PERTEMUAN 10

VIEWING DAN CLIPPING

A. TUJUAN PEMBELAJARAN

Setelah menyelesaikan materi pada pertemuan ini, mahasiswa mampu mengaplikasikan viewing dan clipping.

Pada pertemuan ini akan dijelaskan mengenai :

1. Viewing
2. Clipping

B. URAIAN MATERI

1. Viewing

Sekarang kita mempertimbangkan mekanisme formal untuk merender representasi gambar ke perangkat keluaran. Biasanya, paket grafis memungkinkan pengguna untuk menentukan bagian mana dari pi & ke yang harus ditampilkan dan di mana bagian itu harus ditempatkan pada perangkat tampilan. Setiap sistem koordinat Cartesian yang nyaman yang disebut kotak referensi koordinat dunia dapat digunakan untuk menentukan gambar. Untuk gambar dua dimensi, tampilan dipilih dengan menentukan sebagian area dari total area gambar. Seorang pengguna dapat memilih satu area untuk ditampilkan, atau beberapa area dapat dipilih untuk tampilan simultan atau untuk urutan pan animasi pada sebuah adegan. Bagian gambar dalam area yang dipilih dipetakan ke area tertentu dari koordinat perangkat. Jika beberapa area tampilan dipilih, area ini dapat ditempatkan di lokasi tampilan terpisah, atau beberapa area dapat ditempatkan di area tampilan lain yang lebih besar. Transformasi dari koordinat dunia ke koordinat perangkat mencakup operasi terjemahan, putar, dan skala, serta prosedur untuk menghapus bagian gambar yang berada di luar batas area tampilan yang dipilih.

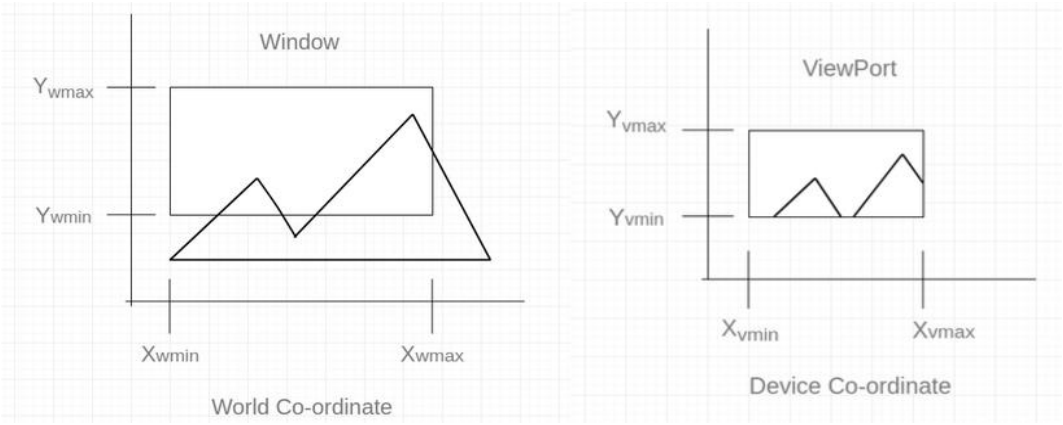
a. Struktur Viewing :

1) Window : Area koordinat dunia (world-coordinate) yang dipilih untuk ditampilkan. Untuk mendefinisikan apa yang ditampilkan.
2) Viewport : Area dari display device (misal:monitor) dimana window dipetakan. Untuk mendefinisikan di mana harus ditampilkan.
3) Viewing Transformation : Pemetaan bagian dari koordinat dunia ke koordinat device.

b. Window

Transformasi Window to Viewport adalah proses mengubah objek koordinat dunia 2D menjadi koordinat perangkat. Objek di dalam world atau clipping window clipping dipetakan ke area pandang yang merupakan area di layar tempat koordinat dunia dipetakan untuk ditampilkan.

World Coordinate Area yang dipilih untuk tampilan disebut Window. Sebagian besar window berbentuk persegi panjang.



Gambar 10. 1 Window

c. Viewing Transformation

Transformasi terakhir yang diterapkan pada objek sebelum ditampilkan dalam gambar adalah transformasi window-to-viewport, yang memetakan jendela tampilan persegi pada bidang xy yang berisi pemandangan ke kisi piksel persegi panjang tempat gambar akan ditampilkan. Saya akan berasumsi di sini bahwa jendela tampilan tidak diputar; bahwa itu, sisi-sisinya sejajar dengan sumbu x dan y. Dalam hal ini, transformasi window-to-viewport dapat diekspresikan dalam bentuk terjemahan dan transformasi penskalaan.

Mari kita lihat kasus umum di mana area pandang memiliki koordinat piksel mulai dari 0 di kiri hingga lebar di kanan dan dari 0 di atas hingga tinggi di bawah. Dan asumsikan bahwa batas pada jendela tampilan adalah kiri, kanan, bawah, dan atas. Dalam hal ini, transformasi window-to-viewport dapat diprogram sebagai:

```
scale( width / (right-left), height / (bottom-top) );
translate( -left, -top )
```

Ini harus menjadi transformasi terakhir yang diterapkan ke suatu titik. Karena transformasi diterapkan ke titik-titik dengan urutan kebalikan dari yang ditentukan dalam program, transformasi tersebut harus menjadi transformasi pertama yang ditentukan dalam program. Untuk melihat bagaimana ini bekerja, pertimbangkan satu titik (x, y) di jendela tampilan. (Titik ini berasal dari beberapa objek dalam pemandangan. Beberapa transformasi pemodelan mungkin telah diterapkan pada objek untuk menghasilkan titik (x, y), dan titik tersebut sekarang siap untuk transformasi terakhirnya menjadi koordinat area pandang.) Koordinat (x, y) pertama kali diterjemahkan oleh (-left, -top) menjadi (x-left, y-top). Koordinat ini kemudian dikalikan dengan faktor skala yang ditunjukkan di atas, sehingga menghasilkan koordinat akhir

```
x1 = width / (right-left) * (x-left)
y1 = height / (bottom-top) * (y-top)
```

Perhatikan bahwa titik (kiri, atas) dipetakan ke (0,0), sedangkan titik (kanan, bawah) dipetakan ke (lebar, tinggi), yang kita inginkan. Masih ada pertanyaan aspek rasio. Seperti disebutkan di Subbagian 2.1.3, jika kita ingin memaksa rasio aspek jendela agar sesuai dengan rasio aspek viewport, mungkin perlu menyesuaikan batas pada jendela. Berikut adalah pseudocode untuk subrutin yang akan melakukannya, dengan asumsi lagi bahwa pojok kiri atas viewport memiliki koordinat piksel (0,0):

```
subroutine applyWindowToViewportTransformation (
          left, right,    // horizontal limits on view window
          bottom, top,    // vertical limits on view window
          width, height, // width and height of viewport
          preserveAspect // should window be forced to match viewport aspect?
       )

   if preserveAspect :
       // Adjust the limits to match the aspect ratio of the drawing area.
       displayAspect = abs(height / width);
       windowAspect = abs(( top-bottom ) / ( right-left ));
       if displayAspect > windowAspect :
           // Expand the viewport vertically.
           excess = (top-bottom) * (displayAspect/windowAspect - 1)
           top = top + excess/2
           bottom = bottom - excess/2
       else if displayAspect < windowAspect :
           // Expand the viewport vertically.
           excess = (right-left) * (windowAspect/displayAspect - 1)
           right = right + excess/2
```

d. Viewing Transformation

Titik *P (x, y)* dan *P0(x_0, y_0)* yang bertindak sebagai vector arah pandang atas bagi pengamat (viewer) atau sebagai sumbu Y_v dari viewing coordinates system, vector satuan v pada arah Y_v pada viewing coordinat system. Rumusnya sebagai berikut :

$$v = \frac{p-p_0}{|p-p_0|} = \frac{(x-x_0\,,y-y_0)}{\sqrt{(x-x_0)^2+(y-y_0)^2}} = (v_x,v_y)$$

Vector satuan *u* pada coordinates system dihitung dengan cara perkalian silang antara vector satuan v dengan sumbu z pada viewing coordinates system:

$$\boldsymbol{u = v * (0,0,1)\ \text{atau}\ u = (v_y, -v_x)}$$

Matriks transformasi M dari World Coordinates System ke Viewing Coordinates System dinyatakan sebagai komposisi matriks berikut :

$$M = R * T(-x_0, -y_0)$$

$$M = \begin{pmatrix} u_x & u_y & 0 \\ v_x & v_y & 0 \\ 0 & 0 & 1 \end{pmatrix} * \begin{pmatrix} 1 & 0 & -x_0 \\ 0 & 1 & -y_0 \\ 0 & 0 & 1 \end{pmatrix}$$

Sehingga diperoleh titik terhadap viewing Coordinates System adalah

P viewer = M * *P* word

Keterangan : *P*word = Posisi titik terhadap World Coordinat System

*P*viewer = Posisi titik terhadap Viewig Coordinat System

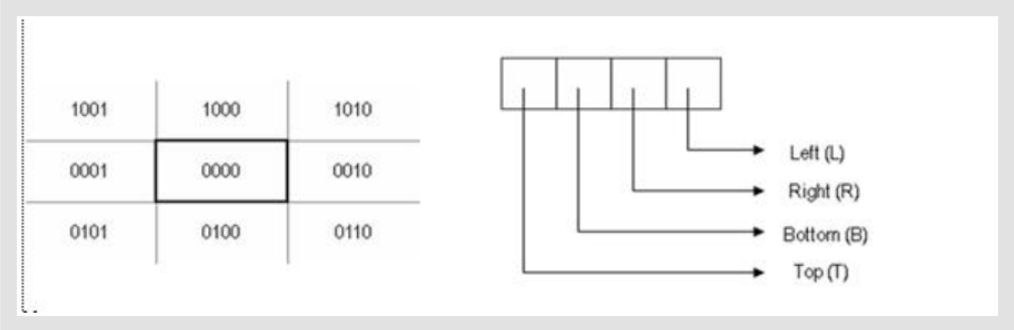
2. Clipping

Metode clipping adalah metode yang digunakan untuk menentukan garis yang perlu digambar atau tidak. Alasan dilakukanna clpping adalah untuk menghindari perhitungan koordinat pixel yang rumit dan interpolasi parameter. Clpping dilakuakn sebelum proses rasterization. Setelah proses clipping selanjutnya dilakukan proses rasterization yang mana dilakukan pengkonversian suatu citra vektor ke citra bitmap.

Tiap metode mempunya beberapa algoritma dan tentunya tiap algoritma memiliki kelebihan dan kekurangan untuk dianalisis. Contohnya pada algoritma clipping didapatkan algoritma Liang-Barsky yang terbaik karena kecepatan waktu yang efisien dan juga stabil. Untuk metode rasterization didapat algoritma Midpoint yang terbaik karena operasi bilangan pada Midpoint dilakukan dengan cara menghilangkan operasi bilangan riel dengan bilangan integer yang mana bilangan integer jauh lebih cepat dibandingkan dengan operasi bilangan riel. Oleh karena itu, komputasi midpoint lebih cepat delapan kali pada pembuatan garis lurus dan lima belas kali pada penggambaran lingkaran. Sedangkan pada metode Hidden Surface Remove, algoritma yang terbaik adalah algoritma scan Line karena pada algoritma ini menggunakan memori yang lebih sedikit dan dari segi kecepatan juga lebih unggul.

a. Point Clipping

Clipping line bisa ditentukan dengan beberapa alghoritma, tapi untuk kasus ini saya menggunakan algoritma Cohen-Suterland. Berikut ini adalah gambar pembagian dari baris segmen window dengan algoritma Cohen-Sutherland.



Keterangan :

L :	1 apabila x < xmin0 apabila x ≥ xmin	B :	1 apabila y < ymin0 apabila y ≥ ymin
R :	1 apabila x > xmax0 apabila x ≤ xmax	T :	1 apabila y > ymax0 apabila y ≤ ymax

Gambar 10. 2 Pembagian dari baris segmen window Sutherland

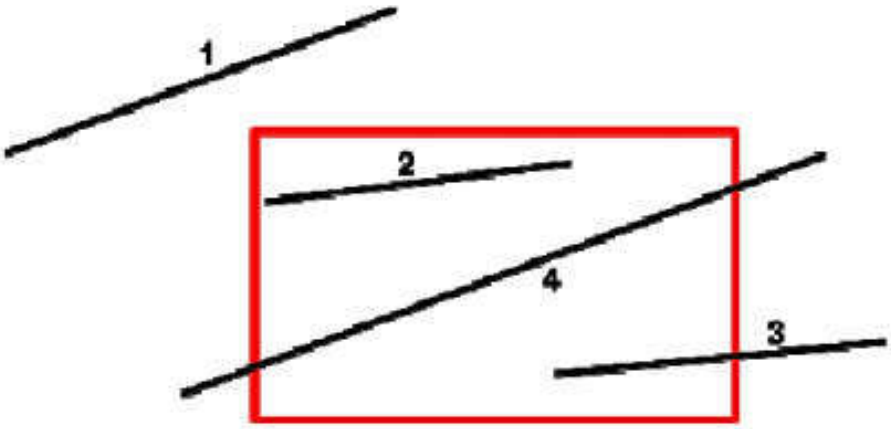
Dengan asumsi bahwa jendela klip adalah persegi panjang dalam posisi standar, kami menyimpan titik P = (x, y) untuk tampilan jika pertidaksamaan berikut terpenuhi:

$$xw_{\min} \leq x \leq xw_{\max}$$
$$yw_{\min} \leq y \leq yw_{\max}$$

Dengan tepi Clipp Window (*xwmin xwmax, ywmin, ywmax*) dapat berupa batas jendela koordinat dunia atau batas area pandang. Jika salah satu dari empat ketidaksamaan ini tidak terpenuhi, poinnya akan dipotong (tidak disimpan untuk ditampilkan).

b. Line Clipping

Line clipping diproses dengan inside-outside test dengan memeriksa endpoint dari garis tersebut.



Gambar 10. 3 Line Clipping

Dari test tersebut dapat dikategorikan sebagai berikut:

1) Garis yang kedua endpoint-nya ada di dalam batas clipping akan disimpan (garis 2)
2) Garis yang kedua endpoint tidak berada dalam batas clipping maka garis tersebut berada di luar window (garis 1).
3) Semua garis lain yang memotong satu atau lebih batas clipping memerlukan algoritma clipping yang dapat mengidentifikasi dengan efisien bahwa garis berada diluar batas clipping window (garis 3 dan 4).

Kondisi garis terhadap clipping window dapat dikategorikan seperti tabel berikut:

Nama	Kondisi
Invisible	Tidak kelihatan, terletak di luar *clipping window*
Visible	Terletak di dalam *clipping window*
Half Partial	Terpotong sebagian oleh *clipping window*
Full Partial	Terpotong penuh oleh *clipping window*, garis melintasi *clipping window*

Untuk kondisi garis yang invisible dan visible tidak perlu dilakukan aksi clipping, dimana untuk:

1) Invisible, tidak perlu ditampilkan
2) Visible, langsung ditampilkan

Persamaan untuk segmen garis dengan endpoint (x1,y1) dan (x2,y2) yang keduanya terletak diluar clipping window adalah:

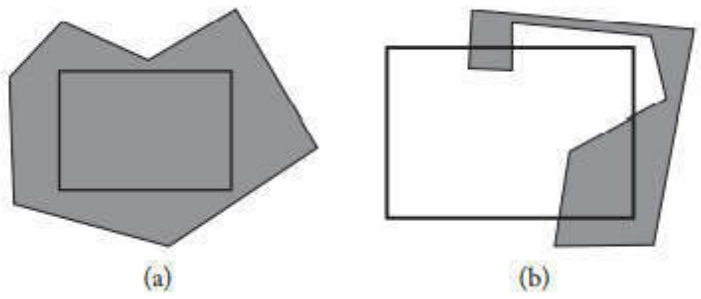
$$x = x1 + u(x2 - x1)$$
$$y = x1 + u(x2 - x1)$$
$$0<=u<=1$$

Persamaan tersebut digunakan untuk mengenali nilai parameter untuk koordinat pemotongan dengan batas *clipping window*.

1) Membaca data garis
2) Membaca data *clipping window*
3) Mengecek koondisi garis terhadap *clipping window*
4) Proses *clipping*

c. Polygon Clipping

Pemotongan tidak hanya untuk simpul dan tepi poligon. Pada Gambar, semua simpul dan tepi berada di luar persegi panjang pemotongan; akan tetapi, poligon tidak berada di wilayah luar dari persegi panjang: kliping adalah persegi panjang itu sendiri.



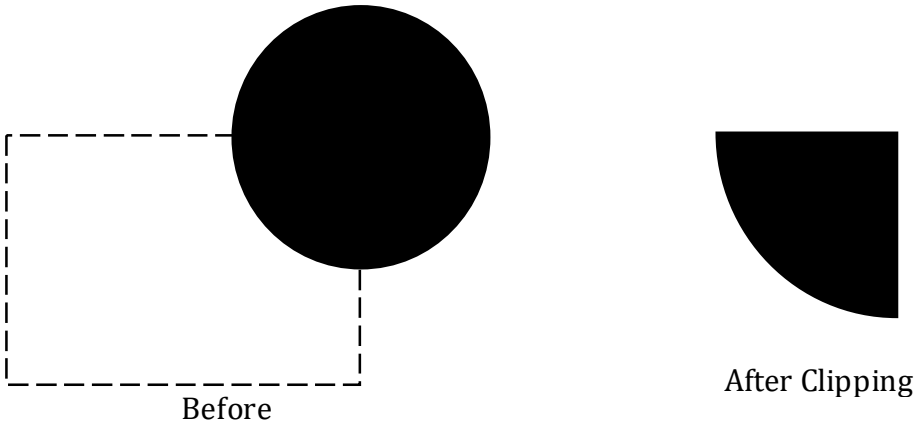
Gambar 10. 4 Polygon Clipping

Dalam hal objek yang akan dipotong adalah sebuah titik, satu-satunya operasi yang harus dilakukan adalah klasifikasi. Jika objek berupa garis lurus, kita juga dapat menghitung persimpangan sebagai cara untuk mengklasifikasikan setiap segmen yang dihasilkan. Dalam kasus ini, kita tidak harus berurusan dengan penataan. Tetapi dalam kasus daerah Polygon, pemotongan melibatkan: persimpangan, klasifikasi, dan penataan.

Secara umum, untuk daerah klip kami melakukan pemotongan batas dan kemudian menentukan kurva Polygon batas dari setiap daerah yang dihasilkan dari operasi pemotongan. Karena pemotongan batas direduksi untuk merekam urutan segmen garis lurus, hal ini tidak menimbulkan kesulitan apa pun — tantangan sebenarnya adalah dalam Polygon data untuk mendapatkan batas setiap wilayah yang dihasilkan dari pemotongan tersebut. Kasus paling sederhana dari pemotongan Polygon adalah pemotongan segitiga, yang akan kita pelajari selanjutnya.

d. Curve Clipping

Area dengan batas melengkung dapat dipotong dengan metode yang mirip dengan yang dibahas di bagian sebelumnya. Prosedur kurva melibatkan persamaan nonlinier, dan ini membutuhkan lebih banyak pemrosesan dari pada untuk objek dengan batas linier.



Gambar 10. 5 Curve Clipping

Persegi panjang pembatas untuk lingkaran atau objek melengkung lainnya dapat digunakan pertama kali untuk menguji tumpang tindih dengan

jendela klip persegi panjang. Jika persegi panjang pembatas untuk objek sepenuhnya berada di dalam jendela, kami menyimpan objek tersebut Jika persegi panjang ditentukan sepenuhnya berada di luar jendela, kami membuang objeknya Dalam kedua kasus tersebut; tidak diperlukan perhitungan lebih lanjut Tetapi jika uji persegi panjang pembatas gagal: kita dapat mencari pendekatan lain yang menghemat komputasi. Untuk lingkaran, kita dapat menggunakan luasan koordinat dari masing-masing kuadran dan kemudian oktan untuk pengujian pendahuluan sebelum menghitung persimpangan kurva-jendela_ Untuk elips, kita dapat menguji luasan koordinat kuadran individu. Gambar 10.5 mengilustrasikan kliping lingkaran pada jendela persegi Panjang Prosedur serupa dapat diterapkan saat memotong objek melengkung ke wilayah klip poligon umum. Pada lintasan pertama, kita dapat memotong persegi panjang pembatas objek terhadap persegi panjang pembatas dari wilayah klip. Jika kedua daerah tumpang tindih, kita perlu menyelesaikan persamaan kurva garis secara simultan untuk mendapatkan titik potong.

C. SOAL LATIHAN/TUGAS

Latihan	Petunjuk Pengerjaan Tugas
Latihan Pertemuan 10	1. Jelaskan definisi dari viewing & clipping?

D. REFERENSI