

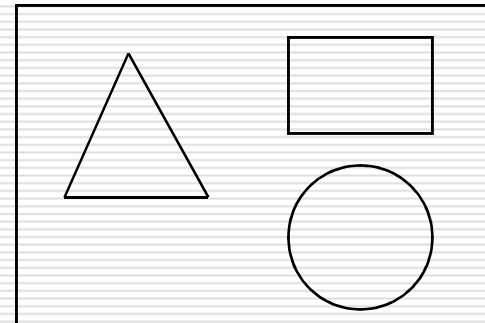
Area Filling

- Setelah berhasil membangun objek grafis maka langkah selanjutnya adalah melakukan customisasi terhadap atribut objek grafis
- Atribut dari objek grafis dapat berupa:
 - Ukuran garis batas
 - Tipe garis batas
 - Warna garis batas
 - Warna objek (Fill color / Area Filling)

Area Filling

- Dalam grafis, selain digunakan primitif-primitif grafis, diperlukan algoritma khusus untuk mengisi warna. Untuk pengisian warna pada grafik diterapkan pada objek yang memiliki *hole*.

- Contoh :



Area Filling

Beberapa algoritma yang dapat digunakan untuk melakukan *area filling*, yaitu :

- Scan Line
- Boundary Fill Algorithm
- Flood Fill Algorithm

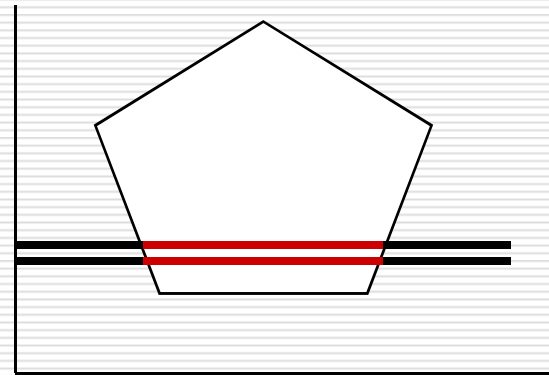
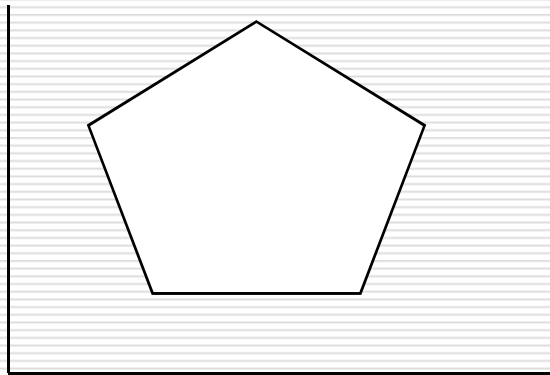
Scan Line Fill Algorithm

- ❑ Dimulai dari kiri atas layar
- ❑ Bergerak ke kanan hingga batas kanan layar
- ❑ Ulangi untuk baris-baris berikutnya
- ❑ Jika ditemukan suatu batas area, maka dilakukan operasi *putpixel* hingga ditemukan batas area berikutnya

- ❑ **Kelemahan :**
 - Algoritma akan gagal jika ditemui titik-titik ujung, sehingga perlu penanganan khusus.
 - → Solusi : periksa semua informasi titik ujung berdasarkan list vektor yang dimiliki setiap primitif.

Scan Line Fill Algorithm

□ Ilustrasi Proses Scan Line



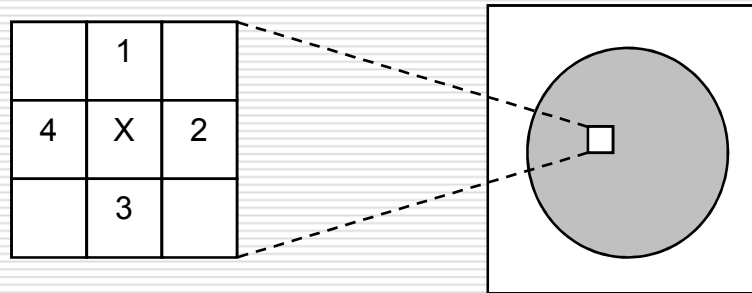
□ Ada masalah ??

Boundary Fill

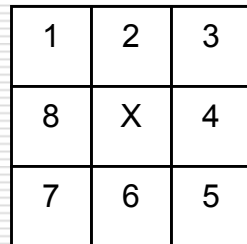
- Dengan teknik ini pengisian warna dilakukan berdasarkan penentuan titik awal.
- Posisi awal diletakkan pada *area filling*.
- Kemudian semua titik tetangga diuji terhadap batas area
- Ada 2 metoda untuk mencari titik tetangga :
 - 4 Connected
 - 8 Connected

Boundary Fill

□ 4 Connected



□ 8 Connected



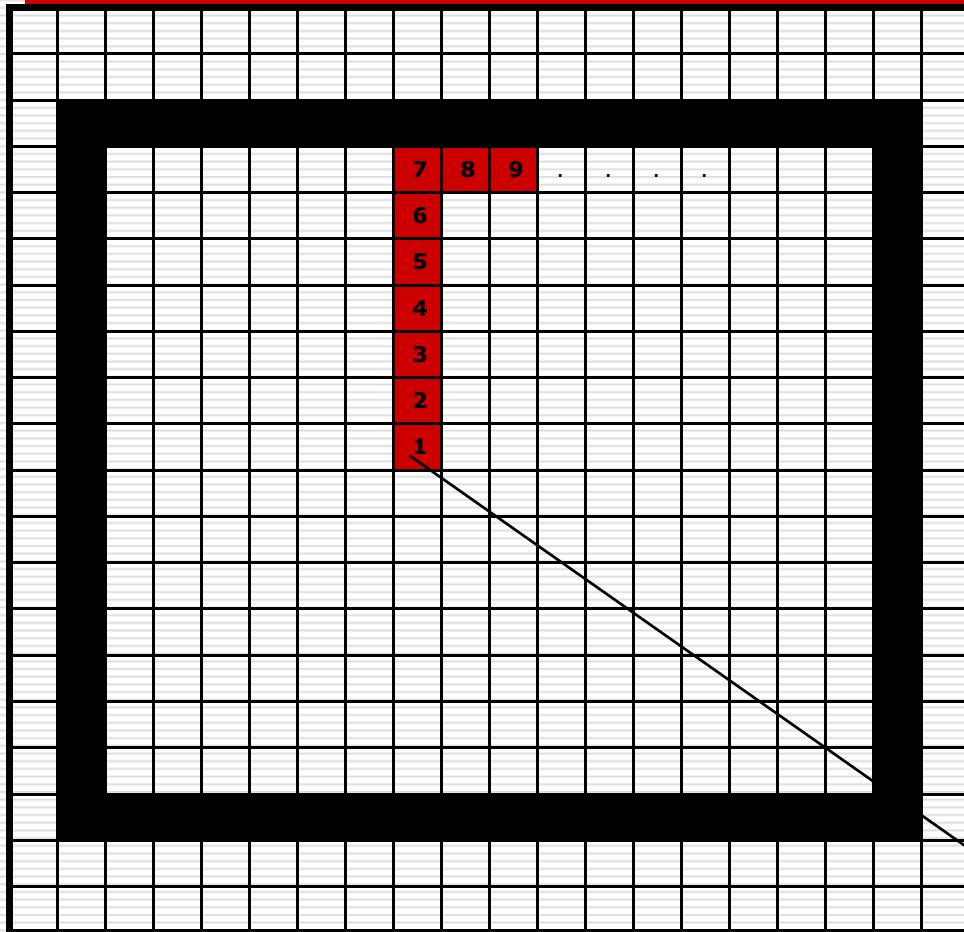
Algoritma 4-Connected

```
Procedure BFill (x,y,fillcol,boundcol);
Var
    Currcol : integer;
Begin
    Currcol ← getpixel(x,y);
    If (Currcol<>boundcol) and (Currcol<>fillcol) then
        Begin
            Putpixel (x,y,fillcol);
            Bfill(x+1,y,fillcol,boundcol);
            Bfill(x-1,y,fillcol,boundcol);
            Bfill(x,y+1,fillcol,boundcol);
            Bfill(x,y-1,fillcol,boundcol);
        End;
    End;
```

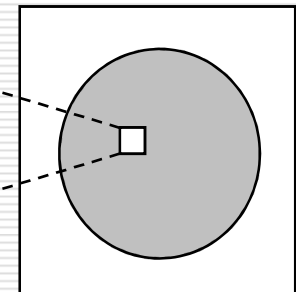

Algoritma 8-Connected

```
Procedure BFill (x,y,fillcol,boundcol);
Var
    Currcol : integer;
Begin
    Currcol ← getpixel(x,y);
    If (Currcol<>boundcol) and (Currcol<>fillcol) then
        Begin
            Putpixel (x,y,fillcol);
            Bfill(x-1,y-1,fillcol,boundcol);
            Bfill(x,y-1,fillcol,boundcol);
            Bfill(x+1,y-1,fillcol,boundcol);
            Bfill(x-1,y-1,fillcol,boundcol);
            Bfill(x+1,y-1,fillcol,boundcol);
            Bfill(x-1,y+1,fillcol,boundcol);
            Bfill(x,y+1,fillcol,boundcol);
            Bfill(x+1,y+1,fillcol,boundcol);
        End;
    End;
```

Boundary Fill



	1	
4	X	2
	3	



BFill (x,y,Red,Black);

Seed position (x,y)

Boundary Fill

- ❑ Bila Area Fill menjadi atribut dari objek maka warna batas dari proses ini akan sama dengan warna batas objek.
- ❑ Ingat bahwa boundry objek tidak boleh ada lubang

Flood Fill Algorithm

- Proses flood fill mirip dengan boundary fill, dimana inti dari proses ini adalah mengubah suatu warna menjadi warna yang lain

Flood Fill Algorithm

```
Procedure floodFill (x,y,fillColor,oldColor);
Begin
  If getpixel(x,y)= oldColor then
  Begin
    Putpixel (x,y,fillColor);
    floodFill (x+1,y,fillColor,oldColor);
    floodFill (x-1,y,fillColor,oldColor);
    floodFill (x,y+1,fillColor,oldColor);
    floodFill (x,y-1,fillColor,oldColor);
  End;
End;
```

-
- ❑ Apa perbedaan dari Boundary Fill dan Flood fill, kapan kita memakai Flood fill dan kapan kita memakai Boundary Fill
 - ❑ Untuk menjamin keamanan dari algoritma fill yang digunakan maka sebaiknya ditambahkan sebuah parameter penghentian yang lain, yaitu posisi dari pixel yang diproses apakah masih didalam objek atau sudah diluar
 - ❑ Ingat bahwa area fill dan garis batas merupakan 2 buah atribut yang berbeda

