Penerapan Steganografi Dengan Menggunakan Metode Least Significant Bit (Lsb) Dan Pixel Value Differencing (Pvd) Pada Citra Warna

Zaim Nabil Alif Tirta Putra[1], Agus Prihanto[2],

[1] Jurusan Teknik Informatika/Teknik Informatika, Universitas Negeri Surabaya
[2] Jurusan Teknik Informatika/Teknik Informatika, Universitas Negeri Surabaya
[1]zp@mhs.unesa.ac.id
[2]agusprihanto@unesa.ac.id

Abstrak **- Steganografi merupakan teknik penyembunyian pesan kedalam suatu media telah menjadi salah satu Teknik yang banyak digunakan dalam melakukan keamanan terhadap pesan, karena pada zaman modern seperti sekarang ini keamanan suatu informasi menjadi isu yang penting. Sebuah informasi akan menjadi sangat penting untuk dijaga kerahasiaannya apabila hal tersebut berkaitan dengan bisnis maupun keamanan negara. Pada penelitian ini bertujuan untuk meningkatkan kapasitas pesan dan sekaligus berusaha menjaga kulitas gambar dengan menggabungkan metode LSB (*Least Significant Bit*) untuk 2 pixel yang memiliki perbedaan nilai rendah dan metode PVD (*Pixel Value Differencing*) untuk 2 pixel yang memiliki perbedaan nilai tinggi. Hasil pengujian menunjukkan bahwa setelah dilakukan perbandingan kapasitas metode yang diusulkan memiliki kapasitas lebih tinggi dari pada metode single PVD dengan angka peningkatan berada diantara 280-300 kb. Sedangkan pada pengujian PSNR diperoleh hasil bahwa metode yang diusulkan sedikit lebih rendah dibandingkan dengan metode single PVD dengan rata-rata selisih 1 (dB).**

***Kata Kunci*— Steganografi, Least Significant Bit (LSB), Pixel Value Differencing (PVD).**

I. PENDAHULUAN

Kerahasiaan dan keamanan informasi menjadi salah satu isu yang penting pada zaman kemajuan teknologi seperti sekarang ini. Sepotong informasi akan dinilai lebih tinggi jika menyangkut aspek keputusan bisnis, keamanan, atau kepentingan publik dan pribadi. Terdapat dua cara yang biasa digunakan untuk melindungi suatu pesan rahasia. Salah satunya adalah kriptografi di mana informasi rahasia dikodekan dalam bentuk lain dengan menggunakan kunci rahasia sebelum mengirim, yang hanya dapat diterjemahkan dengan kunci rahasia. Cara lain adalah steganografi yang merupakan teknik menyembunyikan informasi rahasia ke dalam suatu media. Media tersebut dapat berupa gambar, audio maupun video.

Penggunaan kriptografi cukup aman namun masih mencurigakan karena pesan hanya disandikan dan tetap masih terlihat, sedangkan steganografi melakukan penyembunyian pesan sehingga selain orang yang mengirim dan menerima tidak akan menyadari bahwa ada pesan rahasia di dalamnya. Terdapat banyak metode dalam steganografi, salah satu metode steganografi yang paling umum dan paling sederhana yaitu least significant bit (LSB). LSB merupakan metode yang kerap digunakan, namun algoritma ini dalam melakukan penyisipan disamaratakan dan kadang masih kurang bagus. Pada metode Least Significant Bit (LSB) ini dilakukan menggunakan pendekatan yang sederhana yaitu dengan menyisipkan suatu informasi kedalam suatu media dengan cara mengganti nilai-nilai bit dengan bit data yang akan disisipkan. Metode pixel value differencing (PVD) menjadi metode yang menarik untuk digunakan. Pixel value differencing (PVD) merupakan metode yang digunakan untuk steganografi dengan cara kerja mencari nilai dua pixel yang terdekat. Karena mata manusia kurang peka didaerah kontras. Selain itu, metode ini menawarkan kapasitas penyimpanan pesan yang lebih besar, dengan kualitas gambar yang lebih baik dibandingkan dengan metode lain. Namun pada penilitian ini akan menggabungkan dua metode sekaligus yaitu metode least significant bit (LSB) dengan metode pixel value differencing (PVD) yang dimana dengan melakukan penggabungan dua metode ini akan menghasilkan kapasitas penyimpanan yang lebih besar lagi.

II. PENELITIAN RELEVAN

Cukup banyak yang pernah melakukan penelitian berkaitan dengan penggunaan metode pixel value differencing (PVD) sebelumnya. Beberapa penelitian tersebut diantaranya seperti yang dilakukan oleh Michael Sitorus yang berjudul "*Teknik Steganography Dengan Metode Least Significant Bit (LSB)*". Pada penelitian ini penulis melakukan suatu implementasi pada steganografi text dengan mengenkripsi pesan text menggunakan Teknik kriptografi terlebih dahulu. Pada penelitian ini menggunakan metode *Least Significant Bit Insertion (LSB)*. Hasil yang diperoleh menunjukkan bahwa penyisipan dan ekstraksi pesan dapat dilakukan dengan baik. (Siturus, 2016)

Pada tahun 2016, Dicky Nofriansyah dan Robbi Rahim yang berjudul "*Combination of Pixel Value Differencing Algorithm with Caesar Algorithm For Steganography*". Pada penelitian ini penulis menggunakan kombinasi metode Pixel Value Differencing (PVD) dengan algoritma caesar. Menurut penulis, penelitian dengan cara ini dapat menyisipkan pesan lebih banyak ke piksel yang di pilih nilai kontras tinggi. Untuk meningkatkan tingkat keamanan informasi yang akan disematkan ke dalam gambar, maka digunakan kriptografi seperti algoritma cipher Caesar. (Nofriansyah & Rahim, 2016)

Pada tahun 2016, Shobana Manoharan dan Deepika RajKumar juga melakukan penelitian dengan judul "*Pixel Value Differencing Method Based on CMYK Colour Model*". Pada penelitian ini, cara baru dalam teknik persembunyian telah diusulkan dengan memperkenalkan konsep lapisan warna CMYK dalam gambar di bidang steganografi. Menyembunyikan pesan dalam lapisan warna CMYK memberikan kualitas gambar yang lebih aman dan baik daripada lapisan warna RGB-nya. Nilai PSNR dan MSE berada dalam kisaran yang baik dalam pendekatan CMYK bila dibandingkan dengan model warna RGB. Dalam metode ini, jika penyerang mencoba memecah gambar menjadi lapisan warna RGB juga ia tidak dapat mengambil pesan sepenuhnya. (Manoharan & RajKumar, 2016)

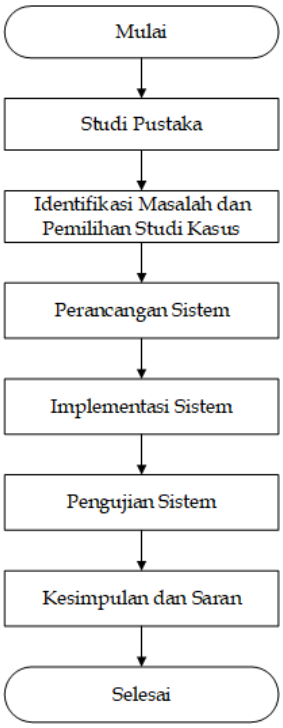
Pada tahun 2015, Avinash K. Gulve dan Madhuri S. Joshi melakukan penelitian dengan judul *"An Image Steganography Method Hiding Secret Data into Coefficients of Integer Wavelet Transform Using Pixel Value Differencing Approach*". Sistem steganografi gambar menggunakan domain spasial atau domain frekuensi untuk menyembunyikan informasi rahasia. Teknik yang diusulkan menggunakan teknik domain spasial untuk menyembunyikan informasi rahasia di domain frekuensi. Gambar sampul ditransformasikan menggunakan integer wavelet transform untuk memperoleh empat subbands: LL, LH, HL, dan HH. Kemudian, pendekatan PVD digunakan untuk menyembunyikan informasi rahasia dalam koefisien wavelet dari keempat subbands. Nilai PSNR yang dihasilkan oleh algoritma mendekati 39,5 yang jauh di atas ambang batas 36 dB setelah menggunakan kapasitas persembunyian penuh dari gambar sampul. Ini membuktikan bahwa gambar stego berkualitas baik. (Gulve & Joshi, 2015).

Pada tahun 2018, Anita Pradhan, K. Raja Sekhar dan Gandharba Swain melakukan penelitian dengan judul "*Digital Image Steganography Using LSB Substitution,PVD, and EMD*". Pada penelitian ini peneliti mengusulkan dua teknik steganografi gambar hibrida dengan kombinasi substitusi LSB, Pixel Value Differencing(PVD), dan Exploiting Modification Directions (EMD). Teknik pertama beroperasi pada blok 2 × 2 piksel dan teknik kedua beroperasi pada blok 3 × 3 piksel. Untuk setiap blok, perbedaan nilai piksel rata-rata, d, dihitung. Jika nilai d lebih besar dari 15, blok berada di area tepi, sehinggan menggunakan penerapan kombinasi subtitusi LSB dan PVD. Jika nilai d kurang dari atau sama dengan 15, blok berada di area yang halus, sehingga menggunakan penerapan kombinasi substitusi LSB dan EMD. Hasil dari penelitian menunjukkan bahwa Teknik tersebut tidak terdeteksi oleh analisis RS. (Pradhan, Sekhar, & Swain, 2018).

Pada tahun 2016, Gandharba swain juga telah melakukan penelitian dengan judul "*A steganographic method combining LSB substitution and PVD in a block*". Makalah ini mengusulkan teknik steganografi dengan menggunakan substitusi LSB dan PVD dengan dalam satu blok. Sebuah teknik steganografi berdasarkan substitusi LSB dan tiga directional PVD dalam blok 2 × 2 piksel diusulkan. Ada dua varian dari teknik yang diusulkan ini. Varian-1 (Tipe 1) yang diusulkan mencapai PSNR lebih tinggi dibandingkan dengan teknik Khodaei & Faez (Tipe 1). Varian-2 yang diusulkan (Tipe 2) mencapai PSNR yang lebih tinggi dan kapasitas yang lebih tinggi dibandingkan dengan teknik Khodaei & Faez (Tipe 2). Namun jika kita membandingkan dua varian dari teknik yang diusulkan, maka varian-1 lebih disukai untuk PSNR yang lebih tinggi dan varian-2 lebih disukai untuk kapasitas persembunyian yang lebih tinggi. (Swain, 2016)

III. METODOLOGI PENELITIAN

Penelitian ini membahas mengenai Teknik penyisipan pesan ke dalam gambar atau yang disebut dengan steganografi. Penelitian ini akan menggunakan metode penggabungan *Least Significant Bit (LSB)* dan *Pixel Value Differencing (PVD)*. Terdapat beberapa tahap alur proses penelitian yang ditunjukkan pada gambar 1 dibawah ini.



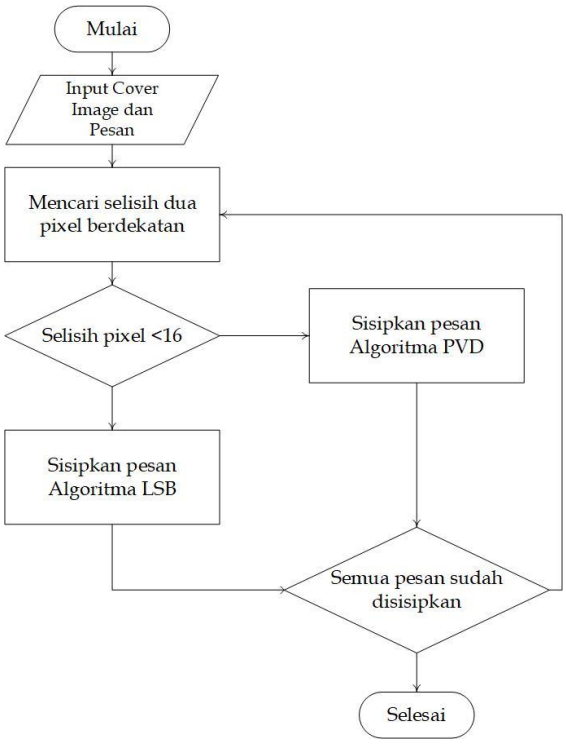
Gbr. 1 Alur Proses Penelitian

Tahap pertama yang dilakukan dalam proses penelitian yaitu melakukan studi pustaka untuk memperoleh referensi berkaitan dengan metode steganografi *Least Significant Bit* (LSB) dan *Pixel Value Differencing* (PVD). Setelah itu melakukan proses identifikasi masalah serta pemilihan studi kasus berkaitan dengan penyisipan pesan kedalam gambar atau yang dikenal dengan proses steganografi. Pada tahap perancangan sistem yaitu penelitian ini yaitu akan dilakukan proses penyisipan pesan tersembunyi kedalam gambar dengan menggabungkan algoritma *Least Significant Bit* (LSB) dan *Pixel Value Differencing* (PVD) dengan menggunakan huruf ASCII 256. Kemudian untuk proses implementasi sistem akan menggunakan Bahasa pemrograman MATLAB. Pada tahap pengujian sistem dilakukan pada uji keamanan yang bertujuan untuk menganalisis keamanan sistem. Tahap terakhir adalah

menarik kesimpulan atas apa yang telah dilakukan dan saran yang dibutuhkan dalam penelitian selanjutnya.

A. Proses Penyisipan Pesan

Tujuan dari proses ini yaitu untuk menyisipkan pesan text kedalam gambar yang dijadikan wadah untuk menyimpan isi pesan tersembunyi.



Gbr. 2 Alur Penyisipan

Berdasarkan gambar 2 diatas menunjukkan alur dari proses penyisipan yaitu :

1. Input cover image dan pesan.
   Pada proses ini cover image/gambar asli akan diubah menjadi blok-blok piksel RGB. Dan pesan tersembunyi akan diubah menjadi bentuk biner. Sebagai contoh :

R = 110 G = 130 B = 192	R = 100 G = 150 B = 164

   LOREM = 01001100 01001111 01010010 01000101 01001101

2. Mencari selisih dua piksel
   Pada proses ini menghitung selisih dua piksel dimana jika selesih blok piksel tersebut kurang dari 16 dan adanya 1 blok piksel terakhir atau paling belakang yang tidak bisa diselisih maka blok piksel tersebut akan disisipkan menggunakan algoritma LSB. Dan jika selisih blok piksel tersebut lebih dari 16 maka akan disisipkan menggunakan PVD.
3. Algoritma LSB
   Penyisipan menggunakan algoritma LSB ini dengan cara mengganti nilai 4 bit belakang piksel dengan nilai 4 bit pesan. Contoh jika hasil selisih piksel R adalah 8 maka piksel R akan disisipkan menggunakan LSB. Dengan cara mengubah piksel ke bentuk biner. Dari piksel 110 dan 100 biner yang didapat ialah 01101110 dan 01100100. Dan karakter pesan yang akan disisipkan adalah L dengan biner 01001100. Maka hasil penyisipan yang didapat 01100100 dan 01101100.
4. Algoritma PVD
   Pada proses ini penyisipan pesan akan dilakukan dengan menghitung selisih dua blok piksel. Contoh pesan yang akan disisipkan yaitu O dengan biner 01001111. Dan piksel G yaitu 130 dan 150. Tahap pertama adalah menghitung selisih piksel |130-150| sehingga didapat d=20. Mencari letak continues range dari nilai difference value pada skema wu&tsai

   R={[0,7],[8,15],[16,31],[32,63],[64,127],[128,255]}.

   Range yang didapat dari d=20 yaitu [16,31] dimana ik=16 dan uk=31. Ambil 4 bit pesan untuk disisipkan yaitu t=0100. Mengubah nilai bit kedalam nilai decimal adalah 4 atau b=4. Kemudian menghitung selisih yang baru d^i=16+4 sehingga didapat nilai d^i=20. Selanjutnya melakukan penyisipan dengan mengubah nilai piksel yang dibandingkn dengan nilai piksel yang baru sesuai dengan aturan yang ada, dimana m=48 didapat dari |20-20| maka aturan yang terpenuhi yaitu P_i=130+|[20-20]/2| dan P_{i+1}=150-|[20-20]/2|. Maka piksel baru yang didapat yaitu P_i=130 dan P_{i+1}=150.
5. Lakukan seperti proses diatas sampai semua piksel berhasil di disisipkan pesan.

```
 image = result_image;
ik = [0, 8, 16, 32, 64, 128];
uk = [7, 15, 31, 63, 127, 255];
possible_values = [3, 3, 4, 5, 6, 7];
ik_index = [0,0,0];
rgb_values = [0,0,0];
char = text(text_index);

if y2 <= size(result_image,1)
   diff = [abs(double(image(y1,x1,1)) -
double(image(y2,x2,1))), abs(double(image(y1,x1,2)) -
double(image(y2,x2,2))), abs(double(image(y1,x1,3)) -
double(image(y2,x2,3)))];
   for i = 1: size(possible_values,2)
      for c =1:size(diff,2)
         if diff(c)>=ik(i) && diff(c)<=uk(i)
            rgb_values(c) = possible_values(i);
            ik_index(c) = i;
         end
      end
   end

   disp('diff');
   disp(diff);
```

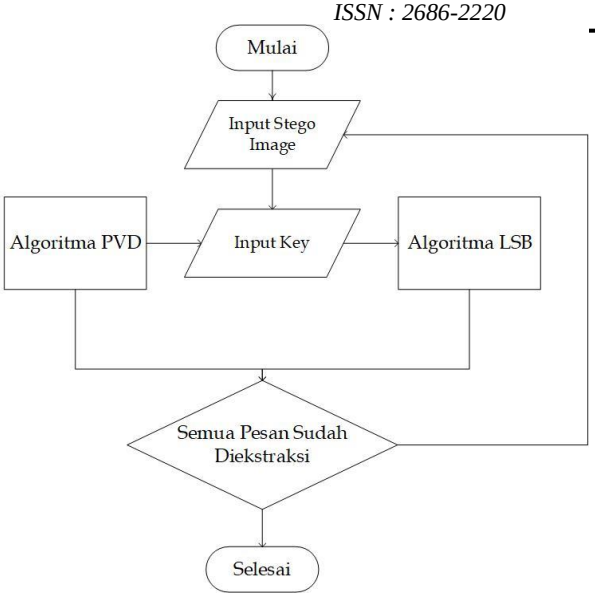
```
    for i=1:size(rgb_values,2)
      if diff(i) >= 16
        disp('16');
        %PVD
        key = strcat(key, '1');
        ascii = double(char);
        char_binary = de2bi(ascii, 8);
        %disp(char1_binary);

        remain_bit = bit_index-rgb_values(i);
        vals = strcat(vals,sprintf('%.0f',rgb_values(i)));
        if remain_bit <=0
          char1_binary = char_binary(1:bit_index);
          bit_index = 8-(abs(remain_bit));
          text_index = text_index+1;
          if text_index > size(text,2)
            complete_text = true;
          else
            char = text(text_index);
            ascii = double(char);
            char_binary = de2bi(ascii, 8);

            if remain_bit<0
              char2_binary = char_binary(bit_index+1:8);
              char1_binary =
cat(2,char2_binary,char1_binary);
            end
          end
        else
          char1_binary =
char_binary(remain_bit+1:bit_index);
          bit_index = remain_bit;
        end
        disp(['ik_index ', num2str(ik(ik_index(i)))]);
        disp('bin');
        %char1 = bi2de(char1_binary);
        disp('chari1 binary origin');
        disp(char1_binary);
        disp('chari1 binary length');
        disp(length(char1_binary));
        if length(char1_binary) < rgb_values(i)
          char_zeros = zeros(1,rgb_values(i) -
length(char1_binary));
          char1_binary = cat(2,char_zeros,char1_binary);
        end

        %char1_binary = de2bi(char1, rgb_values(i));
        disp('result');
        disp(char1_binary);
        d_aks = ik(ik_index(i)) + bi2de(char1_binary);
        disp(['d_aks ', num2str(d_aks), ' diff ',
num2str(diff(i))]);
        m = 0;
        if d_aks > diff(i)
          m = d_aks-diff(i);
        else
          m = diff(i)-d_aks;
        end
        disp(['m ', num2str(m)])

        px1 = 0;
        px2 = 0;
        disp(['x1 ', num2str(image(y1,x1,i)), ' x2 ',
num2str(image(y2,x2,i))]);
        if (result_image(y1,x1,i) < result_image(y2,x2,i) &&
d_aks > diff(i)) || (result_image(y1,x1,i) >=
result_image(y2,x2,i) && d_aks <= diff(i))
          %result_image(y1,x1,i) = abs(image(y1,x1,i)-
floor(m/2));
          %result_image(y2,x2,i) =
abs(image(y2,x2,i)+ceil(m/2));

          px1 = double(image(y1,x1,i))-floor(double(m/2));
          px2 = double(image(y2,x2,i))+ceil(double(m/2));
        else
          %result_image(y1,x1,i) =
abs(image(y1,x1,i)+ceil(m/2));
          %result_image(y2,x2,i) = abs(image(y2,x2,i)-
floor(m/2));

          px1 = double(image(y1,x1,i))+ceil(double(m/2));
          px2 = double(image(y2,x2,i))-floor(double(m/2));
        end

        difference = abs(px1-px2);
        disp(['px1 ', num2str(px1), ' px2 ', num2str(px2)]);
        disp(['ceil ', num2str(ceil(double(m/2))), ' floor ',
num2str(floor(double(m/2)))]);
        disp(['difference ', num2str(difference)]);
        if px1 > px2 && px1 > 255
          px1 = 255;
          px2 = 255-difference;
        elseif px2 > px1 && px2 > 255
          px2 = 255;
          px1 = 255-difference;
        elseif px1 < px2 && px1 < 0
          px1 = 0;
          px2 = 0+difference;
        elseif px2 < px1 && px2 < 0
          px2 = 0;
          px1 = 0+difference;
        end
        result_image(y1,x1,i) = px1;
        result_image(y2,x2,i) = px2;
        disp(['result px1 ', num2str(px1), ' px2 ',
num2str(px2)]);
      else
        %lsb
        key = strcat(key,'0');
        used_value = 8;
        remain_bit = bit_index-used_value;

        if remain_bit <=0
          char1_binary = char_binary(1:bit_index);
```

```
            bit_index = 8-(abs(remain_bit));
            text_index = text_index+1;
            if text_index > size(text,2)
              complete_text = true;
```

```
          else
            char = text(text_index);
            ascii = double(char);
            char_binary = de2bi(ascii, 8);

            if remain_bit<0
               char2_binary = char_binary(bit_index+1:8);
               char1_binary =
cat(2,char2_binary,char1_binary);
            end
          end
        else
          char1_binary =
char_binary(remain_bit+1:bit_index);
          bit_index = remain_bit;
        end
        disp('current char');
        disp(char);
        disp('char1_binary');
        disp(char1_binary);

        if length(char1_binary) < 8
          char_zeros = zeros(1,8 - length(char1_binary));
          char1_binary = cat(2,char_zeros,char1_binary);
        end

        %char1 = bi2de(char1_binary);
        %char1_binary = de2bi(char1, 8);
        disp('new char1_binary');
        disp(char1_binary);
        p1_bin = de2bi(image(y1,x1,i), 8);
        p1_bin = cat(2, char1_binary(5:8), p1_bin(5:8));
        p2_bin = de2bi(image(y2,x2,i), 8);
        p2_bin = cat(2, char1_binary(1:4), p2_bin(5:8));

        result_image(y1,x1,i) = bi2de(p1_bin);
        result_image(y2,x2,i) = bi2de(p2_bin);
      end
      if complete_text == true
        break
      end
    end
end
```

B. Proses Pengambilan Pesan

Selain melakukan penyisipan pesan teks kedalam gambar, pada penelitian ini juga dilakukan proses untuk pengambilan pesan. Berikut merupakan tahapan yang perlu dilakukan :



Gbr. 3 Alur Pengambilan

1. Pilih gambar hasil stego.

R = 100	R = 108
G = 130	G = 150
B = 192	B = 164

2. Input key.
   Pada proses ini key adalah sebuah tanda untuk mengetahui bahwa penyisipan dilakukan menggunakan algoritma PVD atau LSB.
3. Jika key = 0 maka pengambilan pesan dilakukan dengan algoritma LSB.
   Pada proses ini pengambilan pesan dengan algoritma ini cukup mengambil empat bit pesan paling belakang. Contoh jika piksel R digambar tersebut 100 dan 108. Maka diubah ke bentuk biner menjadi 01100100 dan 01101100. Maka bit pesan yang diambil adalah 0100 dan 1100.
4. Jika key = 1 maka pengambilan pesan dilakukan dengan algoritma PVD.
   Pada proses ini pengambilan pesan dilakukan dengan menghitung selisih dua blok piksel. Contoh P_i=130 dan P_{i+1}=150. Menghitung selsisih d=|130-150| sehingga didapat d=20. Mencari letak continues range dari nilai difference value pada skema wu&tsai. (Wu & Tsai, 2003)

   R={[0,7],[8,15],[16,31],[32,63],[64,127],[128,255]}.

   Letak continues range yang didapat dari d=20 yaitu [16,31] dimana ik=16 dan uk=31 yang dimana ada 4 bit pesan yang sudah disisipkan. Selanjutnya menghitung selisih dari |d – ik| yaitu |20-16| didapat hasil decimal atau b=4. Mengubah nilai decimal pesan kedalam bentuk bit, maka didapat bit pesan b =0100.

5. Lakukan seperti proses diatas sampai semua gambar yang telah disisipkan pesan berhasil diekstrak.

```
image          = result_image;
ik             = [0, 8, 16, 32, 64, 128];
uk             = [7, 15, 31, 63, 127, 255];
possible_values = [3, 3, 4, 5, 6, 7];
ik_index       = [0,0,0];
rgb_values     = [0,0,0];

if y2 <= size(result_image,1)
   diff = [abs(double(image(y1,x1,1)) -
double(image(y2,x2,1))), abs(double(image(y1,x1,2)) -
double(image(y2,x2,2))), abs(double(image(y1,x1,3)) -
double(image(y2,x2,3)))];
   for i = 1: size(possible_values,2)
      for c =1:size(diff,2)
         if diff(c)>=ik(i) && diff(c)<=uk(i)
            rgb_values(c) = possible_values(i);
            ik_index(c) = i;
         end
      end
   end
   for i=1:size(rgb_values,2)
      char = key(key_index);
      if char == '1'
         %PVD
         dec = abs(diff(i) - ik(ik_index(i)));
         bin = de2bi(dec, rgb_values(i));
         binary_result = cat(2, bin, binary_result);
      else
         %lsb
         p1_bin = de2bi(image(y1,x1,i), 8);
         p1_bin = p1_bin(1:4);
         p2_bin = de2bi(image(y2,x2,i), 8);
         p2_bin = p2_bin(1:4);
         binary_result = cat(2, cat(2, p2_bin, p1_bin),
binary_result);
      end

      key_index = key_index+1;
      if key_index > size(key,2)
         complete_key = true;
      end
      if complete_key == true
        break
      end
   end
end
```

C. Perhitungan Kapasitas Maximum Pesan

Cara perhitungan max kapasitas pesan ini adalah sebagai berikut :

Tabel 1 Blok Piksel

R = 70	R = 80
G = 100	G = 80
B = 180	B = 70

Pada tabel 1 diatas kita ilustrasikan sebagai gambar 2 blok piksel dengan nilai RGB demikian. Untuk mencdapatkan nilai max kapasitas pesan kita harus menghitung selisih setiap 2 blok piksel diatas untuk mengetahui piksel mana yang disisipkan menggunakan LSB atau PVD. Ketika selisih dibawah 16 maka piksel akan disisipkan menggunakan LSB. Dan sebalikanya ketika selisih piksel adalah 16 atau lebih maka piksel akan disisipkan menggunakan PVD. Contoh :

1. R : 70 – 80 = 10, 10 disini kita akan sisipkan menggunakan LSB dengan cara mengganti 4 bit belakang. Dangan demikian piksel R dapat disisipkan 8 bit atau setara 1 karakter pesan.
2. G : 100 – 80 = 20, 20 disini kita akan sisipkan menggunakan PVD yang dimana 20 berada pada continues range [16,31]. Artinya piksel ini dapat disisipkan 4 bit pesan.
3. B : 180 – 70 = 110. 110 disini penyisipannya menggunakan PVD yang dimana 110 berada pada continues range [64,127]. Artinya piksel ini dapat disisipkan 6 bit pesan.

Perhitungan max kapasitas pesan dengan cara menjumlah dari R+G+B = 18. Aritnya didalam 2 blok piksel tersebut dapat disisipkan maksimal 18 bit pesan. Untuk mengubah kedalam Kb maka harus dibagi 1024. 18 : 1024 = 0.01 Kb

D. Perhitungan PSNR

Ada beberapa cara untuk melihat tingkat keamanan informasi rahasia. Salah satunya yakni dengan menghitung nilai PSNR (*Peak Signal Moise Ratio*). Menurut Alvinash K Gulve dan Madhuri S Joshi Nilai PSNR yang dihasilkan oleh algoritma mendekati 39,5 yang jauh di atas ambang batas 36dB setelah menggunakan kapasitas persembunyian penuh dari gambar sampul. Ini membuktikan bahwa gambar stego berkualitas baik. Hasil juga menunjukkan bahwa perbedaan antara gambar sampul dan gambar stego tidak dapat diperhatikan oleh sistem visual manusia (HVS). (Gulve & Joshi, 2015)

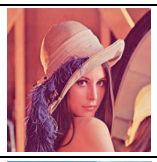
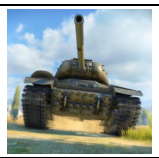
IV. HASIL DAN PEMBAHASAN

A. Data penelitian

Pada tahap pertama dalam melakukan penelitian ini adalah penentuan dan persiapan data yang perlu diuji coba yaitu pesan tersembunyi dengan ukuran [78.7 Kb] yang diambil dari LoremIpsum (Lipsum, Tanpa Tahun) dan gambar jenis RGB yang diambil dari SIPI (Viterbi, tanpa tahun) dan web citra uji (Munir, Tanpa tahun) yang sudah disiapkan untuk menjadi uji coba dalam proses steganografi dengan menggunakan penggabungan algoritma *pixel value differencing* (PVD) dan 4 bit *least significant bit* (LSB). Pengambilan pesan dilakukan

dengan algoritma yang sama seperti proses penyisipan pesan dalam gambar. Pada tabel II berikut merupakan kumpulan data yang diuji coba:

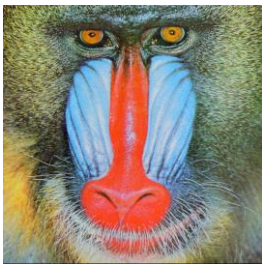
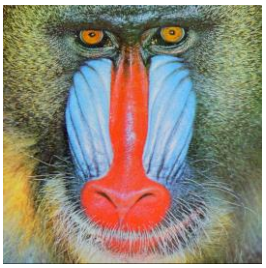
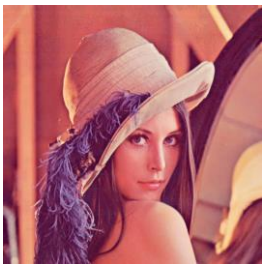
Tabel 2 Kumpulan Data Uji Coba

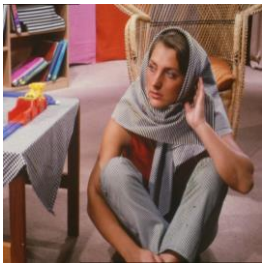
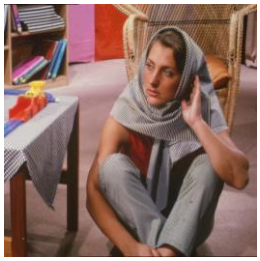
No	Gambar	Nama File	Resolusi	Ukuran	Format
1.		Elaine	512 x 512	47.8 Kb	jpg
2.		Baboon	512 x 512	768 Kb	bmp
3.		Lena	512 x 512	768 Kb	bmp
4.		Tank	512 x 512	519 Kb	png
5.		Pepper	512 x 512	768 Kb	bmp
6.		Barbara	512 x 512	99.8 Kb	jpg
7.		Boat	512 x 512	279 Kb	png

B. Hasil Pengujian

Dalam hal ini telah dilakukan pengujian penyisipan pesan pada gambar, pengambilan pesan pada gambar, maksimal kapasitas pesan dan PSNR (*Peak Signal Noise Ratio*) terhadap sistem yang telah dibuat. Pengujian ini dilakukan dengan menyisipakan pesan tersembunyi dengan ukuran [78.7 Kb] terhadap 3 jenis format gambar yaitu jpg, png, dan bmp. Hasil analisis dan pengujian yang diperoleh dapat dilihat berdasarkan perbandingan gambar sebelum proses penyisipan pesan tersembunyi dan setelah proses penyisipan pesan tersembunyi menggunakan algoritma *Least Significant Bit* (LSB) dan *Pixel Value Differencing* (PVD). Berikut adalah perbandingan pengujian gambar sebelum dan sesudah disisipkan pesan secara kasat mata :

Tabel 3 Perbandingan Citra Asli dan Citra Stego

No	Citra Asli	Citra Stego
1.		
2.		
3.		
4.		
5.		

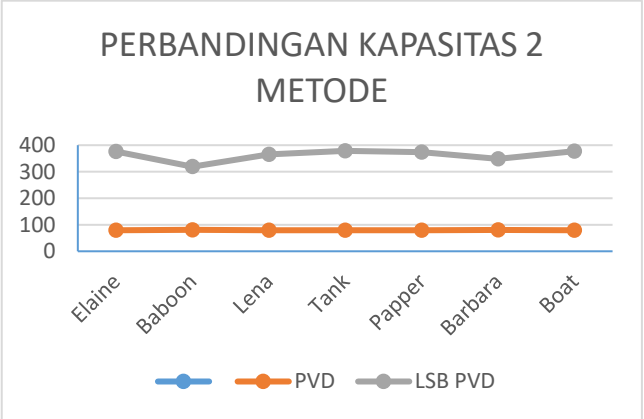
No.	Citra Asli	Citra Stego
6.		
7.		

1. Pengujian Kapasitas Maximum Pesan

Pengujian maksimum kapasitas pesan bisa diperoleh dari proses perhitungan total data yang dapat disispkan tanpa harus menggunakan pesan tersembunyi. Dengan resolusi 512x512 dan menggabungkan antara metode *Least Significant Bit* dan *Pixel Value Differencing* (PVD). Hasil pengujian yang dilakukan untuk mengetahui maksimum kapasitas pesan dapat dituliskan pada tabel berikut ini :

Tabel 4 Hasil Kapasitas Pesan

No	Citra Cover	Max Kapasitas (Kb)	
		PVD	LSB + PVD
1	Elaine	79.42	375.85
2	Baboon	80.99	318.90
3	Lena	79.47	365.21
4	Tank	79.41	378.59
5	Papper	79.48	373.89
6	Barbara	80.35	348.15
7	Boat	79.64	377.48



Gbr. 4 Grafik perbandingan Max Kapasitas pesan untuk metode PVD vs LSB+PVD

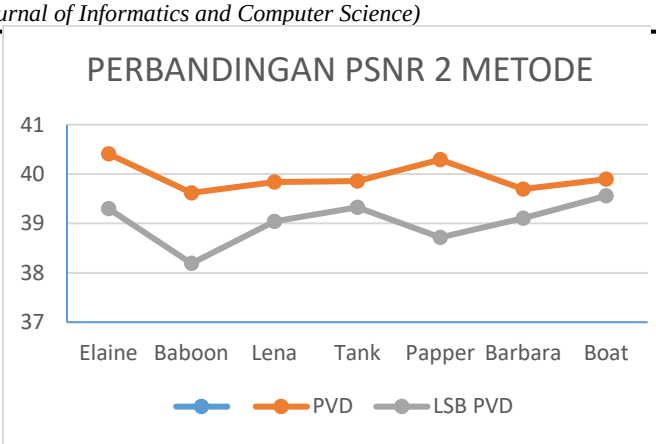
Berdasarkan hasil yang didapat dari tabel dan grafik di atas pada penyisipan pesan ke dalam gambar dengan ukuran yang sama yaitu ukuran pesan [78.7 Kb] dengan menggunakan resolusi gambar 512x512, maka hasil yang didapat adalah penggabungan dua metode antara *Leat Significant Bit* (LSB) dan *Pixel Value Differencing* (PVD) mendapat hasil kapasitas pesan yang lebih besar daripada metode *single Pixel Value Differencing* (PVD).

2. Pengujian PSNR(Peak Signal Noise Ratio)

Berdasarkan hasil pengujian yang telah dilakukan terhadap gambar dengan menggunakan pengujian pada perbandingan antara gambar asli dan gambar yang telah disisipkan pesan menggunakan pengujian PSNR (*Peak Signal Noise Ratio*) yang menghasilkan sebuah grafik perbandingan tersebut. PSNR (*Peak Signal Noise Ratio*) sering juga dinyatakan dalam skala logaritmik dalam satuan decibel (dB). Apabila nilai PSNR (*Peak Signal Noise Ratio*) berada dibawah 30 dB maka itu berarti perbandingan terlihat jelas berbeda dengan gambar aslinya. Akan tetapi jika kualitas dan tingkat kemiripan gambar yang tinggi maka PSNR (*Peak Signal Noise Ratio*) berada pada nilai 40dB dan diatasnya karena sedikit atau tidak ada perbedaan pada gambarnya. Hasil pengujian yang dilakukan pada penyisipan pesan tersembunyi ke dalam gambar warna menghasilkan yang dapat dituliskan pada Tabel berikut ini:

Tabel 5 Hasil pengujian PSNR

No	Citra Cover	PSNR	
		PVD	LSB + PVD
1	Elaine	40.41	39.3
2	Baboon	39.62	38.19
3	Lena	39.84	39.04
4	Tank	39.86	39.33
5	Papper	40.29	38.72
6	Barbara	39.7	39.11
7	Boat	39.9	39.56



Gbr. 5 Grafik perbandingan PSNR untuk metode PVD vs LSB+PVD

Dari hasil Tabel dan grafik datas pada pengujian penyisipan pesan terhadap format gambar (jpg, png, bmp) dengan menggunakan PSNR (*Peak Signal Noise Ratio*) dengan ukuran pesan 78.7 Kb diperoleh hasil bahwa metode single *Pixel Value Differencing* (PVD) memiliki PSNR (*Peak Signal Noise Ratio*) lebih tinggi dari pada metode penggabungan antara *Least Significant Bit* (LSB) dan *Pixel Value Differencing* (PVD) yang artinya metode *single Pixel Value Differencing* (PVD) memiliki kualitas yang lebih baik.

V. KESIMPULAN

Berdasarkan hasil uji coba yang telah dilakukan mengenai penyembunyian pesan pada citra digital dengan menggunakan algoritma *Least Significant Bit* dan *Pixel Value Differencing*, kesimpulan yang diperoleh dari seluruh proses dan hasil pembahasan penelitian yang telah dilakukan adalah sebagai berikut :

1. Kapasitas pesan yang dihasilkan ketika menggunakan penggabungan antara metode *Least Significant Bit* (LSB) dan *Pixel Value Differencing* (PVD) lebih besar dibandingkan menggunakan metode single *Pixel Value Differencing*.
2. Semakin besar kapasitas untuk disisipkan pesan maka kualitas gambar dengan perhitungan PSNR terlihat kurang baik. Namun, secara kasat mata gambar yang telah disisipkan pesan tidak terlihat jauh berbeda dengan gambar asli.

VI. UCAPAN TERIMA KASIH

Ucapan terima kasih yang sebesar-besarnya kepada Allah SWT yang telah memberikan pertolongannya dalam setiap langkah pengerjaan penelitian ini. Terimakasih pula untuk semua pihak yang telah memberikan dukungan sehingga penelitian ini dapat berjalan dan terselesaikan dengan baik.

VII. REFERENSI

[1] Gulve, A. K., & Joshi, M. S. (2015). An Image Steganography Method Hiding Secret Data into Coefficients of Integer Wavelet Transform Using Pixel Value Differencing Approach. *Hindawi Publishing Corporation Mathematical Problems in Engineering*.

[2] Lipsum. (Tanpa Tahun). *Lorem Ipsum*. Retrieved from Lorem Ipsum: https://www.lipsum.com/

[3] Manoharan, S., & RajKumar, D. (2016). Pixel Value Differencing Method Based on CMYK Colour Model. *Int. J. of Electronics and Information Engineering*, 37-46.

[4] Munir, R. (Tanpa tahun). *Citra uji*. Retrieved from Homepage Rinaldi Munir: http://informatika.stei.itb.ac.id/~rinaldi.munir/Koleksi/Citra%20Uji/CitraUji.htm

[5] Nofriansyah, D., & Rahim, R. (2016). COMBINATION OF PIXEL VALUE DIFFERENCING ALGORITHM WITH CAESAR ALGORITHM FOR STEGANOGRAPHY. *International Journal of Research In Science & Engineering*.

[6] Pradhan, A., Sekhar, K. R., & Swain, G. (2018). Digital Image Steganography Using LSB Substitution,PVD, and EMD. *Hindawi Mathematical Problems in Engineering*.

[7] Siturus, M. (2016). Teknik Steganography dengan Metode Least Significant Bit (LSB). *Jurnal Ilmiah Fakultas Teknik LIMIT'S*, 54.

[8] Swain, G. (2016). A steganographic method combining LSB substitution and PVD in a block. *International Conference on Computational Modeling and Security (CMS 2016)*, 39-44.

[9] Viterbi, U. (tanpa tahun). *Volume 3: Miscellaneous*. Retrieved from http://sipi.usc.edu/database/database.php?volume=misc

[10] Wu, D.-C., & Tsai, W.-H. (2003). A steganographic method for images by pixel-value differencing. *Elsevier Science*.