PERTEMUAN 03
ALGORITMA GARIS

ADHI PRAHARA

Teknik Informatika. Fakultas Teknologi Industri

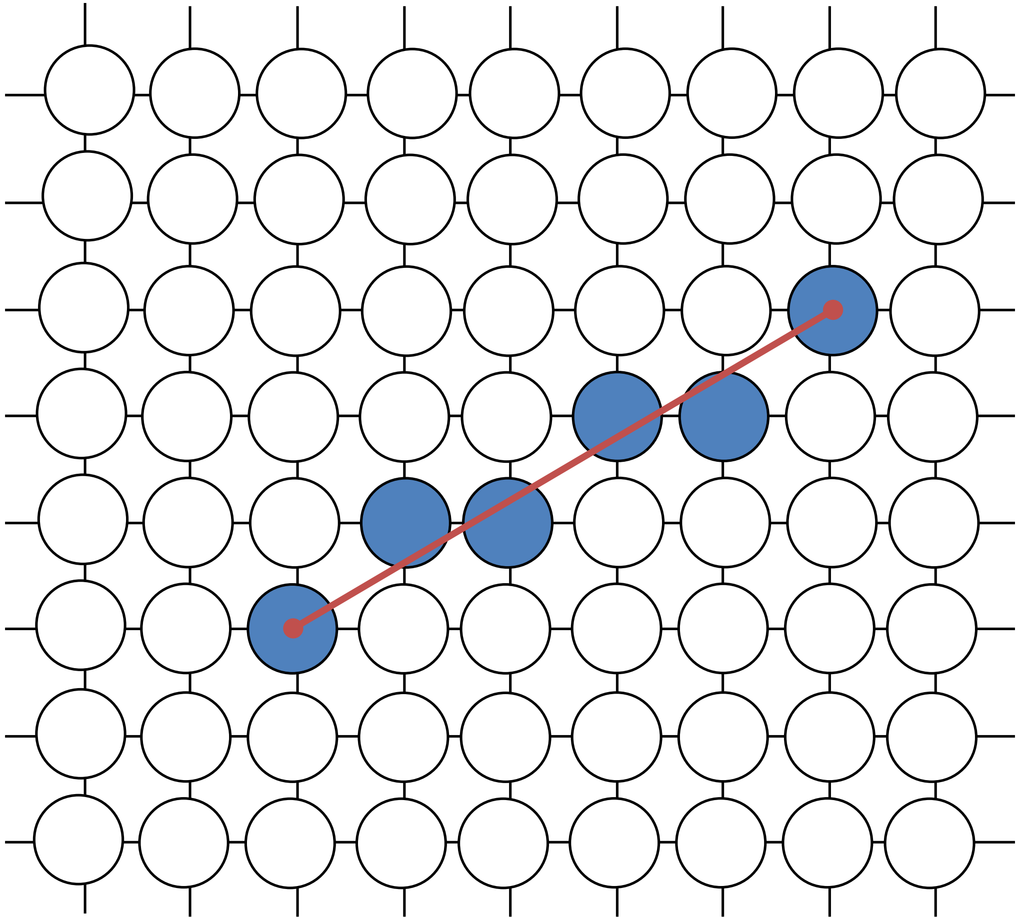
Universitas Ahmad Dahlan

CAPAIAN PEMBELAJARAN

- Menjelaskan konsep menggambar garis dalam komputer grafis
- Menjelaskan algoritma untuk membuat garis
- Mengetahui kelebihan dan kekurangan algoritma garis

GARIS?

- Garis : menghubungkan dua titik
- Menggambar garis itu mudah tapi bila harus menggambar garis dilayar maka muncul permasalahan
- Bila menggambar garis diatas layar yang berbasis piksel
- Piksel mana yang akan digambar selanjutnya?

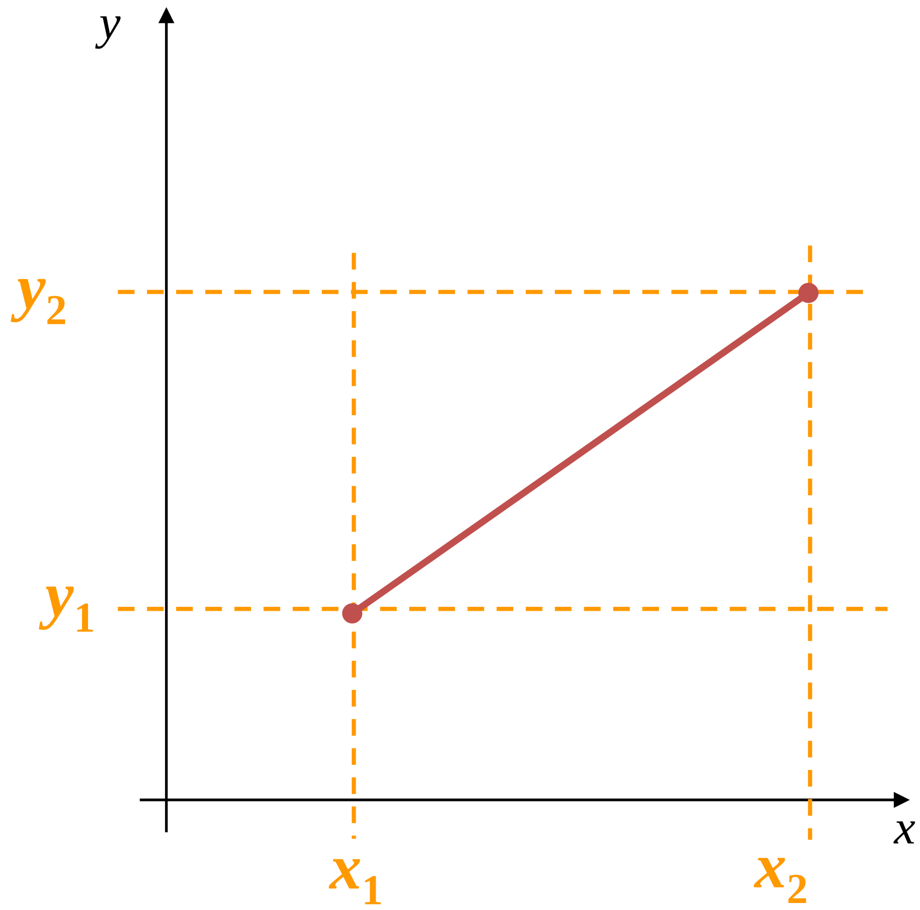


MENGGAMBAR GARIS

- Garis harus tampak bagus dengan meminimalkan jagged effect atau meminimalkan aliasing
- Mempunyai ketebalan garis yang seragam
- Membutuhkan waktu yang singkat

ALGORITMA GARIS

- Persamaan garis lurus:
$$y = mx + b$$
- m = gradient
- b = titik potong dengan sumbu y
- Bila ada dua titik yang akan dihubungkan (x_1, y_1) dan (x_2, y_2)
- Maka gradient (m) dapat dihitung dengan:
$$m = \frac{y_2 - y_1}{x_2 - x_1}$$
- Dan titik potong b dapat dihitung dengan:
$$b = y_1 - m\,x_1$$



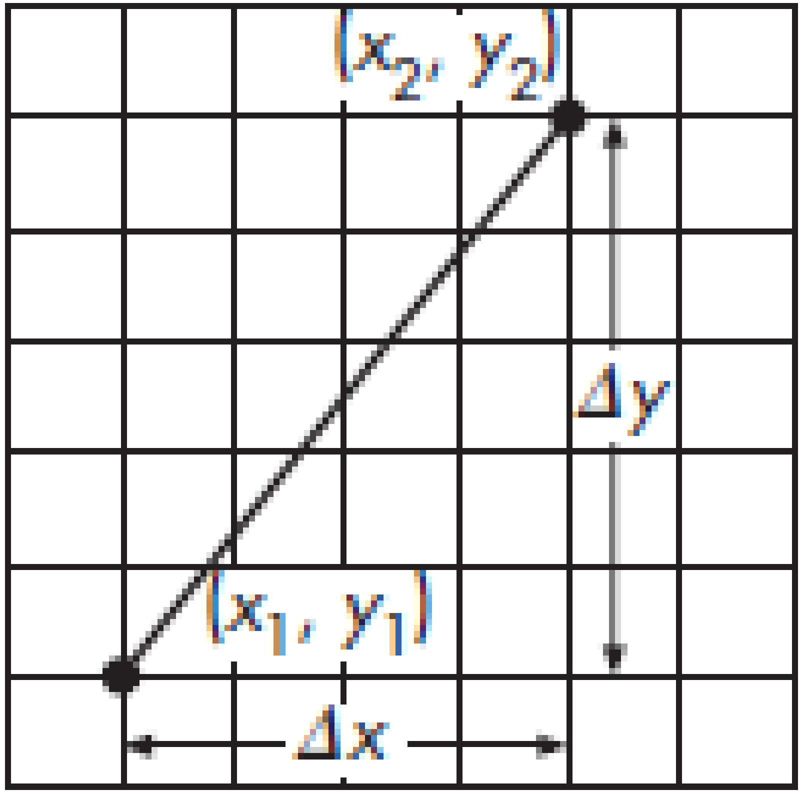
ALGORITMA GARIS

Bila dimisalkan:

$$y_2 - y_1 = \Delta y$$
$$x_2 - x_1 = \Delta x$$

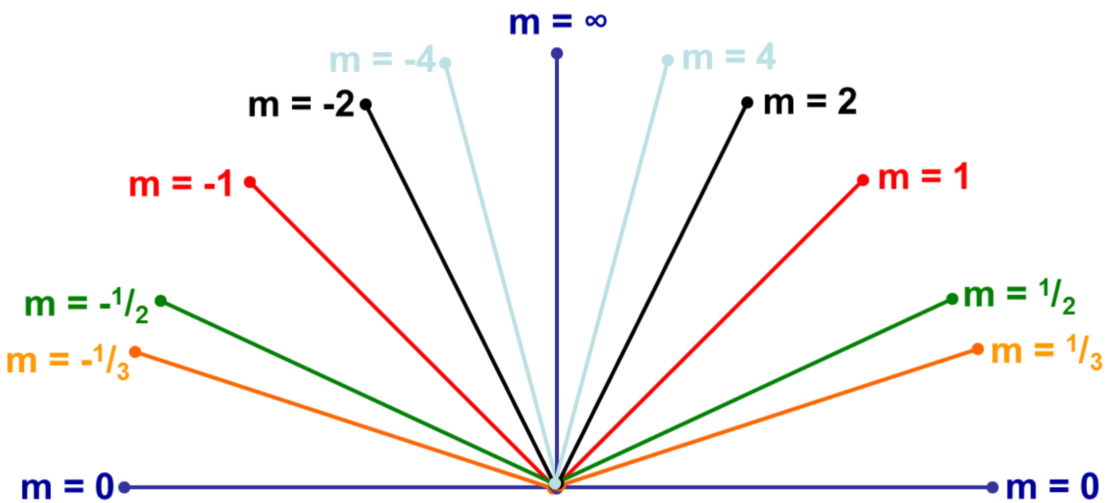
maka:

$$\Delta y = m\,\Delta x$$
$$\Delta x = \frac{\Delta y}{m}$$



SLOPE / KEMIRINGAN (m)

- Bila $|m| < 1$: garis mendekati garis horizontal
- Bila $|m| > 1$: garis mendekati garis vertikal
- Bila $m = 1$: garis miring 45°

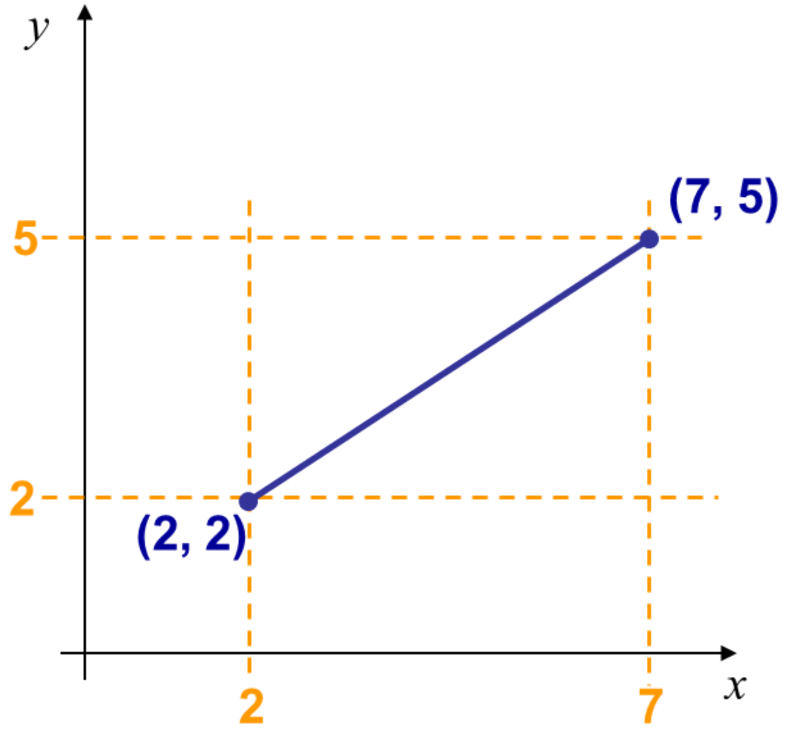


CONTOH

Bila ada dua titik yang akan dihubungkan

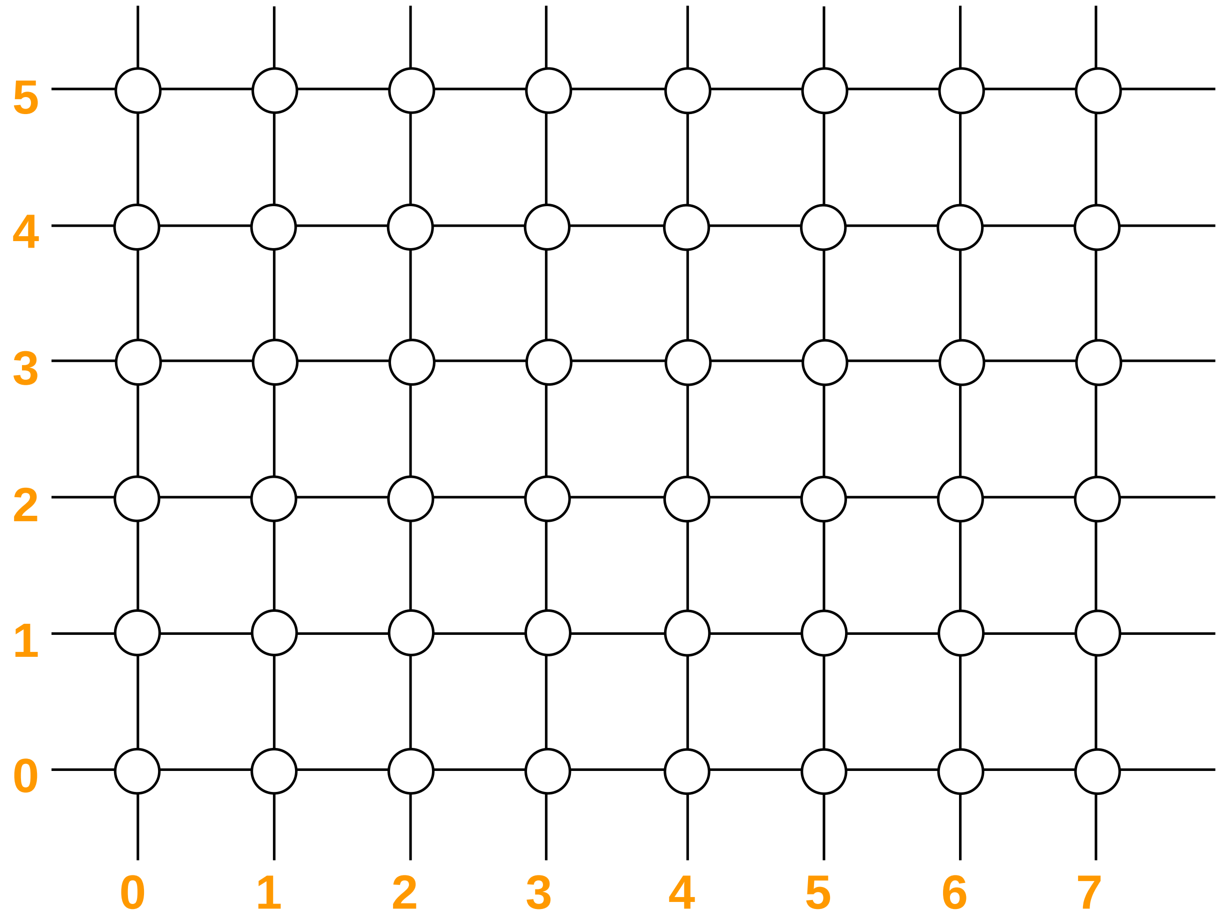
Misal

- $A(2,2)$ dan $B(7,5)$



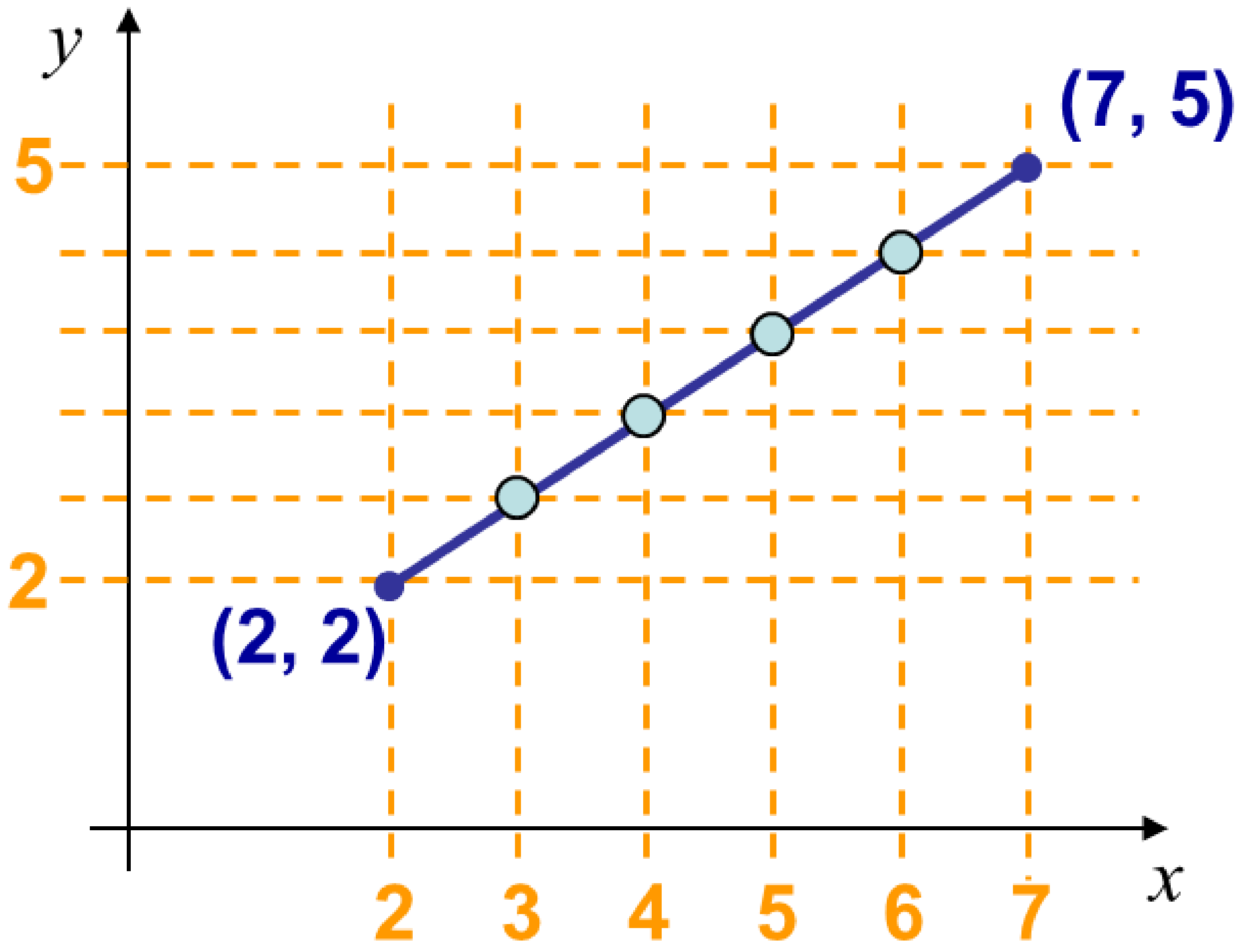
PENYELESAIAN

- $m = \frac{y_2 - y_1}{x_2 - x_1}$
- $m = \frac{5-2}{7-2} = \frac{3}{5}$
- $b = 2 - \frac{3}{5} * 2 = \frac{4}{5}$



PENYELESAIAN

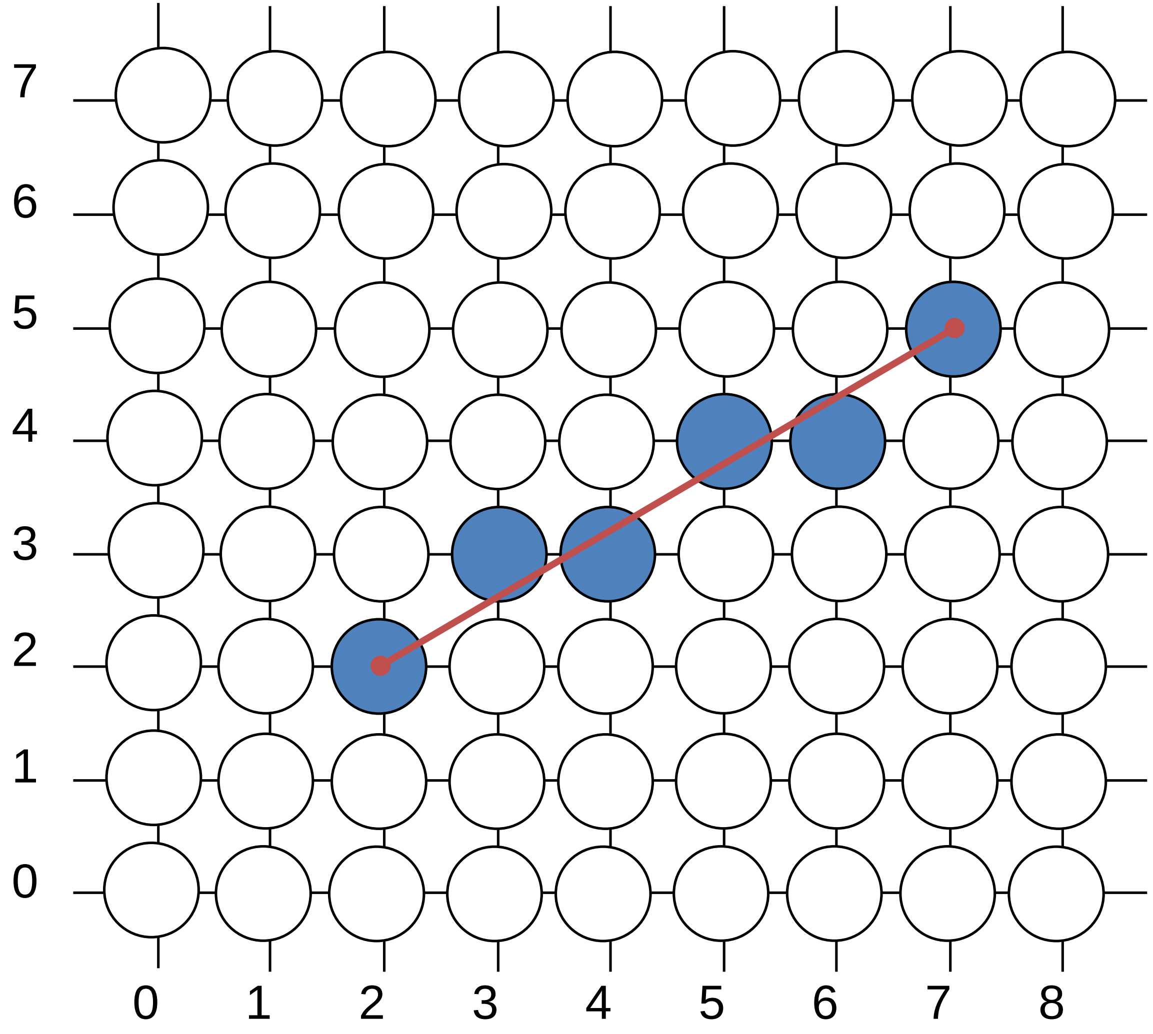
- $y(3) = 2\frac{3}{5}$
- $y(4) = 3\frac{1}{5}$
- $y(5) = 3\frac{4}{5}$
- $y(6) = 4\frac{2}{5}$



PENYELESAIAN

Dibulatkan:

- $y(3) = 2\frac{3}{5} \approx 3$
- $y(4) = 3\frac{1}{5} \approx 3$
- $y(5) = 3\frac{4}{5} \approx 4$
- $y(6) = 4\frac{2}{5} \approx 4$



PERMASALAHAN

- Solusi tersebut sangat lambat
- Dikarenakan:
  - Persamaan $y = mx + b$ membutuhkan operasi perkalian m dengan x
  - Membulatkan hasil dari y

SOLUSI?

Solusi sebelumnya:

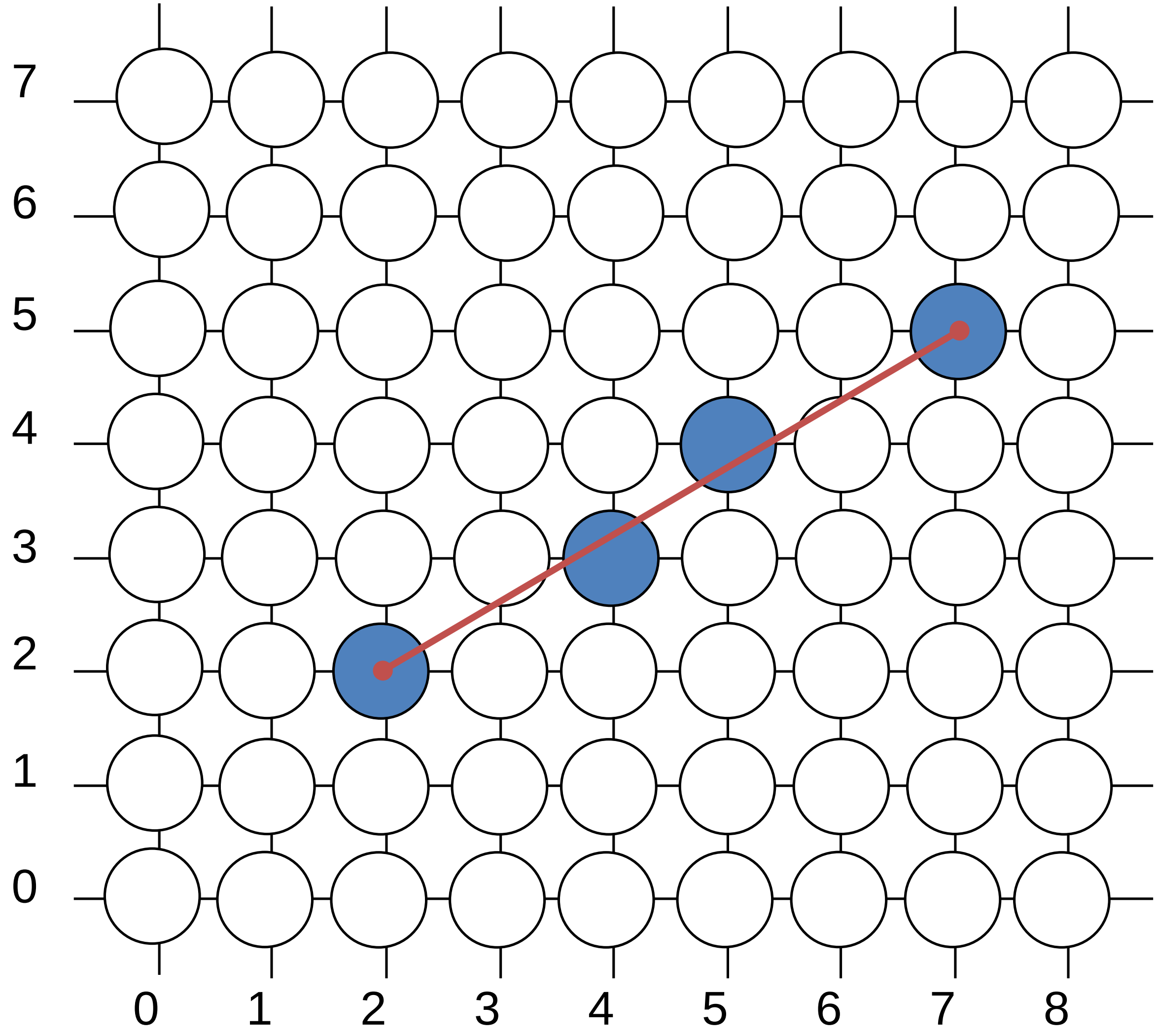
- Mencari nilai y dari setiap nilai x

Solusi lain

- Mencari nilai x dari setiap nilai y
- $x = \frac{y-b}{m}$
- $m = \frac{y_2-y_1}{x_2-x_1}$
- $b = y_1 - mx_1$

PENYELESAIAN

- $x(3) = 3\frac{2}{3} \approx 4$
- $x(4) = 5\frac{1}{3} \approx 5$
- Hasilnya tampak jelek

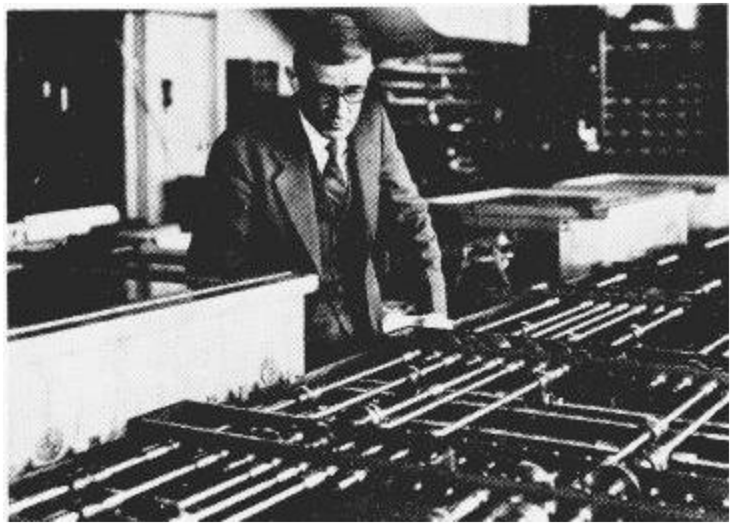


SOLUSI

- Menggunakan slope untuk memilih metode mana yang akan dipakai
- Jika slope pada garis $-1 \leq m \leq 1$
  - Digunakan solusi yang mencari nilai y dari setiap nilai x
- Selain dari nilai slope tersebut :
  - Digunakan solusi yang mencari nilai x dari setiap nilai y

ALGORITMA DDA

- DDA (Digital Differential Analyzer)
- DDA: algoritma scan-konversi garis dengan melakukan sampling pada garis di rentang Δx atau Δy
- Dioptimalkan untuk mempercepat kecepatan menggambar garis



The original differential analyzer was a physical machine developed by Vannevar Bush at MIT in the 1930's in order to solve ordinary differential equations.

ALGORITMA DDA

- Perhatikan contoh menggambar garis sebelumnya
- $(2,2), \left(3, 2\frac{3}{5}\right), \left(4, 3\frac{1}{5}\right), \left(5, 3\frac{4}{5}\right), \left(6, 4\frac{2}{5}\right), (7,5)$
- Koordinat x naik satu demi satu
- Koordinat y naik berdasarkan slope dari garis

ALGORITMA DDA

Ketika slope berada pada nilai $-1 \leq m \leq 1$ dan mulai dari titik awal

- Dengan menaikkan koordinat x dengan 1 maka y dapat dihitung:

$$y_{k+1} = y_k + m$$

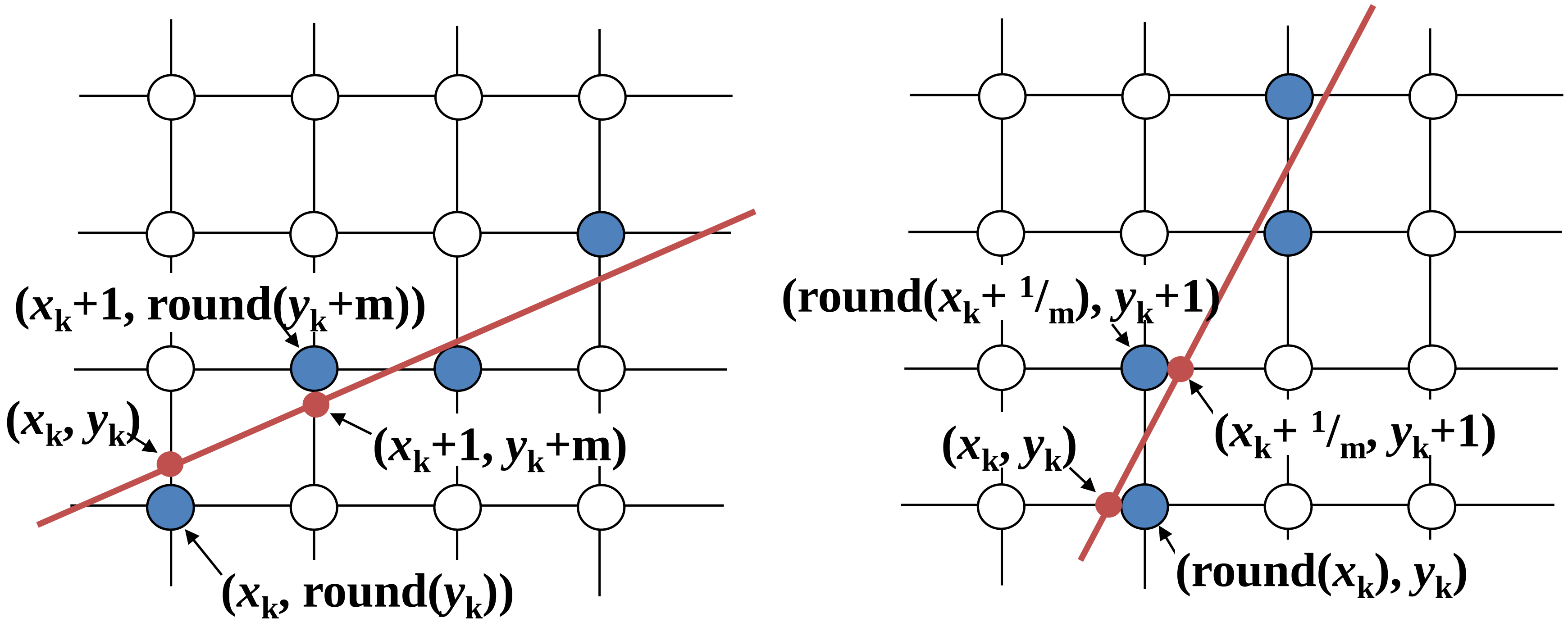
Bila slope diluar rentang nilai $-1 \leq m \leq 1$

- Dengan menaikkan koordinat y dengan 1 maka x dapat dihitung:

$$x_{k+1} = x_k + \frac{1}{m}$$

ALGORITMA DDA

Nilai hasil perhitungan harus dibulatkan agar cocok dengan nilai piksel



KESIMPULAN

- Algoritma DDA lebih cepat dari solusi sebelumnya
- Karena tidak ada operasi perkalian

Tapi ada beberapa masalah:

- Akumulasi error pada pembulatan dapat membuat piksel jauh dari garis
- Operasi pembulatan dan operasi pada floating point memakan waktu

ALGORITMA BRESENHAM

- Merupakan algoritma yang sekarang digunakan di computer grafis modern

Lebih baik dari algoritma DDA, karena:

- Incremental algorithm : nilai sekarang menggunakan nilai sebelumnya
- Digunakan untuk tipe data integer : menghindari operasi floating point
- Algoritma bresenham menggunakan fungsi keputusan untuk menentukan letak koordinat selanjutnya

ALGORITMA BRESENHAM

- Diberikan 2 titik (x_1, y_1) dan (x_2, y_2)
- Dicari titik penghubung di antara kedua titik tersebut
- Asumsi:
- $x_1 < x_2$
- $0 \leq m \leq 1$

ALGORITMA BRESENHAM

- Tentukan titik awal garis (x_1, y_1) dan akhir garis (x_2, y_2)

- Inisialisasi awal, hitung:
- Selisih lebar = $\Delta x = x_2 - x_1$
- Selisih tinggi = $\Delta y = y_2 - y_1$
- $2\Delta y = 2(y_2 - y_1)$
- Inisial parameter keputusan = $p_0 = 2\Delta y - \Delta x$

ALGORITMA BRESENHAM

- Setiap x_k di sepanjang garis, mulai dari $k = 0$, cek kondisi berikut:
- Jika $p_k < 0$ maka titik selanjutnya untuk digambar di : $(x_k + 1, y_k)$

$$p_{k+1} = p_k + 2\Delta y$$

- Selain itu maka titik selanjutnya untuk digambar di : $(x_k + 1, y_k + 1)$

$$p_{k+1} = p_k + 2\Delta y - 2\Delta x$$

- Ulangi langkah diatas sebanyak Δx

ALGORITMA BRESENHAM

- Bila $m > 1$
- Inisial parameter keputusan = $p_0 = 2\Delta x - \Delta y$
- Setiap y_k di sepanjang garis, mulai dari $k = 0$, cek kondisi berikut:
- Jika $p_k < 0$ maka titik selanjutnya untuk digambar di : $(x_k, y_k + 1)$

$$p_{k+1} = p_k + 2\Delta x$$

- Selain itu maka titik selanjutnya untuk digambar di : $(x_k + 1, y_k + 1)$

$$p_{k+1} = p_k + 2\Delta x - 2\Delta y$$

- Ulangi langkah diatas sebanyak Δy

CONTOH

- Bila diberikan 2 titik yang akan dihubungkan garis
- $A(20,10)$ dan $B(30,18)$
- Hitung:
- $m = \frac{y_2 - y_1}{x_2 - x_1} = \frac{\Delta y}{\Delta x} = \frac{8}{10} = 0.8$
- $\Delta x = 10$
- $\Delta y = 8$
- $p_0 = 2\Delta y - \Delta x = 6$ // inisial parameter keputusan
- $2\Delta y = 16$ // increment
- $2\Delta y - 2\Delta x = -4$

CONTOH

k	p_k	(x_{k+1}, y_{k+1})	k	p_k	(x_{k+1}, y_{k+1})
0	6	(21,11)	5	6	(26,15)
1	2	(22,12)	6	2	(27,16)
2	-2	(23,12)	7	-2	(28,16)
3	14	(24,13)	8	14	(29,17)
4	10	(25,14)	9	10	(30,18)