka koordinat baru dari hasil translasi adalah :

x’ = x + tx

y’ = y + ty

Sebaiknya tidak memikirkan rumus diatas karena java telah memberikan kemudahan kepada programer untuk melakukan proses translasi, yaitu dengan

menggunakan method yang telah disediakan oleh Graphics2D seperti dibawah ini:

```
g2d.translate(nilai1, nilai2);
g2d.fill(shape);
```

Dengan demikian maka object yang akan digambar dengan fill atau draw akan dipidah dari koordinat lama ke koordinat yang baru, diamana nilai1 dan nilai2 akan ditambahkan ke posisix dan posisiy yang pertama kali di definisikan dan menghasilkan posisi baru hasil penjumlahan tadi yaitu (posisix+nilai1) dan (posisiy+nilai2).

7.4 Rotasi

Rotasi adalah bentuk transformasi dengan dengan memutar sebuah objek, ada dua macam cara atau rumus untuk merotasi object di java yaitu berpatokan pada titik acuan (0,0) atau (p,q) dibawah ini adalah rumus yang digunakan untuk operasi tersebut.

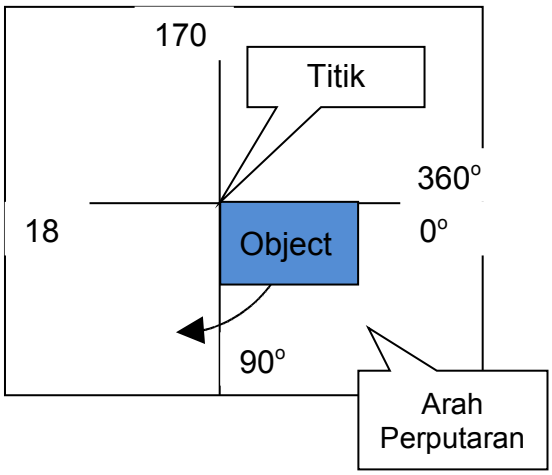
Rotasi dengan sudut putar θ dengan titik acuan (0,0) adalah:

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

Maka, koordinat baru hasil rotasi adalah :

x' = x cosθ – y sinθ

y' = x sin y + y cosθ



Gambar 7.1 Arah Perputaran Object

Penggunaan sin,cos yang sangat membungungkan, di java telah menyediakan method rotate dari class Graphics2D dengan pendefinisian sebagai berikut:

```
g2d.rotate(Math.toRadians(1));
g2d.fill(shape);
```

Radian dan sudut adalah dua cara untuk menghitung sesuatu yang sama. Pada sebuah lingkaran terdapat 360 derajat dan juga 2*PI radian dalam sebuah lingkaran. Sebagai contoh:

Jika diketahui : Sudut 180° = PI = 3.141592653589793 Radian

Maka dapat dituliskan :

```
Radian = (Sudut/180) * PI
```

Untuk 90° = (90 / 180) * PI = 1.5707963267948966 Radian

Untuk 300° = (300 / 180) * PI = 5.235987755982989 Radian

Supaya parameter yang dilewatkan melalui method rotate(...) tepat sesuai dengan sebuah lingkaran maka kita harus menggunakan nilai Radian,

yang dapat terlihat pada contoh diatas bahwa object akan di rotasi sebanyak 1 derajat atau = 0.017453292519943295 Radian.

Rotasi dengan sudut putar θ dengan titik acuan (p,q) adalah :

$$\begin{bmatrix} cos\theta & -sin\theta & p - p\,cos\theta + q\,sin\theta \\ sin\theta & cos\theta & q - p\,sin\theta - q\,cos\theta \\ 0 & 0 & 1 \end{bmatrix}$$

Dengan demikian, koordinat baru hasil rotasinya adalah:

x' = x cosθ - y sinθ + p - p cosθ + q sinθ
y' = x sin y + y cosθ + q – p sinθ – q cosθ

Untuk rotasi dengan sudut acuan parameternya ada 3 yang dapat dituliskan sebagai berikut:

```
g2d.rotate(Math.toRadians(1),nilai1,nilai2);
g2d.fill(shape);
```

Parameter pertama digunakan untuk menentukan sudut, sedangkan parameter yang kedua dan ketiga digunakan untuk menentukan titik pusat perputaran berdasarkan posisix dan posisiy sebelumnya atau (posisix+nilai1) dan (posisiy+nilai2).

7.5 Skala

Skala adalah operasi untuk membuat sebuah objek lebih besar atau lebih kecil dari bentuk aslinya. Matriks transformasi untuk penskalaan dengan faktor skala terhadap sumbu x dan sumbu y adalah:

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

Maka, koordinat baru yang dihasilkan adalah:

x' = sx . x

y' = sy . y

Jangan sekali-kali terpancing dengan rumus diatas yang sangat rumit dan membingungkan itu hanya sebagai gambaran nyata dari operasi matematisnya yang sebenarnya dilakukan oleh class Graphics2D. Java telah menyediakan methodnya dimana seorang programer dipermudah tanpa harus menghitung perkalian matriksnya.

```
g2d.scale(nilai1, nilai2);
g2d.fill(shape);
```

Perlu diingat bahwa skala digunakan untuk membuat suatu object di perbesar atau diperkecil. Parameter nilai1 diatas dikalikan dengan lebar object sedangkan nilai2 dikalikan dengan tinggi object.

7.6 Menggunakan Class AffineTransform

Transformasi sudah dipelajari langsung menggunakan method yang disediakan Graphics2D. Setiap operasi transformasi hanya bisa dilakukan sekali untuk sebuah shape. Sayangnya shape tidak digunakan dalam game sebenarnya. Untuk keperluan tersebut maka akan menggunakan class AffineTransform. Class AffineTransform menyediakan operasi transformasi matrix seperti yang telah dibahas sebelumnya. Selain itu bisa menggabungkan beberapa transformasi matrix sekaligus sebelum di terapkan pada sebuah gambar. AffineTransform Menyediakan beberapa transformasi instant seperti rotasi, translasi serta scalasi.

Untuk keperluan transformasi maka akan dibuat sebuah class sederhana (class ImageEfek). Class ini menyediakan 3 buah method yaitu getRotatedImage() untuk transformasi rotasi, getScaledImage() untuk

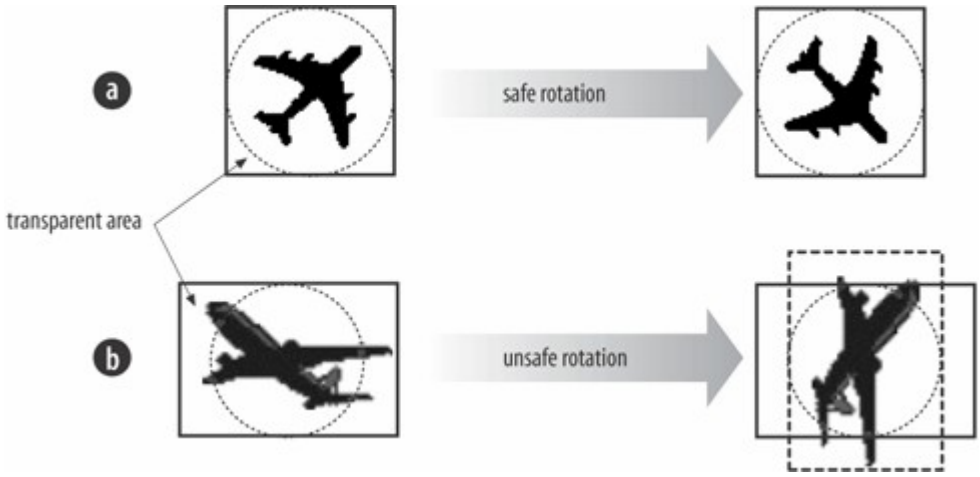
transformasi scala dan getTranslatedImage() untuk transformasi transalasi. Masing masing method sebenarnya memiliki cara kerja yang sama kecuali pemberian AffineTransform sesuai dengan method yang bersangkutan. Method getRotatedImage() (dan kedua method lainya) dibawah ini bekerja dengan membuat image penampung yang akan di gunakan sebagai tempat hasil transformasi.

```
public BufferedImage getRotatedImage(BufferedImage src, int angle)
    {
      if (src == null) {
        System.out.println("Image kosong");
        return null;
      }

      int transparency = src.getColorModel( ).getTransparency( );
      BufferedImage dest =
gc.createCompatibleImage( src.getWidth( ),     src.getHeight( ),
transparency );
      Graphics2D g2d = dest.createGraphics( );
      AffineTransform origAT = g2d.getTransform( );
      AffineTransform rot = new AffineTransform( );
      rot.rotate( Math.toRadians(angle), src.getWidth( )/2,
src.getHeight( )/2);
      g2d.transform(rot);
      g2d.drawImage(src, 0, 0, null);
      g2d.setTransform(origAT);
      g2d.dispose( );

      return dest;
  }
```

Secara teknis anda hanya perlu menerapkan satu atau lebih transformasi kedalam sebuah AffineTransform sebelum digunakan dalam transformasi gambar. Namun satu hal yang perlu diberi perhatian lebih adalah ketika menggunakan rotasi. Perhatikan ilustrasi berikut :



Gambar 7.2 Transformasi Object

Ketika menggunakan transformasi yang langsung digambarkan oleh graphics ke layar tanpa disimpan dalam object image, maka jangan khawatir akan terjadinya clipping gambar. Namun masalahnya tujuan sebenarnya adalah membuat sebuah class yang menyediakan fungsi transormasi secara langsung. Artinya perlu menggunakan sebuah image lagi untuk menampung hasil dari transformasi. Ketika membuat object image paling tidak perlu menentukan panjang dan lebar image tersebut. Penentuan perluasan image penampung ini yang sedikit bermasalah, hal itu karena kita tidak tahu bentuk citra yang terdapat pada object image yang di transformasi. Seperti ilustrasi diatas, bagian a menunjukan proses transformasi sempurna karena luasan image penampung sesuai dengan image yang di transformasi. Hal tersebut berkebalikan dengan bagian b, luasan image penampung tidak sesuai sehingga terjadi clipping.

Untuk mengantisipasi hal tersebut anda perlu memastikan besar transparant area dari sebuah gambar yang akan ditransformasi sesuai dengan kebutuhan. Atau anda juga bisa merubah secara langsung luasan image

penampung dengan nilai maximum antara panjang dan lebar image yang akan di transformasi.

Method getRotatedImage() diatas merotasi image pada titik tengah gambar atau dapat menambahkan beberapa method lagi seperti rotasi terhadap sumbu tertentu dan penggabungan beberapa transformasi sekaligus.