

Praktikum 11: Modeling & Simulasi

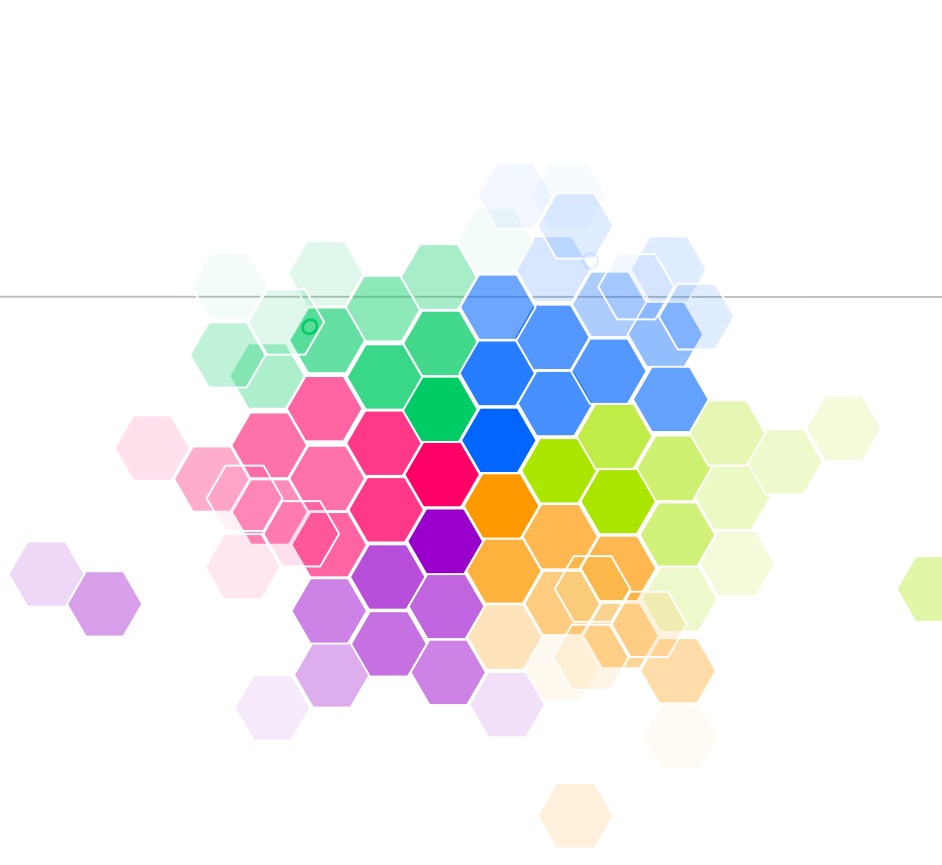


CELLULAR AUTOMATA 2D

Tujuan

- 1 Cellular Automata 2D
- 2 Tetangga
- 3 Aturan Dalam CA 2D
- 4 Game of Life
- 5 Pattern Generator

CELLULAR AUTOMATA 2D



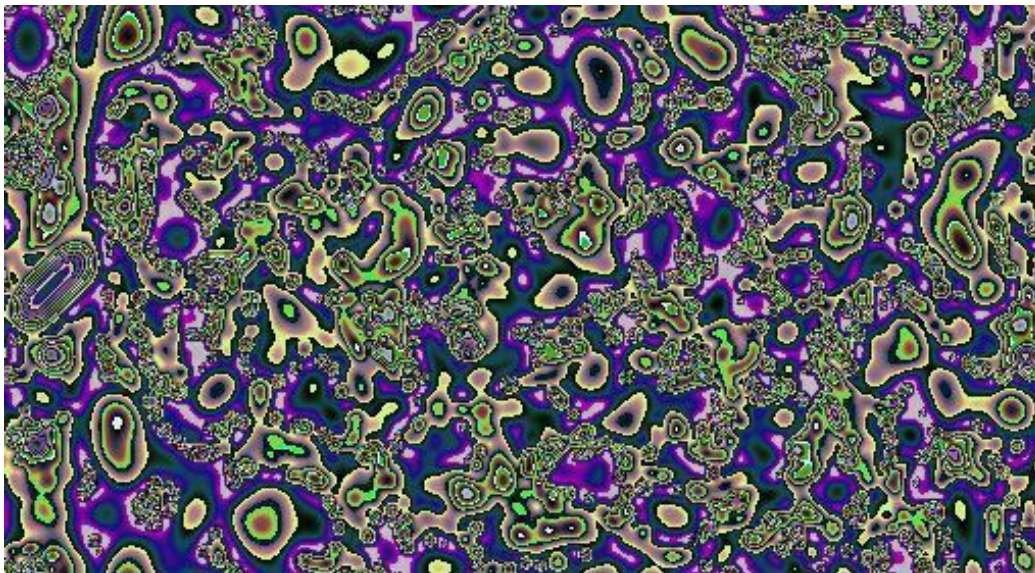
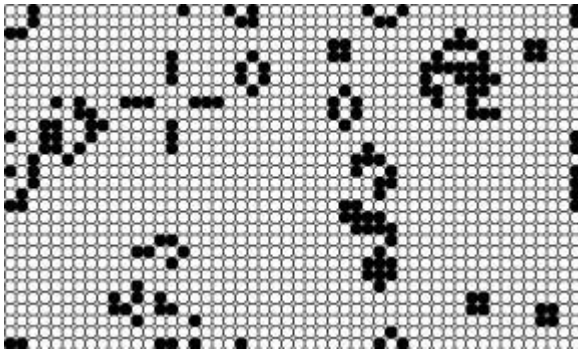
Cellular Automata 2D adalah bentuk planar dari susunan sel-sel yang mengikuti aturan-aturan perubahan akibat hubungan sel dengan sel-sel tetangganya.

CA 2D banyak digunakan dalam:

1. Pattern Generator
2. Sebaran Fluida
3. Simulasi Pedestrian dan Antrian

CA 2D menggunakan data spasial/peta.

Contoh-contoh CA 2D



TETANGGA

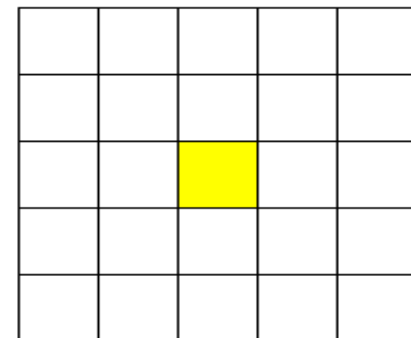
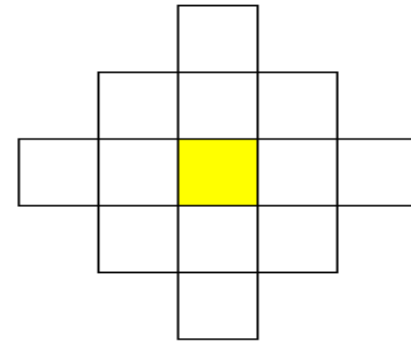
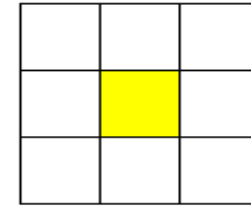
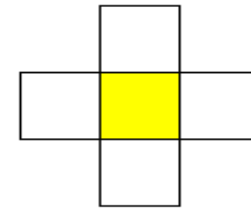
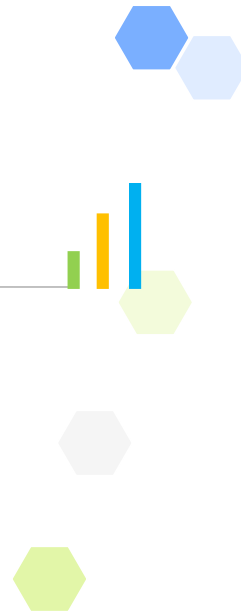
Dalam CA 2D adalah beberapa bentuk tetangga yang fundamental antara lain:

4 tetangga (3x3)

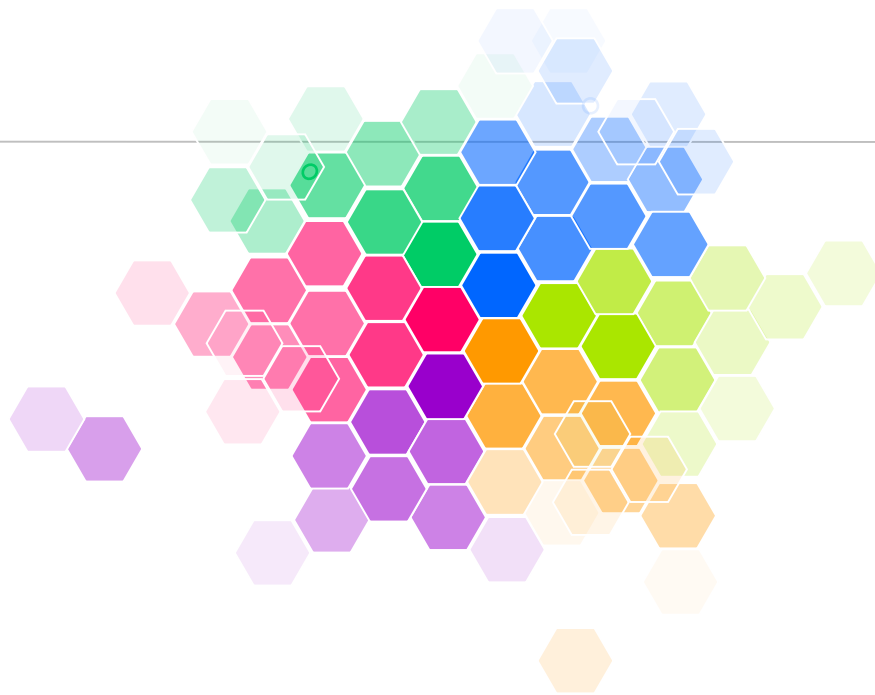
8 tetangga (3x3)

12 tetangga (5x5)

24 tetangga (5x5)



ATURAN CELLULAR AUTOMATA 2D



Aturan dalam CA 2D dinyatakan sebagai “perubahan nilai sel akibat keadaan sel dan tetangganya”



Aturan dibentuk berdasarkan model dari kehidupan nyata atau permasalahan yang akan disimulasikan.



Aturan dasar dalam CA 2D dinyatakan dalam GAME OF LIFE.

GAME OF LIFE

Game of Life adalah bentuk Cellular Automata 2D yang sederhana.

Setiap sel mempunyai 2 state (keadaan): 0 = mati dan 1 = hidup.

Aturan dalam Game of Life:

1. Sebuah sel hidup akan terus hidup bila mempunyai 2 atau 3 tetangga yang hidup
2. Sebuah sel mati menjadi hidup jika mempunyai 3 tetangga yang hidup.





Game of Life

```
n=input('Lattice size = ');
pd=input('Density (probability) = ');

%Initial state
s=zeros(n+2,n+2);
for i=2:n+1
    for j=2:n+1
        r=rand;
        if(r<pd)
            s(i,j)=1;
        end
    end
end

u=255*s(2:n+1,2:n+1);
figure(1), image(u), colormap(gray(255))
```



```

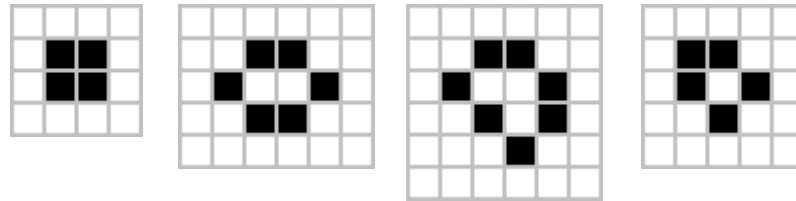
%Iterasi sebanyak 100 kali
for iter=1:100
    %Update state dengan aturan Game of Life
    st=s;
    for i=2:n+1
        for j=2:n+1
            t_hidup=st(i-1,j-1)+st(i-1,j)+st(i-1,j+1);
            t_hidup=t_hidup+st(i,j-1)+st(i,j)+st(i,j+1);
            t_hidup=t_hidup+st(i+1,j-1)+st(i+1,j)+st(i+1,j+1);
            if(st(i,j)==1)
                if((t_hidup<2)|| (t_hidup>3))
                    s(i,j)=0;
                end
            else
                if(t_hidup==3)
                    s(i,j)=1;
                end
            end
        end
    end
end
u=255*s(2:n+1,2:n+1);
figure(1), image(u), colormap(gray(255))
drawnow;
end

```

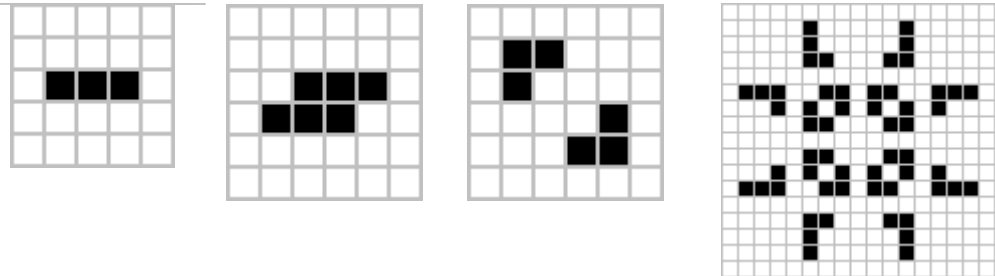
GAME OF LIFE

Perhatikan hasil dari Game of Life. Ada beberapa keadaan yang bisa diperoleh dari hasil ini:

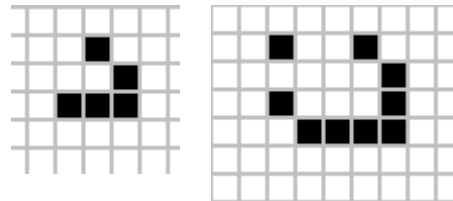
1. Still Life



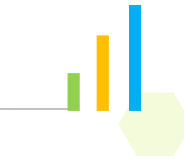
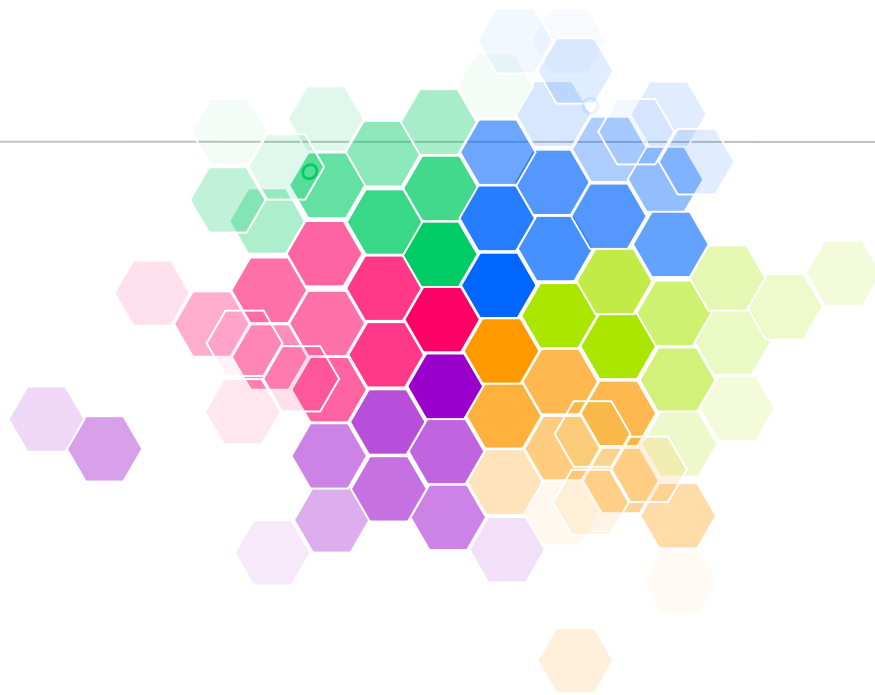
2. Oscillator



3. Spaceship (Glitter)



TUGAS 1



Buatlah initial state sesuai dengan keadaan yang still life, oscillator dan glitter. Kemudian simulasikan dengan aturan Game of Life.

TUGAS 2

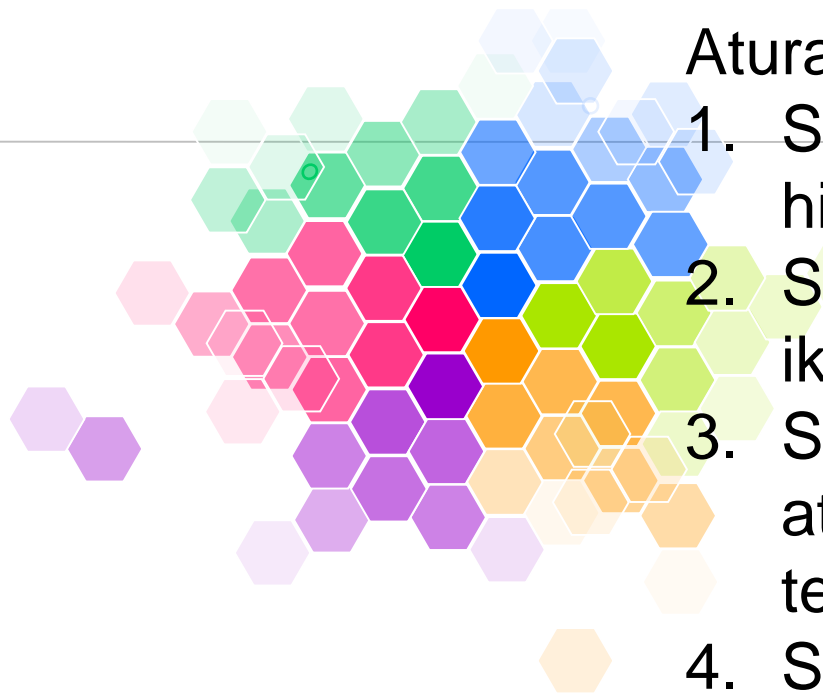
Buatlah Game of Life dengan mensimulasikan keadaan pemangsa (hiu) dan yang dimangsa (ikan).

State:

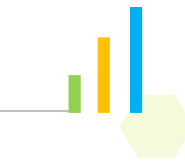
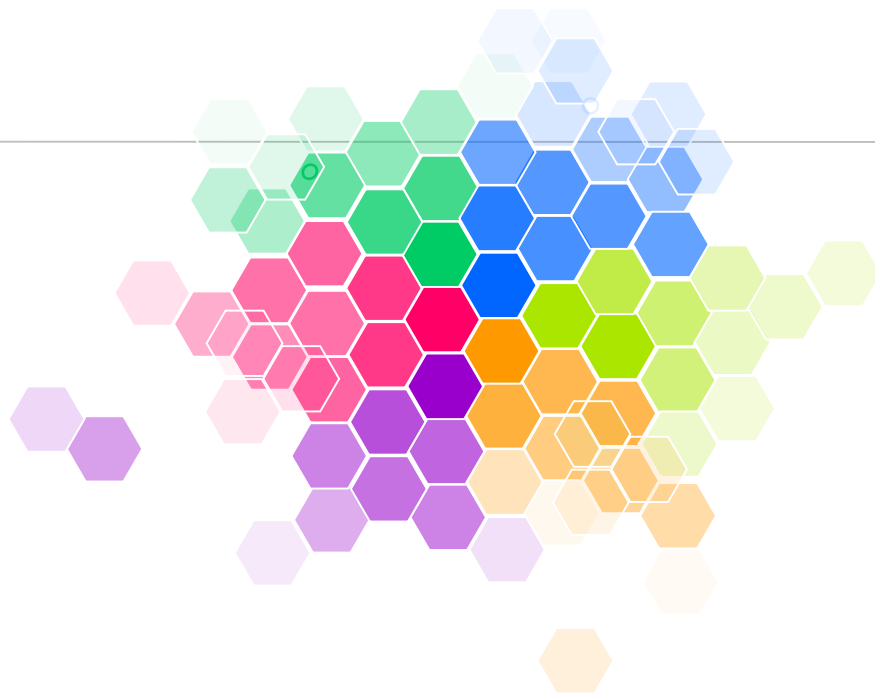
0 = kosong, 1 = hiu; 2 = ikan.

Aturan:

1. Sebuah sel kosong akan menjadi hiu bila ada 3 sel tetangga hiu
2. Sebuah sel kosong akan menjadi ikan bila ada 3 sel tetangga ikan
3. Sel hiu akan tetap hidup bila ada 2 atau 3 hiu tetangga dan ada ikan tetangga.
4. Sel ikan akan mati bila ada maksimal 3 hiu.



PATTERN GENERATOR



Pattern Generator tidak berbeda jauh dengan Game of Life, hanya pada bagaimana mendefinisikan initial state dan aturannya.



Pattern Generator

- Initial State
 - Semua nilai 0 kecuali pada titik tengah bernilai 1.
- Aturan
 - Sebuah titik 0 menjadi 1 jika ada 1 atau 3 tetangga yang bernilai 1
 - Sebuah titik 1 menjadi 0 jika tetangga yang bernilai 1 kurang dari 2 dan lebih dari 5



Pattern Generator

```
n=input('Lattice size = ');

%Initial state
s=zeros(n+2,n+2);
m=floor((n+2)/2);
s(m,m)=1;

u=255*s(2:n+1,2:n+1);
figure(1), image(u), colormap(gray(255))

%Iterasi sebanyak 100 kali
for iter=1:100
    %Update state dengan aturan Game of Life
    st=s;
    for i=2:n+1
        for j=2:n+1
            t_hidup=st(i-1,j-1)+st(i-1,j)+st(i-1,j+1);
            t_hidup=t_hidup+st(i,j-1)+st(i,j)+st(i,j+1);
```



Pattern Generator

```
t_hidup=t_hidup+st(i+1,j-1)+st(i+1,j)+st(i+1,j+1);
    if(st(i,j)==0)
        if((t_hidup==1)|| (t_hidup==4))
            s(i,j)=1;
        end
    else
        if((t_hidup>5)|| (t_hidup<2))
            s(i,j)=0;
        end
    end
end
end
end
u=255*s(2:n+1,2:n+1);
figure(1), image(u), colormap(gray(255))
drawnow;
end
```




Hasilnya adalah

