

TUGAS TENGAH SEMESTER

IF3260 Grafika Komputer



Oleh

Firdaus Ibnu Romadhon	13510079
Mamat Rahmat	13512007
Arieza Nadya Sekariani	13512017
Jonathan	13512031
Fahmi Dumadi	13512047
Rakhmatullah Yoga Sutrisna	13512053
Aurelia	13512099

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung

2015

I. Dasar Teori

Tugas 1 – Point (Color picker)

Titik adalah lokasi di dalam suatu ruang yang tidak memiliki karakteristik. Titik tidak memiliki panjang, lebar, atau ketebalan. Dalam grafika komputer, titik direpresentasikan dalam koordinat x dan y. Titik origin dari x adalah ujung kiri layar komputer dan titik origin dari y adalah ujung atas layar komputer.

Dasar untuk membuat sebuah point pada layar komputer menggunakan frame buffer. Untuk sistem raster hitam putih, sebuah point dibentuk dengan mengubah bit value pada sebuah posisi layar yang terdapat pada frame buffer menjadi 1. Pada sistem RGB, frame buffernya menyimpan nilai kode warna untuk intensitas yang akan di-display pada sebuah pixel yang terdapat pada posisi layar tertentu.

Tugas 2 – Garis

Garis adalah suatu kumpulan titik yang menghubungkan antara dua titik ujung. Dalam garis, terdapat pengertian segmen garis (line segment). Segmen garis adalah suatu garis lurus yang menghubungkan antara dua titik. Segmen garis tidak memiliki ketebalan. Antara dua titik yang dihubungkan dengan garis lurus, hanya boleh terdapat satu segmen garis.

Secara teori, garis bersifat kontinu. Akan tetapi, dalam praktiknya, garis tidak bisa digambar secara kontinu di dalam sebuah layar karena layar merupakan salah satu media grafis diskrit. Oleh karenanya, biasanya dilakukan pendekatan untuk menentukan titik-titik yang membentuk segmen garis. Metode untuk menggambar garis yang paling simpel adalah dengan algoritma naive line drawing seperti dituliskan di bawah ini.

$$Dx = x2 - x1;$$

$$Dy = y2 - y1;$$

For x from x1 to x2 {

$$Y = y1 + dy * (x - x1) / dx$$

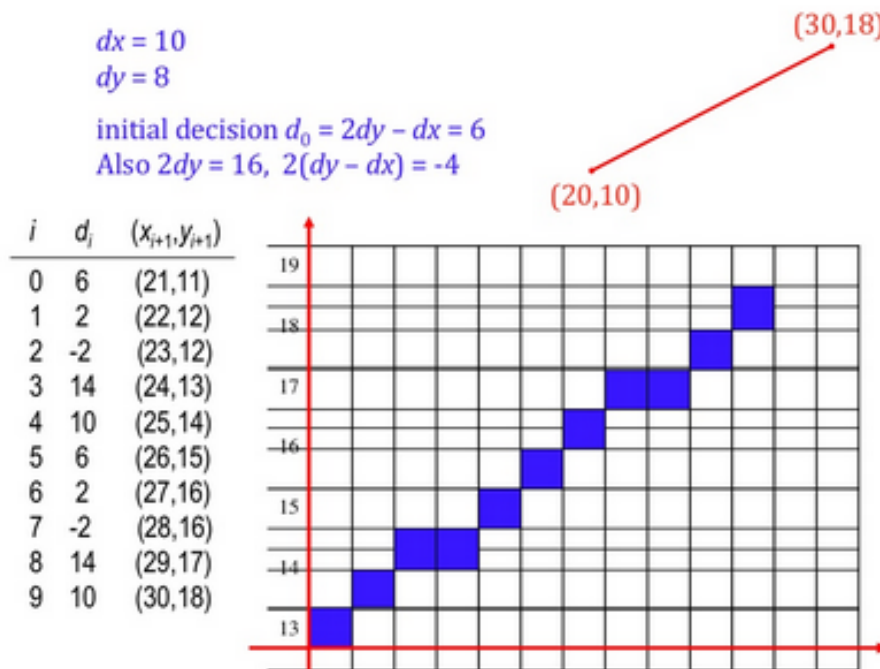
Plot(x,y)

}

Akan tetapi, algoritma naive line drawing tidak efisien dan memakan biaya yang mahal. Ketidakefisiensian algoritma ini terutama dikarenakan memakan banyak operasi untuk melakukan perhitungan bilangan real. Algoritma yang lebih baik dibandingkan naive line drawing ini adalah algoritma Bresenham dan algoritma Wu.

Pembuatan garis pada komputer menggunakan algoritma garis Bresenham. Contohnya point (10.48, 20.51) akan dibulatkan menjadi (10,21). Algoritma Bresenham menggunakan parameter keputusan $d_i = (s_i - t_i)$ untuk menentukan pixel berikutnya yang akan ditandai berdasarkan garis aslinya. Jika nilai $d_i < 0$ maka pixel selanjutnya yang ditandai berada di bawah garis asli (s_i lebih kecil), sedangkan jika nilai $d_i \geq 0$ maka pixel berikutnya berada di atas garis asli atau t_i lebih kecil.

Contoh untuk membuat garis dari point (20,10) ke (30,18) dengan menggunakan algoritma Bresenham.

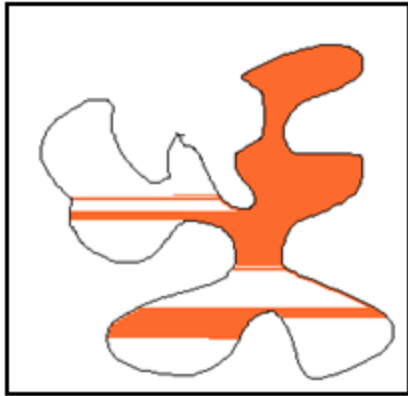


Tugas 3 – Filling

Algoritma filling yang digunakan adalah *flood fill* dan *scan line*.

Flood fill adalah teknik mewarnai sebuah bidang dengan menggunakan analogi menuang cat warna di suatu titik kemudian cat bergerak meluas sampai memenuhi semua area yang dibatasi bidang. Cat akan mengisi daerah yang memiliki warna tertentu. Algoritma flood fill dideskripsikan secara rekursif. Cara kerja algoritma ini adalah dengan pertama-tama menentukan warna dan titik awal lalu memeriksa titik-titik tetangganya, jika warna titik tetangga tersebut berbeda, maka warnanya akan diubah. Pemeriksaan dan pengubahan warna ini dilakukan secara rekursif sampai seluruh area di dalam bidang selesai diproses.

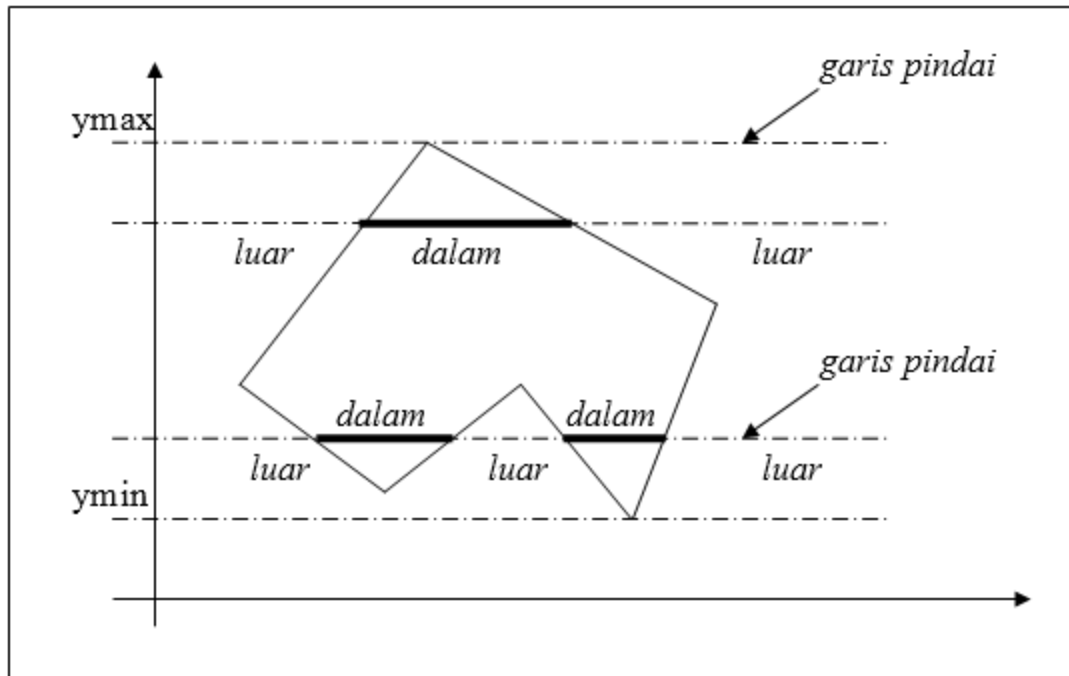
Dalam algoritma flood fill, harus dipastikan bahwa area yang akan diwarnai merupakan suatu bidang yang tertutup. Karena jika tidak tertutup dan ada satu saja titik yang terbuka, maka pengisian akan keluar dari area yang diinginkan.



Scan line adalah teknik mewarnai dengan scanning/memindai untuk setiap baris dari layar bidang gambar. Metode scan line dilakukan dengan menentukan overlap interval yang melintasi area. Algoritma ini berdasarkan pada titik potong terhadap garis dari gambar untuk menentukan titik-titik pada gambar. Titik potong diurutkan dari kiri ke kanan sehingga algoritma ini bekerja dimulai dari kiri atas layar dan bergerak ke kanan hingga batas kanan layar. Proses ini diulang untuk semua baris yang ada. Ketika sampai di suatu batas area, akan dilakukan operasi untuk menemukan batas area berikutnya, yakni operasi putpixel.

Implementasi pertama dari algoritma scan line ini adalah membentuk sebuah garis vertikal. Pembuatan garis-garis ini dilakukan berulang untuk sisi kiri dan kanan sampai bidang terisi penuh. Oleh karena itu, biasanya algoritma pengisian warna ini bekerja secara horizontal.

Dalam algoritma scan line, dapat ditemui kegagalan jika ditemui titik-titik ujung sehingga memerlukan penanganan khusus. Hal ini dapat ditangani dengan memeriksa semua informasi titik ujung berdasarkan vektor yang dimiliki primitif.



Tugas 4 – Translasi, rotasi, scaling dua dimensi

TRANSLASI

Translasi pada suatu objek dilakukan dengan menambahkan parameter jarak translasi pada koordinat awal posisi objek yang akan ditranslasikan. Bentuk translasi ini dapat dituliskan dalam bentuk matriks. Koordinat baru (x', y') didapat dari:

$x' = x + t_x$ dan $y' = y + t_y$ dimana (x, y) merupakan koordinat awal dari objek tersebut. Sehingga translasi objek dapat dituliskan seperti $\mathbf{P}' = \mathbf{P} + \mathbf{T}$ dimana $\mathbf{P} = [x \ y]$, $\mathbf{T} = [t_x \ t_y]$. Translasi dilakukan dengan memindahkan koordinat lama ke koordinat baru dan menggambar kembali garis serta atribut yang ada di antara koordinat-koordinat barunya.

ROTASI

Rotasi pada suatu objek dilakukan dengan mendefinisikan sudut rotasi θ dan posisi titik sumbu rotasinya. Rotasi dapat dinyatakan dengan:

$$x' = r \cos (\Phi + \theta) = r \cos \Phi \cos \theta - r \sin \Phi \sin \theta$$

$$y' = r \sin (\Phi + \theta) = r \cos \Phi \sin \theta + r \sin \Phi \cos \theta$$

dimana pada koordinat polar, $x = x \cos \Phi$ dan $y = r \sin \Phi$,

$$x' = x \cos \theta - y \sin \theta$$

$$y' = x \sin \theta + y \cos \theta, \text{ sehingga rotasi objek dapat didefinisikan dengan } \mathbf{P}' = \mathbf{R} \cdot \mathbf{P}$$

Rotasi objek dilakukan dengan memindahkan (merotasikan) setiap poin pada objek dan menggambarkan kembali garis-garisnya.

SKALA

Untuk mengskalakan suatu objek poligon, operasi dilakukan dengan mengalikan koordinat vertex (x, y) dengan faktor skala s_x dan s_y , dimana s_x mengskalakan objek pada sumbu x dan s_y mengskalakan objek pada sumbu y . Pengskalakan objek dalam matriks dapat dituliskan seperti:

$P' = S \cdot P$, dimana S merupakan matriks pengkalaannya. Lokasi objek tersebut dapat diatur dengan menentukan suatu fixed point. Polygon diskalakan relatif terhadap fixed point tersebut.

Tugas 5 – Clipping

Algoritma Cohen-Sutherland digunakan untuk melakukan clipping pada garis. Algoritma ini membagi ruang dua dimensi menjadi 9 bagian (8 bagian “outside” dan 1 bagian “inside”) dan menentukan garis dan bagian dari garis yang terlihat pada sebuah viewport. Pembagian dilakukan dengan meng-assign 4 bit nilai dengan pembacaan (left, right, bottom, top).

Contoh pembagian layar:

1001	0001	0101
1000	0000	0100
1010	0010	0110

Nilai 0000 merupakan viewportnya (bagian inside), sedangkan 8 bagian lainnya merupakan bagian outside.

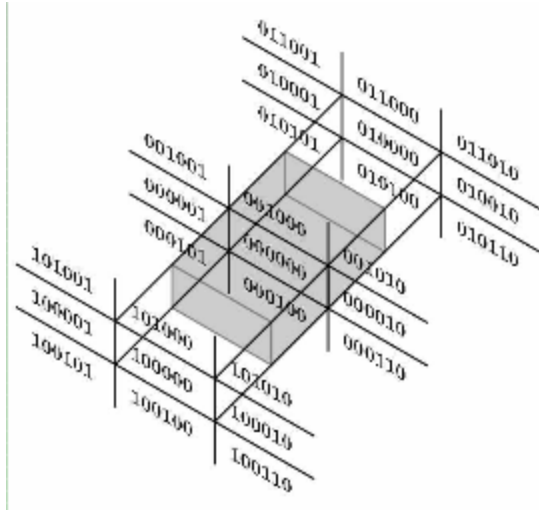
Apabila endpoints dari suatu garis berada pada bagian viewport → trivial accept, garis tidak perlu di-clip. Sedangkan jika endpoints garis berada di salah satu sisi viewport → trivial reject, garis tidak perlu di-clip dan tidak perlu di-display.

Operasi logical AND pada endpoints digunakan untuk menentukan apabila garis berada di luar viewport. Contoh: endpoint1 memiliki kode 1001 dan endpoint2 memiliki kode 1010. $1001 \text{ AND } 1010 = 1000$, yang berarti segmen garis tersebut berada di luar viewport. Apabila hasil logical AND pada kedua endpoints menghasilkan 0000 maka garis tersebut tidak dapat di trivial reject.

Operasi logical OR dipakai untuk menentukan apabila suatu garis berada di dalam viewport. Jika hasil logical OR nya menghasilkan 0000, garis tersebut dapat di trivial accept.

Tugas 6 – 3D

Algoritma Cohen-Sutherland juga dapat digunakan untuk clipping pada ruang 3 dimensi, namun pada ruang 3 dimensi, bagian pada layar dibagi menjadi 27 bagian. Nilai pada setiap bagiannya terdiri atas 6 bit kode (atas, bawah, kanan, kiri, belakang, depan). Metode clipping yang dilakukan sama seperti pada ruang 2 dimensi.



II. Story

Pada suatu waktu saat terjadi perang dunia pada tahun xxxx Indonesia diserang oleh negara x. Tentara negara x yang menyerang Indonesia membuat markas negara tersebut yang berlokasi di daerah yang sekarang merupakan daerah Bandung. User diminta untuk bergabung bersama TNI AU menjadi pilot untuk menghancurkan kapal-kapal musuh yang ingin menginvasi teritori negara Indonesia serta menghancurkan markas musuh yang terletak di Bandung lama. Markas musuh hancur berkat bantuan user dan teroris dari negara Indonesia. Bertahun-tahun setelahnya di bekas lokasi markas musuh tersebut berdirilah sebuah perguruan tinggi terkemuka yang sekarang menjadi Institut Teknologi Bandung.

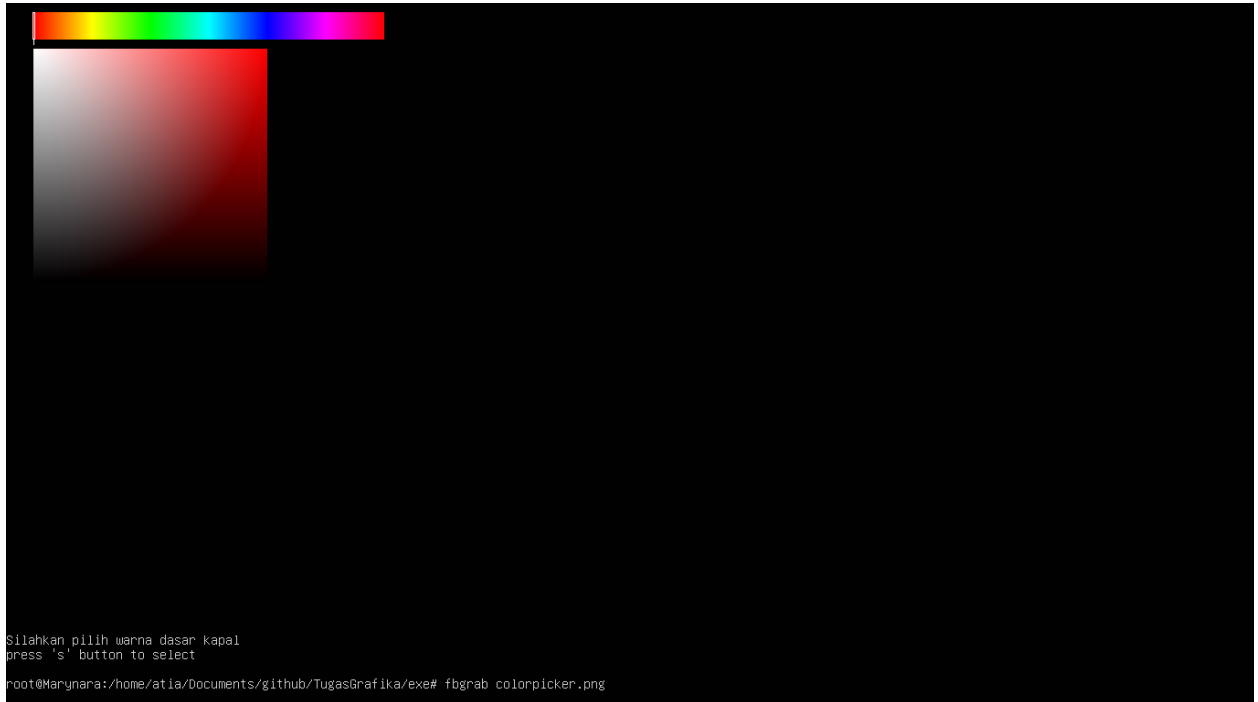
III. Cara dan Contoh Penggunaan

Cara penggunaan

1. Buat direktori "exe" dulu pada repository ini untuk lokasi executable files (sesuai spesifikasi deliverable)
2. Compile pada direktori src/mainprogram
3. Untuk menjalankan program, tekan ctrl+alt+f1 untuk masuk mode console, kemudian login dan masukkan perintah "sudo su" untuk akses sebagai root
4. Jalankan dengan perintah ./<nama executable>

Contoh penggunaan

User memasuki aplikasi dan memilih warna dasar untuk pesawat dan kapal yang akan dipakai dengan menggunakan color picker.



Setelah itu akan ditampilkan animasi pesawat dan kapal yang saling menembak serta narasi untuk menjelaskan situasi kepada user. User juga disuguhkan dengan peta Indonesia yang dapat di-zoom di lokasi yang diinginkan user. Setelah itu ditampilkan peta 3D ITB.