LAMPIRAN A

Program Visual Kriptografi 2 subpixel

Main.m

```
% Program pembentukan skema kriptografi visual (2,2) 2 subpixel
% Input citra untuk program ini harus berupa citra biner

inImg = imread('gambar\face\face1.bmp');
figure;imshow(inImg);title('Secret Image');

%Visual Cryptography
[share1, share2, share12] = VisCrypt(inImg);

%Outputs
figure;imshow(share1);title('Share 1');
figure;imshow(share2);title('Share 2');

figure;imshow(share12);title('Overlapping Share 1 & 2');

imwrite(share1,'Share1.bmp');
imwrite(share2,'Share2.bmp');
imwrite(share12,'Overlapped.bmp');
```

GenerateShare.m

```
% Program untuk mengacak permutasi generasi share
% Program ini menghasilkan share 1 dan share 2 secara acak untuk setiap pixel
function out = generateShare(a,b)
```

```
a1 = a(1);
a2 = a(2);
b1 = b(1);
b2 = b(2);

in = [a
      b];
out = zeros(size(in));
randNumber = floor(1.9*rand(1));

if (randNumber == 1)
   out = in;
elseif (randNumber == 0)
   a(1) = a2;
   a(2) = a1;
   b(1) = b2;
   b(2) = b1;
   out = [a
         b];
end
```

Viscrypt.m

```
%Kegunaan:
%Input: inImg – Sebuah citra biner
%Output:        share 1  - Generated share 1
%               share 2  - Generated share 2
```

```
%                   share12 – Hasil overlapped share 1 dan share 2
function [share1, share2, share12] = VisCrypt(inImg)

s = size(inImg);
share1 = zeros(s(1), (2 * s(2)));
share2 = zeros(s(1), (2 * s(2)));

%White Pixel Processing
%White Pixel share combinations
disp('White Pixel Processing...');
s1a=[1 0];
s1b=[1 0];
[x y] = find(inImg == 1);
len = length(x);

for i=1:len
   a=x(i);b=y(i);
   pixShare=generateShare(s0a,s0b);
   share1((a),(2*b-1):(2*b))=pixShare(1,1:2);
   share2((a),(2*b-1):(2*b))=pixShare(2,1:2);
end

%Black Pixel Processing
%Black Pixel share combinations
disp('Black Pixel Processing...');
s0a=[1 0];
s0b=[0 1];
```

```
[x y] = find(inImg == 0);
len = length(x);

for i=1:len
   a=x(i);b=y(i);
   pixShare=generateShare(s1a,s1b);
   share1((a),(2*b-1):(2*b))=pixShare(1,1:2);
   share2((a),(2*b-1):(2*b))=pixShare(2,1:2);
end

share12=bitxor(share1, share2);
share12 = ~share12;
disp('Share Generation Completed.');
```

Main.m

```
% Program pembentukan skema kriptografi visual (2,2) 2 subpixel
% Input citra untuk program ini harus berupa citra biner

inImg = imread('gambar\face\face1.bmp');
figure;imshow(inImg);title('Secret Image');

%Visual Cryptography
[share1, share2, share12] = VisCrypt(inImg);

%Outputs
figure;imshow(share1);title('Share 1');
figure;imshow(share2);title('Share 2');

figure;imshow(share12);title('Overlapping Share 1 & 2');

imwrite(share1,'Share1.bmp');
imwrite(share2,'Share2.bmp');
imwrite(share12,'Overlapped.bmp');
```

GenerateShare.m

```
% Program untuk mengacak permutasi generasi share
% Program ini menghasilkan share 1 dan share 2 secara acak untuk setiap pixel
function out = generateShare(a,b)

a1 = a(1);
```

```
a2 = a(2);
b1 = b(1);
b2 = b(2);

in = [a
      b];
out = zeros(size(in));
randNumber = floor(1.9*rand(1));

if (randNumber == 1)
   out = in;
elseif (randNumber == 0)
   a(1) = a2;
   a(2) = a1;
   b(1) = b2;
   b(2) = b1;
   out = [a
         b];
end
```

Viscrypt.m

```
%Kegunaan:
%Input: inImg – Sebuah citra biner
%Output:       share 1  - Generated share 1
%              share 2  - Generated share 2
%              share12 – Hasil overlapped share 1 dan share 2
function [share1, share2, share12] = VisCrypt(inImg)
```

```
s = size(inImg);
share1 = zeros(s(1), (2 * s(2)));
share2 = zeros(s(1), (2 * s(2)));

%White Pixel Processing
%White Pixel share combinations
disp('White Pixel Processing...');
s1a=[0 1];
s1b=[0 1];
[x y] = find(inImg == 1);
len = length(x);

for i=1:len
   a=x(i);b=y(i);
   pixShare=generateShare(s0a,s0b);
   share1((a),(2*b-1):(2*b))=pixShare(1,1:2);
   share2((a),(2*b-1):(2*b))=pixShare(2,1:2);
end

%Black Pixel Processing
%Black Pixel share combinations
disp('Black Pixel Processing...');
s0a=[0 1];
s0b=[1 0];
[x y] = find(inImg == 0);
len = length(x);
```

```
for i=1:len
    a=x(i);b=y(i);
    pixShare=generateShare(s1a,s1b);
    share1((a),(2*b-1):(2*b))=pixShare(1,1:2);
    share2((a),(2*b-1):(2*b))=pixShare(2,1:2);
end

share12=bitxor(share1, share2);
share12 = ~share12;
disp('Share Generation Completed.');
```

LAMPIRAN B

Program Visual Kriptografi 4 subpixel

Main.m

```
% Program pembentukan skema kriptografi visual (2,2) 4 subpixel
% Input citra untuk program ini harus berupa citra biner

inImg = imread('gambar\face\face1.bmp');
figure;imshow(inImg);title('Secret Image');

%Visual Cryptography
[share1, share2, share12] = VisCrypt(inImg);

%Outputs
figure;imshow(share1);title('Share 1');
figure;imshow(share2);title('Share 2');

figure;imshow(share12);title('Overlapping Share 1 & 2');

imwrite(share1,'Share1.bmp');
imwrite(share2,'Share2.bmp');
imwrite(share12,'Overlapped.bmp');
```

GenerateShare.m

```
% Program untuk mengacak permutasi generasi share
% Program ini menghasilkan share 1 dan share 2 secara acak untuk setiap pixel
function out = generateShare(a,b)
```

```
a1 = a(1);
a2 = a(2);
a3 = a(3);
a4 = a(4);

b1 = b(1);
b2 = b(2);
b3 = b(3);
b4 = b(4);


in = [a ; b ];
out = zeros(size(in));
randNumber = floor(1.9*rand(1));

if (randNumber == 0)
   out = in;
elseif (randNumber == 1)
   a(1) = a2;
   a(2) = a1;
   a(3) = a4;
   a(4) = a3;

   b(1) = b2;
   b(2) = b1;
   b(3) = b4;
```

```
    b(4) = b3;


    out = [a ; b ];
end
```

Viscrypt.m

```
%Kegunaan:
%Input: inImg – Sebuah citra biner
%Output:        share 1  - Generated share 1
%               share 2  - Generated share 2
%               share12 – Hasil overlapped share 1 dan share 2
function [share1, share2, share12] = VisCrypt(inImg)


s = size(inImg);
share1 = zeros(2*s(1), (2 * s(2)));
share2 = zeros(2*s(1), (2 * s(2)));

%%White Pixel Processing
%White Pixel share combinations
disp('White Pixel Processing...');
s1a=[1 0;0 1];
s1b=[1 0;0 1];



[x y] = find(inImg == 1);
len = length(x);
```

```
for i=1:len
    a=x(i)*2;b=y(i)*2;
    pixShare=generateShare(s1a,s1b);
    share1((a:a+1),(b:b+1))=pixShare(1:2,1:2);
    share2((a:a+1),(b:b+1))=pixShare(3:4,1:2);

end

%Black Pixel Processing
%Black Pixel share combinations
disp('Black Pixel Processing...');
s0a=[1 0;0 1];
s0b=[0 1;1 0];

[x y] = find(inImg == 0);
len = length(x);

for i=1:len
    a=x(i)*2;b=y(i)*2;
    pixShare=generateShare(s0a,s0b);
    share1((a:a+1),(b:b+1))=pixShare(1:2,1:2);
    share2((a:a+1),(b:b+1))=pixShare(3:4,1:2);

end

share12=bitxor(share1, share2);
share12 = ~share12;
```

```
disp('Share Generation Completed.');
```

Main.m

```
% Program pembentukan skema kriptografi visual (2,2) 4 subpixel
% Input citra untuk program ini harus berupa citra biner

inImg = imread('gambar\face\face1.bmp');
figure;imshow(inImg);title('Secret Image');

%Visual Cryptography
[share1, share2, share12] = VisCrypt(inImg);

%Outputs
figure;imshow(share1);title('Share 1');
figure;imshow(share2);title('Share 2');

figure;imshow(share12);title('Overlapping Share 1 & 2');

imwrite(share1,'Share1.bmp');
imwrite(share2,'Share2.bmp');
imwrite(share12,'Overlapped.bmp');
```

GenerateShare.m

```
% Program untuk mengacak permutasi generasi share
% Program ini menghasilkan share 1 dan share 2 secara acak untuk setiap pixel
function out = generateShare(a,b)

a1 = a(1);
```

```
a2 = a(2);
a3 = a(3);
a4 = a(4);

b1 = b(1);
b2 = b(2);
b3 = b(3);
b4 = b(4);



in = [a ; b ];
out = zeros(size(in));
randNumber = floor(1.9*rand(1));

if (randNumber == 0)
   out = in;
elseif (randNumber == 1)
   a(1) = a2;
   a(2) = a1;
   a(3) = a4;
   a(4) = a3;

   b(1) = b2;
   b(2) = b1;
   b(3) = b4;
   b(4) = b3;
```

```
    out = [a ; b ];
end
```

Viscrypt.m

```
%Kegunaan:
%Input: inImg – Sebuah citra biner
%Output:        share 1  - Generated share 1
%               share 2  - Generated share 2
%               share12 – Hasil overlapped share 1 dan share 2
function [share1, share2, share12] = VisCrypt(inImg)

s = size(inImg);
share1 = zeros(2*s(1), (2 * s(2)));
share2 = zeros(2*s(1), (2 * s(2)));

%%White Pixel Processing
%White Pixel share combinations
disp('White Pixel Processing...');
s1a=[1 0;1 0];
s1b=[1 0;1 0];


[x y] = find(inImg == 1);
len = length(x);

for i=1:len
    a=x(i)*2;b=y(i)*2;
```

```
    pixShare=generateShare(s1a,s1b);
    share1((a:a+1),(b:b+1))=pixShare(1:2,1:2);
    share2((a:a+1),(b:b+1))=pixShare(3:4,1:2);

end

%Black Pixel Processing
%Black Pixel share combinations
disp('Black Pixel Processing...');
s0a=[1 1;0 0];
s0b=[0 0;1 1];

[x y] = find(inImg == 0);
len = length(x);

for i=1:len
    a=x(i)*2;b=y(i)*2;
    pixShare=generateShare(s0a,s0b);
    share1((a:a+1),(b:b+1))=pixShare(1:2,1:2);
    share2((a:a+1),(b:b+1))=pixShare(3:4,1:2);

end

share12=bitxor(share1, share2);
share12 = ~share12;
disp('Share Generation Completed.');
```

LAMPIRAN C

```
inImg = imread('face1.bmp'); %citra asli%
[m,n] = size(inImg); %matriks ukuran m*n sebesar inImg%
outImg = imread('face.bmp'); %citra hasil dari hasil 2 subpixel%
[k,l]= size(outImg); %matriks ukuran k*l = m*2n hasil dari 2 subpixel%

aa= size(inImg); %matriks untuk menyimpan hasil outImg seukuran array inImg%

for k = 1:k %looping k kali%
   for l = 1:2:l %looping sebanyak l kali count number 2
      aa(k,ceil(l/2)) = outImg(k,l); %simpan nilai outImg kedalam array ukuran mn%
   end
end

NCC = sum(sum(inImg .* aa)) / sum(sum(inImg .* inImg))
```

LAMPIRAN D

```
inImg = imread('view13.bmp'); %citra asli%
[m,n] = size(inImg); %matriks ukuran m*n sebesar inImg%
outImg = imread('view13a.bmp'); %citra hasil dari hasil 4 subpixel%
[k,l]= size(outImg); %matriks ukuran k*l = 2m*2n hasil dari 4 subpixel%

aa= size(inImg); %matriks untuk menyimpan hasil outImg seukuran array inImg%

for k = 2:2:k %looping k kali%
   for l = 2:2:l %looping sebanyak l kali count number 2
      aa( ceil(k/2),ceil(l/2)) = outImg(k,l); %simpan nilai outImg kedalam array ukuran mn%
   end
end

mse = sum(sum(aa - inImg).^2)/(m*n) %rumus MSE = mean square error%
%secara logis jika kedua nilai dari citra adalah sama/tidak berubah maka hasil MSE adalah 0%

psnr = 10*(log(1^2/mse))

NCC = sum(sum(inImg .* aa)) / sum(sum(inImg .* inImg))
```