

DAFTAR PUSTAKA

- [1] Coltuc, Dinu dan Jean-Marc Chassery, “*Very Fast Watermarking by Reversible Contrast Mapping*”, IEEE Signal Processing Letters Vol. 14, 2007.
- [2] Maity, Hirak Kumar, Santi P. Maityy, dan Tapasi Bhattacharjeez, “*Prediction based Reversible Watermarking with Contrast Modification*”, IEEE International Image Processing Applications and Systems Conference, 2014.
- [3] Han-Min Tsai, Long-Wen Chang. “*A High Secure Reversible Visible Watermarking Scheme*”. IEEE International Conference on Multimedia and Expo, 2007.
- [4] Ying Yang, Xingming Sun, Hengfu Yang, Chang-Tsun Li, and Rong Xiao, “*A Contrast-Sensitive Reversible Visible Image Watermarking Technique*”, IEEE Transactions on Circuits and Systems for Video Technology Vol. 19, 2009.
- [5] Pei-Yu Lin, Yi-Hui Chen, Chin-Chen Chang, dan Jung-San Lee, ”*Contrast Adaptive Removable Visible Watermarking (CARVW) mechanism*”, Image and Vision Computing 31, 2013.
- [6] Retno, Y., A. Harjoko, “*Reversible Watermarking dengan Metode Modifikasi Histogram pada Difference Image*”, IJCCS, Vol.5, 2011.
- [7] Gonzalez, Rafael C, Richard E. Woods, dan Steven L. Eddins, “*Digital Image Processing Using Matlab*”. Prentice Hall: New Jersey, 2003.
- [8] Tyas, L.A., “*Watermarking Citra Digital Berbasis TWD-SVD dengan Detektor Non-Blind*”, Semarang: Universitas Diponegoro, 2011.
- [9] Terzija, Nataša, “*Robust Digital Image Watermarking Algorithms for Copyright Protection*”. Universität Duisburg-Essen, 2006.
- [10] Hariyadi, Fitri, “*Kompresi Citra Digital Menggunakan Transformasi Wavelet*”, Jakarta: Universitas Mercu Buana, 2009.

tripathi, Deepika, “*Efficient Implementations of Discrete Wavelet transform using FPGAs*”, Florida State University, 2003.



- [12] Alfatwa, D.F., “*Watermarking Pada Citra Digital Menggunakan Discrete Transformasi Wavelet*”, Bandung: Institut Teknologi Bandung, 2007.
- [13] Rahmaningtyas, Deisna, Setiawan Hadi, dan Akmal, “*Analisis Perbandingan Kinerja Metode Binerisasi Terhadap Citra Lontar Sunda Kuno*”, Jatinangor: Universitas Padjadjaran, 2016.
- [14] Wulandari, Meirista, “*Index Quality Assesment Citra Terinterpolasi (SSIM dan FSIM)*”, JUTEI Vol. I, 2017.
- [15] Polikar, Robi, “*The Wavelet Tutorial: Second Edition*”, Durham Computation Center, Iowa State, 1998.



LAMPIRAN



Lampiran I Program prosedur penyisipan

```
citra=uigetfile('*.','pilih gambar');
O=imread(num2str(citra));
figure,imshow(uint8(O));
info=imfinfo(citra);
nama=info.FileName;
O=double(O);
R=O(:,:,1);G=O(:,:,2);B=O(:,:,3);
R1=subsampling1(R);G1=subsampling1(G);B1=subsampling1(B);
R2=subsampling2(R);G2=subsampling2(G);B2=subsampling2(B);
R3=subsampling3(R);G3=subsampling3(G);B3=subsampling3(B);
R4=subsampling4(R);G4=subsampling4(G);B4=subsampling4(B);

[LL1R,LH1R,HL1R,HH1R]=dwt2(R1,'db1');
[LL2R,LH2R,HL2R,HH2R]=dwt2(R2,'db1');
[LL3R,LH3R,HL3R,HH3R]=dwt2(R3,'db1');
[LL4R,LH4R,HL4R,HH4R]=dwt2(R4,'db1');
[LL1G,LH1G,HL1G,HH1G]=dwt2(G1,'db1');
[LL2G,LH2G,HL2G,HH2G]=dwt2(G2,'db1');
[LL3G,LH3G,HL3G,HH3G]=dwt2(G3,'db1');
[LL4G,LH4G,HL4G,HH4G]=dwt2(G4,'db1');
[LL1B,LH1B,HL1B,HH1B]=dwt2(B1,'db1');
[LL2B,LH2B,HL2B,HH2B]=dwt2(B2,'db1');
[LL3B,LH3B,HL3B,HH3B]=dwt2(B3,'db1');
[LL4B,LH4B,HL4B,HH4B]=dwt2(B4,'db1');

scan={'Kunci Rahasia1(ak)','Kunci Rahasia2(bk)','Kekuatan
Watermark(wu)','Kekuatan Bagian Penting dari Gambar(wv)'};
judul='Masukkan Faktor(wu,wv) dan Kunci(ak,bk)';
pjpgInput=1;
inputan={' ',' ',' ',' '};
options.Resize='on';
options.WindowStyle='normal';
options.Interpreter='tex';
jawaban=inputdlg(scan,judul,pjpgInput,inputan,options);
ak=str2num(jawaban{1});
bk=str2num(jawaban{2});
wu=str2num(jawaban{3});
wv=str2num(jawaban{4});
[xr,yr]=kunci(ak,bk,LL1R,LL2R,LL3R,LL4R);
[xg,yg]=kunci(ak,bk,LL1G,LL2G,LL3G,LL4G);
[xb,yb]=kunci(ak,bk,LL1B,LL2B,LL3B,LL4B);
[LL1R_w,LL2R_w,LL3R_w,LL4R_w,qkr,dkr,wkr]=sisip(
xr,yr,wu,wv,LL1R,LL2R,LL3R,LL4R);
[LL1G_w,LL2G_w,LL3G_w,LL4G_w,qkg,dkg,wkg]=sisip(
xg,yg,wu,wv,LL1G,LL2G,LL3G,LL4G);
[LL1B_w,LL2B_w,LL3B_w,LL4B_w,qkb,dkb,wkb]=sisip(
xb,yb,wu,wv,LL1B,LL2B,LL3B,LL4B);
```

```
dwt2(LL1R_w,LH1R,HL1R,HH1R,'db1');
dwt2(LL2R_w,LH2R,HL2R,HH2R,'db1');
dwt2(LL3R_w,LH3R,HL3R,HH3R,'db1');
dwt2(LL4R_w,LH4R,HL4R,HH4R,'db1');
dwt2(LL1G_w,LH1G,HL1G,HH1G,'db1');
dwt2(LL2G_w,LH2G,HL2G,HH2G,'db1');
```



```

O3G_w=idwt2(LL3G_w,LH3G,HL3G,HH3G,'db1');
O4G_w=idwt2(LL4G_w,LH4G,HL4G,HH4G,'db1');
O1B_w=idwt2(LL1B_w,LH1B,HL1B,HH1B,'db1');
O2B_w=idwt2(LL2B_w,LH2B,HL2B,HH2B,'db1');
O3B_w=idwt2(LL3B_w,LH3B,HL3B,HH3B,'db1');
O4B_w=idwt2(LL4B_w,LH4B,HL4B,HH4B,'db1');

R_w=invers_subsampling(O1R_w,O2R_w,O3R_w,O4R_w);
G_w=invers_subsampling(O1G_w,O2G_w,O3G_w,O4G_w);
B_w=invers_subsampling(O1B_w,O2B_w,O3B_w,O4B_w);
O_w(:,:,1)=R_w;
O_w(:,:,2)=G_w;
O_w(:,:,3)=B_w;

imwrite(uint8(O_w),'gambar ber-watermark.png');
formatSpec = 'Watermarking dengan kunci %d dan %d nilai wu = %d
dan wv = %d.\n';
fprintf(formatSpec,ak,bk,wu,wv);
disp(nama);

mser = sum((R(:)-R_w(:)).^2) / prod(size(O));
psnr = 10*log10(255*255/mser);
mseg = sum((G(:)-G_w(:)).^2) / prod(size(O));
psnrG = 10*log10(255*255/mseg);
mseb = sum((B(:)-B_w(:)).^2) / prod(size(O));
psnrB = 10*log10(255*255/mseb);
psnr=(psnr+psnrG+psnrB)/3;
fprintf('Nilai PSNR adalah %0.4f.\n',psnr);
[ssimval, ssimmap] = ssim(O,O_w);
fprintf('Nilai SSIM adalah %0.4f.\n',ssimval);

figure,imshow(uint8(O_w));
title(['watermarking dengan nilai kunci ( ',num2str(ak),' ',
',num2str(bk),' '), wu = ',num2str(wu),' dan wv = ',num2str(wv)]);
mytext=['PSNR = ',num2str(psnr),' dan SSIM = ', num2str(ssimval)];
annotation('textbox',[0.3, 0.09, 1, 0], 'string',mytext);

```

Lampiran II Program prosedur penghapusan

```

citra=uigetfile('*.','pilih gambar');
O_w=imread(num2str(citra));
figure, imshow(O_w);
O_w=double(O_w);
Rw=O_w(:,:,1);Gw=O_w(:,:,2);Bw=O_w(:,:,3);
R1w=subsampling1(Rw);G1w=subsampling1(Gw);B1w=subsampling1(Bw);
R2w=subsampling2(Rw);G2w=subsampling2(Gw);B2w=subsampling2(Bw);
R3w=subsampling3(Rw);G3w=subsampling3(Gw);B3w=subsampling3(Bw);
R4w=subsampling4(Rw);G4w=subsampling4(Gw);B4w=subsampling4(Bw);

[LL1Rw,LH1Rw,HL1Rw,HH1Rw]=dwt2(R1w,'db1');
[LH2Rw,HL2Rw,HH2Rw]=dwt2(R2w,'db1');
[LH3Rw,HL3Rw,HH3Rw]=dwt2(R3w,'db1');
[LH4Rw,HL4Rw,HH4Rw]=dwt2(R4w,'db1');
[LH1Gw,HL1Gw,HH1Gw]=dwt2(G1w,'db1');
[LH2Gw,HL2Gw,HH2Gw]=dwt2(G2w,'db1');
[LH3Gw,HL3Gw,HH3Gw]=dwt2(G3w,'db1');

```



```

[LL4Gw, LH4Gw, HL4Gw, HH4Gw]=dwt2 (G4w, 'db1' );
[LL1Bw, LH1Bw, HL1Bw, HH1Bw]=dwt2 (B1w, 'db1' );
[LL2Bw, LH2Bw, HL2Bw, HH2Bw]=dwt2 (B2w, 'db1' );
[LL3Bw, LH3Bw, HL3Bw, HH3Bw]=dwt2 (B3w, 'db1' );
[LL4Bw, LH4Bw, HL4Bw, HH4Bw]=dwt2 (B4w, 'db1' );

scan={'Kunci Rahasia(ak)', 'Kunci Rahasia2(bk)', 'Kekuatan
Watermark(wu)', 'Kekuatan Bagian Terpenting dari Gambar(wv)'};
judul='Masukkan Faktor(wu,wv) dan Kunci(ak,bk)';
pjpgInput=1;
inputan={' ',' ',' ',' '};
options.Resize='on';
options.WindowStyle='normal';
options.Interpreter='tex';
jawaban=inputdlg (scan, judul, pjpgInput, inputan, options);
ak=str2num (jawaban{1});
bk=str2num (jawaban{2});
wu=str2num (jawaban{3});
wv=str2num (jawaban{4});

[xrw, yrw]=kunci (ak, bk, LL1Rw, LL2Rw, LL3Rw, LL4Rw );
[xgw, ygw]=kunci (ak, bk, LL1Gw, LL2Gw, LL3Gw, LL4Gw );
[xbw, ybw]=kunci (ak, bk, LL1Bw, LL2Bw, LL3Bw, LL4Bw );
[LL1R_h, LL2R_h, LL3R_h, LL4R_h, qkr, dkr, wkr]=hapus (
xrw, yrw, wu, wv, LL1Rw, LL2Rw, LL3Rw, LL4Rw );
[LL1G_h, LL2G_h, LL3G_h, LL4G_h, qkg, dkg, wkg]=hapus (
xgw, ygw, wu, wv, LL1Gw, LL2Gw, LL3Gw, LL4Gw );
[LL1B_h, LL2B_h, LL3B_h, LL4B_h, qkb, dkb, wkb]=hapus (
xbw, ybw, wu, wv, LL1Bw, LL2Bw, LL3Bw, LL4Bw );

O1R_h=idwt2 (LL1R_h, LH1Rw, HL1Rw, HH1Rw, 'db1' );
O2R_h=idwt2 (LL2R_h, LH2Rw, HL2Rw, HH2Rw, 'db1' );
O3R_h=idwt2 (LL3R_h, LH3Rw, HL3Rw, HH3Rw, 'db1' );
O4R_h=idwt2 (LL4R_h, LH4Rw, HL4Rw, HH4Rw, 'db1' );
O1G_h=idwt2 (LL1G_h, LH1Gw, HL1Gw, HH1Gw, 'db1' );
O2G_h=idwt2 (LL2G_h, LH2Gw, HL2Gw, HH2Gw, 'db1' );
O3G_h=idwt2 (LL3G_h, LH3Gw, HL3Gw, HH3Gw, 'db1' );
O4G_h=idwt2 (LL4G_h, LH4Gw, HL4Gw, HH4Gw, 'db1' );
O1B_h=idwt2 (LL1B_h, LH1Bw, HL1Bw, HH1Bw, 'db1' );
O2B_h=idwt2 (LL2B_h, LH2Bw, HL2Bw, HH2Bw, 'db1' );
O3B_h=idwt2 (LL3B_h, LH3Bw, HL3Bw, HH3Bw, 'db1' );
O4B_h=idwt2 (LL4B_h, LH4Bw, HL4Bw, HH4Bw, 'db1' );

R_h=invers_subsampling (O1R_h, O2R_h, O3R_h, O4R_h);
G_h=invers_subsampling (O1G_h, O2G_h, O3G_h, O4G_h);
B_h=invers_subsampling (O1B_h, O2B_h, O3B_h, O4B_h);

O_h (:, :, 1)=R_h;
O_h (:, :, 2)=G_h;
O_h (:, :, 3)=B_h;
O_h (:, :, 4)=
(uint8 (O_h), 'gambar hasil restorasi.png');
getfile ('*.png', 'pilih gambar'); %citra asal
d(num2str (citra));
dec = 'Watermarking dengan kunci %d dan %d nilai wu = %d
= %d.\n';

```



```

fprintf(formatSpec,ak,bk,wu,wv);

A = double(O); P = double(O_h);
mse = sum((A(:)-P(:)).^2) / prod(size(A));
psnr = 10*log10(255*255/mse);
fprintf('Nilai PSNR adalah %0.4f.\n',psnr);
[ssimval, ssimap] = ssim(O,O_h);
fprintf('Nilai SSIM adalah %0.4f.\n',ssimval);

figure;imshow(uint8(O_h)); title('Hasil Restorasi');
title(['restorasi dengan nilai kunci ( ',num2str(ak),' ',
',num2str(bk),' ), wu = ',num2str(wu),' dan wv = ',num2str(wv)]);
mytext=['PSNR = ',num2str(psnr),' dan SSIM = ', num2str(ssimval)];
annotation('textbox', [0.3, 0.09, 1, 0], 'string',mytext);

```

Lampiran III Fungsi subsampling1

```

function output = subsampling1(img)
[r, c] = size(img);
output = zeros(r/2,c/2,class(img));
[r2, c2] = size(output);
for x = 1:r2
    for y = 1:c2
        j = 2*x-1;
        i = 2*y-1;
        output(x,y) = img(j,i);
    end
end
end

```

Lampiran IV Fungsi subsampling2

```

function output = subsampling2(img)
[r, c] = size(img);
output = zeros(r/2,c/2,class(img));
[r2, c2] = size(output);
for x = 1:r2
    for y = 1:c2
        j = 2*x-1;
        i = 2*y;
        output(x,y) = img(j,i);
    end
end
end

```



Lampiran V Fungsi subsampling3

```
function output = subsampling3(img)
[r, c] = size(img);
output = zeros(r/2,c/2,class(img));
[r2, c2] = size(output);
for x = 1:r2
    for y = 1:c2
        j = 2*x;
        i = 2*y-1;
        output(x,y) = img(j,i);
    end
end
end
```

Lampiran VI Fungsi subsampling4

```
function output = subsampling4(img)
[r, c] = size(img);
output = zeros(r/2,c/2,class(img));
[r2, c2] = size(output);
for x = 1:r2
    for y = 1:c2
        j = 2*x;
        i = 2*y;
        output(x,y) = img(j,i);
    end
end
end
```

Lampiran VII Fungsi kunci

```
function [ x,y ] = kunci( ak,bk,LL1,LL2,LL3,LL4)
switch ak
    case 1
        switch bk
            case 2
                x=LL1;y=LL2;
            case 3
                x=LL1;y=LL3;
            case 4
                x=LL1;y=LL4;
        end
    case 2
        switch bk
            case 1
                x=LL2;y=LL1;
            case 3
                x=LL2;y=LL3;
            case 4
                x=LL2;y=LL4;
        end
    case 3
        switch bk
            case 1
                x=LL3;y=LL1;
```




```

        case 2
            x=LL3;y=LL2;
        case 4
            x=LL3;y=LL4;
        end
    case 4
        switch bk
            case 1
                x=LL4;y=LL1;
            case 2
                x=LL4;y=LL2;
            case 3
                x=LL4;y=LL3;
        end
    end
end
end

```

Lampiran VIII Fungsi sisip

```

function [LL1_w,LL2_w,LL3_w,LL4_w,qk,dk,wk] =
sisip(x,y,wu,wv,LL1,LL2,LL3,LL4)
m=imread('topi.png');%citra watermark
mk=im2double(m);
[wt,t]=size(mk);
for i=1:wt
    for j=1:wt
        dk(i,j)=abs(x(i,j)-y(i,j));
    end
end
dmax=max(max(dk));
dmin=min(min(dk));
for i=1:wt
    for j=1:wt
        if mk(i,j)==0
            qk(i,j)=(dk(i,j)-dmin)/(dmax-dmin);
            wk(i,j)=wu+floor((wv*qk(i,j)));
            LL1_w(i,j)=LL1(i,j)-wk(i,j);
            LL2_w(i,j)=LL2(i,j)-wk(i,j);
            LL3_w(i,j)=LL3(i,j)-wk(i,j);
            LL4_w(i,j)=LL4(i,j)-wk(i,j);
        elseif mk(i,j)==1
            qk(i,j)=(dk(i,j)-dmin)/(dmax-dmin);
            wk(i,j)=wu+floor((wv*qk(i,j)));
            LL1_w(i,j)=LL1(i,j);
            LL2_w(i,j)=LL2(i,j);
            LL3_w(i,j)=LL3(i,j);
            LL4_w(i,j)=LL4(i,j);
        end
    end
end
end
end

```



Lampiran IX Fungsi hapus

```
function [LL1_w,LL2_w,LL3_w,LL4_w,qk,dk,wk] =
hapus(x,y,wu,wv,LL1,LL2,LL3,LL4 )
m=imread('topi.png');
mk=im2double(m);
[wt,t]=size(mk);
dmax=0;
dmin=x(1,1);
for i=1:wt
    for j=1:wt
        dk(i,j)=abs(x(i,j)-y(i,j));
        if dk(i,j)> dmax
            dmax=dk(i,j);
        end
        if dk(i,j)< dmin
            dmin=dk(i,j);
        end
    end
end
for i=1:wt
    for j=1:wt
        if mk(i,j)==0
            qk(i,j)=(dk(i,j)-dmin)/(dmax-dmin);
            wk(i,j)=wu+floor((wv*qk(i,j)));
            LL1_w(i,j)=LL1(i,j)+wk(i,j);
            LL2_w(i,j)=LL2(i,j)+wk(i,j);
            LL3_w(i,j)=LL3(i,j)+wk(i,j);
            LL4_w(i,j)=LL4(i,j)+wk(i,j);
        elseif mk(i,j)==1
            qk(i,j)=(dk(i,j)-dmin)/(dmax-dmin);
            wk(i,j)=wu+floor((wv*qk(i,j)));
            LL1_w(i,j)=LL1(i,j);
            LL2_w(i,j)=LL2(i,j);
            LL3_w(i,j)=LL3(i,j);
            LL4_w(i,j)=LL4(i,j);
        end
    end
end
end
end
```

Lampiran X Fungsi invers subsampling

```
function output = invers_subsampling(img1,img2,img3,img4)
[r, c] = size(img1);
output = zeros(r*2,c*2,class(img1));
[r2, c2] = size(output);
for x = 1:2:r2
    for y = 1:2:c2
        j = (x+1)/2;
        i = (y+1)/2;
        output(x,y) = img1(j,i);
    end
end
y = 2:2:c2
```



```
        j = (x+1)/2;
        i = y/2;
        output(x,y) = img2(j,i);
    end
end
for x = 2:2:r2
    for y = 1:2:c2
        j = x/2;
        i = (y+1)/2;
        output(x,y) = img3(j,i);
    end
end
for x = 2:2:r2
    for y = 2:2:c2
        j = x/2;
        i = y/2;
        output(x,y) = img4(j,i);
    end
end
end
end
```

