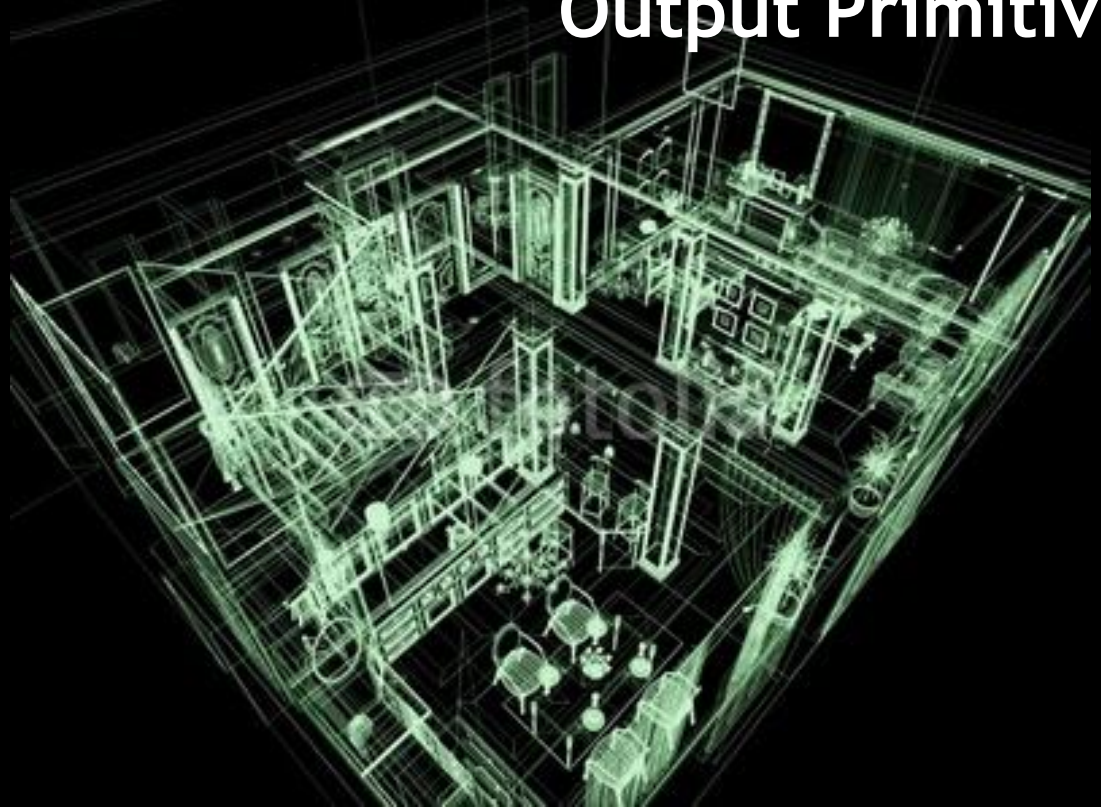




# GRAFIKA KOMPUTER

## Output Primitive



Heri Setiawan, S.Kom

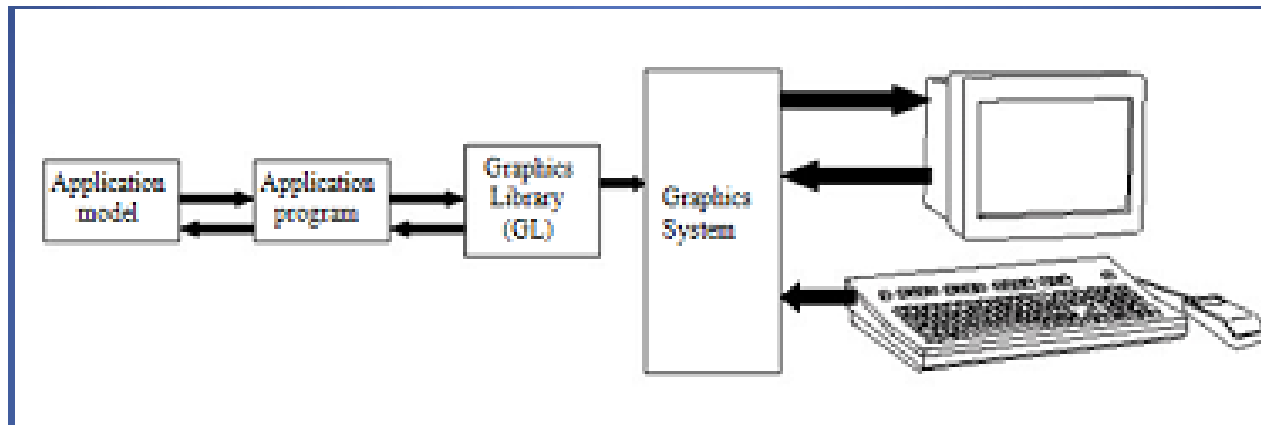
# Pendahuluan

- ❑ Gambar dapat dijelaskan dengan beberapa cara, bila menggunakan raster display, gambar ditentukan oleh satu set intensitas untuk posisi display pada display. Sedangkan dengan scene tampilan gambar dengan loading array dari pixel ke dalam buffer atau dengan mengkonversikan scan dari grafik geometri tertentu ke dalam pola pixel.
- ❑ Paket grafika dilengkapi dengan fungsi untuk menyatakan scene dalam bentuk struktur . Paket pemrograman grafika dilengkapi dengan fungsi untuk menyatakan scene dalam bentuk struktur dasar geometri yang disebut output primitif, dengan memasukkan output primitif tersebut sebagai struktur yang lebih kompleks.

# Kerangka Grafik Komputer Interaktif

- ❑ Graphics library/package (contoh: OpenGL) adalah perantara aplikasi dan display hardware (Graphics System)
- ❑ Application program memetakan objek aplikasi ketampilan/citra dengan memanggil graphics library
- ❑ Hasil dari interaksi user menghasilkan/modifikasi citra
- ❑ Citra merupakan hasil akhir dari sintesa, disain, manufaktur, visualisasi dll.

# Kerangka Grafik Komputer Interaktif

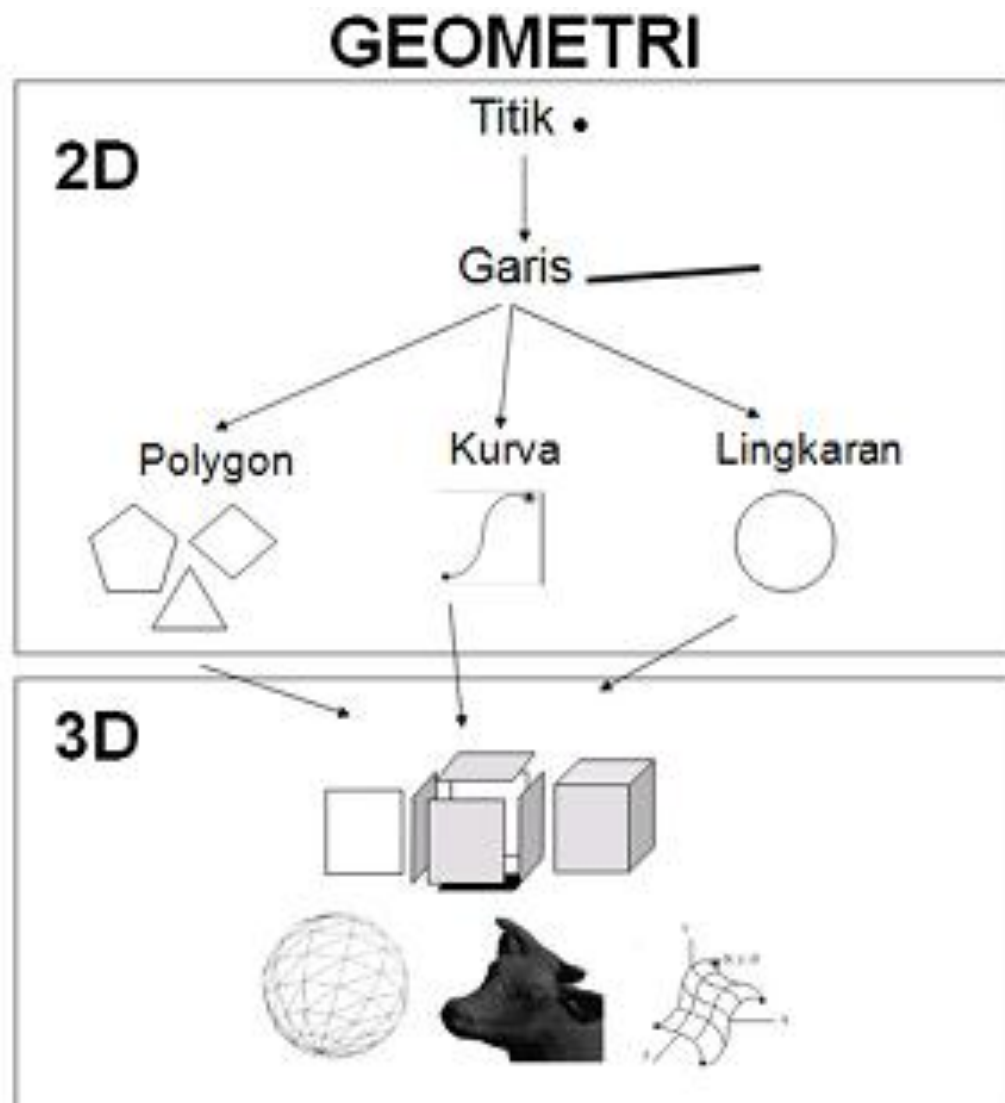


- ❑ Graphics library/package (contoh: OpenGL) adalah perantara aplikasi dan display hardware (Graphics System)
- ❑ Application program memetakan objek aplikasi ketampilan/citra dengan memanggil graphics library
- ❑ Hasil dari interaksi user menghasilkan/modifikasi citra
- ❑ Citra merupakan hasil akhir dari sintesa, disain, manufaktur, visualisasi dll.

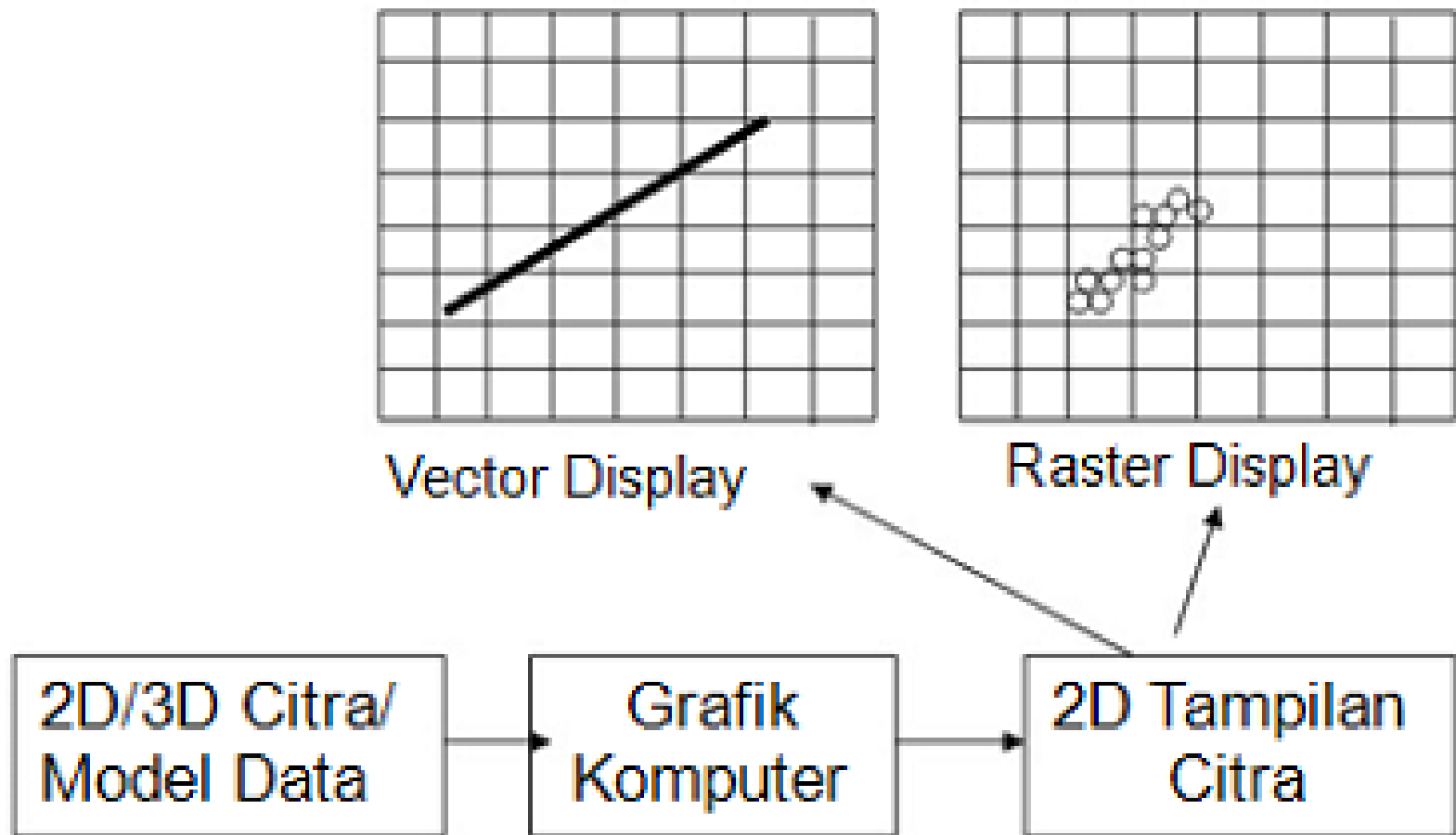
# Pemodelan Geometris

- ❑ Transformasi dari suatu konsep (atau suatu benda nyata) ke suatu model geometris yang bisa ditampilkan pada suatu komputer :
  - Shape/bentuk
  - Posisi
  - Orientasi (cara pandang)
  - Surface Properties / Ciri-ciri Permukaan (warna, tekstur)
  - Volumetric Properties / Ciri-ciri volumetric (ketebalan/pejal, penyebaran cahaya)
  - Lights/cahaya (tingkat terang, jenis warna), Dan lain-lain ...
- ❑ Pemodelan Geometris yang lebih rumit :
  - Jala-Jala segi banyak: suatu koleksi yang besar dari segi bersudut banyak, dihubungkan satu sama lain.

# Elemen-elemen Pembentuk Grafik:

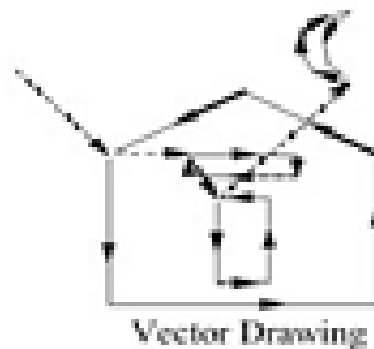


# Pemrosesan Citra untuk Ditampilkan di Layar

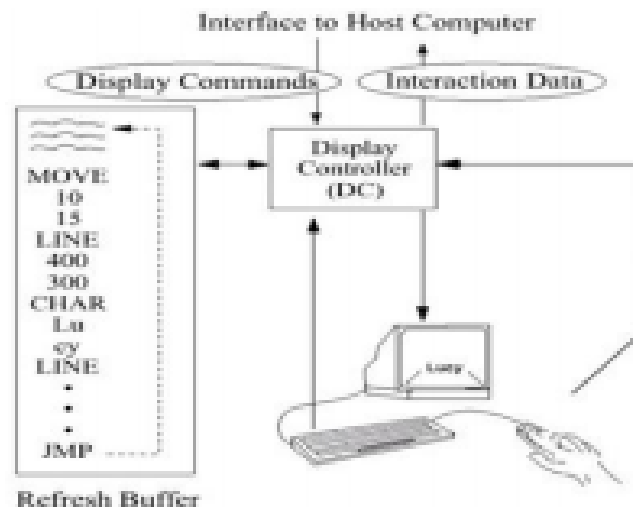


# Hardware Display Grafik : Vektor

- Vetor (calligraphic, stroke, random-scan)



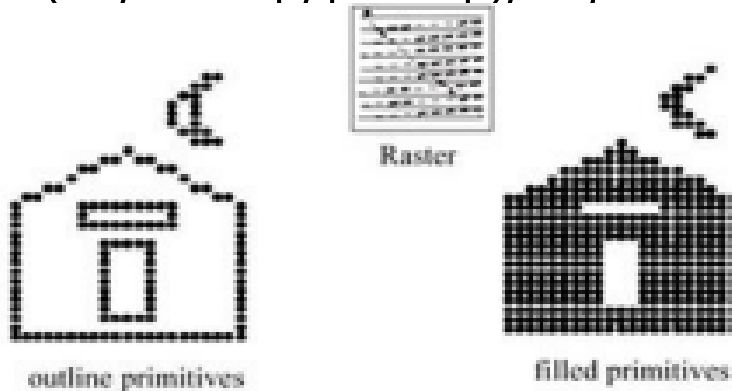
- Arsitektur Vektor



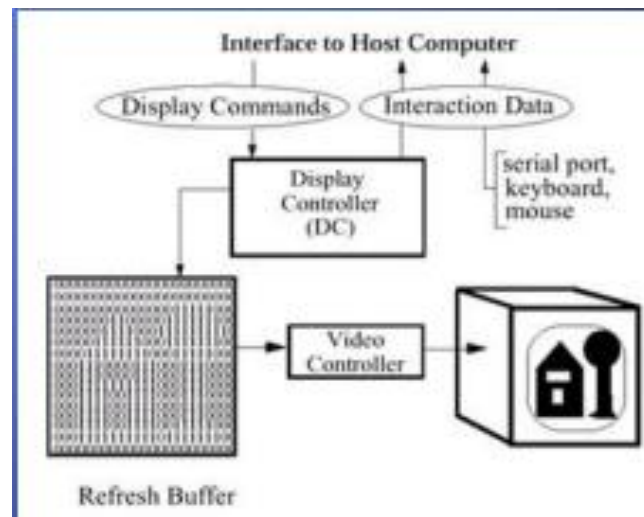


# Hardware Display Grafik : Raster

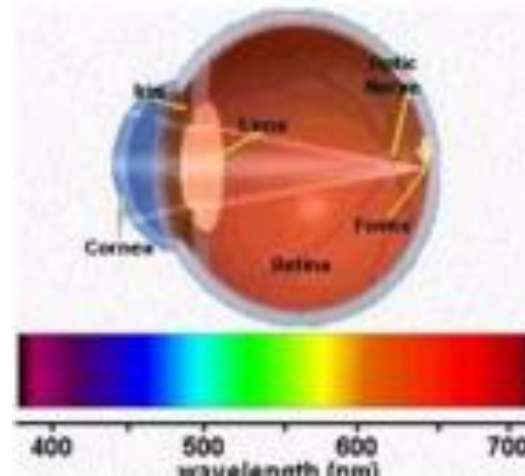
- Raster (TV, bitmap, pixmap), digunakan dalam layar dan laser printer



- Arsitektur Vektor

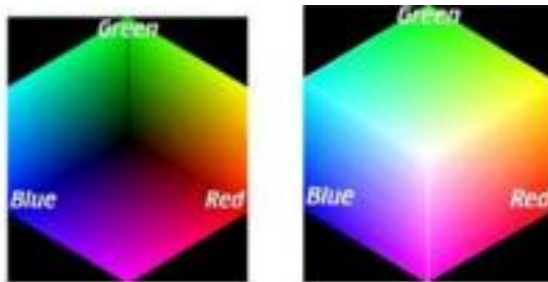


- Sistem Visual Manusia



- Pembentukan Citra oleh Sensor Mata
  - Intensitas cahaya ditangkap oleh diagram iris dan diteruskan ke bagian retina mata.
  - Bayangan obyek pada retina mata dibentuk dengan mengikuti konsep sistem optik dimana fokus lensa terletak antara retina dan lensa mata.
  - Mata dan syaraf otak dapat menginterpretasi bayangan yang merupakan obyek pada posisi terbalik.

- Kubus Warna RGB



-Sistem Koordinat dengan R,G, B sebagai axes.

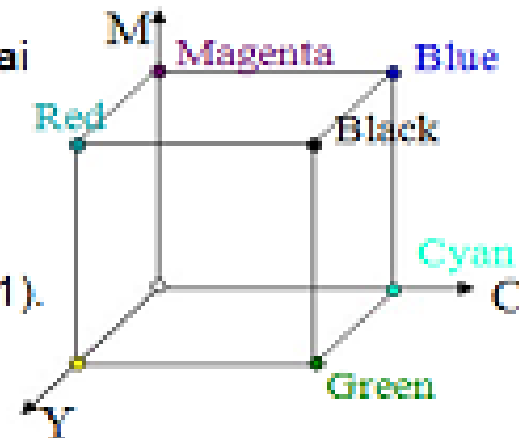
- Model Warna CMY

## Model Warna CMY

-Sistem Koordinat dengan C, M, Y sebagai axes; banyak digunakan untuk menggambarkan warna pada perangkat output hard-copy

- Grayscale axis runs from (0,0,0) to (1,1,1).

- Color : proses subtractive.



# TRANSFORMASI

Menurut Suyoto (2003), transformasi adalah memindahkan objek tanpa merusak bentuk. Contoh transformasi adalah transisi, penskalaan, putaran/rotasi, balikan, shearing dan gabungan.

Tujuan transformasi adalah

- ✓ Merubah atau menyesuaikan komposisi pemandangan
- ✓ Memudahkan membuat objek yang simetris
- ✓ Melihat objek dari sudut pandang yang berbeda
- ✓ Memindahkan satu atau beberapa objek dari satu tempat ke tempat lain. Ini biasa dipakai untuk animasi computer

# TRANSFORMASI

Untuk dapat menggunakan transformasi dengan baik maka diperlukan pengetahuan operasi matriks dan vektor.

- **Operasi matriks:**

- Penambahan dan pengurangan
- Perkalian
- Determinan
- Transpos
- Kebalikan (inverse)

- **Operasi vektor**

- Penambahan dan pengurangan
- Perkalian titik (dot product)
- Perkalian silang (cross product)

# Titik dan Garis

- Pembentukan titik dilakukan dengan mengkonversi suatu posisi titik koordinat dengan program aplikasi ke dalam suatu operasi tertentu menggunakan output. Random-scan (vektor ) system menyimpan instruksi pembentukan titik pada display list dan nilai koordinat menentukan posisi pancaran electron ke arah lapisan fosfor pada layer. Garis dibuat dengan menentukan posisi titik diantara titik awal dan akhir dari suatu garis.

# Algoritma pembentukan garis

- Persamaan garis menurut koordinat Cartesian adalah :  
 $y = m.x + b$
- Dimana  $m$  adalah slope (kemiringan) dari garis yang dibentuk oleh dua titik yaitu  $(x_1, y_1)$  dan  $(x_2, y_2)$ . Untuk penambahan  $x$  sepanjang garis yaitu  $dx$  akan mendapatkan penambahan  $y$  sebesar :  $\Delta y = m. \Delta x$

# Algoritma garis DDA

- DDA adalah algoritma pembentukan garis berdasarkan perhitungan  $\Delta x$  dan  $\Delta y$ , menggunakan rumus  $\Delta y = m \cdot \Delta x$ . Garis dibuat dengan menentukan dua endpoint yaitu titik awal dan titik akhir. Setiap koordinat titik yang membentuk garis diperoleh dari perhitungan, kemudian dikonversikan menjadi nilai integer.



# Algoritma garis DDA

Langkah-langkah pembentukan menurut algoritma DDA, yaitu :

1. Tentukan dua titik yang akan dihubungkan.
2. Tentukan salah satu titik sebagai titik awal ( $x_0, y_0$ ) dan titik akhir ( $x_1, y_1$ ).
3. Hitung  $x = x_1 - x_0$  dan  $y = y_1 - y_0$ .
4. Tentukan step, yaitu jarak maksimum jumlah penambahan nilai  $x$  maupun nilai  $y$  dengan cara :
  - ✓ bila nilai  $|y| > |x|$  maka  $\text{step} = \text{nilai } |y|$ .
  - ✓ bila tidak maka  $\text{step} = |x|$ .
5. Hitung penambahan koordinat pixel yaitu  $x\_increment = x / \text{step}$  dan  $y\_increment = y / \text{step}$ .
6. Koordinat selanjutnya ( $x+x\_inceregment, y+y\_increment$ ).
7. Posisi pixel pada layer ditentukan dengan pembulatan nilai koordinasi tersebut.
8. Ulangi step 6 dan 7 untuk menentukan posisi pixel selanjutnya, sampai  $x = x_1$  dan  $y = y_1$ .

# Contoh : DDA

Untuk menggambarkan algoritma DDA dalam pembentukan suatu garis yang menghubungkan titik (10,10) dan (17,16), pertama-tama ditentukan dx dan dy, kemudian dicari step untuk mendapatkan x\_increment dan y\_increment.

- ✓ Mencari  $\Delta x$  dan  $\Delta y$

$$\Delta x = x_1 - x_0 = 17 - 10 = 7$$

$$\Delta y = y_1 - y_0 = 16 - 10 = 6$$

- ✓ selanjutnya hitung dan bandingkan nilai absolutnya.

$$|x| = 7$$

$$|y| = 6$$

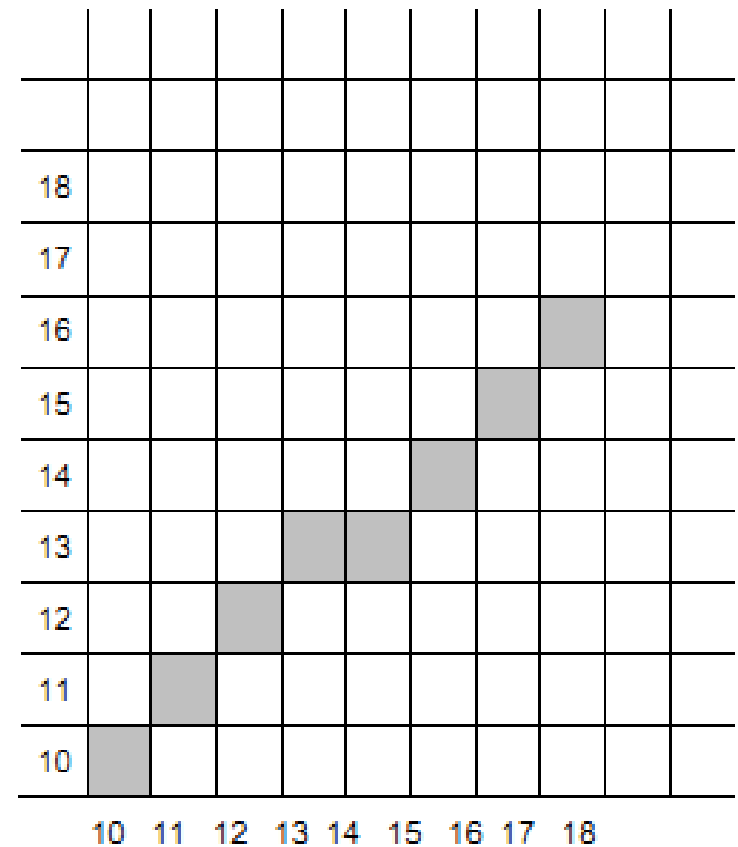
- ✓ Hitung step, karena  $|x| > |y|$ , maka  $\text{step} = |x| = 7$ , maka diperoleh :

$$x_{\text{inc}} = \Delta x / \text{Step} = 7 / 7 = 1$$

$$y_{\text{inc}} = \Delta y / \text{Step} = 6 / 7 = 0,86$$

# Contoh : DDA

k	x	y	round(x),round(y)
			(10,10)
0	11	10,86	(11,11)
1	12	11,72	(12,12)
2	13	12,58	(13,13)
3	14	13,44	(14,13)
4	15	14,3	(15,14)
5	16	15,16	(16,15)
6	17	16,02	(17,16)



# Algoritma Garis Bresenhem

- ✓ Prosedur untuk menggambar kembali garis dengan membulatkan nilai  $x$  atau  $y$  ke bilangan integer membutuhkan waktu, serta variabel  $x, y$  dan  $m$  merupakan bilangan real karena kemiringan merupakan nilai pecahan
- ✓ Bresenham mengembangkan algoritma klasik yang lebih menarik, karena hanya menggunakan perhitungan matematika dengan bilangan integer. Dengan demikian tidak perlu membulatkan nilai posisi setiap pixel setiap waktu.
- ✓ Algoritma garis Bresenham disebut juga midpoint line algorithm adalah algoritma konversi penambahan nilai integer yang juga dapat diadaptasi untuk menggambar sebuah lingkaran.

# Algoritma Garis Bresenhem

- ✓ Langkah-langkah untuk membentuk garis menurut algoritma ini adalah:
  1. Tentukan dua titik yang akan dihubungkan dalam pembentukan garis.
  2. Tentukan salah satu titik disebelah kiri sebagai titik awal ( $x_0, y_0$ ) dan titik lainnya sebagai titik akhir ( $x_1, y_1$ ).
  3. Hitung  $x$ ,  $y$ ,  $2x$ , dan  $2y - 2x$ .
  4. Hitung parameter  $p_0 = 2y - x$ .
  5. Untuk setiap  $x_k$  sepanjang jalur garis, dimulai dengan  $k = 0$  bila  $p_k < 0$  maka titik selanjutnya ( $x_{k+1}, y_k$ ) dan  $p_{k+1} = p_k + 2y$  bila tidak maka titik selanjutnya adalah ( $x_k + 1, y_k + 1$ ) dan  $p_{k+1} = p_k + 2y - 2x$ .
  6. Ulangi langkah nomor 5 untuk menentukan posisi pixel selanjutnya, sampai  $x = x_1$  dan  $y = y_k$ .

# Contoh: Bresenham

- ✓ Untuk menggambarkan algoritma Bresenham dalam pembentukan suatu garis yang menghubungkan titik  $(10,10)$  dan  $(17,16)$ , pertama-tama ditentukan bahwa titik  $(10,10)$  berada disebelah kiri merupakan titik awal, sedangkan  $(17,16)$  merupakan titik akhir

# Contoh: Bresenham

- ✓  $\Delta x = x_1 - x_0$  dan  $\Delta y = y_1 - y_0$
- ✓  $\Delta x = 7$  dan  $\Delta y = 6$
- ✓ parameter  $p_0 = 2 \Delta y - \Delta x$ 
  - ✓  $p_0 = 5$
- ✓ increment
  - ✓  $2 \Delta y = 12$
  - ✓  $2 \Delta y - 2 \Delta x = -2$

# TUGAS\_

1. Buat sebuah garis yang menghubungkan dari titik (20,10) sampai dengan titik (30,18) dengan menggunakan algoritma DDA.
2. Buat sebuah garis yang menghubungkan dari titik (20,10) sampai dengan titik (30,18) dengan menggunakan algoritma Bresenham.





ADA PERTANYAAN..????