Mata Kuliah : Grafik Komputer

KONVERSI PEMINDAIAN

Karmilasari

Konversi Pemindaian/Konversi Scan

- Konversi pemindaian atau rasterisasi adalah proses menemukan piksel layar yang besinggungan dengan garis/poligon/ kurva.
- Caranya adalah dengan menemukan piksel yang bersinggungan tersebut kemudian menyalinnya ke suatu ruang citra dengan skala yang sesuai degnan piksel dalam layar.

Jenis Konversi Pemindaian

- Konversi Pemindaian Garis
  - Algoritma Incremental
  - Algoritma Midpoint
- Konversi Pemindaian Lingkaran
  - Algoritma Incremental
  - Algoritma Midpoint

Konversi Pemindaian Garis

- Menggambar GARIS
  - Proses menghubungkan dua titik pada layar raster
- Masalah :
  - Bila terdapat dua titik, P dan Q, pada suatu bidang datar dengan koordinat integer, maka masalahnya adalah bagaimana menentukan piksel-piksel berikutnya pada layar raster yang menghubungkan setiap unit segmen dari P dan berakhir di Q.

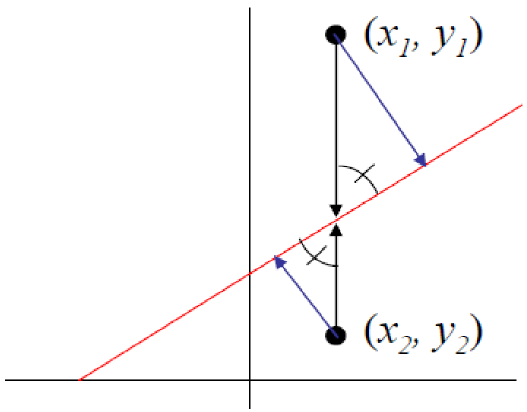
Konversi Pemindaian Garis : Menemukan Piksel Berikutnya

- Tergantung dari Jenis Garis :
  - **Horisontal** , gambarkan piksel P dan **inkremen/tambahkan nilai koordinat x**, satu persatu untuk mendapatkan piksel berikutnya
  - **Vertikal**, gambarkan piksel P dan **inkremen/tambahkan nilai koordinat y** , satu persatu untuk mendapatkan piksel berikutnya
  - **Diagonal**, gambarkan piksel P dan **inkremen/tambahkan nilai koordinat x dan y** , satu persatu untuk mendapatkan piksel berikutnya
- Secara umum yang perlu dilakukan :
  - Inkremen/tambahkan nilai koordinat x dengan 1 dan pilih titik terdekat dengan garis.
  - Bagaimana menemukan titik terdekat dengan garis ?

Konversi Pemindaian Garis :

Jarak Vertikal

- Mengapa menggunakan jarak vertikal sebagai ukuran mencari titik terdekat ?
  - Karena jarak vertikal sebanding dengan jarak yang sebenarnya
  - Ditunjukan dengan segitiga kongruen



- Pada gambar di atas, dengan segitiga yang sama terlihat bahwa jarak untuk garis (warna biru) adalah berbanding lurus dengan jarak vertikal ke baris (hitam) untuk setiap titik
- Oleh karena itu, titik dengan jarak vertikal yang lebih kecil untuk garis adalah yang paling dekat dengan garis

Konversi Pemindaian Garis : Algoritma Inkremental

- Dasar Algoritma
  - Gunakan persamaan garis yang menghubungkan titik P dan Q.
  - Mulailah dengan titik terkiri P, inkremen/tambahkan xi dengan 1 untuk menghitung $yi = mxi + B$

    dimana m = slope/kemiringan, B = intercept/memotong y
  - Lekatkan pixel pada (xi, Round(yi)), dimana

    $$\text{Round}\ (yi) = \text{Dasar}(0.5 + yi)$$

Konversi Pemindaian Garis : Algoritma Inkremental

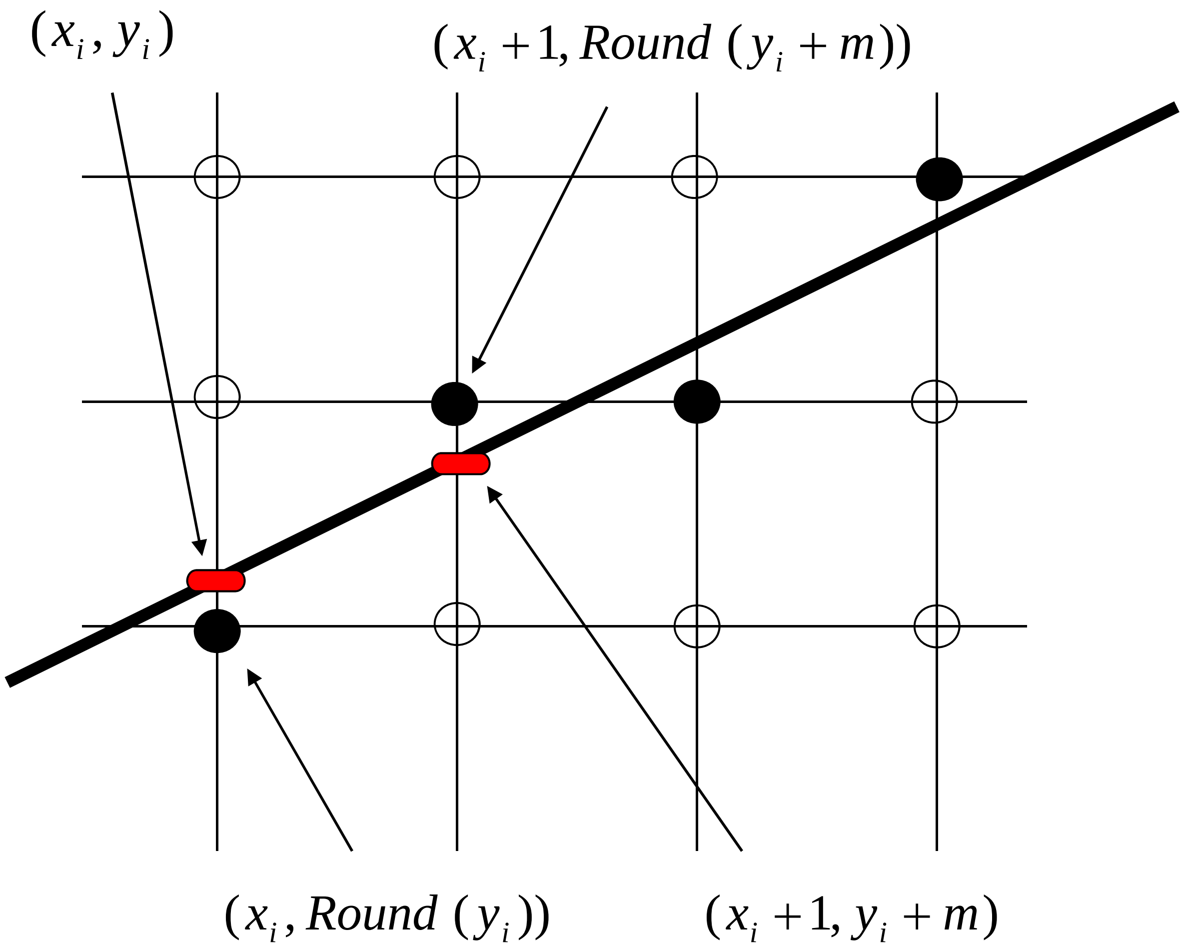
Algoritma Inkremental

- Setiap iterasi membutuhkan perkalian floating point
- Diperlukan modifikasi algoritma
  - $y_{i+1} = mx_{i+1} + B = m(x_i + \Delta x) + B = y_i + m\Delta x$

  Jika $\Delta x = 1$, maka $y_{i+1} = y_i + m$

- Pada setiap tahap, hitung inkremen/penambahan berdasarararkan tahap sebelumnya untuk menemukan nilai y berikutnya

Konversi Pemindaian Garis :
Algoritma Inkremental



Masalah :

- Pembulatan bilangan bulat membutuhkan waktu
- Variabel Y dan m harus bilangan real atau biner karena kemiringan adalah sebuah pecahan
- Diperlukan penyelesaian khusus untuk kasus garis-garis vertikal

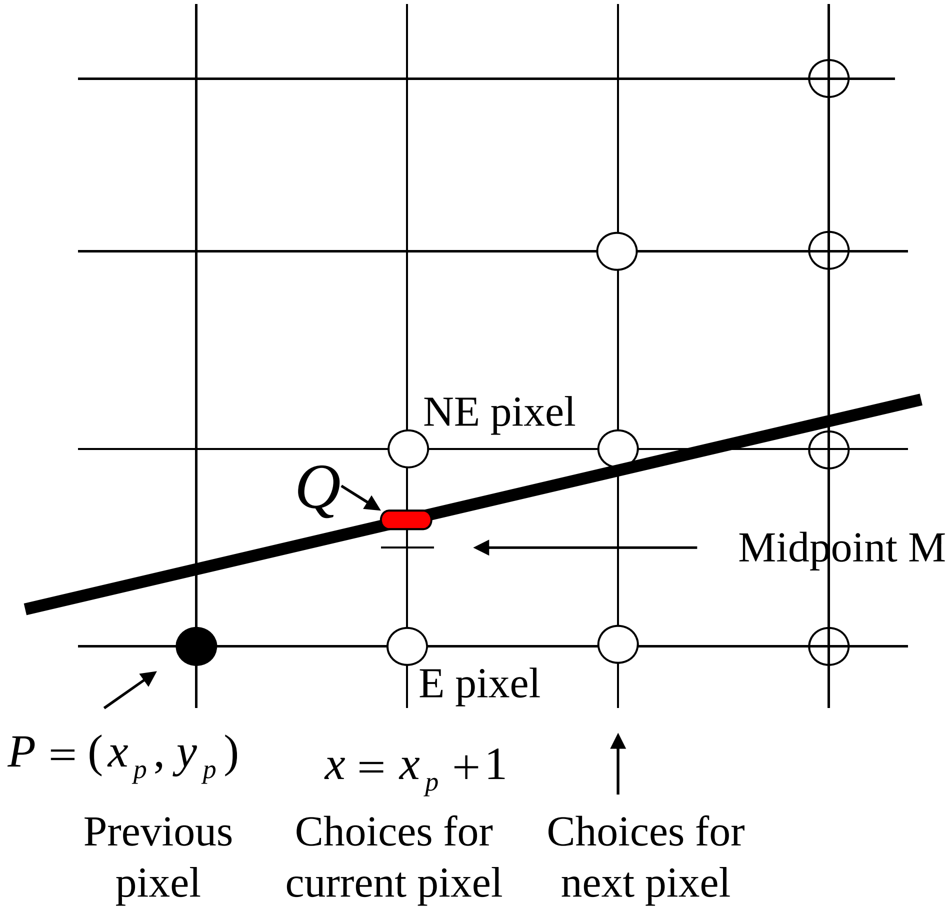
Konversi Pemindaian Garis :

Algoritma Midpoint Line/Titik Tengah Garis

- Asumsikan bahwa kemiringan garis adalah landai dan positif (0 < kemiringan < 1); kemiringan lain dapat ditangani dengan refleksi yang cocok berdasarkan prinsip axis/sumbu
- Sebut titik ujung kiri (x0,y0) dan titik ujung atas kanan (x1,y1)
- Asumsikan, pilih piksel P (xp, yp)
- Selanjutnya, pilih piksel arah kanan (piksel E) atau piksel arah kanan atas (piksel NE)
- Titik Q adalah titik perpotongan garis konversi pemindaian dengan garis grid $x = xp + 1$

Konversi Pemindaian Garis :

Algoritma Midpoint Line/Titik Tengah Garis



- Jalur ini melewati antara E dan NE
- Titik yang lebih dekat ke titik persinggungan Q harus dipilih
- Amati di sisi mana dari garis titik tengah M terletak:

  - E lebih dekat dengan garis jika titik tengah M terletak di atas garis, yaitu garis menyilang bagian bawah

  - NE lebih dekat dengan garis jika titik tengah M terletak di bawah garis, yaitu garis melintasi bagian atas
- Kesalahan, jarak vertikal antara pixel yang dipilih dan garis yang sebenarnya, adalah selalu <= ½
- Algoritma ini memilih NE sebagai pixel berikutnya untuk baris yang ditampilkan
- Sekarang, temukan cara untuk menghitung di sisi mana garis titik tengah terletak

GARIS

- Persamaan garis dituliskansebagai fungsi *f(x)*:

  $f(x) = m*x + B = dy / dx*x + B$

- Persamaan garis secara impllisit dituliskan :

  $F(x, y) = a*x + b*y + c = 0$ untuk koefisine a, b, c, dimana $a, b \neq 0$

  dari persamaan di atas : $y*dx = dy*x + B*dx$

  maka $a = dy$, $b = -dx$, $c = B*dx$, $a > 0$ for $y0 < y1$

- Properti (berdasarkan analisis)
  - $F(xm, ym) = 0$ dimana setiap titik M berada pada garis
  - $F(xm, ym) < 0$ dimana setiap titik M berada di atas garis
  - $F(xm, ym) > 0$ dimana setiap titik M berada di bawah garis
- Keputusan didasarkan pada nilai fungsi dari midpoint M pada $(xp + 1, yp + ½)$

Garis :

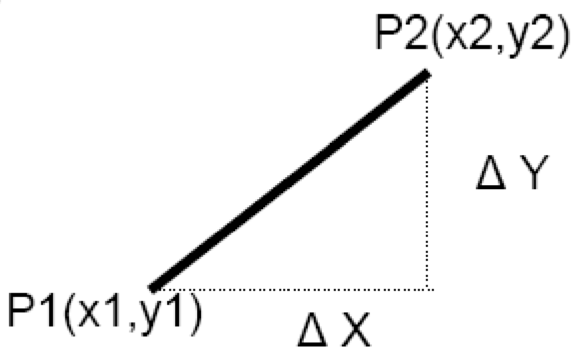
Algoritma Membuat Garis

- Algoritma DDA, adalah suatu algoritma pengkonversian suatu himpunan pixel–pixel menjadi suatu garis yang didasari atas perhitungan delta(x) dan delta(y);
- Algoritma *Bresenham* merupakan suatu algoritma yang dibuat oleh Bresenham yang tidak kalah akurat dan efisien dengan algoritma primitif lainnya.

Membuat Garis Bebas (Simple Digital Differential Analyzer/DDA)

- Garis yang membentang antara 2 titik, P1 dan P2, selalu membentuk sudut yang besarnya sangat bervariasi.
- Sudut yang terbentuk menentukan kemiringan suatu garis atau disebut ***gradient/ slop*** atau disimbolkan dengan parameter ***m***. Jika titik-titik yang membetuk garis adalah : **(x1,y1)** dan **(x2,y2)** maka

$$m = \frac{\Delta y}{\Delta x}, \mathrm{m} = \frac{\mathrm{y2 - y1}}{\mathrm{x2 - x1}}$$



Membuat Garis Bebas (Simple Digital Differential Analyzer/DDA)

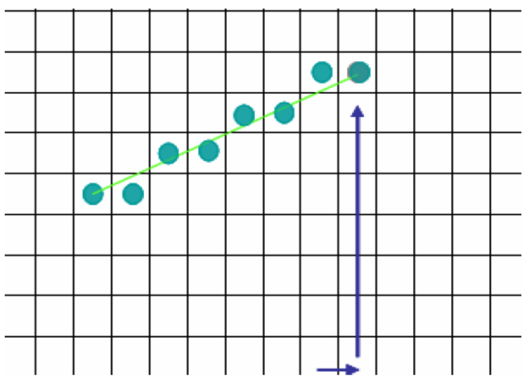
- Algoritma DDA bekerja atas dasar penambahan nilai x dan nilai y.
- Pada garis lurus, turunan pertama dari x dan y adalah konstanta.
- Sehingga untuk memperoleh suatu tampilan dengan ketelitian tinggi, suatu garis dapat dibangkitkan dengan menambah nilai x dan y masing-masing sebesar Δx dan Δy.
- Kondisi ideal ini sukar dicapai, karenanya pendekatan yang mungkin dilakukan adalah berdasarkan piksel-piksel yang bisa dialamati/dicapai atau melalui penambahan atau pengurangan nilai x dan y dengan suatu besaran dan membulatkannya ke nilai integer terdekat.

$$\begin{aligned} x_{k+1} &= x_k + \partial x \\ &= x_k + 1 \end{aligned}$$

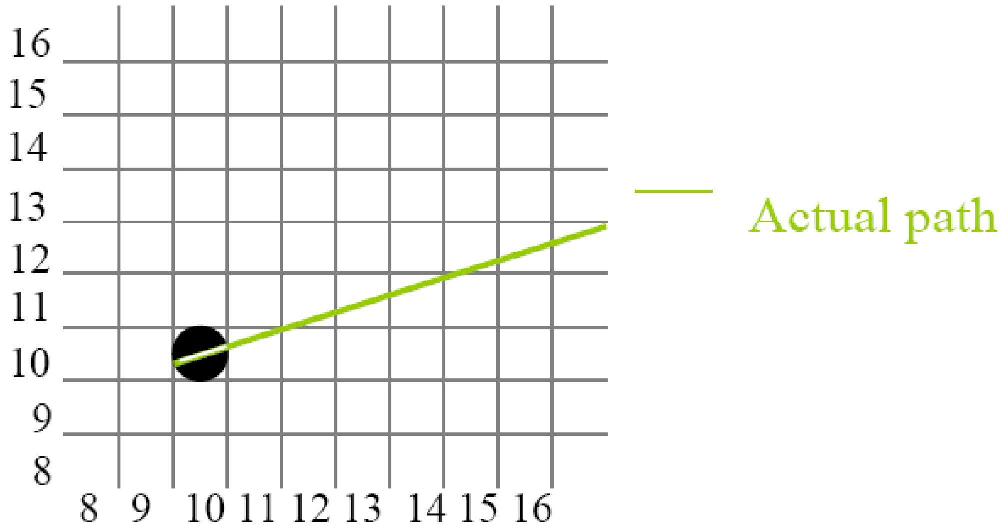
$$\begin{aligned} y_{k+1} &= y_k + \partial y \\ &= y_k + m \cdot \partial x \\ &= y_k + m \cdot (1) \end{aligned}$$

Endpoint :

P1(0,0), P2(7,4)

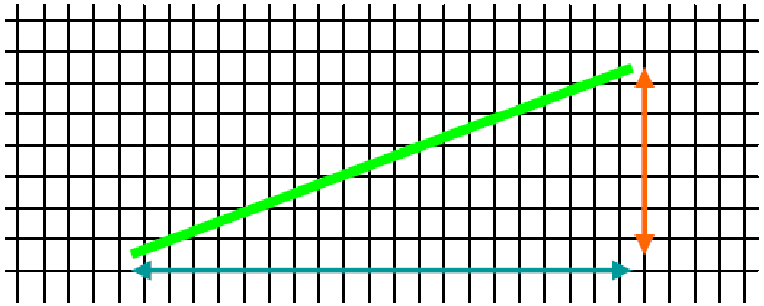


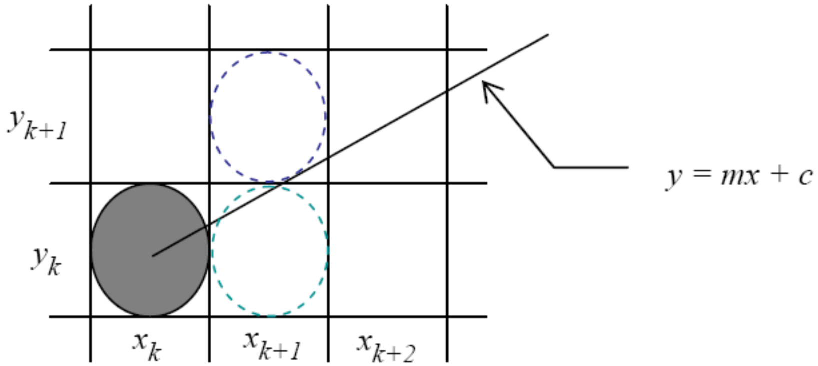
Menggambar Garis : Algoritma Bresenham

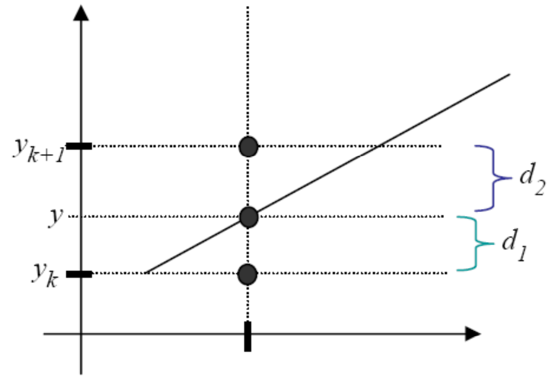


- Pixel selanjutnya ?
- Algoritma Bresenham memilih titik terdekat dari actual path
- Setiap sampling akan diinkremen menjadi 1 atau 0.

Menggambar Garis : Algoritma Bresenham







- Kondisi awal :Jika m < 1, maka m bernilai positif
- Bresenham melakukan inkremen 1 untuk x dan 0 atau 1 untuk y.
- Jika current pixel (xk,yk)
- Dimanakah pixel berikutnya akan di-plot, apakah di (xk+1, yk+1), (xk+1, yk), atau (xk, yk+1)?

Menggambar Garis : Algoritma Bresenham

- Parameter keputusan, *pk:*

$$p_k = \Delta x(d_1 - d_2)$$

p_k	*negatif*	$d_1 < d_2$; pixel pada scanline y_k adalah di dekat actual path • plot (x_{k+1}, y_k)
p_k	*positif*	$d_1 > d_2$; pixel pada scanline y_{k+1} adalah di dekat actual path • plot (x_{k+1}, y_{k+1})

Algoritma Bresenham untuk |m| < 1:

1. Input 2 endpoints, simpan endpoints kiri sebagai $(x0, y0)$.
2. Panggil frame buffer (plot titik pertama)
3. Hitung konstanta Δx, Δy, $2\Delta y$, $2\Delta y - 2\Delta x$dan nilai awal parameter keputusan $p0 = 2\Delta y - \Delta x$
4. Pada setiap xk sepanjang garis, dimulai dari $k=0$, ujilah :

   Jika $pk < 0$, maka plot$(xk+1, yk)$ dan

   $$pk+1 = pk + 2\Delta y$$

   Jika tidak maka plot $(xk+1, yk+1)$ dan

   $$pk+1 = pk + 2\Delta y - 2\Delta x$$

5. Ulangi tahap 4 Δxkali

Menggambar Garis :
Algoritma Bresenham

- Latihan : Hitunglah posisi piksel hingga membentuk sebuah garis yang menghubungkan titik (12,10) dan (17,14) !

Jawab :

1. $(x0, y0) = (12, 10)$

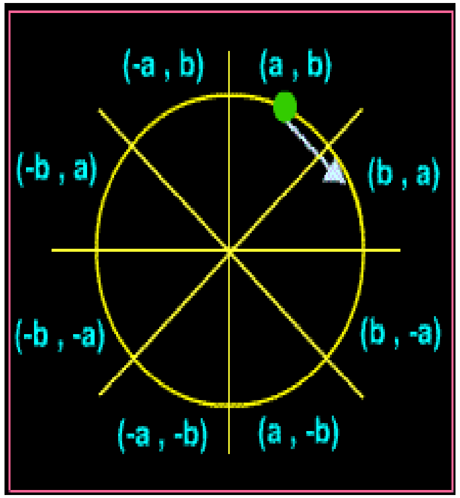
2. $\Delta x = 5, \Delta y = 4, 2\Delta y = 8, 2\Delta y - 2\Delta x = -2$

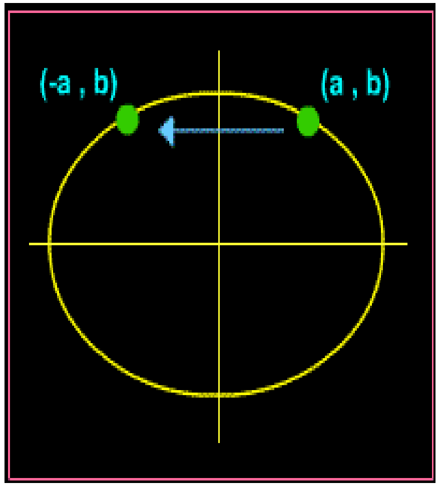
3. $p0 = 2\Delta y - \Delta x = 3$

k	p_k	(x_{k+1}, y_{k+1})
0	3	(13, 11)
1	1	(14, 12)
2	-1	(15, 12)
3	7	(16, 13)
4	5	(17, 14)

Konversi Pemindaian Lingkaran

- Algoritma :
  - Algoritma Simetri
  - Algoritma Inkremental
  - Algoritma Midpoint
- Konsepnya : Bila diketahui lingkaran dengan radius r dan posisi tengah pixel (xc ,yc), selanjutnya dapat diatur atau ditentukan sesuai kondisi tertentu suatu algoritma perhitungan yang bertitik pusat pada koordinat origin (0, 0).

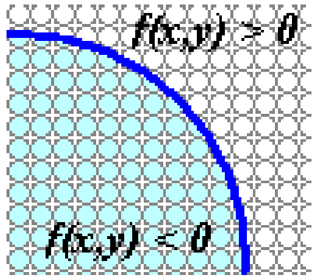




Konversi Pemindaian Lingkaran

Fungsi Discriminator

Diketahui bahwa : $(x - x_0)^2 + (y - y_0)^2 = r^2$

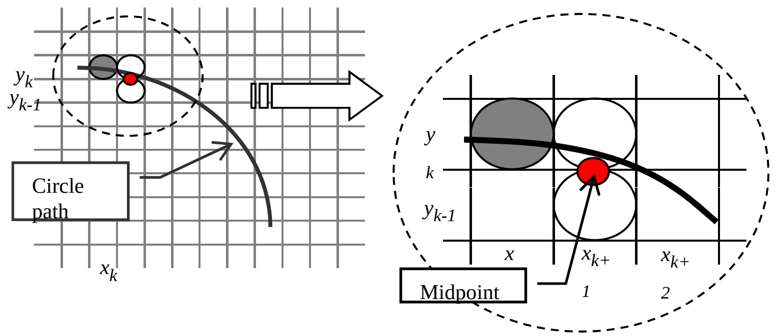


dan dapat ditulis sebagai suatu fungsi : $f(x,y) = x^2 + y^2 - r^2$

Fungsi Discriminator :

$f(x, y)$
- **f(x,y) < 0 untuk titik di dalam lingkaran**
- **f(x,y) > 0 untuk titik di luar lingkaran**
- **f(x,y) = 0 untuk titik yang terletak pada lingkaran**

- **Algoritma Titik Tengah Lingkaran (Midpoint Circle Algorithm)**
  - Bila diketahui suatu titik : (x_k,y_k), maka titik berikutnya apakah di(x_k+1, y_k), or (x_k+1, y_k-1) ?
  - Misal titik tengahnya (midpoint) : $(x_k+1, y_k) = 0.5$
  - Gunakan fungsi discriminator untuk mendapatkan :
    $f(x,y) = x^2 + y^2 - r^2$



Konversi Pemindaian Lingkaran

- Algoritma Titik Tengah Lingkaran ....*(lanjutan)*

Dengan menggunakan *midpoint* di antara 2 kandidat pixel, kita dapat mencari *Parameter Keputusan,* P_k, untuk mendapatkan plot pixel berikutnya :

$$P_k = f(x_k + 1, y_k - ½) \quad = \quad (x_k + 1)^2 + (y_k - ½)^2 - r^2$$

Jika -ve, titik tengah berada di dalam lingkaran,

plot = $(\mathbf{x_k+1, y_k})$,

Update *P* : $f(x+1, y) = (x + 1)^2 + y^2 - r^2$

$f(x+1, y) = (x^2 + 2x + 1) + y^2 - r^2$

$f(x+1, y) = f(x, y) + 2x + 1$

P_{k+1} P_k

Inkremen : $P +\ = \mathbf{2x + 1}$

Jika + ve, titik tengah berada di luar lingkaran,

plot = $(\mathbf{x_k+1, y_k-1})$

Update *P*: $f(x+1, y-1) = (x + 1)^2 + (y-1)^2 - r^2$

$f(x+1, y-1) = (x^2 + 2x + 1) + (y^2 - 2y + 2 - r^2)$

$f(x+1, y-1) = f(x, y) + 2x - 2y + 1$

P_{k+1} P_k

Inkremen : $P +\ = \mathbf{2x - 2y + 1}$

Terima Kasih