**BAB IV**

**HASIL DAN PEMBAHASAN**

* 1. **Data Penelitian**

Dalam proses steganografi menggunakan citra *lossy* yang berformat \*.JPEG dengan ukuran 255x255 piksel sebagai *cover* dan *file* \*.TXT sebagai pesan yang disembunyikan.

*File* teks yang disembunyikan berisi tiga kalimat paragraf pertama dari latar belakang bab 1 dengan jumlah karakter adalah sebanyak 247 karakter yaitu “Keamanan data dan informasi sangat penting bagi para pengguna internet saat ini. Banyak cara yang dapat dilakukan untuk mengamankan pesan yang akan dikirim. Pertama, teknik kriptografi yakni menyandikan pesan dengan menggunakan algoritma tertentu.”. Untuk *sampel imag*e yang digunakan sebagai media atau *cover-image* ada tiga seperti yang dapat dilihat pada Gambar 4.1.

(a) (b) (c)

**Gambar 4.1** *Cover-Image* (a) peppers.jpeg, (b) lena.jpeg (c) baboon.jpeg

* 1. **Menghitung Daya Tampung *Cover***

Sebelum dilakukan proses penyembunyian pesan ke dalam citra dilakukan proses menghitung daya tampung *cover-image* untuk mengetahui banyak pesan yang dapat disembunyikan. Proses menghitung daya tampung *cover-image* dilakukan dengan menghitung terlebih dahulu jumlah piksel kemudian hasil banyaknya piksel tersebut dibagi dengan jumlah bilangan biner dari karakter pesan yaitu delapan. Selanjutnya hasil tersebut dikalikan dengan jumlah banyaknya komponen pada *cover-image* adalah tiga, maka akan didapatkan hasil banyaknya data yang dapat ditampung di *cover-image.*

Proses perhitungan daya tampung *cover-image* pada penelitian ini adalah ((255 x 255) / 8) x 3 = 8128 x 3 = 24384. Jadi pada penelitian ini *cover-image* yang digunakan dapat menampung pesan adalah sebanyak 24384 karakter.

* 1. ***Embedding* Dengan Substitusi Bit**

Pada tahapan ini dilakukan proses penyembunyian pesan ke dalam *cover image* berformat *lossy* dengan menggunakan substitusi bit. Yang dimaksud dengan substitusi bit adalah penyembunyian dilakukan dengan mengganti bit-bit data dari segmen citra dengan bit-bit data rahasia. Pada proses penyembunyian ini pesan disembunyikan ke dalam bit-bit citra di bit satu sampai bit delapan dan di semua komponen secara bergantian.

Proses penyembunyian pesan dilakukan dengan membaca pesan dalam satu karakter, kemudian mengubah karakter pesan tersebut ke dalam biner. Selanjutnya biner dari karakter pesan diambil satu bit kemudian disembunyikan ke dalam citra yang sebelumnya sudah diubah ke dalam biner. Penyembunyian dilakukan dengan cara mengubah satu bit citra dengan satu bit karakter pesan. Sebagai contoh, media citra yang digunakan untuk penyembunyian adalah citra peppers.jpeg dan nilai biner yang diambil adalah pada piksel 0,0 sampai 0,32. Representasi data citra peppers.jpeg dalam bentuk biner dapat dilihat pada Gambar 4.2

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 10110110 | 11000000 | 11000010 | 10111111 | 11000101 | 11001001 | 11000111 | 11000111 |
| 11001001 | 11001110 | 11010000 | 11001011 | 11001010 | 11001001 | 11001011 | 11001101 |
| 11001111 | 11010000 | 11010011 | 11010011 | 11010011 | 11011010 | 11011110 | 11011100 |
| 11010111 | 11010011 | 11001101 | 10110110 | 10000110 | 01011100 | 01010000 | 01000010 |

**Gambar 4.2** Nilai Biner *Cover-Image*

Setelah didapatkan nilai biner citra, kemudian dilakukan proses penyembunyian pesan karakter KAMU. Sebelum dilakukan proses penyembunyian perlu dilakukan perubahan karakter pesan ke dalam biner seperti yang dapat dilihat pada Tabel 4.1.

**Tabel 4.1 Nilai Biner Karakter Pesan**

|  |  |  |
| --- | --- | --- |
| Karakter | ASCII (desimal) | Biner |
| K | 75 | 01001011 |
| A | 65 | 01000001 |
| M | 77 | 01001101 |
| U | 85 | 01010101 |

Setelah nilai biner dari setiap karakter didapatkan maka dapat dilakukan proses penyembunyian pesan ke dalam citra. Penyembunyian pesan ke dalam citra dilakukan dengan cara memasukkan tiap-tiap *byte* dari data pesan ke dalam biner citra di posisi bit ke delapan. Hasil dari proses penyembunyian tersebut akan menjadi seperti yang dapat dilihat pada Gambar 4.3.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1011011**0** | 1100000**1** | 1100001**0** | 1011111**0** | 1100010**1** | 1100100**0** | 1100011**1** | 1100011**1** |
| 1100100**0** | 1100111**1** | 1101000**0** | 1100101**0** | 1100101**0** | 1100100**0** | 1100101**0** | 1100110**1** |
| 1100111**0** | 1101000**1** | 1101001**0** | 1101001**0** | 1101001**1** | 1101101**1** | 1101111**0** | 1101110**1** |
| 1101011**0** | 1101001**1** | 1100110**0** | 1011011**1** | 1000011**0** | 0101110**1** | 0101000**0** | 0100001**1** |

**Gambar 4.3** *Stego* Bit Ke Delapan

Pada penelitian ini proses penyembunyian pesan dilakukan di semua komponen pada citra dan di bit satu sampai bit delapan secara bergantian, maka contoh proses penyembunyian selanjutnya adalah dengan menggunakan data pesan dan citra yang sama akan dilakukan proses penyembunyian data pesan di posisi bit ketiga pada citra. Hasil dari proses penyembunyian tersebut dapat dilihat pada Gambar 4.4.

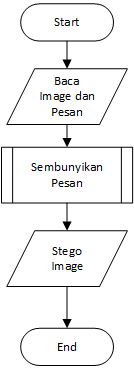
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 10**0**10110 | 11**1**00000 | 11**0**00010 | 10**0**11111 | 11**1**00101 | 11**0**01001 | 11**1**00111 | 11**1**00111 |
| 11**0**01001 | 11**1**01110 | 11**0**10000 | 11**0**01011 | 11**0**01010 | 11**0**01001 | 11**0**01011 | 11**1**01101 |
| 11**0**01111 | 11**1**10000 | 11**0**10011 | 11**0**10011 | 11**1**10011 | 11**1**11010 | 11**0**11110 | 11**1**11100 |
| 11**0**10111 | 11**1**10011 | 11**0**01101 | 10**1**10110 | 10**0**00110 | 01**1**11100 | 01**0**10000 | 01**1**00010 |

**Gambar 4.4** *Stego* Bit Ke Tiga

Seperti yang terlihat pada Gambar 4.3 dan Gambar 4.4, perubahan hanya terjadi pada bit-bit yang ditandai dengan warna merah. Perubahan tersebut dilakukan dengan cara menggantikan bit-bit pada citra dengan bit-bit data pesan yang disembunyikan.

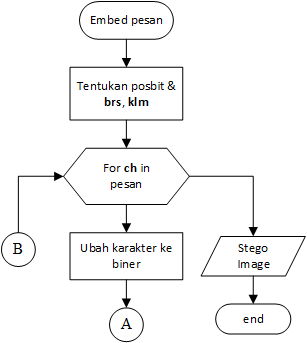
Pesan yang disembunyikan pada penelitian ini dilakukan pada bit satu sampai bit delapan citra dan di semua komponen secara bergantian. Setelah semua pesan telah selesai disembunyikan ke dalam citra, selanjutnya citra disimpan ke dalam format .JPEG *lossy* yang disebut sebagai *stego-image*.

Flowchart tahapan penyembunyian pesan ke dalam gambar berformat *lossy* dengan menggunakan substitusi bit dapat diliihat pada Gambar 4.5.



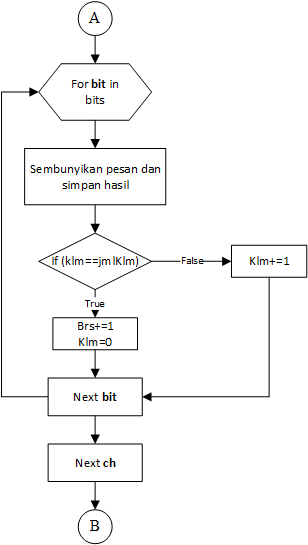
**Gambar 4.5** Flowchart Tahapan Penyembunyian Pesan

Pada Gambar 4.5, dapat dilihat tahapan penyembunyian pesan ke dalam *cover-image*. Sebelum melakukan penyembunyian pesan, terlebih dahulu dilakukan membaca cover image dan pesan. Selanjutnya melakukan proses penyembunyian pesan ke dalam citra. Kemudian menyimpan citra ke dalam format .JPEG. Berikut flowchart proses penyembunyian pesan ke dalam citra yang ditunjukkan pada Gambar 4.6.



**Gambar 4.6** Flowchart Proses Penyembunyian Pesan

Posbit adalah posisi bit untuk menentukan posisi bit pesan disembunyikan. Brs dan klm adalah menentukan awal baris dan kolom pesan disembunyikan. Ch adalah karakter dari pesan. Setelah menentukan posisi bit, baris dan kolom, selanjutnya adalah membaca pesan ke dalam satu karakter kemudian mengubah karakter tersebut ke bentuk biner. Setelah itu akan masuk ke proses berikutnya adalah proses A yaitu proses penyembunyian bit karakter pesan ke dalam biner citra dengan cara membaca satu bit karakter kemudian menyembunyikan bit tersebut ke dalam biner citra sesuai dengan posisi bit yang telah ditentukan.

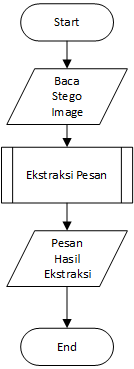


**Gambar 4.7** Flowchart Proses Penyembunyian Pesan A

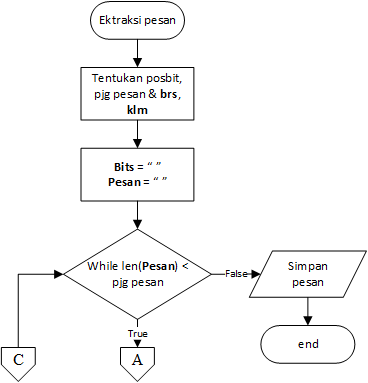
* 1. **Ekstraksi Pesan**

Pada proses ini akan dilakukan tahapan pengambilan pesan teks dari dalam *stego-image*. Proses ekstraksi ini merupakan kebalikan dari proses *embedding*. Tahap pertama adalah membaca *stego-image* yang telah dilakukan proses *embedding* pesan. Selanjutnya dilakukan pengambilan isi pesan pada bit dimana bit pesan disembunyikan pada *stego-image*.

Setelah itu dilakukan ubah data dengan mengkonversi data biner pesan yang telah diambil menjadi teks. Flowchart tahapan ekstraksi pesan dari citra berformat *lossy* yang ditunjukkan pada Gambar 4.8.

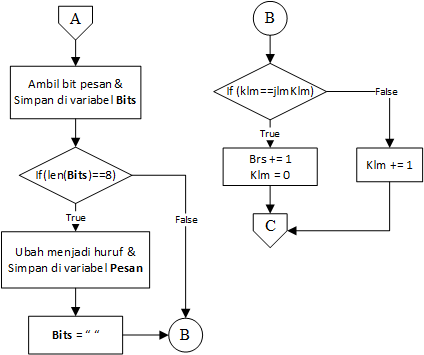


**Gambar 4.8** Flowchart Tahapan Ekstraksi Pesan



**Gambar 4.9** Flowchart Proses Ekstraksi Pesan

Posbit adalah posisi bit untuk menentukan posisi bit pesan disembunyikan. Pjg pesan adalah jumlah panjang pesan yang telah disembunyikan. Brs dan klm adalah awal baris dan kolom letak pesan disembunyikan. Bits adalah variabel untuk menyimpan bit-bit pesan rahasia yang telah diambil. Pesan adalah variabel untuk menyimpan bit-bit pesan yang telah diubah menjadi karakter.



**Gambar 4.10** Flowchart Proses Ekstraksi Pesan A

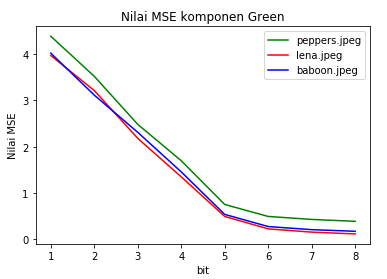
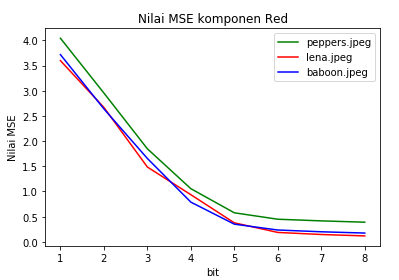
Pada Gambar 4.9, dapat dilihat proses pengambilan pesan dari *stego-image*. Setelah membaca *file* *stego-image* dan menentukan jumlah panjang pesan, selanjutnya pada Gambar 4.10 proses A adalah mengambil bit pesan sesuai dengan posisi bit yang telah ditentukan. Kemudian bit yang didapatkan disimpan ke dalam variabel untuk disimpan sementara sampai memenuhi syarat yaitu jumlah panjang bit sama dengan delapan. Setelah variabel memenuhi syarat, maka bit-bit tersebut diubah menjadi huruf atau karakter. Proses tersebut dilakukan berulang-ulang sampai pesan yang didapatkan melebihi jumlah panjang pesan yang telah ditentukan. Setelah selesai maka akan didapatkan pesan yang disembunyikan di dalam citra. Proses B adalah memindahkan kolom atau baris berikutnya dari piksel citra.

* 1. **Data Hasil**

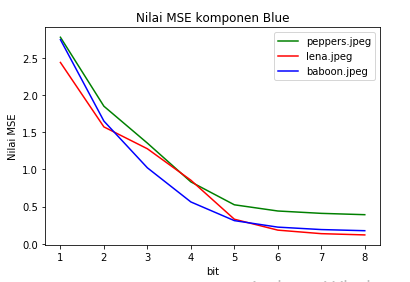
Kualitas gambar yang baik dari citra *stego* adalah syarat penting suksesnya steganografi. Untuk mengetahui perbandingan kualitas gambar sebelum dan sesudah dilakukan penyembunyian pesan, perlu dilakukan perhitungan *Means Square Error* (MSE) dan *Peak Signal to Noise Ratio* (PSNR). Berikut adalah hasil perhitungan MSE menggunakan Persamaan 2.1 yang dapat dilihat pada Tabel 4.2.

**Tabel 4.2 Nilai MSE**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Bit ke** | **Peppers.jpeg** | | | **Lena.jpeg** | | | **Baboon.jpeg** | | | |
| **R** | **G** | **B** | **R** | **G** | **B** | **R** | **G** | **B** |
| **1** | 4,0413 | 4,3871 | 2,7787 | 3,5968 | 3,9708 | 2,4393 | 3,7162 | 4,0217 | 2,7466 |
| **2** | 2,9539 | 3,5216 | 1,8502 | 2,6678 | 3,2139 | 1,5719 | 2,6393 | 3,1152 | 1,6503 |
| **3** | 1,8469 | 2,4838 | 1,3535 | 1,4848 | 2,1814 | 1,2775 | 1,6552 | 2,3115 | 1,0211 |
| **4** | 1,0552 | 1,6986 | 0,8317 | 0,9359 | 1,3500 | 0,8545 | 0,7883 | 1,4569 | 0,5617 |
| **5** | 0,5765 | 0,7535 | 0,5233 | 0,3773 | 0,4944 | 0,3291 | 0,3510 | 0,5391 | 0,3102 |
| **6** | 0,4490 | 0,4950 | 0,4398 | 0,1868 | 0,2263 | 0,1822 | 0,2349 | 0,2768 | 0,2226 |
| **7** | 0,4162 | 0,4308 | 0,4080 | 0,1460 | 0,1564 | 0,1347 | 0,1999 | 0,2099 | 0,1898 |
| **8** | 0,3901 | 0,3901 | 0,3901 | 0,1178 | 0,1178 | 0,1178 | 0,1745 | 0,1745 | 0,1745 |



(a) (b)



(c)

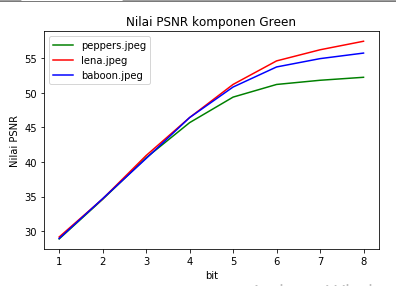
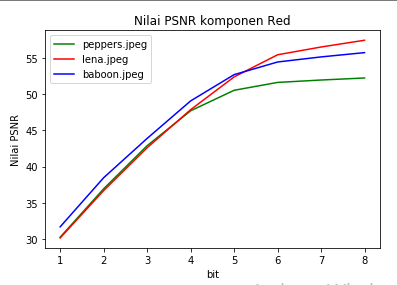
**Gambar 4.11** Grafik MSE (a) Komponen R (b) Komponen G (c) Komponen B

Seperti yang terlihat pada Gambar 4.11, dapat diketahui bahwa nilai MSE pada bit-bit *Most Significant Bit* (MSB) terlihat lebih tinggi dengan nilai tertinggi adalah 4,3871 dibandingkan dengan nilai MSE pada bit-bit *Least Significant Bit* (LSB) dengan nilai tertinggi adalah 0,7535. Hal tersebut terjadi di semua komponen ketiga citra. Berdasarkan hal tersebut juga dapat diketahui bahwa semakin kecil nilai maka semakin besar nilai PSNR*.*

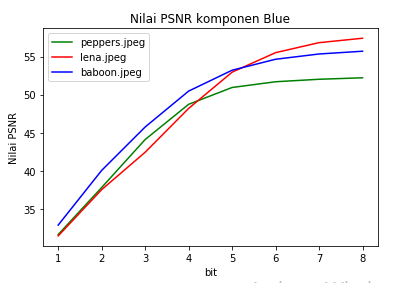
Setelah diperoleh nilai MSE maka perhitungan PSNR dapat dilakukan dan hasil perhitungan PSNR menggunakan Persamaan 2.2 dapat dilihat pada Tabel 4.3.

**Tabel 4.3 Nilai PSNR**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Bit ke** | **Peppers.jpeg** | | | **Lena.jpeg** | | | **Baboon.jpeg** | | |
| **R** | **G** | **B** | **R** | **G** | **B** | **R** | **G** | **B** |
| **1** | 30,2801 | 28,9009 | 31,6032 | 30,1811 | 29,1771 | 31,4243 | 31,7090 | 28,9672 | 32,8436 |
| **2** | 36,9495 | 34,6170 | 37,7885 | 36,7047 | 34,6787 | 37,5076 | 38,4733 | 34,7059 | 40,0364 |
| **3** | 42,8861 | 40,6049 | 44,0962 | 42,6023 | 40,9294 | 42,4241 | 43,8898 | 40,5026 | 45,7442 |
| **4** | 47,6967 | 45,6749 | 48,7300 | 47,8485 | 46,4091 | 48,1871 | 49,0566 | 46,4072 | 50,4662 |
| **5** | 50,5222 | 49,3596 | 50,9425 | 52,3628 | 51,1892 | 52,9563 | 52,6771 | 50,8137 | 53,2131 |
| **6** | 51,6074 | 51,1840 | 51,6976 | 55,4154 | 54,5830 | 55,5238 | 54,4205 | 53,7091 | 54,6551 |
| **7** | 51,9368 | 51,7880 | 52,0234 | 56,4865 | 56,1872 | 56,8370 | 55,1208 | 54,9098 | 55,3475 |
| **8** | 52,2182 | 52,2182 | 52,2182 | 57,4183 | 57,4183 | 57,4183 | 55,7111 | 55,7111 | 55,7111 |



(a) (b)



(c)

**Gambar 4.12** Grafik PSNR (a) Komponen R (b) Komponen G (c) Komponen B

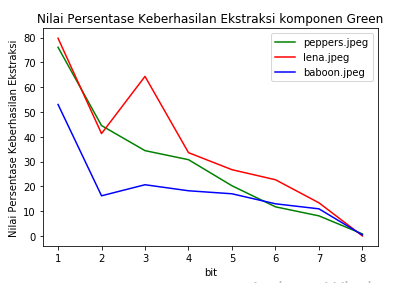
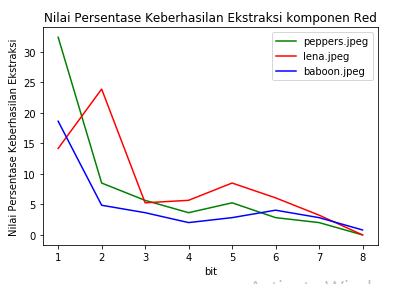
Seperti yang terlihat pada Gambar 4.12, dapat diketahui bahwa nilai PSNR pada bit-bit MSB terlihat lebih rendah dibandingkan dengan nilai PSNR pada bit-bit LSB. Nilai PSNR tertinggi di bit-bit MSB adalah 50,4662 yang berada di bit empat komponen B dari citra Baboon.jpeg sedangkan nilai PSNR tertinggi di bit-bit LSB nilai tertinggi adalah 57,4183 yang berada di bit delapan komponen R,G, dan B dari citra Lena.jpeg. Nilai PSNR tersebut terjadi di tiap-tiap komponen pada ketiga citra. Dan semakin rendah nilai PSNR maka akan semakin rendah kualitas citra.

Selanjutnya adalah menghitung nilai persentase keberhasilan ekstraksi. Perhitungan ini dilakukan untuk mengetahui seberapa baik kesamaan antara pesan hasil ekstraksi dengan pesan asli. Perhitungan persentase keberhasilan ekstraksi ini dilakukan dengan membagi jumlah huruf yang sama dengan panjang pesan. Kemudian hasil dari pembagian tersebut di ubah dalam bilangan persen.

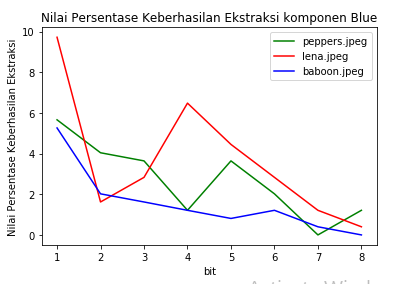
Perhitungan ini dilakukan pada semua pesan hasil ekstraksi yang telah didapatkan dari pesan yang telah disembunyikan sebelumnya di bit satu sampai bit delapan dan disemua komponen pada ketiga citra. Nilai persentase keberhasilan ekstraksi dari citra RGB dapat dilihat pada Tabel 4.4.

**Tabel 4.4 Nilai Persentase Keberhasilan Ekstraksi Citra RGB**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Bit ke** | **Peppers.jpeg** | | | **Lena.jpeg** | | | **Baboon.jpeg** | | |
| **R** | **G** | **B** | **R** | **G** | **B** | **R** | **G** | **B** |
| **1** | 32,38 | 76,11 | 5,66 | 14,17 | 79,76 | 9,72 | 18,62 | 53,04 | 5,26 |
| **2** | 8,50 | 44,53 | 4,04 | 23,89 | 41,30 | 1,62 | 4,86 | 16,19 | 2,02 |
| **3** | 5,66 | 34,41 | 3,64 | 5,26 | 64,37 | 2,83 | 3,64 | 20,65 | 1,62 |
| **4** | 3,64 | 30,76 | 1,21 | 5,67 | 33,60 | 6,48 | 2,02 | 18,22 | 1,21 |
| **5** | 5,26 | 20,24 | 3,64 | 8,50 | 26,72 | 4,45 | 2,83 | 17,00 | 0,81 |
| **6** | 2,83 | 11,74 | 2,02 | 6,07 | 22,67 | 2,83 | 4,05 | 12,96 | 1,21 |
| **7** | 2,02 | 8,09 | 0,00 | 3,24 | 13,36 | 1,21 | 2,83 | 10,93 | 0,40 |
| **8** | 0,00 | 0,80 | 1,21 | 0,00 | 0,00 | 0,40 | 0,81 | 0,40 | 0,00 |



(a) (b)



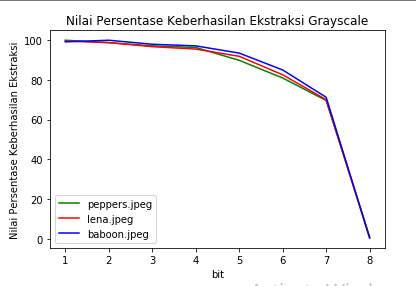
(c)

**Gambar 4.13** Grafik Persentase Keberhasilan Ekstraksi (a) Komponen R (b) Komponen G (c) Komponen B

Dapat dilihat bahwa pada Gambar 4.13 nilai persentase keberhasilan ekstraksi pada ketiga citra di komponen RGB pada bit-bit MSB, nilai persentase lebih tinggi dibandingkan dengan pesan yang disembunyikan di bit-bit LSB. Dimana nilai persentase tertinggi di bit MSB adalah 79,76% yang berada di bit satu komponen G dari citra Lena.jpeg sedangkan nilai persentase tertinggi di bit LSB adalah 26,72% yang berada di bit lima komponen G dari citra Lena.jpeg.

**Tabel 4.5 Nilai Persentase Keberhasilan Ekstraksi Citra Grayscale**

| **Bit ke** | **Peppers.jpeg** | **Lena.jpeg** | **Baboon.jpeg** |
| --- | --- | --- | --- |
| **1** | 100 | 99,60 | 99,19 |
| **2** | 98,79 | 98,79 | 100 |
| **3** | 97,17 | 96,76 | 97,98 |
| **4** | 96,36 | 95,55 | 97,17 |
| **5** | 89,88 | 91,90 | 93,52 |
| **6** | 80,97 | 82,59 | 85,02 |
| **7** | 69,64 | 70,04 | 71,26 |
| **8** | 0,81 | 0,40 | 0,40 |



**Gambar 4.14** Grafik Persentase Keberhasilan Ekstraksi Citra Grayscale

Dapat dilihat bahwa pada Tabel 4.5 nilai persentase keberhasilan ekstraksi citra grayscale pada bit satu sampai kelima memiliki rata-rata diatas 90%. Nilai persentase tersebut lebih baik dibandingkan dengan nilai persentase keberhasilan ekstraksi pada bit-bit ke enam, tujuh, dan delapan.

* 1. **Analisa**

Dari penelitian yang telah dilakukan pada ketiga citra yaitu Peppers.jpeg, Lena.jpeg, dan Baboon.jpeg maka dapat diketahui bahwa nilai MSE pada bit-bit MSB terlihat lebih tinggi dibandingkan dengan nilai MSE dimana pesan disembunyikan pada bit-bit LSB. Hal tersebut terjadi di ketiga citra dan di semua komponen citra yaitu komponen R, G, dan B. Dimana nilai MSE tertinggi di bit-bit MSB adalah 4,3871 yang berada di bit satu komponen G dari citra Peppers.jpeg sedangkan di bit-bit LSB nilai tertinggi adalah 0,7535 yang berada di bit lima komponen G dari citra Peppers.jpeg.

Kualitas gambar yang baik dari citra *stego* adalah syarat penting suksesnya steganografi. Untuk mengetahui perbandingan kualitas gambar sebelum dan sesudah dilakukan penyembunyian pesan, maka perlu dilakukan perhitungan PSNR. Nilai PSNR yang didapatkan menunjukkan berbanding terbalik dengan nilai MSE yaitu nilai PSNR pada bit-bit MSB terlihat lebih rendah dibandingkan dengan nilai PSNR pada bit-bit LSB. Hal tersebut terjadi di ketiga citra dan di semua komponen citra yaitu komponen R, G, dan B. Dimana nilai PSNR tertinggi di bit-bit MSB adalah 50,4662 yang berada di bit empat komponen B dari citra Baboon.jpeg sedangkan nilai PSNR tertinggi di bit-bit LSB nilai tertinggi adalah 57,4183 yang berada di bit delapan komponen R,G, dan B dari citra Lena.jpeg.

Pada penelitian ini nilai MSE yang tinggi dan nilai PSNR yang rendah membuat nilai persentase keberhasilan ekstraksi yang baik. Seperti yang telah diketahui nilai persentase keberhasilan ekstraksi pada bit-bit MSB menunjukkan nilai yang lebih baik dibandingkan dengan nilai persentase pada bit-bit LSB. Akan tetapi hal tersebut berbanding terbalik dengan kualitas citra secara visual, dimana pada bit-bit MSB perubahan citra dapat terlihat secara langsung berbeda dengan penyembunyian yang dilakukann pada bit-bit LSB perubahan citra tidak terlihat.

Berdasarkan penelitian yang telah dilakukan, dapat diketahui bahwa nilai persentase keberhasilan ekstraksi berbeda-beda di setiap bit pesan disembunyikan. Hal tersebut disebabkan dari pengaruh posisi bit dimana pesan yang disembunyikan ke dalam *cover-image*. Seperti yang telah diketahui, nilai persentase keberhasilan ekstraksi pada bit-bit MSB lebih tinggi dibandingkan dengan pesan yang disembunyikan di bit-bit LSB. Selain pengaruh dari posisi bit, *cover-image* yang digunakan adalah citra berformat *lossy* juga memiliki pengaruh yang besar terhadap nilai persentase keberhasilan ekstraksi, dimana citra *lossy* memiliki kompresi yang secara langsung dapat memengaruhi nilai piksel, sehingga merusak informasi yang disembunyikan. Kompresi *lossy* dapat membuang atau menghilangkan data yang bersifat redundant, sehingga suatu data hanya terdiri atas sebagian besar informasi yang dibutuhkan dari data itu sendiri.