

Perancangan Dan Simulasi Kompresi Citra Digital Dengan Metode *Discrete Cosine Transform* Menggunakan Matlab Institut Teknologi Nasional Bandung

FIRMAN OKTA NUGRAHA ¹, LITA LIDYAWATI,S.T.,M.T. ¹,

LUCIA JAMBOLA,S.T.,M.T ¹

¹Institut Teknologi Nasional
Email : firmankt@mhs.itenas.ac.id

Received DD MM YYYY | *Revised* DD MM YYYY | *Accepted* DD MM YYYY

ABSTRAK

Pada penelitian ini, kompresi citra menggunakan metoda Discrete Cosine Transform dengan perangkat lunak MATLAB bertujuan untuk menganalisis ukuran citra asli dengan citra hasil kompresi, menghitung presentase rasio kompresi dan untuk mengetahui kualitas citra yang didapat. Nilai MSE yang semakin rendah mengakibatkan kualitas citra semakin baik sedangkan nilai PSNR yang semakin besar juga menunjukkan kualitas citra semakin baik. Dari hasil penelitian ini dapat disimpulkan bahwa untuk kompresi tanpa serangan dengan menggunakan gambar jpg dan bmp terdapat perbedaan terhadap citra yang dikompresi. Citra hasil kompresi tanpa serangan memiliki rasio kompresi terendah dengan presentase sebesar 23.7469% dan terbesar 43.3534%. Sedangkan untuk citra dengan noise, rasio kompresi terendah 0.1% dan terbesar 49.5439%.

Kata kunci: *Kompresi Citra, DCT, Matlab, Lossy compression ,Lossless Compression.*

ABSTRACT

In this study, image compression using the Discrete Cosine Transform method with MATLAB software aims to analyze the size of the original image with the compressed image, calculate the percentage of compression ratio and to know the quality of the image obtained. The lower the MSE value, the better the image quality, while the higher the PSNR value also indicates the better image quality. From the results of this study, it can be concluded that for compression without attack using jpg and bmp images, there are differences in compressed images. The compressed image without attack has the lowest compression ratio with a percentage of 23.7469% and the largest is 43.3534%. As for the image with noise, the lowest compression ratio is 0.1% and the largest is 49.5439%.

Keywords: *Image Compression, DCT, Matlab, Lossy compression ,Lossless Compression*

1. PENDAHULUAN

Data atau informasi sekarang ini tidak hanya disajikan dalam bentuk teks saja, tetapi juga dapat berupa gambar, suara dan video. Ke-empat macam data ini sering disebut multimedia. Citra merupakan salah satu media yang sering digunakan untuk pertukaran informasi, citra biasanya dihasilkan dari setiap perangkat yang mempunyai kamera atau perangkat kamera itu sendiri. Citra yang dihasilkan biasanya mempunyai ukuran yang relatif besar untuk itu diperlukan penyimpanan yang besar juga (**Rafsyam, 2010**).

Masalah pada citra adalah besarnya ruang penyimpanan yang diperlukan karena file-file gambar yang didapat memakan *space* yang cukup besar. Selain itu, data citra yang berukuran besar jika dikirim melalui jaringan akan membuat pengiriman citra dari satu tempat ke tempat lain menjadi lambat. Untuk mengatasi masalah tersebut maka dibutuhkan suatu teknik yang dapat memperkecil ukuran data gambar yaitu dengan mengurangi duplikasi data, teknik tersebut dinamakan teknik kompresi. Kompresi terhadap citra digital memiliki tujuan untuk menghilangkan duplikasi data pada citra digital serta meningkatkan efisiensi penyimpanan dan transmisi. Kompresi ini dilakukan untuk mendapatkan gambar yang memiliki jumlah bit yang lebih kecil dibandingkan citra aslinya, dan harus mampu merepresentasikan citra aslinya tanpa menyebabkan penurunan kualitas secara kasat mata (**Rina, Triasmoro, & Wirayuda, 2009**).

Adapun tujuan dari kompresi citra ini yaitu : 1) mengecilkan ukuran file; 2) melihat hasil MSE, PSNR dan ukuran file setelah terkompresi; 3) melihat pengaruh kuantisasi faktor kualitas terhadap file citra.

Discrete Cosine Transform merupakan salah satu metode transformasi yang digunakan untuk kompresi data citra yang mempunyai sifat *lossy* (**Hidayatullah, 2017**). Metode kompresi DCT ini menggunakan pendekatan nilai kosinus. Pada dasarnya DCT akan mengubah detail warna dari gambar asli, namun karena keterbatasan indra manusia, maka perubahan yang terjadi tidak begitu terlihat. Dalam operasi DCT ini yang digunakan adalah nilai *real*. Ada 2 macam persamaan yang bisa digunakan yaitu DCT 1 dimensi yang digunakan untuk menghitung data vektor, dan DCT 2 dimensi yang digunakan untuk data matriks. Untuk proses kompresi digunakan DCT 2 dimensi dengan persamaan (1) ditunjukkan sebagai berikut :

$$F(u, v) = C(u)C(v) \sum_{Y=0}^{N-1} \sum_{X=0}^{N-1} f(x, y) \cos \frac{x(2x+1)u}{2N} \cos \frac{x(2y+1)v}{2N} \quad (1)$$

Keterangan:

1. $F(u,v)$ adalah titik koordinat dari matriks yang telah mengalami transformasi DCT 2 dimensi.
2. N dan M adalah banyak kolom dan baris. Apabila ukuran matriks adalah 8×8 , maka nilai N dan M adalah 8.
3. $C(u)$ dan $C(v)$ adalah himpunan hasil yang nilainya ditentukan dari nilai koefisien u dan v .
4. $f(x,y)$ adalah nilai pixel dari matrik pada titik (x,y) . π bernilai 180° .

Frekuensi *horizontal* meningkat dari kiri ke kanan, dan frekuensi *vertical* meningkat dari atas ke bawah. Konstanta bernilai fungsi dasar di kiri atas sering disebut fungsi dasar DC, dan koefisien DCT pada masing-masing *array* dimensi. Sebagai contoh, jika himpunan bilangan *real* disajikan dalam *array* 2 dimensi, maka DCT dua dimensi dilakukan dengan cara melakukan DCT satu dimensi terhadap masing-masing baris dan kemudian melakukan DCT satu dimensi terhadap masing-masing kolom dari hasil DCT tersebut.

Cara kerja transformasi domain frekuensi DCT yaitu pertama, DCT membagi gambar menjadi blok-blok yang besar setiap blok-nya $N \times N$ piksel sehingga tidak ada variasi warna

yang cukup berarti dalam ukuran seperti ini. Kemudian, format gambar RGB yang dimiliki oleh gambar pada umumnya dikonversi menjadi *YCbCr*. Manusia memiliki sensitifitas yang lebih tinggi terhadap *luminance* dibandingkan terhadap *chrominance* sehingga kita dapat memanfaatkan perbedaan-perbedaan yang terdapat pada *chrominance* tersebut untuk melakukan perubahan. Setelah itu, masing-masing nilai *YCbCr* tersebut dikurangi, hal ini dilakukan karena transformasi ke domain frekuensi DCT yang di desain untuk bekerja pada piksel dengan interval tersebut. Kemudian dilakukan perubahan terhadap blok 8 x 8 piksel tersebut menjadi domain frekuensi menggunakan *Discrete Cosine Transform* tipe kedua untuk dua dimensi sehingga kita memperoleh representasi warna dari *YCbCr* dalam bentuk domain frekuensi. Jika DCT diterapkan blok 8 x 8 maka dengan $N = 8$ dan x, y bernilai 0 sampai 7 (**Wijaya, 2012**). Setelah dilakukan proses kompresi dilakukan pengukuran terhadap citra untuk melihat output pada citra dengan menghitung MSE, PSNR dan rasio kompresi. *Peak Signal Noise to Ratio* (PSNR) merupakan parameter yang paling sering digunakan untuk mengukur kualitas citra hasil rekonstruksi pada kompresi *lossy*. PSNR merupakan pengukuran citra secara kualitatif menggunakan *Mean Square Error* (MSE) dari rekonstruksi citra. Berdasarkan kedua parameter ini, kualitas rekonstruksi citra dapat dikatakan baik atau mendekati nilai citra aslinya. Nilai MSE yang kecil dan PSNR yang tinggi akan memberikan nilai kualitas citra yang semakin baik. PSNR diukur dalam decibel. PSNR didefinisikan dengan rumus (2) sebagai berikut (**Alfiah, 2017**):

$$PSNR = 10 \log_{10} \left(\frac{L^2}{MSE} \right) \quad (2)$$

sedangkan untuk MSE didefinisikan dengan rumus (3) :

$$MSE = \frac{1}{M \times N} \sum_{x=1}^M \sum_{y=1}^N (S(x, y) - r(x, y))^2 \quad (3)$$

Rasio kompresi dalam kompresi citra adalah perbandingan antara besar file citra hasil kompresi dengan besar citra asli. Hal ini dilakukan untuk melihat seberapa efektif kompresi dilakukan, semakin besar presentase nilai rasio maka citra hasil kompresi akan semakin terlihat menyerupai citra asli. Demikian pula sebaliknya, semakin kecil presentase nilai rasio maka citra hasil kompresi akan semakin buruk. Perhitungan rasio kompresi dan nilai presentasinya dapat dilakukan dengan persamaan (4) dan (5) berikut (**Noviardhi, 2008**) :

$$Rasio\ Kompresi = \frac{ukuran\ citra\ asli}{ukuran\ citra\ terkompresi} \quad (4)$$

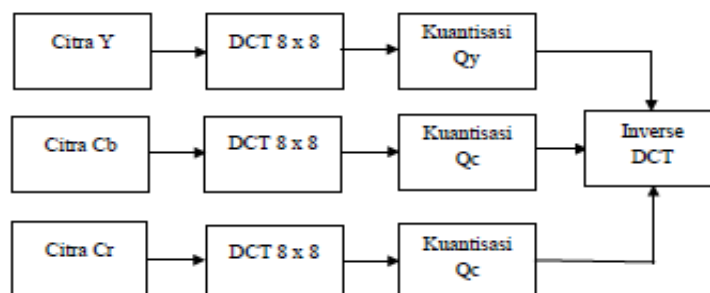
$$Presentase\ rasio\ kompresi = \frac{ukuran\ hasil\ kompresi}{ukuran\ citra\ asli} \times 100\% \quad (5)$$

Noise pada citra tidak hanya terjadi karena ketidaksempurnaan dalam proses *capturing*, tetapi bisa juga disebabkan oleh kotoran-kotoran yang terjadi pada citra. *Noise* muncul biasanya sebagai akibat dari pembelokan *pixel* yang tidak baik. Gangguan tersebut umumnya berupa variasi intensitas suatu *pixel* yang tidak berkorelasi dengan *pixel* tetangganya. Secara visual, gangguan mudah dilihat oleh mata karena tampak berbeda dengan *pixel* tetangganya. *Pixel* yang mengalami gangguan umum memiliki frekuensi tinggi. Komponen citra yang berfrekuensi rendah umumnya mempunyai nilai *pixel* konstan atau berubah sangat lambat (**Sastia & Firman, 2016**). 1) *Gaussian Noise*, merupakan jenis *noise* yang mengikuti distribusi normal standar dengan rata-rata 0 dan standar deviasi 1. *Gaussian noise* dapat dibangkitkan dengan cara membangkitkan bilangan acak dengan nilai berikisar antara 0 dan 1. Kemudian pada titik-titik yang terkena *noise*, nilai fungsi citra ditambahkan dengan *noise* yang ada. Efek dari *gaussian noise* pada citra adalah munculnya

titik-titik berwarna yang jumlahnya sama dengan presentase *noise* (Prihatini & Tuti, 2010). 2) *Salt & Pepper Noise*, biasa dinamakan sebagai *noise impuls* positif dan negative. *Noise* ini disebabkan karena terjadinya error bit dalam pengiriman data, *pixel-pixel* yang tidak berfungsi dan kerusakan pada lokasi memori. Bentuk *noise* ini berupa bintik-bintik hitam atau putih di dalam citra (Kadir & Susanto, 2013). 3) *Speckle Noise*, merupakan model *noise* yang memberikan warna hitam pada titik yang terkena *noise*. *Speckle noise* dapat dibangkitkan dengan membangkitkan bilangan 0 (warna hitam) pada titik-titik yang secara probabilitas lebih kecil dari nilai probabilitas *noise* (Prihatini & Tuti, 2010).

2. METODOLOGI PENELITIAN

Metode kompresi yang digunakan pada penelitian ini, dijelaskan dalam Gambar 1 :



Gambar 1 Diagram Proses Kompresi

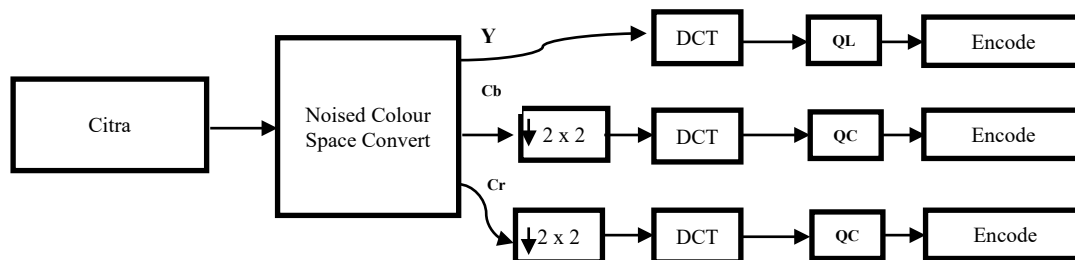
2.1. Proses Kompresi

Proses kompresi citra diawali dengan memanggil citra kedalam *program* citra yang digunakan ada 2 yaitu citra lena yang bertipe JPG dengan 256 x 256 pixel dan citra *cat* yang bertipe bmp dengan 256 x 256 pixel, dengan cara mencari pada sistem *drive* yang digunakan. Setelah penyisipan citra berhasil, citra akan digunakan sebagai *input* proses DCT, dimana blok 8 x 8 akan dipanggil untuk diubah menjadi fungsi matriks *cosinus*



Gambar 2. Konversi RGB ke YCbCr.

Kemudian citra diserang dengan *gaussian*, *salt & pepper* dan *speckle*, setelah diserang citra masuk kedalam proses *sampling* yaitu proses pengkonversian *data pixel* dari RGB ke YCbCr seperti pada Gambar 2, Y adalah komponen *luminance*, Cb dan Cr adalah komponen *chrominance* dari citra. Pada komponen *chrominance* dilakukan proses subsampling untuk mengurangi ukuran *file*.



Gambar 3. Simulasi Kompresi *Discrete Cosine Transform*.



Selanjutnya, program akan melakukan ekstraksi citra YCbCr yang sudah diserang. Citra yang sudah terserang ini masuk ke dalam kuantisasi yaitu proses pembersihan koefisien DCT yang tidak penting untuk pembentukan citra baru. Hal ini yang menyebabkan kompresi citra JPG bersifat *lossy* karena ada informasi yang hilang akibat pembulatan Gambar 3.

Kemudian dilanjutkan dengan proses dekuantisasi dan IDCT. Proses dari IDCT ini yaitu sebuah citra yang telah direkonstruksi, dan nilai matriks citra yang telah direkonstruksi juga. Untuk tahap ekstraksi citra yang sudah dikompresi dan diberi *noise* akan menghasilkan 3 buah gambar.

3. DATA PENGAMATAN DAN ANALISA

Simulasi dibuat dengan melakukan pemograman yang menggunakan *software* MATLAB R2015b, untuk percobaan simulasi dilakukan 2 kali yang pertama tidak memakai *noise* pada citra dan yang kedua memakai *noise*. Untuk citra yang digunakan ada dua tipe yaitu JPG dan bmp seperti pada Tabel 1 :

Tabel 1 Citra Kompresi Yang Akan Dipakai

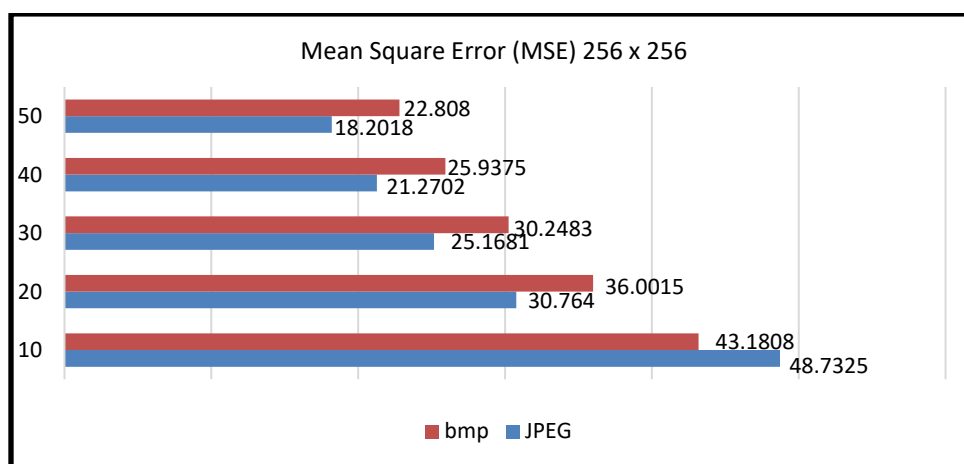
No	<i>Pixel</i> dan Jenis Gambar	Gambar
1	256 x 256 (jpg)	
2	256 x 256 (bmp)	

Faktor kualitas yang dilakukan pada percobaan pertama yaitu 10 sampai 50 dan untuk percobaan kedua memakai faktor kualitas 30, 60 dan memakai α 0.01, 0.02. Langkah pertama. dalam pengujian ini yaitu memanggil citra kedalam program MATLAB, kemudian untuk pengujian citra yang pertama setelah citra dipanggil lalu dimasukan kedalam program

DCT yang sudah dibuat dan memilih kuantisasi yang akan dipakai, untuk percobaan yang kedua setelah gambar dipanggil memilih kuantisasi, α dan *noise* yang akan dipakai pada citra. Langkah terakhir yaitu menghitung kualitas citra yang didapat yaitu dengan menghitung MSE, PSNR dan rasio kompresi.

3.1 Pengujian Data MSE

Pada kompresi biasa pada saat percobaan dengan gambar jpg kuantisasi faktor kualitas 10 mendapatkan MSE = 48.7325 dan untuk gambar bmp MSE =43.1808 ini bisa dikategorikan kualitas citra buruk karena nilai yang tinggi tetapi ketika kuantisasi faktor kualitas dinaikan MSE semakin turun faktor kualitas menunjukkan tingkat kualitas pemampatan dengan tujuan untuk mendapatkan beberapa variasi hasil dari kompresi. Hasil MSE yang didapatkan dari kompresi citra tanpa serangan terlihat pada Gambar 4:

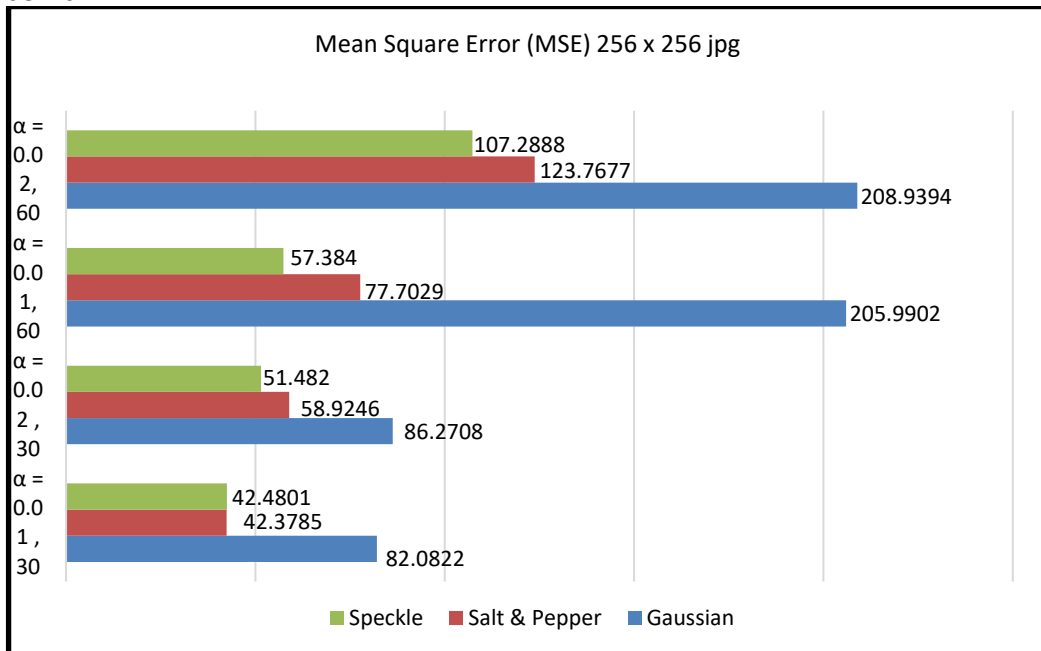


Gambar 4. Grafik Pengukuran Nilai MSE pada Citra Lena dan Cat jpg 256 x 256.

Jika dilihat dari segi citranya terlihat perbedaan antara citra bmp dan jpg yaitu : 1) Hasil citra bmp menghasilkan citra berbentuk *lossless* yang artinya dalam proses kompresinya tidak ada informasi yang hilang dan menghasilkan citra hasil kompresi yang sama dengan citra semula biasanya citra bmp ini digunakan di citra medis, citra biner. 2) Hasil citra jpg menghasilkan citra berbentuk *lossy* yang artinya menghasilkan citra hasil kompresi yang hampir sama dengan citra semula namun ada informasi yang hilang dalam batas toleransi tertentu, teknik dari kompresi *lossy* yaitu mengubah detail pada warna citra menjadi lebih sederhana.

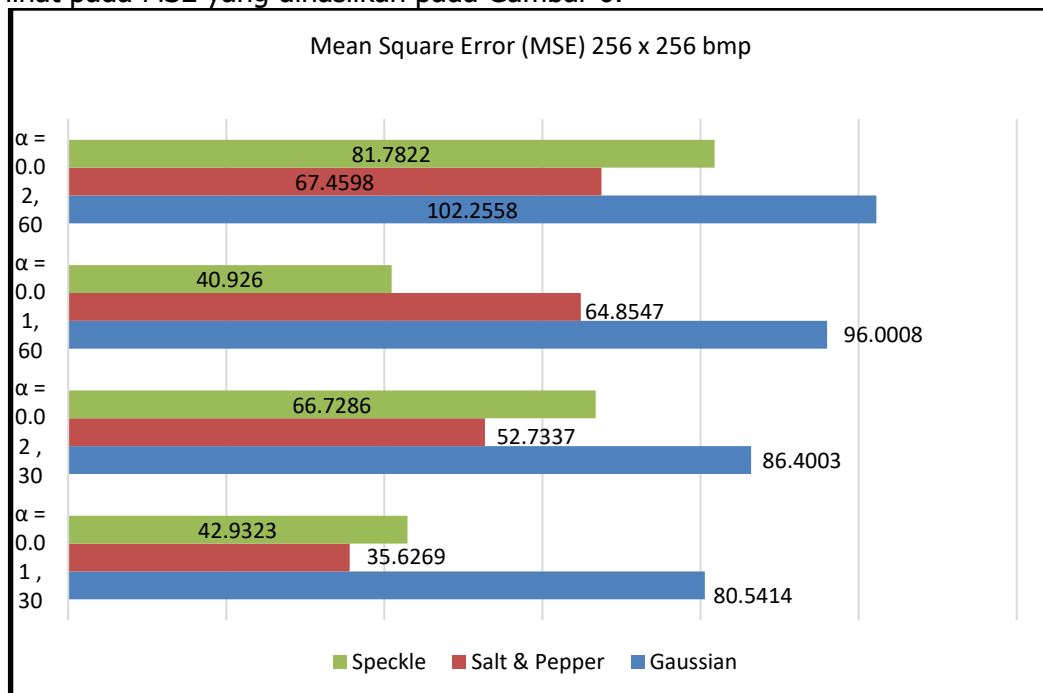
Pada percobaan citra lena kompresi dengan *noise gaussian*, *salt & pepper* dan *speckle* terdapat perbedaan yang sangat signifikan terhadap masing masing citra yaitu pada saat citra jpg terkena *noise gaussian* saat citra ditambahkan bit dari *noise* tersebut citra masih bisa dikompresi namun perekdusian bit tidak begitu efektif dengan kompresi DCT sehingga output dari citra tersebut hanya menyamakan *noise gaussian* itu sendiri. MSE yang dihasilkan sangat buruk yang buruk untuk suatu kualitas citra bisa terlihat pada histogram dan simulasi $\alpha = 0.02$ dan faktor kualitas 60 bahwa pemampatan ini tidak tahan dengan *noise gaussian* dan warna yang dihasilkan menjadi lebih buruk terlihat pada Gambar 5, percobaan *noise salt & pepper* hasil MSE tidak terlalu besar seperti *noise gaussian* ini terjadi karena *noise* yang dihasilkan oleh *gaussian* itu berupa *distortion* warna putih sehingga derau yang dihasilkan cukup besar dampaknya, berbeda dengan *salt & pepper* yaitu warna solid dari putih dan hitam tetapi karena citra ini RGB maka *noise* yang diperoleh bintik merah, hijau dan biru. Hasil output citra pada percobaan $\alpha = 0.01$ dan faktor kualitas 30 menghasilkan MSE = 42.3785 yang artinya citra tidak terlalu buruk, untuk percobaan citra cat dengan format bmp hasil citra cukup bagus ada pada *noise salt & pepper* dengan $\alpha =$

0.01 dan faktor kualitas 30 menghasilkan MSE = 35.6259, untuk percobaan dengan noise lain format bmp ini dalam proses kompresinya tidak ada informasi yang hilang namun pada histogram terjadi perubahan karena warna yang tercampur oleh *noise* terlihat pada MSE yang dihasilkan.



Gambar 5. Grafik Pengukuran Nilai MSE 256 x 256 Lena (JPG) dengan Noise.

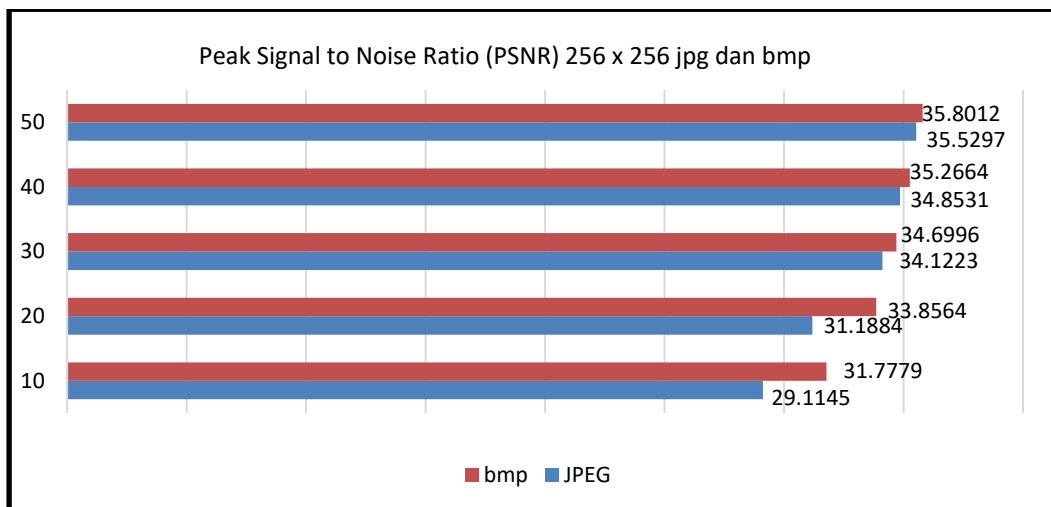
lalu percobaan citra cat dengan format bmp hasil citra cukup bagus ada pada *noise salt & pepper* dengan $\alpha = 0.01$ dan faktor kualitas 30 menghasilkan MSE = 35.6259, untuk percobaan dengan noise lain format bmp ini dalam proses kompresinya tidak ada informasi yang hilang namun pada histogram terjadi perubahan karena warna yang tercampur oleh *noise* terlihat pada MSE yang dihasilkan pada Gambar 6.



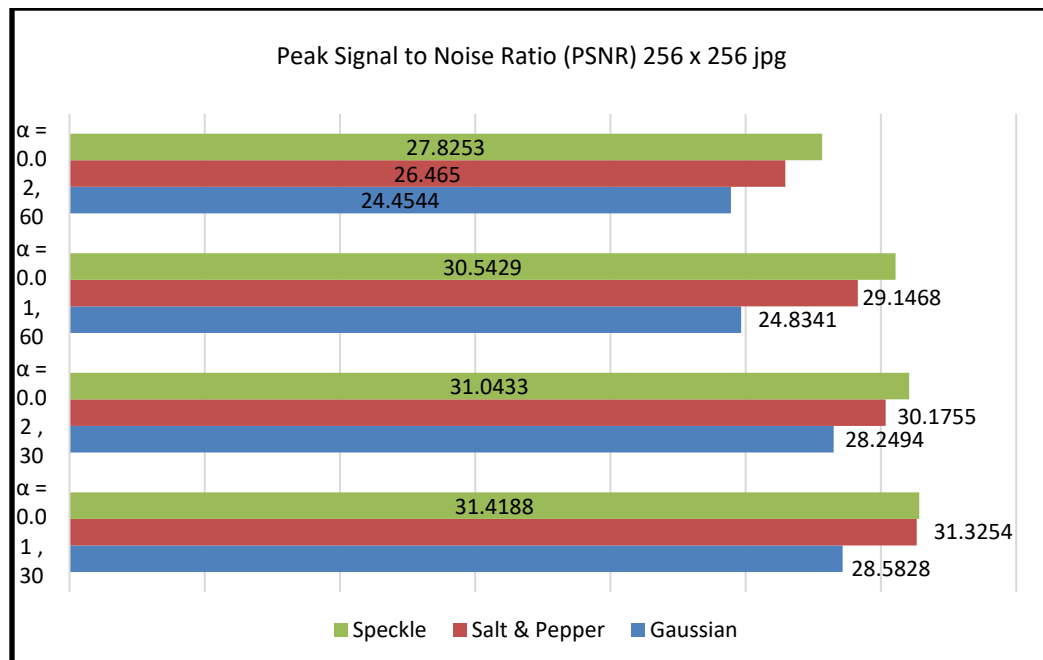
Gambar 6. Grafik Pengukuran Nilai MSE 256 x 256 Cat (JPG) dengan Noise.

3.2 Pengujian Data PSNR

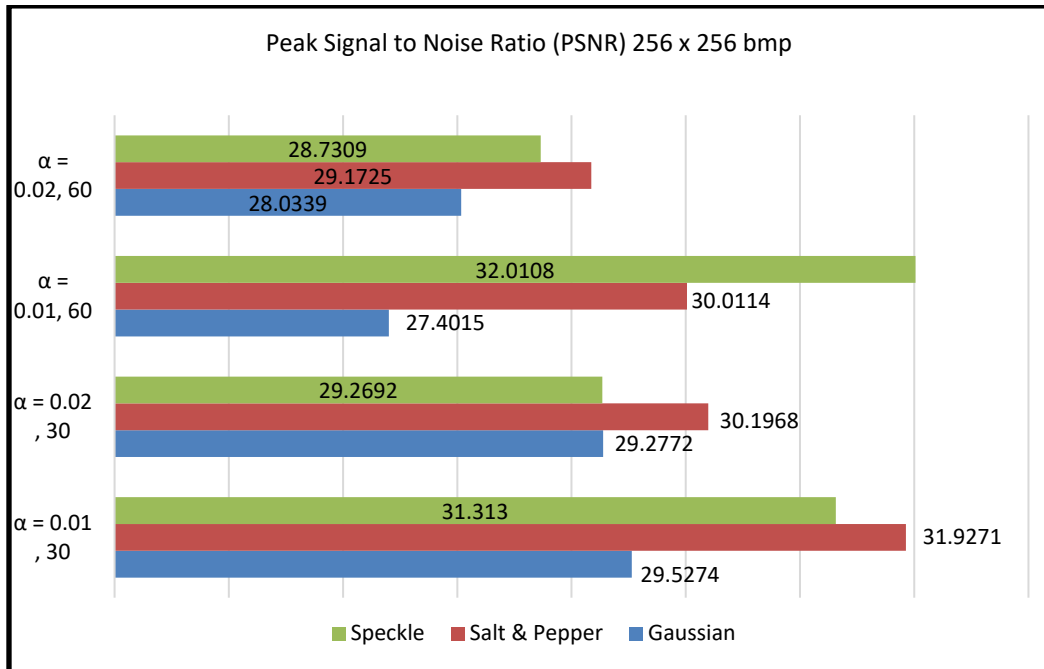
Dari hasil simulasi yang didapat dari kompresi biasa dan kompresi dengan *noise gaussian*, *salt & pepper* dan *speckle* terdapat hasil PSNR yang bervariasi untuk kompresi tanpa serangan semakin dinaikan faktor kualitas maka PSNR akan semakin naik artinya kualitas citra semakin baik terlihat pada Gambar 7, sebaliknya dengan citra yang sudah terkena *noise* ketika α dinaikan dan faktor kualitas dinaikan maka PSNR akan turun hal ini dipengaruhi oleh bit tambahan dari *noise* sehingga bit error semakin bertambah namun kompresi DCT tidak dapat menghilangkan bit yang tidak diperlukan secara efektif terlihat pada Gambar 8 dan Gambar 9.



Gambar 7. Grafik Pengukuran Nilai PSNR 256 x 256 pada Kompresi Tanpa Serangan.



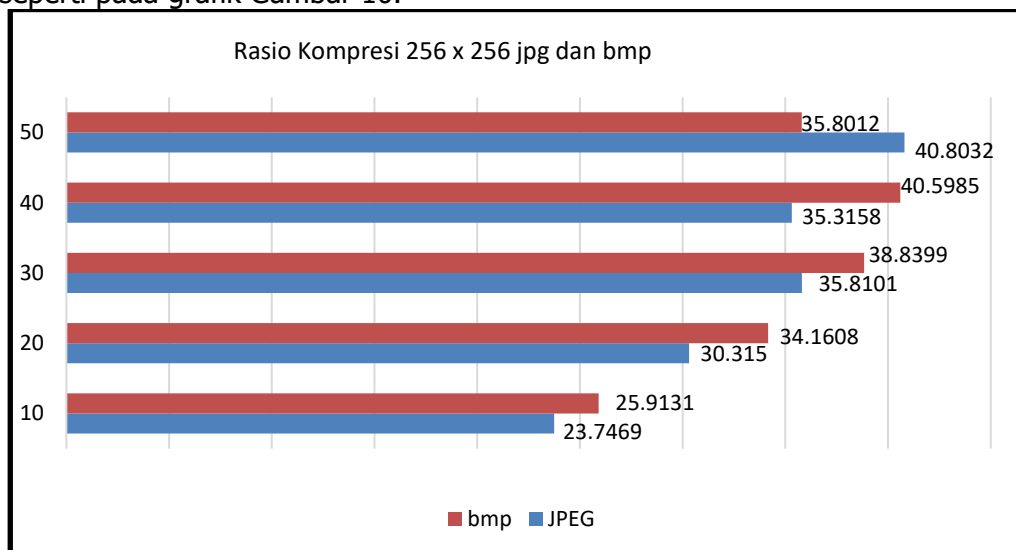
Gambar 8. Grafik Pengukuran Nilai PSNR 256 x 256 pada Gambar JPG.



Gambar 9. Grafik Pengukuran Nilai PSNR 256 x 256 pada Gambar BMP.

3.3 Pengujian Data Rasio Kompresi

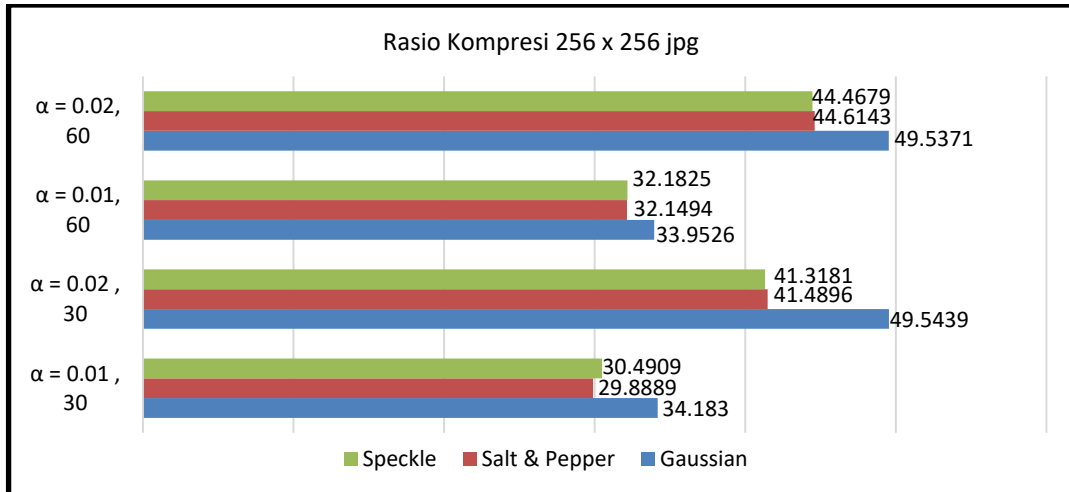
Hasil simulasi yang didapat pengukuran rasio kompresi dari file jpg menggunakan kompresi tanpa *noise* hasilnya ketika kuantisasi faktor kualitas dinaikan rasio kompresi semakin naik dalam kompresinya ada informasi yang hilang namun dalam batas toleransi tertentu yaitu kuantisasi, untuk kompresi dengan *lossy compression* biasanya digunakan pada citra yang tidak memerlukan detail citra dimana ketika bit hilang tidak berpengaruh hal ini mempengaruhi pada ukuran file citra pada citra berbeda dengan format bmp dalam kompresinya sangat rendah karena citra bmp adalah citra grafis yang terdiri dari susunan titik maka pada saat dilihat dari jauh citra yang terkompresi hampir tidak terlihat bahwa suatu data telah hilang dari informasi tersebut tetapi ketika di perbesar terlihat pengurangan datanya seperti pada grafik Gambar 10.



Gambar 10. Grafik Pengukuran Rasio Kompresi 256 x 256 Kompresi Tanpa Serangan

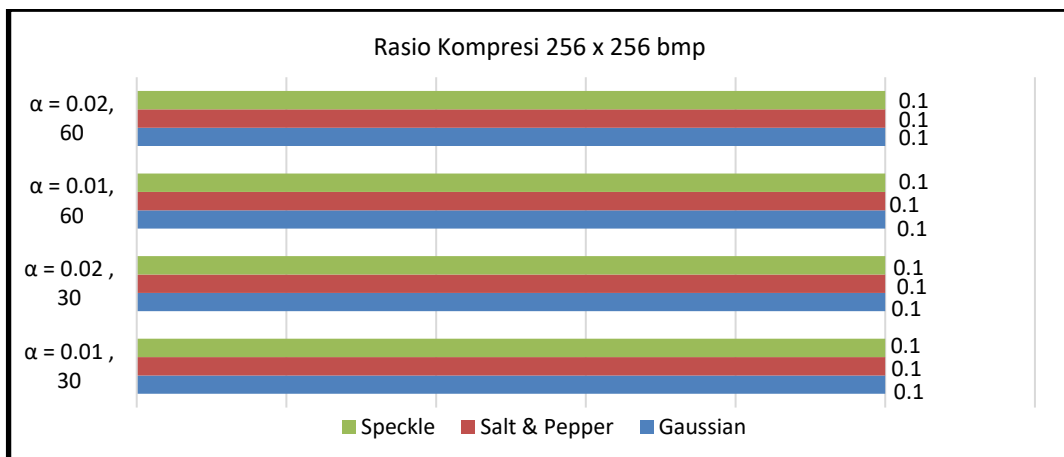
Untuk pengukuran rasio kompresi dengan *noise* Gambar 11 terdapat perbedaan yang sangat besar karena adanya penambahan bit pada citra, ketika noise ditambahkan ukuran file citra

menjadi bertambah pada format JPG sehingga menghasilkan rasio kompresi menjadi besar. Jika dilihat hasil citra *output* kompresi dengan serangan ini hasilnya kurang baik karena tidak semua *noise* yang timbul di gambar tidak hilang. Untuk perbandingan citra dengan kompresi tidak menggunakan serangan keluaran citra lebih baik karena PSNR yang naik dan menurunnya MSE.



Gambar 11. Grafik Pengukuran Rasio Kompresi 256 x 256 JPG dengan Noise

Untuk rasio kompresi dengan serangan pada citra BMP ini hasil rasio kompresi yang dihasilkan sangat rendah karena sifat dari citra BMP ini sendiri yaitu *lossless* yang artinya rasio kompresi yang didapatkan pada citra ini sangat rendah dan terbatas dan dalam kompresinya seperti pada Gambar 12 Untuk kualitas gambar yang didapat pixel tidak pecah namun *noise* pada gambar masih ada tidak sepenuhnya hilang. Hal ini dikarenakan dari masing masing tipe citra mempunyai sifat yang berbeda. Perbedaan *lossy compression* dan *lossless compression* : 1) Pada *Lossless Compression* akan menghasilkan citra hasil kompresi yang hampir sama dengan semula , Dalam proses kompresinya, informasi yang hilang hanya sedikit dan Rasio kompresi sangat rendah. 2) Pada *lossy compression* akan menghasilkan rasio kompresi tinggi, menghasilkan citra hasil kompresi yang hampir sama dengan citra semula, dalam proses kompresinya, ada informasi yang hilang namun dalam batas toleransi tertentu dan biasanya digunakan pada citra foto atau image lain yang tidak terlalu memerlukan detail citra, dimana kehilangan bit rate foto tidak berpengaruh pada citra.



Gambar 12. Grafik Pengukuran Rasio Kompresi 256 x 256 BMP dengan Noise

4. KESIMPULAN

Berdasarkan hasil simulasi terlihat bahwa semakin naik PSNR maka citra yang dikeluarkan semakin baik begitu pula dengan pengukuran MSE semakin turun nilainya maka citra yang dikeluarkan juga semakin baik. Untuk kompresi tanpa serangan pengaruh kualitas citra ada pada kuantisasi faktor kualitas semakin nilai tinggi nilai kuantisasi faktor kualitas citra yang dikeluarkan juga semakin bagus, berbeda dengan pengujian kompresi dengan serangan hasil yang dikeluarkan setelah kompresi bit yang ditambahkan pada citra tidak sepenuhnya hilang sehingga *noise* yang diberikan pada citra masih ada pada output citra namun dari ketiga *noise* yang dipakai *gaussian*, *salt & pepper* dan *speckle* hasil citra yang tidak terlalu buruk ada pada *salt & pepper* dengan Hasil output citra pada percobaan $\alpha = 0.01$ dan faktor kualitas 30 menghasilkan MSE = 42.3785. Citra pada format bmp yang mempunyai sifat *lossless* jika dikompresi dan disimpan ke format bmp kembali maka hasilnya akan sama seperti sama hanya kualitas citranya saja berubah rasio kompresi format bmp pada kompresi dengan *noise* = 0.1%

DAFTAR PUSTAKA

- Alfiah, F., Setiadi, A., Supriadi, A (2017). *Discrete Cosine Transform DCT Methods on Compression RGB and Grayscale image, International Journal of Computer Techniques*, vol.4, no.6.
- Hidayatullah, P. (2017). *Pengolahan Citra Digital Teori dan Aplikasinya*. Bandung: Informatika Bandung.
- Kadir, A., Susanto, A. (2013). *Teori dan Aplikasi Pengolahan Citra*. Yogyakarta : ANDI.
- Noviardhi, A. G. (2008). *Kompresi Citra Menggunakan Metode Discrete Cosine Transform (DCT)*. Yogyakarta : Universitas Sanata Dharma.
- Prihatini, Tuti Adi. (2010). Analisis dan Implementasi Low Pass Filter untuk Mereduksi Noise pada Citra Digital . Universitas Sumatera Utara.
- Rafsyam, Y. (2010). Kompresi Citra Menggunakan Teknik Lossy Dengan Metode Algoritma JPEG. *Elektron* : Vol. 2 No. 1. ISSN No. 2085-6989.
- Rina, A., Triasmoro, I.I., Wirayuda, T.A.B. (2009). Analisis Dan Simulasi Kompresi Citra Digital Dengan Metode Fractal Dan Discrete Cosine Transform (DCT) Berbasis Entropi. Universitas Telkom Bandung.
- Sastia, H.W, Firman, S. (2016). Penerapan Metode *Gaussian Smoothing* Untuk Mereduksi *noise* Pada Citra Digital. Universitas Muhammadiyah Bengkulu.
- Wijaya, A. R. (2012). *Kompresi Citra Berwarna Dengan Penerapan Discrete Cosine Transform (DCT)*. Surakarta: Universitas Muhammadiyah Surakarta.