ALGORITMA KRIPTOGRAFI DAN STEGANOGRAFI UNTUK PENGAMANAN PESAN KE DALAM CITRA

SKRIPSI

OLEH
KHILMI HANI
NIM. 16610093

JURUSAN MATEMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2020

ALGORITMA KRIPTOGRAFI DAN STEGANOGRAFI UNTUK PENGAMANAN PESAN KE DALAM CITRA

SKRIPSI

Diajukan Kepada
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang
untuk Memenuhi Salah Satu Persyaratan dalam
Memperoleh Gelar Sarjana Matematika (S.Mat)

Oleh
Khilmi Hani
NIM. 16610093

JURUSAN MATEMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2020

ALGORITMA KRIPTOGRAFI DAN STEGANOGRAFI UNTUK PENGAMANAN PESAN KE DALAM CITRA

SKRIPSI

Oleh
Khilmi Hani
NIM. 16610093

Telah Diperiksa dan Disetujui untuk Diuji
Tanggal 12 Agustus 2020

Pembimbing I,	Pembimbing II,
Muhammad Khudzaifah, M.Si	Juhari, M.Si
NIDT. 19900511 20160801 1 057	NIP. 19840209 20160801 1 055

Mengetahui
Ketua Jurusan Matematika

Dr. Usman Pagalay, M.Si
NIP. 19650414 200312 1 001

ALGORITMA KRIPTOGRAFI DAN STEGANOGRAFI UNTUK PENGAMANAN PESAN KE DALAM CITRA

SKRIPSI

Oleh
KHILMI HANI
NIM. 16610093

Telah Dipertahankan di Depan Dewan Penguji Skripsi

dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
untuk Memperoleh Gelar Sarjana Matematika (S.Mat)
Tanggal 13 Mei 2020

Penguji Utama	: Mohammad Jamhuri, M.Si	.....
Ketua Penguji	: Mohammad Nafie Jauhari, M.Si	.....
Sekretaris Penguji	: Muhammad Khudzaifah, M.Si	.....
Anggota Penguji	: Juhari, M.Si	.....

Mengesahkan,
Ketua Jurusan Matematika

Dr. Usman Pagalay, M.Si
NIP. 19650414 200312 1 001

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini:

Nama : Khilmi Hani

NIM : 16610093

Jurusan : Matematika

Fakultas : Sains dan Teknologi

Judul Skripsi : Algoritma Kriptografi dan Steganografi untuk Pengamanan Pesan ke dalam Citra,

menyatakan dengan sebenarnya bahwa skripsi yang saya tulis ini benar-benar merupakan hasil karya sendiri, bukan merupakan pengambilan data, tulisan, atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar pustaka. Apabila di kemudian hari terbukti atau dapat dibuktikan skripsi ini hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 12 Agustus 2020
Yang membuat pernyataan,

Khilmi Hani
NIM. 16610093

MOTO

“Jangan membandingkan prosesmu dengan yang lain, ingat bunga tidak selalu merekah bersama dalam satu waktu, bunga akan merekah jika waktunya sudah tepat, jangan pernah menyerah!”

PERSEMBAHAN

Penulis persembahkan skripsi ini kepada:

Kedua orang tua tercinta yang tidak kenal lelah memberikan dukungan baik fisik maupun psikis kepada penulis, tidak pernah terlewatkan memanjatkan doa kepada Allah SWT, dan berbagai pengorbanan yang tak ternilai. Serta kepada kakak penulis M. Uman yang selalu sabar memberikan support kepada penulis.

KATA PENGANTAR

Assalamu'alaikum Warahmatullahi Wabarakatuh

Segala puji bagi Allah Swt, yang telah melimpahkan rahmat, taufik serta hidayahnya kepada penulis, sehingga penulis dapat menyelesaikan skripsi yang berjudul "Algoritma Kriptografi dan Steganografi untuk Pengamanan Pesan ke dalam Citra" dengan baik, dan sebagai syarat untuk memperoleh gelar sarjana di bidang matematika di Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang. Sholawat dan salam tetap tercurahkan kepada nabi Muhammad Saw yang telah membimbing kita dari jalan yang sesat menuju jalan yang benar dan diridhoi.

Pada penulisan skripsi ini, penulis mendapat banyak sekali bimbingan dan arahan dari berbagai pihak. Untuk itu, penulis mengucapkan terima kasih yang amat besar dan penghargaan setinggi-tingginya kepada:

1. Prof. Dr. Abd. Haris, M.Ag, selaku rektor Universitas Islam Negeri Maulana Malik Ibrahim Malang.
2. Dr. Hj. Sri Harini, M.Si, selaku dekan Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang.
3. Dr. Usman Pagalay, M.Si, selaku ketua Jurusan Matematika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang.
4. Muhammad Khudzaifah, M.Si, selaku dosen pembimbing I yang telah memberikan arahan, nasihat, saran dan pengalaman yang berharga kepada penulis.

5. Juhari, M.Si, selaku dosen pembimbing II dan dosen wali yang telah memberikan nasihat, saran dan motivasi kepada penulis.
6. Segenap sivitas akademika Jurusan Matematika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang dan seluruh dosen yang telah memberikan ilmu dalam proses perkuliahan.
7. Kedua orang tua dan kakak tercintah yang selalu memberikan doa dan semangat kepada penulis.
8. Teman-teman mahasiswa Jurusan Matematika angkatan 2016 atas dukungannya untuk menggapai impian kepada penulis.
9. Semua pihak yang telah membantu dalam menyelesaikan skripsi ini baik moril maupun materil.

Semoga Allah SWT, melimpahkan rahmat dan karunia-Nya kepada kita semua. Penulis berharap agar skripsi bisa bermanfaat bagi penulis dan pembaca.

Wassalamu'alaikum Warahmatullahi Wabarakatuh.

Malang, 12 Agustus 2020

Penulis

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

ABSTRAK

Hani, Khilmi, 2020. **Algoritma Kriptografi dan Steganografi untuk Pengamanan Pesan ke dalam Citra**. Skripsi. Jurusan Matematika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) Muhammad Khudzaifah, M.Si, (II) Juhari M.Si.

Kata kunci: Kriptografi, RSA, Steganografi, MSB, LSB, Citra.

Keamanan sangat penting untuk menjaga pesan atau informasi yang bersifat pribadi dan rahasia. Maka dari itu pengamanan pesan sangat diperlukan agar pesan yang bersifat rahasia tersebut tidak mudah diakses oleh pihak lain yang tidak berkepentingan. Berdasarkan keterangan di atas, penelitian ini menghasilkan kombinasi antara kriptografi dan steganografi pada media citra. Kriptografi yang digunakan dalam penelitian ini adalah metode RSA, dan steganografi yang digunakan yaitu Metode LSB dan MSB, di mana pesan yang telah terenkrip akan disembunyikan ke dalam citra dengan steganografi. Penelitian ini menghasilkan *stego image* metode MSB dengan nilai eror 0,00119 dan PSNR 77,3737 db. Menurut tabel kualitas citra, nilai PSNR pada citra tersebut baik, karena tidak banyak *noise* yang bisa menimbulkan kecurigaan pihak lain. Kemudian, *stego image* metode LSB memiliki nilai eror 1,1394 dan nilai PSNR 97,3638 db. Artinya kualitas *cover image* sangat baik karena tidak ada *noise*, sehingga kerahasiaan pesan terjaga.

ABSTRACT

Hani, Khilmi, 2020. **Cryptography and Steganography Algorithms for Securing Messages into Image**. Thesis. Department of Mathematics, Faculty of Science and Technology, Islamic State University of Maulana Malik Ibrahim Malang. Advisors: (I) Muhammad Khudzaifah, M.Si, (II) Juhari M.Si.

Keywords: Confidential, Cryptography, RSA, Steganography, MSB, LSB, Encrypted.

Security code is the most important way to keep messages or information privately and confidentially. Therefore, the security of messages is very necessary to make confidential messages doleful to be accessed by the others. The purpose of this research is to produce a combination of cryptography and steganography on image media. The cryptography used in this research is the RSA method, and the steganography used in this research is the LSB and MSB methods. Where encrypted messages will be hidden e n into the image by steganography.

This study produces a stego image of the MSB method with an error value of 43.302 and PSNR of 31.765 dB. Based on the quality table image, the PSNR value on the image show isn't a good result, because there Is a lot of noise which can be caused by the suspicions of others. Then, the LSB method of stego image obtained an error value of 0,0002 and the PSNR value of 83,502 dB. These results showing good image quality because no faults were found, so the message was kept confidential.

ملخص

هاني ، خيلمي ، ٢٠٢٠. التشفيرو الاحتوال الخوارزميات الرسانك في الصور.بحث جامعي. قسم الرياضيات ، كلية العلوم والتكنولوجيا ، جامعة مولانا مالك إبراهيم الإسلامية الحكومية في مالانغ. المشرف (١) : محمد خضيفة الماجستير ، (٢) ماجستير جهاري الماجستير.

الكلمات المفتاحيه :سري, التشفير ، RSA، الاحتوال,MSB, LSB، صورة.

الأمن ضروري جدًا لحفظ على الرسائل أو المعلومات الخاصة والسرية. لذلك ، يعد أمن الرسائل أمرًا ضروريًا للغاية بحيث لا يمكن الوصول إلى الرسائل السرية بسهولة من قبل الأطراف الأخرى غير المهتمة. استنادًا إلى المعلومات الواردة أعلاه ، أنتجت هذه الدراسة مزيجًا من التشفير وإخفاء المعلومات على وسائط الصور. التشفير المستخدم في هذه الدراسة هو طريقة RSA ، الاحتوال المستخدم هو أساليب LSB و MSB ، حيث سيتم إخفاء الرسائل المشفرة في الصورة عن تعسر المعلومات.

أنتجت هذه الدراسة صورة ستيجو لطريقة MSB بقيمة خطأ٠,٠٠١٩ و PSNR٧٧,٣٧٣٧ ديسيبل. وفقًا لجدول جودة الصورة ، فإن قيمة PSNR على الصورة ليست جيدة ، لأن هناك الكثير من الضوضاء التي يمكن أن تسبب الشك في الأطراف الأخرى. بعد ذلك ، حصلت صورة stego لأسلوب LSB على قيمة خطأ ١,١٣٩٤ وقيمة PSNR بقيمة ٩٧,٥٦٣٨ ديسيبل. تم الحصول على جودة صورة جيدة ، لأنه لا يوجد ضوضاء حتى يتم الحفاظ على سرية الرسالة.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi terutama bidang informasi pada zaman sekarang menjadi kebutuhan masyarakat umum, dengan berkembangnya teknologi informasi tersebut ada banyak keuntungan yang didapat dalam berbagai bidang, baik pendidikan, bisnis maupun bidang lainnya. Selain itu, informasi apapun dapat dengan mudah diperoleh dan tersebar dengan cepat melalui media internet (Ibrahim, 2017). Internet adalah sekumpulan jaringan komputer yang terhubung dengan berbagai situs pemerintah, grup, maupun perorangan dengan tujuan memberikan informasi kepada pengguna (Handoyo dkk, 2018). Selain banyak keuntungan, ada banyak juga kerugiannya, seperti pencurian dan *hacking* beberapa situs, penyadapan data penting akun perseorangan, sehingga perlu adanya pengamanan data dan informasi.

Keamanan data sangat diperlukan bagi perusahaan, institusi, organisasi, maupun perseorangan yang memiliki informasi rahasia. Penggunaan keamanan tersebut, ditujukan agar informasi tidak dapat dicuri oleh orang lain. Ada beberapa metode yang digunakan untuk mengamankan pesan dari zaman dahulu, seperti menyembunyikan pesan ke dalam media lain agar orang lain terkecoh dengan tampilannya, ilmu ini disebut *steganography*. Selain pesan yang disembunyikan ke dalam media, ada juga ilmu pengamanan dengan menyandikan atau mengubah makna pesan tidak terbaca dengan menggunakan berbagai perhitungan, ilmu itu disebut *cryptography* (Ibrahim, 2017).

Kriptografi adalah ilmu dan seni pengamanan pesan di mana pesan disandikan dengan menggunakan berbagai perhitungan pesan sehingga pesan tersebut tidak dimengerti maknanya oleh orang lain. Kriptografi pertama kali ditemukan oleh bangsa Mesir pada tahun 3000 SM. Kriptografi berasal dari bahasa yunani yaitu *kriptos* dan *graphia* yang artinya "tulisan yang tersembunyi". Beberapa hal yang membedakan antar kriptografi yaitu metode persandiannya. Semakin rumit metode yang digunakan maka pengamanan pesan akan lebih susah dipecahkan. Penelitian ini menggunakan algoritma RSA dalam mengamankan pesan.

Algoritma RSA merupakan algoritma yang menggunakan kunci asimetri, di mana kunci enkripsi berbeda dengan kunci dekripsi. Algoritma ini di ambil dari nama penemunya yaitu Rivest, Shamir dan Adleman dari MIT. Algoritma ini mempuyai kerumitan yang cukup tinggi, karena sulitnya mencari hasil faktor-faktor prima dari bilangan yang besar. Hasil pemfaktoran tersebut, digunakan untuk mencari kunci privat. Ada tiga proses inti dalam algoritma RSA yaitu pembangkitan kunci, proses enkripsi dan proses dekripsi.

Dari penelitian sebelumnya yang dilakukan oleh (Saputra.2016) ditemukan bahwa metode RSA memiliki perhitungan yang rumit dalam pembentukan kuncinya. Kemudian dari penelitian (Benny.2017) dapat disimpulkan bahwa sistem algoritma RSA mudah dalam proses enkripsi, tetapi ketika pesan sudah terenkrip maka akan sulit dibobol. Keamanan pesan dengan kriptografi RSA yang diketahui rumit, ternyata masih ada celah terbobolnya suatu pesan.

Sehingga penelitian ini menambahkan teknik steganografi agar pesan lebih terjaga kerahasiaannya.

Steganografi merupakan teknik penyembunyian pesan pada media lain, misal teknik menyembunyikan pesan ke dalam media gambar agar tidak ada yang curiga bahwa gambar tersebut menyimpan pesan rahasia. Metode steganografi yang digunakan dalam penelitian ini adalah metode LSB (*Least Significant Bit*) dan MSB (*Most Significant Bit*). Metode LSB adalah metode yang bekerja dengan merubah bit terakhir dari citra, sedangkan metode MSB adalah metode yang bekerja pada bit pertama dari citra.

Metode dalam penelitian ini diharapkan dapat menjaga kerahasiaan isi pesan dari orang yang tidak berkepentingan dengan cara mengkombinasikan teknik kriptografi dan steganografi.

Berdasarkan paparan di atas, penelitian ini mengambil judul "Algoritma Kriptografi dan Steganografi untuk Pengamanan Pesan ke dalam Citra".

Al-Qur'an menganjurkan manusia untuk menjaga rahasia yang diketahui dan harus menyimpan rahasia tersebut dengan baik, karena dalam surat An-Nisa ayat 58

إِنَّ اللهَ يَأْ مُرُكُمْ أَنْ تُؤَدُّواالْآمَانَاتِ إِلَى أَهْلِهَا وَإِذَاحَكَمْتُمْ بَيْنَ النَّاسِ أَنْ تَحْكُمُوْابِالْعَدْل،إِنَّ اللهَ نِعِمَّايَعِظُكُمْ بِهِ،إِنَّ اللهَ كَنَاسَمِيْعًا بَصِيْرًا

artinya :

"sesungguhnya Allah menyuruh kamu menyampaikan amanat kepada yang berhak menerimanya, dan (menyuruh kamu) apabila menetapkan hukum di antara manusia supaya kamu menetapkan dengan adil. Sesungguhnya Allah memberi pengajaran yang sebaik-baiknya kepadamu. Sesungguhnya Allah adalah Maha Mendengar lagi Maha Melihat"(QS. An-Nisa;/4:58).

Dari ayat tersebut kita dianjurkan untuk menjaga dan menyampaikan pesan sesuai dengan yang ada, tanpa mengurangi atau menambahi isi pesan dan memastikan pesan tersebut tersampaikan kepada orang yang tepat. Dengan menerapkan teknik kriptografi RSA diharapkan pesan yang akan disampaikan terjaga kerahasiaannya, dan lebih meningkatkan kerahasiaan pesan tersebut, diterapkan pula teknik steganografi metode LSB dan MSB pada citra.

1.2 Rumusan Masalah

Berdasarkan paparan latar belakang di atas, dapat diketahui rumusan masalahnya yaitu:

1. Bagaimana proses enkripsi kode pesan dengan kriptografi RSA dan menyisipkan kode *ciphertext* ke dalam citra dengan steganografi MSB dan LSB serta menghitung kualitas gambar yang digunakan?
2 Bagaimana proses ekstraksi *stego image* dengan steganografi MSB dan LSB serta proses dekripsi kode *ciphertext* dengan kriptografi RSA?

1.3 Tujuan Penelitian

Tujuan yang didapat dengan adanya rumusan masalah di atas adalah:

1. Untuk mengetahui proses enkripsi kode pesan dengan kriptografi RSA dan proses menyisipkan kode *ciphertext* ke dalam citra dengan steganografi MSB dan LSB serta mengetahui kualitas gambar yang digunakan.
2. Untuk mengetahui proses ekstraksi *stego image* dengan steganografi MSB dan LSB serta proses dekripsi kode *ciphertext* dengan kriptografi RSA.

1.4 Manfaat Penelitian

Penelitian ini diharapkan bermanfaat dalam beberapa hal, sebagai berikut:

1. Pesan yang telah terenkrip akan terjaga kerahasiaannya dan keasliannya.
2. Keberadaan pesan akan tersamarkan oleh bentuk *stego image*, sehingga akan terjaga kerahasiaannya.

1.5 Batasan Masalah

Batasan masalah pada penelitian ini yaitu:

1. Pesan yang digunakan berbentuk teks.
2. Citra yang digunakan adalah citra *grayscale* dengan format .png.
3. Metode kriptografi yang digunakan adalah metode RSA, dan metode steganografinya adalah metode MSB dan LSB.

1.6 Metode Penelitian

Langkah-langkah yang digunakan dalam penelitian ini sebagai berikut:

1. Merumuskan masalah.
2. Mencari data dan studi literatur.
3. Menerapkan ke dalam python seperti:
    a. Proses enkripsi dan dekripsi dengan kriptografi RSA.
    b. Proses penyisipan pesan ke dalam gambar dengan steganografi LSB dan MSB.
    c. Proses ekstraksi dengan steganografi LSB dan MSB.

1.7 Sistematika Penulisan

Penulisan penelitian ini dibagi menjadi empat bab dan setiap bab terdiri dari beberapa subbab. Sistematika tersebut dimaksudkan agar penulisan lebih terarah dan mudah dipahami. Adapun sistematika tersebut yaitu:

BAB I PENDAHULUAN

Pendahuluan meliputi: latar belakang, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah, metode penelitian, dan sistematika penulisan.

BAB II KAJIAN PUSTAKA

Kajian pustaka berisi mengenai teori-teori yang berkaitan dengan permasalahan. Pada penelitian ini teori yang digunakan meliputi: kriptografi dan steganografi. Pada kriptografi terdapat subbab yaitu algoritma RSA pada steganografi dan steganografi terdapat subbab yaitu metode LSB dan metode MSB.

BAB III PEMBAHASAN

Pembahasan berisi mengenai penyelesaian terhadap permasalahan penyandian pesan menggunakan metode RSA, kemudian penyelesaian penyisipan kode *ciphertext* ke dalam citra dengan metode MSB, dan penyelesaian penyisipan kode *ciphertext* ke dalam citra dengan metode LSB serta proses dekripsi metode RSA.

BAB IV PENUTUP

Penutup berisi kesimpulan dari hasil pembahasan dan saran untuk penelitian selanjutnya.

BAB II

KAJIAN PUSTAKA

2.1 Kriptografi

Kriptografi berasal dari bahasa Yunani, menurut bahasa dibagi menjadi dua yaitu *Crypto* dan *graphia*. *Crypto* berarti *secret* (rahasia) dan *graphia* yaiu *writing* (tulisan). Jadi secara estimologi kriptografi dapat diartikan sebagai penulisan pesan rahasia. Secara terminologi kriptografi adalah ilmu dan seni yang digunakan untuk menjaga keamanan pesan, agar pesan tetap terjaga kerahasiaannya (Munir,2019).

2.1.1 Istilah dalam Kriptografi

Kriptografi mempunyai beberapa istilah seperti berikut:

1. ***Plaintext*** adalah pesan asli atau pesan yang maknanya masih jelas.
2. ***Ciphertext*** adalah hasil dari enkripsi, bisa disebut dengan teks tersandi.
3. ***Enkripsi*** adalah algoritma untuk mentransformasikan *plaintext* menjadi *ciphertext*.
4. ***Dekripsi*** adalah algoritma untuk memulihkan kembali *ciphertext* menjadi *plaintext* (Sadikin,2012).
5. **Kunci (*Key*)** adalah parameter yang digunakan untuk transformasi enkripsi dan dekripsi. Kunci biasanya berupa *string* atau deretan bilangan.
6. **Sistem kriptografi (*Cryptosystem*)** adalah kumpulan yang terdiri dari algoritma kriptografi, semua *plaintext*, *ciphertext* dan kunci.
7. **Kriptanalisis** yaitu bidang yang berlawanan dengan kriptografi. Kriptanalisis adalahseni dan ilmu untuk memecahkan *ciphertext* menjadi *plaintext* tanpa mengetahui kunci yang digunakan. Pelakunya disebut

kriptanalis. Jika seorang kriptografer mentranformasikan *plaintext* menjadi *ciphertext* dengan suatu algoritma dan kunci, maka sebaliknya seorang kriptanalis berusaha untuk memecahkan *ciphertext* untuk menemukan *plaintext* atau kunci. (Munir,2019).

8. **Kriptologi *(Cryptology)*** adalah studi mengenai kriptografi dan kriptanalis, baik kriptografi dan kriptanalis keduanya saling berkaitan.

2.1.2 Macam-Macam Algoritma Kriptografi

Berdasarkan kunci yang digunakan, algoritma kriptografi dibagi menjadi dua bagian yaitu :

1. Algoritma Simetri

Algoritma Simetri bisa disebut dengan algoritma klasik, karena memakai kunci yang sama antara enkripsi dan dekripsi. Keamanan dari algoritma ini tergantung kerumitan dari kuncinya, jika kunci dari algoritma ini mudah ditebak maka kerahasiaan pesan terancam diketahui oleh orang lain. Algoritma yang memakai kunci simetri diantaranya: DES, AES, RC2, RC4, RC5, RC6, dan lain sebagainya (Ariyus,2006).

Algoritma kunci simetri memiliki kelebihan dan kekurangan yaitu :

Kelebihan :

1. Waktu yang dibutuhkan dalam proses enkripsi dan dekripsi relatif cepat, karena efisiensi yang terjadi dalam proses pembangkitan kunci.
2. Ukuran kunci simetri relatif pendek.
3. Kunci simetri digunakan pada sistem secara *real time* seperti saluran telepon digital, karena cepatnya proses enkripsi dan dekripsi.

Kekurangannya :

1. Terdapat banyak kunci yang digunakan dalam penggunaanya. Karena setiap pasang pengguna membutuhkan kunci yang berbeda, sehinga sulit dalam hal manajemen kunci, karena banyak kunci yang harus diingat.
2. Adanya kesepakatan jalur untuk pendistribusian khusus untuk kunci, karena tidak mudah dalam menentukan jalur yang aman, seperti pengiriman melalui jalur tertentu yang telah disepakati ataupun bisa bertemu secara langsung.
3. Kunci harus sering diubah, mungkin disetiap sesi komunikasi.

2. Algoritma Asimetri

Algoritma ini sering disebut dengan algoritma kunci publik, dengan arti kunci yang digunakan untuk enkripsi dan dekripsi berbeda, dalam algoritma asimetri kunci dibagi menjadi dua bagian yaitu :

1. Kunci umum (*public key):* kunci yang di publik atau semua orang boleh mengetahuinya.
2. Kunci Privat (*Privat Key)*: kunci yang dirahasiakan yang hanya diketahui oleh pembuat kunci itu sendiri.

Algoritma yang memakai kunci asimetri diantaranya: DSA (*digital signatur algorithm*), RSA (Rivest, Shamir, Adleman), DH (Diffie-Hellman) (Ariyus,2008).

Algoritma kunci asimetri juga mempunyai kelebihan dan kekurangan antara lain:

Kelebihan:

1. Keamanan dalam pendistribusian kunci dapat diatasi karena tidak ada jalur khusus dalam pendistribusiannya.
2. Manajemen kunci dapat diatasi, karena hanya kunci privat saja yang perlu dijaga kerahasiaannya.
3. Dapat digunakan untuk mengamankan kunci simetri.

Kekurangan:

1. Proses enkripsi dan dekripsi lebih lambat dibandingkan dengan sistem simetri, karena enkripsi dan dekripsi menggunakan bilangan yang besar dan melibatkan operasi perpangkatan yang besar.
2. Untuk tingkat keamanan, algoritma ini menggunakan kunci yang relatif besar.

2.1.3 Tujuan Kriptografi

Ilmu kriptografi mempunyai empat tujuan mendasar dan merupakan aspek keamanan informasi yaitu :

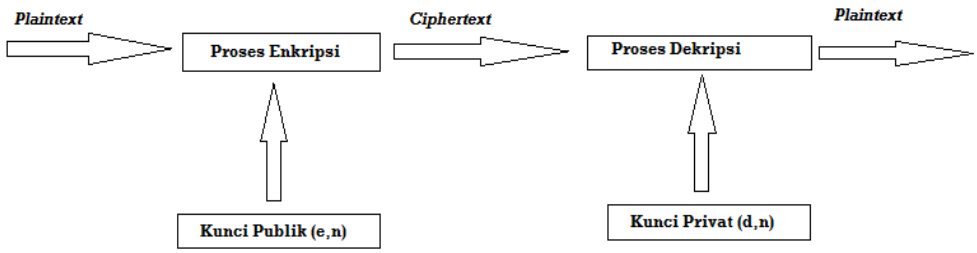
1. Kerahasiaan: sebuah layanan untuk melindungi isi informasi dari siapapun yang tidak memiliki kunci rahasia untuk membuka informasi.
2. *Integrity*: keaslian pesan yang dikirim melalui media sosial dapat dipastikan bahwa pesan tidak di modifikasi oleh orang yang tidak berhak.
3. *Authentication*: berhubungan dengan identifikasi atau pengenalan agar penerima dapat memastikan keaslian dan isi pesan, dan pesan tersebut datang dari orang yang dimintai informasi.

4. *Non-repudiation*: usaha untuk mencegah terjadinya penyangkalan terhadap pengiriman informasi oleh orang yang mengirimkan.

2.2 Algoritma RSA

RSA diambil dari penemunya yaitu Ron Rivest, Adi Shamir dan Leonard Adleman dari MIT (*Massachussets Institute of Technology*) pada tahun 1976. Algoritma ini dipatenkan pada tahun 1983 di Amerika Serikat, algoritma RSA merupakan algoritma kriptografi kunci publik yang populer karena kehandalannya dalam enkripsi. Algoritma ini melakukan pemfaktoran bilangan yang sangat besar menjadi faktor-faktor prima, untuk memperoleh kunci privat. Oleh karena itu, algoritma RSA dianggap paling aman. (Munir,2004).

Algoritma RSA termasuk dalam kategori algoritma kunci asimetris, dimana kunci yang digunakan untuk mengenkripsi berbeda dengan yang digunakan untuk mendekripsi. Pada gambar 2.1 dijelaskan diagram proses enkripsi dan dekripsi algoritma asimetri.



Gambar 2.1 Diagram Proses Enkripsi dan Dekripsi Algoritma Asimetri

Dari Gambar 2.1 tersebut, diketahui proses kriptografi RSA, dimana *plaintext* dienkripsi dengan kunci publik yang menghasilkan *ciphertext*, kemudian dari *ciphertext* akan didekripsi dengan kunci privat dan menghasilkan *plaintext* atau pesan yang semula.

2.2.1 Besaran Algoritma RSA

Algoritma RSA memiliki besaran-besaran sebagai berikut :

Besaran	**Sifat**
p dan q (bilangan prima)	Rahasia
$n = p\,x\,q$	Tidak Rahasia
$\varphi(n) = (p-1)(q-1)$	Rahasia
e (Kunci Enkripsi)	Tidak Rahasia
d (Kunci Dekripsi)	Rahasia
m (Kode *plaintext*)	Rahasia
c (Kode *ciphertext*)	Tidak Rahasia

Tabel 2.1 Tabel besaran algoritma RSA

2.2.2 Perumusan Algoritma RSA

Algoritma RSA didasarkan pada teorema Euler yang menyatakan bahwa

$$a^{\phi(n)} \equiv 1(mod\ n) \qquad (2.1)$$

yang harus memenuhi syarat:

1. a harus relatif prima dengan n
2. $\phi(n) = n\left(1 - 1/p_1\right)\left(1 - 1/p_2\right) \ldots \left(1 - 1/p_r\right)$, yang dalam hal ini $p_1, p_2, \ldots . p_r$ adalah faktor prima dari n.

$\phi(n)$ adalah fungsi *totient Euler* yang menentukan berapa banyak dari bilangan-bilangan 1,2,3, … , n yang relatif prima dengan n.

Berdasarkan sifat $a^k \equiv b^k\ (mod\ n)$ untuk k bilangan bulat lebih dari 1, maka persamaan (2.1) dapat ditulis menjadi

$$a^{k\phi(n)} \equiv 1^k (mod\ n),$$

atau

$$a^{k\phi(n)} \equiv 1(mod\ n). \qquad (2.2)$$

Jika a diganti dengan m maka persamaan (2.2) menjadi

$$m^{k\phi(n)} \equiv 1(mod\ n) \qquad (2.3)$$

Berdasarkan sifat $ac \equiv bc(mod\ n)$, maka persamaan (2.3), jika dikali dengan m menjadi :

$$m^{k\phi(n)+1} \equiv m(mod\ n) \qquad (2.4)$$

dalam hal ini m relatif prima terhadap n.

Misalkan e dan d dipilih sedemikian sehingga

$$e \cdot d \equiv 1(mod\ \phi(n)) \qquad (2.5)$$

atau

$$e \cdot d \equiv k\phi(n) + 1 \qquad (2.6)$$

Substitusikan (2.6) ke (2.4) menjadi:

$$m^{e \cdot d} \equiv m(mod\ n) \qquad (2.7)$$

Persamaan (2.7) dapat ditulis menjadi :

$$(m^e)^d \equiv m(mod\ n) \qquad (2.8)$$

yang artinya, perpangkatan *m* dengan *e* diikuti dengan perpangkatan dengan *d* menghasilkan *m* semula.

Berdasarkan persamaan (2.8), maka enkripsi dan dekripsi dirumuskan sebagai berikut:

$$E_e(m) = c = m^e\ mod\ n \qquad (2.9)$$

$$D_d(m) = c = m^d\ mod\ n \qquad (2.10)$$

Karena $e.d = d.e$, maka enkripsi diikuti dengan dekripsi ekivalen dengan dekripsi diikuti enkripsi:

$$D_d\big(E_e(m)\big) = E_e(D_d(m) \equiv m^d\ mod\ n \qquad (2.11)$$

Oleh karena $m^d mod\ n \equiv (m + jn)^d mod\ n$ untuk sebarang bilangan bulat j, maka setiap *plaintext* $m, m + n, m + 2n,$ ...menghasilkan *ciphertext* yang sama (Munir,2019).

2.2.3 Algoritma Pembangkit Pasangan Kunci

Teknik mendapatkan sepasang kunci algoritma RSA yang terdiri dari kunci publik dan kunci privat dapat dilakukan dengan cara sebagai berikut :

1. Pilih p dan q yaitu sebarang dua bilangan prima acak dengan $p \neq q$.
2. Hitung $n = p \times q$
3. Hitung $\phi(n) = (p - 1) \times (q - 1)$.
4. Pilih satu bilangan bulat e untuk kunci publik, dimana e relatif prima terhadap $\phi(n)$.
5. Bangkitkan kunci privat dengan menggunakan persamaan

$$d \equiv\ e^{-1}\ mod\ (\phi(n))$$

Hasil dari algoritma di atas adalah :

1. Kunci publik(e, n)
2. Kunci privat (d, n)

2.3 Steganografi

Kata steganografi berasal dari bahasa yunani yaitu "*Steganos*" yang berarti tersembunyi dan "*Graphein*" yang berarti tulisan (Ibrahim,2017). Secara estimologi, steganografi dapat diartikan tulisan tersembunyi. Secara terminologi steganografi adalah ilmu dan seni menyembunyikan pesan rahasia sedemikian sehingga keberadaan pesan tidak terdeteksi oleh indera manusia (Munir,2004). Steganografi bisa disebut sebagai kelanjutan dari kriptografi, dimana pesan dari

kriptografi disandikan menjadi *ciphertext*, sedangkan steganografi menyembunyikan ke dalam media lain, sehingga pesan rahasia tersebut tersamarkan.

Dalam steganografi terdapat dua proses yang harus dilewati, seperti proses *encode* yaitu proses penyisipan kode pesan ke dalam citra, hasil dari penyisipan disebut *stego image* dan proses *decode* yaitu proses pemisahan kode pesan dengan *cover image*.

2.3.1 Sejarah Steganografi

Sejarah dalam steganografi terbagi menjadi empat bagian, antara lain :

1. **Steganografi kuno (*ancient steganography)***

   Steganografi kuno ditulis oleh Herodatus dan cara yang digunakan untuk menyembunyikan pesan adalah dengan dipilihnya beberapa budak, kemudian kepala budak di botaki, dan ditulisi pesan dengan di tato, kemudian rambut dibiarkan tumbuh, lalu budak dikirim, sampai di tempat tujuan, kepala budak di gunduli kembali, agar pesan dapat dibaca. Selain itu, ada yang menuliskan pesan di tablet kayu, kemudian melapisinya dengan lilin yang terbuat dari lebah.

2. **Steganografi zaman renaisans (*renaissance steganography)***

   Tahun 1499, Johannes Trithemius menulis buku tentang steganografi berbasis karakter. Kemudian Giovanni Battista Porta menggambarkan cara

menyembunyikan pesan ke dalam telur rebus dengan tinta khusus, dengan prinsip tinta tersebut menembus kulit telur yang berpori tanpa meninggalkan jejak yang terlihat. Tulisan dari tinta tersebut akan membekas pada permukaan isi telur yang telah direbus sebelumnya. Pesan dapat dibaca dengan membuang kulit telur.

3. Steganografi zaman perang dunia

Selama perang dunia II, agen spionanse menggunakan steganografi dengan melalui tinta tak nampak (*invisible ink*). Tinta tersebut terbuat dari campuran susu, cuka, sari buah, dan urine. Cara membaca pesan tersebut dengan memanaskan kertas di atas pemanas (api, lampu), sehingga tulisan tersebut akan nampak.

4. Steganografi modern

Pada tahun 1983, steganografi modern diperkenalkan oleh Simmon Wiseman, terdapat beberapa langkah untuk menjaga pesan rahasia tersebut seperti :

a. Mengenkripsinya terlebih dahulu, kemudian
b. Menyembunyikan pesan ke dalam tulisan lain, dan
c. Menyembunyikan pesan dengan media digital seperti citra, audio ataupun video (Munir,2004).

2.3.2 Metode Steganografi

Dalam ranah operasinya, metode steganografi dibagi menjadi dua bagian yaitu ranah spasial dan ranah transformasi seperti berikut:

2.3.2.1 Ranah Spasial (Waktu)

Teknik steganografi ini yaitu dengan memodifikasi langsung nilai *byte* dari *cover object.* Pada sebuah gambar, nilai *byte* merepresentasikan intensitas atau warna pixel sedangkan *amplitude* pada audio. Contoh metode yang termasuk dalam ranah spasial adalah metode LSB (*Least Significant Bit)* dan MSB (*Most Significant Bit).*

2.3.2.2 Ranah Transformasi

Metode steganografi ranah transformasi yaitu dengan memodifikasi hasil transformasi sinyal ke ranah transform, seperti transformasi dari ranah spasial ke ranah frekuensi, contohnya metode *spread spectrum*. Adapun transformasi dari ranah spasial ke ranah lain, antara lain metode *Discrete Cosine Transfom* (DCT), *Fast Fourier Transform* (FFT), *Discrete Wavelet Transform* (DWT), dan lain sebagainya(Munir,2019).

2.3.3 Kriteria Steganografi yang Baik

Pesan yang disembunyikan tidak hanya berbentuk teks, tetapi bisa berbentuk gambar, audio dan video. Selain itu, media yang digunakan bisa berupa gambar, teks, audio atau video.Penyembunyian pesan ke dalam citra digital akan mengubah kualitas dari citra tersebut, maka ada kriteria yang harus diperhatikan dalam penyembunyian pesan sebagai berikut :

1. ***Imperceptibility***

   Keberadaan pesan rahasia tidak dapat dilihat secara inderawi. Misalnya jika cover berupa gambar maka penyisipan pesan menghasilkan *stego image* yang secara visual susah dibedakan dengan *cover image*nya. Jika *cover object*

berupa audio seperti mp3, wav, dan sebagainya, maka indera telinga tidak dapat mendeteksi pesan yang tersembunyi dalam *stego-audio*.

2. *Fidelity*

Kualitas *cover* yang digunakan sebagai media tidak jauh berubah setelah penyisipan pesan rahasia. Misalnya *cover* berupa gambar maka pesan tidak boleh membuat *cover-image* terdegradasi.

3. *Recovery*

Pesan yang disembunyikan harus bisa diungkap kembali, karena tujuan steganografi adalah menyembunyikan data, maka kapanpun pesan rahasia dapat dikembalikan untuk digunakan lebih lanjut.

4. *Payload*

Pesan yang disembunyikan kedalam cover kalau bisa sebanyak mungkin namun tidak mengurangi *fidelity*.(Munir,2019)

2.4 Metode LSB

Teknik untuk menyembunyikan pesan pada media digital cukup beragam, misalnya menyembunyikan pesan dilakukan dengan cara mengganti bit pesan di dalam citra dengan bit kode rahasia, salah satunya dengan menggunakan metode LSB. Metode LSB yaitu teknik penyisipan dengan mengganti bit paling rendah dengan bit kode pesan. Ada dua teknik penyisipan yang digunakan dalam metode LSB, yaitu penyisipan pesan secara sekuensial dan penyisipan pesan secara acak. Penyisipan secara sekuensial dilakukan secara berurutan dan penyisipan secara acak dilakukan dengan menggunakan kunci. Dalam penelitian ini proses penyisipan ke dalam citra menggunakan penyisipan sekuensial.

Metode LSB adalah proses mengganti bit terkahir dari *cover image* dengan dengan bit kode *ciphertext*. Contohnya, terdapat bilangan biner dari suatu gambar sebagai berikut:

00100111 11101001 11001000

00100111 11001000 11101001

11001000 00100111 11101001

dan representasi biner huruf A adalah 01000001, dengan menyisipkan biner dari huruf A secara sekuensial maka didapat hasil berikut:

00100111 11101001 11001000

00100110 11001000 11101000

11001000 00100111 11101001

Kapasitas *cover image* untuk menampung pesan rahasia untuk metode LSB yaitu $\frac{1}{8}$ dari keseluruhan. Kelebihan metode LSB yaitu ukuran citra untuk *cover image* tidak jauh berbeda dengan ukuran *stego image* dan kekurangan metode LSB yaitu terbatasnya kapasitas utuk penyisipan pesan ke dalam *cover image*.

2.5 Metode MSB

Metode MSB (*Most Significant Bit*) adalah kebalikan dari metode LSB. MSB bisa disebut dengan Urutan Terbesar Bit (*High-Order Bit*) yang merupakan bit terbesar dalam bilangan biner. Letak bit MSB berada di paling kiri. Misalnya dalam sebuah *byte* **1**1011000, maka bit terbesarnya adalah "1" (Jatmoko,

dkk.2018). Cara keja metode MSB adalah dengan mengganti bit awal dari *pixel* dengan bit pesan rahasia. Untuk melakukan penyisipan pesan rahasia ke dalam citra, maka citra dan pesan diubah terlebih dahulu menjadi bilangan biner. Perubahan dari citra menjadi citra biner bertujuan untuk mendapatkan satu nilai pada satu *pixel* citra di mana satu *pixel* citra akan digantikan dengan satu bit dari delapan bit karakter pesan yang akan disisipkan, teknik tersebut disebut dengan teknik *Dynamic Cell Spreading* (DCS) (Wahyuni,dkk 2017). Adapun contoh algoritma penyisipan metode MSB adalah sebagai berikut :

a. Misalkan bit yang akan disisipkan adalah "1" , dan data yang disisipi adalah 01100001 Maka :

   Data : 01100001

   255 : 11111111 di AND kan

   Hasil 1 : 01100001

b. Kemudian nilai 'Hasil 1' di OR kan dengan bit yang akan disisipkan maka :

   Hasil 1 : 01100001

   Bit : 1 di OR kan menghasilkan

   Data baru : 11100001

   (Manalu,2013).

2.6 Teknik Ekstraksi Pesan Metode LSB

Pesan yang disembunyikan di dalam citra dapat dibaca kembali dengan proses ekstraksi. Proses ekstraksi dilakukan dengan cara membaca byte-byte di dalam citra, kemudian mengambil bit ke-8 dan merangkainya menjadi bit-bit

pesan yang telah disisipkan. Misalkan byte-byte dari *stego-image* adalah seperti berikut :

00100111 11101001 11001000

00100110 11001000 11101000

11001000 00100111 11101001

Dengan mengambil setiap bit ke-8 dari setiap *byte*, maka diperoleh kembali bit-bit pesan rahasia yang disisipkan yaitu 01000001, jika diubah menjadi bilangan desimal menghasilkan 65 yang merepresentasikan huruf "A".

Jika bit-bit pesan yang disisipkan ke dalam *pixel* yang dipilih secara acak, maka untuk mengetahui posisi *pixel* tersebut dapat diketahui dari bilangan acak yang dibangkitkan PRNG (*pseudo random-number-generator*). Jika kunci yang dibangkitkan pada waktu ekstraksi sama dengan kunci pada waktu penyisispan maka bilangan acak yang dibangkitkan sama. Maka bit-bit pesan rahasia yang acak bisa dikumpulkan kembali. (Munir,2019)

2.7 Teknik Ekstraksi Metode MSB

Pesan yang telah disembunyikan ke dalam citra akan di ekstraksi dengan cara di bawah ini:

00001010 AND $0 = 0$ maka kode yang diperoleh $0 \times 2^7 = 0$

11100001 AND $1 = 1$ maka kode yang diperoleh $1 \times 2^6 = 64$

10110110 AND $1 = 1$ maka kode yang diperoleh $1 \times 2^5 = 32$

01100101 AND $0 = 0$ maka kode yang diperoleh $0 \times 2^4 = 0$

00101000 AND $0 = 0$ maka kode yang diperoleh $0 \times 2^3 = 0$

01000011 AND $0 = 0$ maka kode yang diperoleh $0 \times 2^2 = 0$

01001000 AND $0 = 0$ maka kode yang diperoleh $0 \times 2^1 = 0$

11100100 AND $1 = 1$ maka kode yang diperoleh $1 \times 2^0 = 1$

Jumlahkan untuk memperoleh kode *plaintext* dan diperoleh $= 97$

Proses di atas merupakan ekstraksi metode MSB dengan mengambil setiap bit pertama dari citra biner. Bit pertama yang diambil, jika dikumpulkan membentuk bilangan biner "01100001" yang mempunyai nilai desimal "97" melambangkan huruf "a". (Manalu,2013).

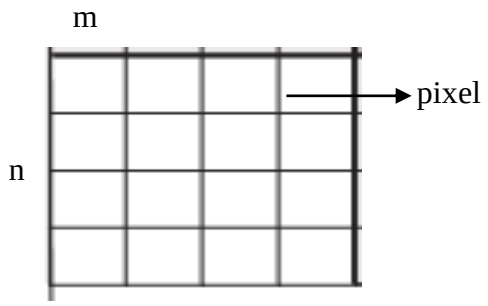
2.8 Citra Digital

Secara harfiah citra (*image*) adalah kombinasi antara titik, garis, bidang dan warna untuk menghasilkan imitasi dari suatu objek. Terdapat dua jenis citra yaitu:

a. Citra diam (*Still Images*) adalah citra tunggal yang tidak bergerak. Contohnya adalah foto.
b. Citra bergerak (*Moving Images*) adalah rangkaian citra diam yang ditampilkan secara beruntun, sehingga memberi kesan bahwa citra tersebut bergerak. Contohnya adalah video. (Munir,2006).

Sedangkan citra digital adalah matriks dua dimensi yang terdiri dari baris m dan kolom n, dimana setiap pasangan indeks baris dan kolom menyatakan suatu titik citra. Nilai matriksnya menjelaskan tingkat kecerahan, dan disebut pixel atau elemen terkecil dari sebuah citra (Kusmanto,2011).

Contoh representasi matriks digambarkan pada gambar 2.2 sebagai berikut



Gambar 2.2 Representasi Matriks Citra

Citra digital yang digunakan dalam penelitian ini yaitu citra hitam putih (*grayscale*) yang setiap *pixel*nya memiliki gradasi warna dari hitam sampai putih. Setiap *pixel* terdiri dari 8 bit. Citra *grayscale* cocok digunakan untuk proses steganografi, karena warna yang tidak terlalu mengalami perubahan dan sulit untuk membedakannya dengan indera penglihatan.

Format citra yang digunakan dalam penelitian ini adalah format png (*Portable Network Graphics*). Format png termasuk *lossless compression* yang merupakan kompresi yang memungkinkan data asli dapat disusun kembali ke bentuk semula. Metode lossless menghasilkan data yang identik dengan data aslinya, maka dari itu dalam penelitian ini menggunakan citra *grayscale* dengan format png, agar data dapat disembunyikan dan dapat diungkap kembali.

2.9 Pengujian Metode

Pengujian metode dilakukan dengan tujuan untuk mengetahui kelebihan dan kekurangan pada penelitian. Selain itu, pengujian metode juga digunakan untuk mengetahui seberapa baik kualitas citra yang digunkan dengan alat ukur *Mean Square Error* (MSE) dan *Peak Signal To Noise Ratio* (PSNR).

Nilai MSE digunakan untuk mengetahui nilai kesalahan kuadrat rata-rata dengan membandingkan nilai *pixel* dari *cover image* dengan nilai dari *stego image*

dengan ketentuan posisi *pixel* yang sama. MSE dihitung dengan menggunakan persamaan berikut :

$$MSE = \frac{1}{mn}\sum_{i}^{m}\sum_{j}^{n}\left(I_{i,j} - K_{i,j}\right)^2,$$

Di mana

MSE = *Mean Square Error*

i, j = Koordinat Masing-masing Pixel, dan

mn = panjang dan lebar citra.

Sedangkan PSNR digunakan untuk mengetahui kualitas citra dalam satuan desibel (dB). Kualitas citra dapat dihitung dengan persamaan

$$PSNR = 10\log\left(\frac{Max_i}{MSE}\right)$$

keterangan :

$PSNR$ = *Peak Signal to Noise Ratio*

Max_i = Nilai Maximum ke-i dari pixel

Adapun tabel kualitas citra berdasarkan nilai PSNR sebagai berikut,

PSNR (dB)	Kualitas Citra
20	Rusak (Tidak dapat digunakan)
30	Kurang Baik (Banyak Noise)

40	Cukup Baik (terdapat butiran halus dalam citra)
50	Baik (Terdapat sejumlah Noise, tetapi kualitas citra masih bagus)
≥ 60	Sangat baik (tanpa noise)

Tabel 2.2 Kualitas Citra

2.10 Python

Python merupakan bahasa pemrograman yang mendukung paradigma *object oriented programming* atau *scripting*. Python dibuat oleh Guido van Rossum dari Amsterdam, Belanda. Di dalam pemrograman python terdapat banyak sekali *packages* yang digunakan, salah satunya dalam bidang matematika seperti numpy, math, stegano, docopt dan lain sebagainya untuk mempermudah perhitungan.

2.11 Kajian Keagamaan

Menjaga rahasia dan melindungi pesan tersebut dari pihak yang tidak berwenang merupakan amanah. Seperti yang dijelaskan dalam Al-Qur'an surah Al-Anfal ayat 27 yang berbunyi,

يَأَيُّهَاالَّذِيْنَءَامَنُوْالاَتَخُوْنُوْاالله وَالرَّسُوْلَ وَتَخُوْنُوْاأَمَنَتِكُمْ وَأَنْتُمْ تَعْلَمُوْنَ،

Artinya : "*hai orang-orang yang beriman, janganlah kamu mengkhianati Allah dan Rasul (Muhammad) dan (juga) janganlah kamu mengkhianati amanat-amanat yang dipercayakan kepadamu, sedang kamu mengetahui"*(QS. Al-Anfal :27).

Menurut tafsir Jalalayn bahwa orang-orang beriman dilarang mengkhianati Allah dan Rasulnya dan larangan mengkhianati amanat apapun yang dipercayakan kepada kalian. Maka dari itu, untuk menghindari sifat khianat terhadap amanah baik berupa pesan ataupun informasi lain yang disampaikan perlu dilakukan teknik pengamanan pesan seperti teknik kriptografi dan teknik steganografi.

Teknik kriptografi yaitu teknik mengubah makna pesan (*plaintext*) menjadi *ciphertext*, kemudian kode dari *ciphertext* di sembunyikan ke dalam media gambar, agar pesan tersebut tetap terjaga kerahasiaannya dari orang lain yang tidak berhak, dengan hasil *stego image* dari metode steganografi diharapkan orang yang tidak berkepentingan akan terkecoh dengan tampilan *stego image* tersebut.

BAB III

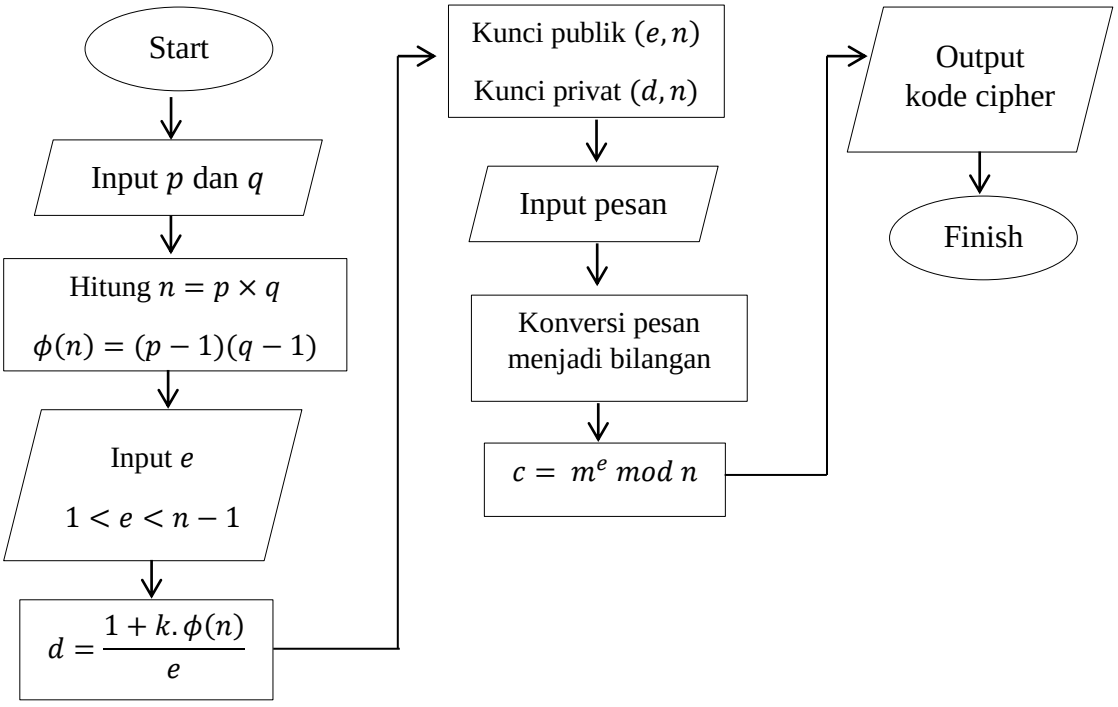
PEMBAHASAN

3.1 Kriptografi dengan Algoritma RSA

Kriptografi adalah teknik pengamanan informasi rahasia dengan menggunakan berbagai perhitungan sesuai dengan aturan. Agar makna pesan tersebut tidak dipahami oleh orang lain. Pada penelitian ini menggunakan kriptografi RSA dalam proses pengamanan pesan, dimana algoritma RSA mempunyai pemecahan kode yang sulit, karena pemfaktoran dua bilangan yang besar menjadi faktor-faktor prima. Penelitian ini menggabungkan teknik kriptografi RSA dan teknik steganografi MSB dan LSB untuk memperoleh tingkat keamanan pesan yang lebih baik.

3.1.1 Proses Enkripsi Metode RSA

Proses enkripsi dengan metode RSA dijelaskan dalam flowchart berikut:



Gambar 3.1 Flowchart enkripsi kriptografi RSA

Berdasarkan Gambar 3.1, proses enkripsi metode RSA, dimulai dengan cara memilih sebarang dua bilangan prima, kemudian dihitung nilai n dan $\phi(n)$, setelah itu pilih nilai e yang relatif prima dengan $\phi(n)$ sebagai kunci publik, setelah didapat nilai e maka kunci privat dapat dicari dengan persaman

$$d = \frac{1 + k.\phi(n)}{e}$$

dengan $k = 1,2,3,..n$. Setelah kunci publik dan privat didapatkan, maka langkah selanjutnya adalah menginputkan pesan yang akan di enkrip, sebelum pesan di enkrip konversikan pesan menjadi bilangan seperti dibawah ini:

A	B	C	D	E	F	G	H	I	J	K	L	M
0	1	2	3	4	5	6	7	8	9	10	11	12

N	O	P	Q	R	S	T	U	V	W	X	Y	Z
13	14	15	16	17	18	19	20	21	22	23	24	25

Tabel 3.1 Konversi abjad ke bilangan

Setelah pesan di konversi menjadi bilangan, maka proses enkripsi dapat dilakukan dengan persamaan $c = m^e \ mod \ n$ dan kunci publik yang telah dipilih. Setelah proses enkripsi selesai maka didapatkan output kode *cipehertext*.

Contoh enkripsi seperti berikut, dengan pesan "**Kabur Mendaki**" melalui proses di bawah ini,

1. Pilih p dan q , dengan $p \neq q$, misalnya

$$p = 11 \text{ dan } q = 13$$

2. Hitung nilai n

$$n = p \times q$$

$$n = 11 \times 13 = 143$$

3. Hitung nilai fungsi *totient Euler* $\phi(n)$

$$\phi(n) = (p-1) \times (q-1)$$

$$= (11-1) \times (13-1)$$

$$= (10) \times (12) = 120$$

4. Pilih bilangan bulat e secara acak yang relatif prima dengan $\phi(n)$, misalnya dipilih $e = 53$, maka diperoleh kunci publik $(53,143)$.

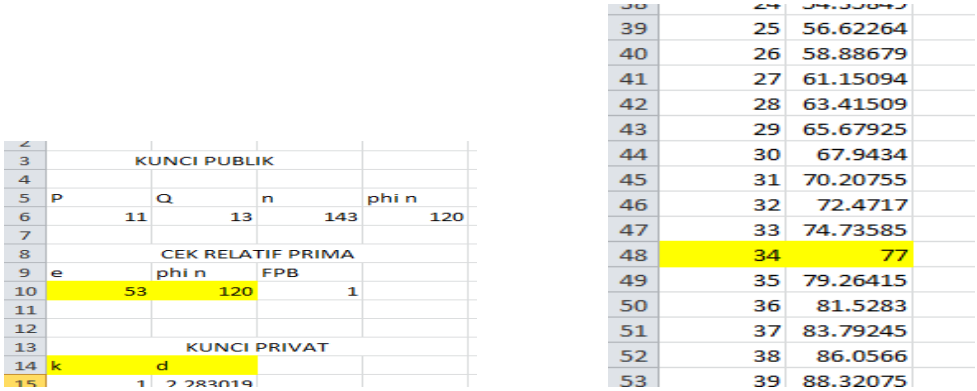
5. Setelah kunci publik diperoleh, maka hitung kunci privat dengan persamaan berikut:

$$d = \frac{1+(k \times \phi(n))}{e} \text{ dengan } k = 1,2,3, \ldots$$

$$d = \frac{1 + (34 \times 120)}{53} = 77$$

didapatkan kunci privat $(77,143)$

kunci privat didapat dengan perhitungan di excel, seperti berikut:



3	KUNCI PUBLIK			
4				
5	P	Q	n	phi n
6	11	13	143	120
7				
8	CEK RELATIF PRIMA			
9	e	phi n	FPB	
10	53	120	1	
11				
12				
13	KUNCI PRIVAT			
14	k	d		
15	1	2.283019		

39	25	56.62264
40	26	58.88679
41	27	61.15094
42	28	63.41509
43	29	65.67925
44	30	67.9434
45	31	70.20755
46	32	72.4717
47	33	74.73585
48	34	77
49	35	79.26415
50	36	81.5283
51	37	83.79245
52	38	86.0566
53	39	88.32075

Gambar 3.2 Hasil pencarian kunci privat

6. Konversi pesan menjadi bilangan seperti berikut :

	K	a	b	u	r	M	e	n	d	a	k	i
M	10	0	1	20	17	12	4	13	3	0	10	8

Tabel 3. 2 Kode *Plaintext*

7. Setelah dikonversi, dilakukan proses enkripsi dengan menggunakan kunci publik $(53,143)$ dan persamaan $C_i = M^e mod\ n$ seperti berikut:

$$C_1 = 10^{53} mod\ 143 = 43$$

$$C_2 = 0^{53} mod\ 143 = 0$$

$$C_3 = 1^{53} mod\ 143 = 1$$

$$C_4 = 20^{53} mod\ 143 = 102$$

$$C_5 = 17^{53} mod\ 143 = 62$$

$$C_6 = 12^{53} mod\ 143 = 12$$

$$C_7 = 4^{53} mod\ 143 = 75$$

$$C_8 = 13^{53} mod\ 143 = 52$$

$$C_9 = 3^{53} mod\ 143 = 126$$

$$C_{10} = 0^{53} mod\ 143 = 0$$

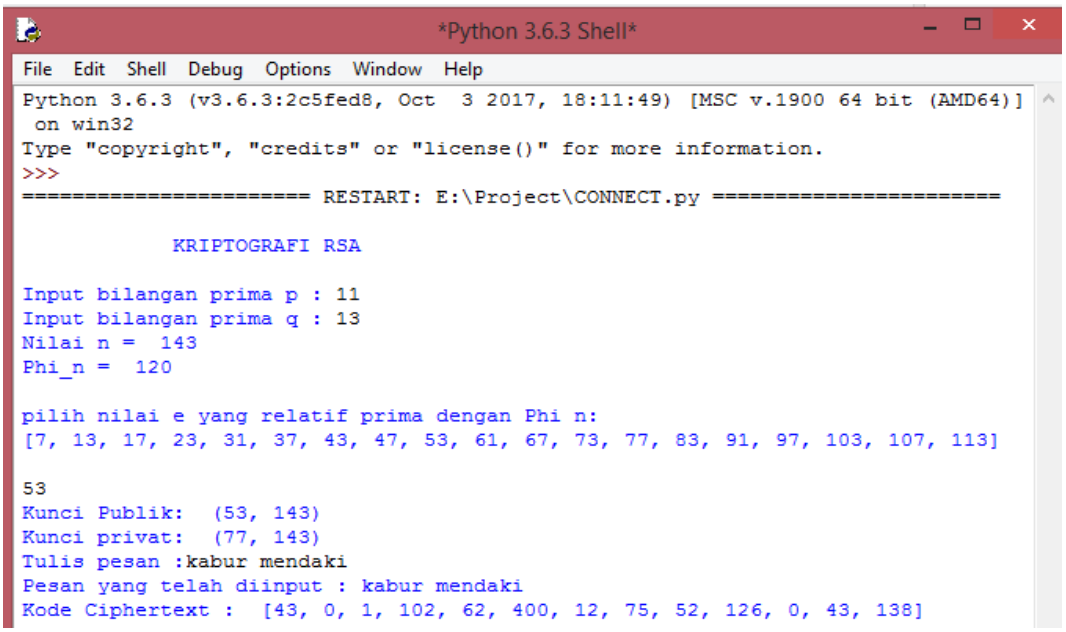
$$C_{11} = 10^{53} mod\ 143 = 43$$

$$C_{12} = 8^{53} mod\ 143 = 138$$

didapatkan kode *ciphertext* :$43, 0, 1, 102, 62, 12, 75, 52, 126, 0, 43, 138$.

3.1.2 Simulasi Proses Enkripsi dengan Python

Proses enkripsi dengan metode RSA dapat disimulasikan ke dalam python dengan hasil seperti berikut :



Gambar 3.3 Simulasi proses enkripsi

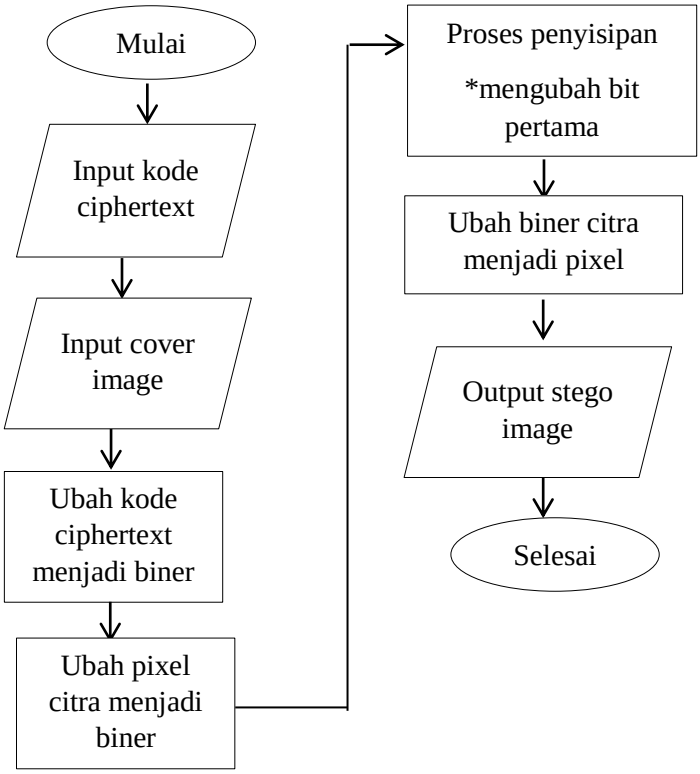
Berdasarkan Gambar 3.3 yaitu simulasi proses enkripsi dengan langkah input dua bilangan prima p dan q, dari nilai p dan q didapatkan nilai n dan $\phi(n)$, kemudian pilih nilai e yang relatif prima dengan $\phi(n)$, dan nilai e akan menjadi kunci enkripsi, kemudian inputkan pesan rahasia dan outputnya adalah kode *ciphertext*.

3.2 Steganografi Metode MSB dan LSB

Steganografi adalah teknik penyembunyian pesan ke dalam media seperti media gambar, video maupun audio. Penyembunyian pesan dilakukan untuk mengamankan pesan rahasia, sehingga orang lain tidak mengetahui bahwa media yang digunakan tersebut memiliki pesan rahasia. Seperti dalam penelitian ini pesan disembunyikan ke dalam media gambar, dengan tujuan agar orang lain terkecoh dengan gambar yang telah disisipkan pesan tersebut.

3.2.1 Proses Penyisipan Metode MSB

Proses penyisipan metode MSB dapat dilihat dalam flowchart berikut:



Gambar 3.4 Flowchart proses encode metode MSB

Berdasarkan Gambar 3.4 dalam proses penyisipan metode MSB terdapat beberapa proses yaitu, menginputkan kode *ciphertext* dan *cover image*, kemudian mengubah kode *ciphertext* dan pixel dari citra menjadi bilangan biner, setelah itu dilakukan proses penyisipan dengan mengganti setiap bit pertama dari *pixel cover image* dengan bit kode *ciphertext*, setelah semua bit kode *ciphertext* tersisipkan, ubah kembali bilangan biner menjadi bilangan desimal dan outputnya berupa gambar yang telah tersisipkan kode *ciphertext* disebut dengan *stego image*.

Adapun contohnya seperti berikut:

1. konversikan kode *ciphertext* dan *cover image* ke bentuk bilangan biner seperti berikut :

43 = 00010101	0 = 00000000	1 = 00000001	102 = 01100110
62 = 00111110	12 = 00001100	75 = 01001011	52 = 00110100
126 = 01111110	0 = 00000000	43 = 00101011	138 = 10001010

Tabel 3. 3 Konversi *Ciphertext* ke Biner (1)

2. Inputkan *cover image* dan konversi ke dalam bentuk matriks seperti berikut :

Gambar 3.5 *Cover Image* metode MSB

didapatkan sebagian matriks dari Gambar 3.5 seperti berikut :

$$\begin{bmatrix} 163 & 163 & 163 & 162 & 162 & 162 & 162 & 162 & 162 & 162 & 162 & 163 \\ 163 & 163 & 163 & 162 & 162 & 162 & 162 & 164 & 164 & 164 & 164 & 164 \\ 164 & 164 & 164 & 164 & 164 & 164 & 164 & 164 & 164 & 163 & 163 & 163 \\ 163 & 163 & 163 & 163 & 163 & 163 & 163 & 162 & 162 & 163 & 163 & 163 \\ 163 & 164 & 164 & 164 & 164 & 164 & 164 & 164 & 164 & 163 & 163 & 163 \\ 163 & 163 & 164 & 162 & 162 & 164 & 163 & 162 & 162 & 163 & 164 & 164 \\ 163 & 164 & 164 & 162 & 162 & 163 & 164 & 164 & 164 & 163 & 163 & 163 \\ 162 & 163 & 162 & 163 & 163 & 164 & 164 & 164 & 164 & 164 & \dots & \dots \end{bmatrix}$$

3. Kemudian matriks tersebut diubah menjadi bilangan biner

$$\begin{bmatrix} 10100011 & 10100011 & 10100011 & 10100010 & 10100010 & 10100010 \\ 10100010 & 10100010 & 10100010 & 10100010 & 10100010 & 10100011 \\ 10100011 & 10100011 & 10100011 & 10100010 & 10100010 & 10100010 \\ 10100010 & 10100010 & 10100100 & 10100100 & 10100100 & 10100100 \\ 10100100 & 10100100 & 10100100 & 10100100 & 10100100 & 10100100 \\ 10100100 & 10100100 & 10100100 & 10100100 & \dots & \dots \end{bmatrix}$$

4. Kemudian dilakukan proses penyisipan dengan metode MSB yaitu dengan mengubah bit pertama dari *cover image* dengan bit kode *ciphertext* seperti berikut:

$$\begin{bmatrix} \mathbf{0}0100011 & \mathbf{0}0100011 & \mathbf{0}0100011 & \mathbf{1}0100010 & \mathbf{0}0100010 & \mathbf{1}0100010 \\ \mathbf{0}0100010 & \mathbf{1}0100010 & \mathbf{0}0100010 & \mathbf{0}0100010 & \mathbf{0}0100010 & \mathbf{0}0100011 \\ \mathbf{0}0100011 & \mathbf{0}0100011 & \mathbf{0}0100011 & \mathbf{0}0100010 & \mathbf{0}0100010 & \mathbf{0}0100010 \\ \mathbf{0}0100010 & \mathbf{0}0100010 & \mathbf{0}0100100 & \mathbf{0}0100100 & \mathbf{0}0100100 & \mathbf{1}0100100 \\ \mathbf{0}0100100 & \mathbf{1}0100100 & \mathbf{1}0100100 & \mathbf{0}0100100 & \mathbf{0}0100100 & \mathbf{1}0100100 \\ \mathbf{1}0100100 & \mathbf{0}0100100 & \mathbf{0}0100100 & \mathbf{0}0100100 & \dots & \dots \end{bmatrix}$$

5. Kemudian konversi kembali ke bentuk matriks seperti berikut:

$$\begin{bmatrix} 42 & 42 & 42 & 162 & 34 & 162 & 34 & 162 & 34 & 34 & 34 & 35 \\ 35 & 35 & 34 & 34 & 34 & 34 & 34 & 36 & 36 & 36 & 164 & 36 \\ 164 & 164 & 36 & 36 & 164 & 164 & 36 & 36 & 36 & 163 & 163 & 163 \\ 163 & 163 & 163 & 163 & 163 & 163 & 163 & 162 & 162 & 163 & 163 & 163 \\ 163 & 164 & 164 & 164 & 164 & 164 & 164 & 164 & 164 & 163 & 163 & 163 \\ 163 & 163 & 164 & 162 & 162 & 164 & 163 & 162 & 162 & 163 & 164 & 164 \\ 163 & 164 & 164 & 162 & 162 & 163 & 164 & 164 & 164 & 163 & 163 & 163 \\ 162 & 163 & 162 & 163 & 163 & 164 & 164 & 164 & 164 & 164 & \dots & \dots \end{bmatrix}$$

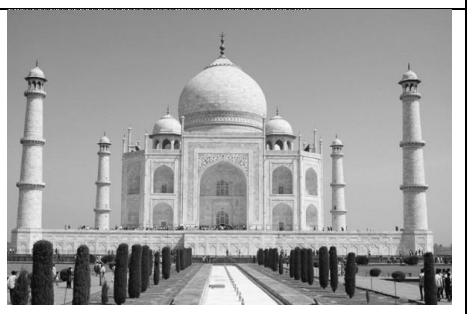
3.2.2 Simulasi Metode MSB dengan Python

Proses steganografi metode MSB disimulasikan dengan python seperti berikut:

```
        PROSES STEGANOGRAFI MSB

input Cover Image dan ekstensinya : E:/COVER/PNG/B/cover.png
Input Kode Ciphertext : 43, 0, 1, 102, 62, 400, 12, 75, 52, 126, 0, 43, 138
Simpan stego image dengan ekstensi :E:/COVER/PNG/B/smsb.png
Input stego image dan ekstensinya :E:/COVER/PNG/B/smsb.png

       PENGUJIAN METODE (MSE DAN PSNR) UNTUK MSB
```

Gambar 3.6 Simulasi penyisipan metode MSB

Dengan menginputkan *cover image* dan kode ciphertext menghasilkan stego image dibawah ini,

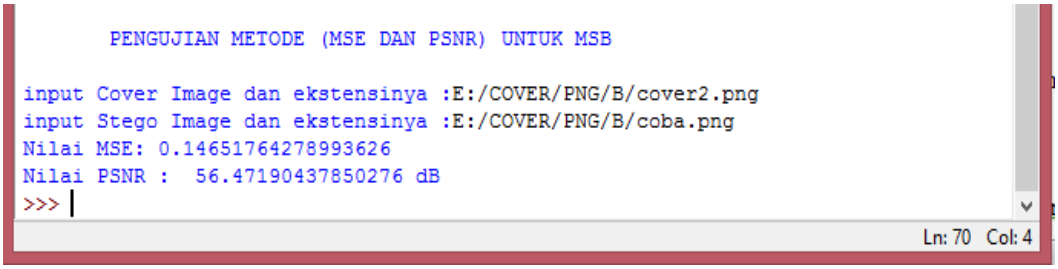
Cover image	*Stego image*
	

Tabel 3.4 Hasil metode MSB

Berdasarkan Tabel 3.4 secara penglihatan tidak mengalami perubahan, hal ini dapat menjaga keberadaan pesan di dalam citra tersebut dari orang yang tidak berkepentingan.

3.2.3 Pengujian Metode MSB

Untuk mendapatkan kualitas cover image yang baik, dilakukan pengujian dengan menghitung nilai MSE dan PSNR seperti berikut,



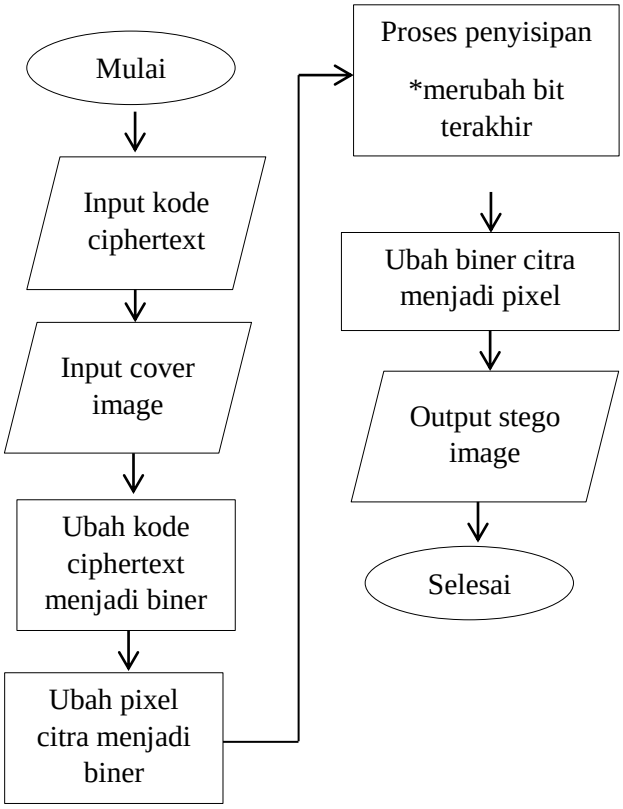
Gambar 3.7 pengujian metode MSB

Berdasarkan Gambar 3.7, diperoleh nilai $MSE = 0{,}1465$ dan nilai $PSNR = 56{,}4719$, berdasarkan tabel kualitas citra nilai PSNR untuk *cover image* Gambar 3.5 dengan metode MSB dapat disimpulkan bahwa citra tersebut baik karena tidak terdapat banyak *noise* dan nilai eror yang cukup kecil sehingga

kualitas citra dari *cover image* tidak banyak mengalami perubahan dengan citra dari *stego image*.

3.2.4 Proses Penyisipan metode LSB

Proses penyisipan metode LSB dapat diketahui dengan flowchart berikut:



Gambar 3.8 Flowchart metode LSB

Berdasarkan gambar 3.8 Proses penyisipan metode LSB dengan langkah menginputkan kode *ciphertext* dan *cover image*, kemudian kode *ciphertext* di konversi menjadi bilangan biner, selain kode *ciphertext*, konversi *cover image* menjadi bilangan biner, setelah keduanya terkonversi, maka dilakukan proses penyisipan dengan mengubah bit terakhir dengan bit kode *ciphertext* sampai semua bit kode *ciphertext* tersisipkan, kemudian konversi kembali menjadi matriks dan outputnya berupa *stego image*.

Adapun contoh penyisipan metode LSB seperti berikut:

1. Konversikan terlebih dahulu kode *ciphertext* dan *cover image* menjadi bilangan biner seperti berikut :

43 = 00010101	0 = 00000000	1 = 00000001	102 = 01100110
62 = 00111110	12 = 00001100	75 = 01001011	52 = 00110100
126 = 01111110	0 = 00000000	43 = 00101011	138 = 10001010

Tabel 3. 5 Konversi kode *Ciphertext* ke Biner (2)

2. Inputkan *cover image* dan konversi ke dalam bentuk matriks seperti berikut :

Gambar 3. 9 *Cover Image* metode LSB

didapatkan sebagian matriks dari Gambar 3.9 seperti berikut :

$$\begin{bmatrix} 163 & 163 & 163 & 162 & 162 & 162 & 162 & 162 & 162 & 162 & 162 & 163 \\ 163 & 163 & 163 & 162 & 162 & 162 & 162 & 164 & 164 & 164 & 164 & 164 \\ 164 & 164 & 164 & 164 & 164 & 164 & 164 & 164 & 164 & 163 & 163 & 163 \\ 163 & 163 & 163 & 163 & 163 & 163 & 163 & 162 & 162 & 163 & 163 & 163 \\ 163 & 164 & 164 & 164 & 164 & 164 & 164 & 164 & 164 & 163 & 163 & 163 \\ 163 & 163 & 164 & 162 & 162 & 164 & 163 & 162 & 162 & 163 & 164 & 164 \\ 163 & 164 & 164 & 162 & 162 & 163 & 164 & 164 & 164 & 163 & 163 & 163 \\ 162 & 163 & 162 & 163 & 163 & 164 & 164 & 164 & 164 & 164 & \dots & \dots \end{bmatrix}$$

3. Kemudian matriks tersebut diubah menjadi bilangan biner

$$\begin{bmatrix} 10100011 & 10100011 & 10100011 & 10100010 & 10100010 & 10100010 \\ 10100010 & 10100010 & 10100010 & 10100010 & 10100010 & 10100011 \\ 10100011 & 10100011 & 10100011 & 10100010 & 10100010 & 10100010 \\ 10100010 & 10100010 & 10100100 & 10100100 & 10100100 & 10100100 \\ 10100100 & 10100100 & 10100100 & 10100100 & 10100100 & 10100100 \\ 10100100 & 10100100 & 10100100 & 10100100 & \dots & \dots \end{bmatrix}$$

4. Kemudian dilakukan proses penyisipan dengan metode LSB yaitu dengan mengganti bit terakhir dari *cover image* dengan bit kode *ciphertext*, sampai semuan bit kode tersisipkan seperti berikut:

$$\begin{bmatrix} 10100010 & 10100010 & 10100010 & 10100011 & 10100010 & 10100011 \\ 10100010 & 10100011 & 10100010 & 10100010 & 10100010 & 10100010 \\ 10100010 & 10100010 & 10100010 & 10100010 & 10100010 & 10100010 \\ 10100010 & 10100010 & 10100100 & 10100100 & 10100100 & 10100101 \\ 10100100 & 10100101 & 10100101 & 10100101 & 10100100 & 10100100 \\ 10100101 & 10100101 & 10100100 & 10100100 & \dots & \dots \end{bmatrix}$$

5. Kemudian konversi kembali ke bentuk matriks seperti berikut:

$$\begin{bmatrix} 162 & 162 & 162 & 163 & 162 & 163 & 162 & 163 & 162 & 162 & 162 & 162 \\ 162 & 162 & 162 & 162 & 162 & 162 & 162 & 164 & 164 & 164 & 165 & 164 \\ 165 & 165 & 165 & 164 & 164 & 164 & 164 & 164 & 164 & 163 & 163 & 163 \\ 163 & 163 & 163 & 163 & 163 & 163 & 163 & 162 & 162 & 163 & 163 & 163 \\ 163 & 164 & 164 & 164 & 164 & 164 & 164 & 164 & 164 & 163 & 163 & 163 \\ 163 & 163 & 164 & 162 & 162 & 164 & 163 & 162 & 162 & 163 & 164 & 164 \\ 163 & 164 & 164 & 162 & 162 & 163 & 164 & 164 & 164 & 163 & 163 & 163 \\ 162 & 163 & 162 & 163 & 163 & 164 & 164 & 164 & 164 & 164 & \dots & \dots \end{bmatrix}$$

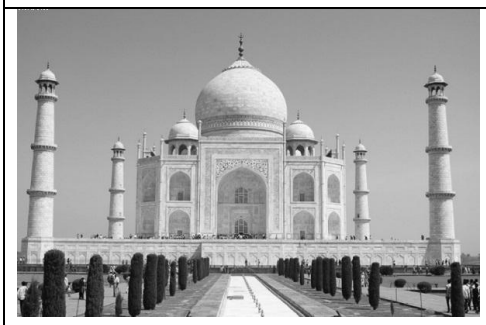
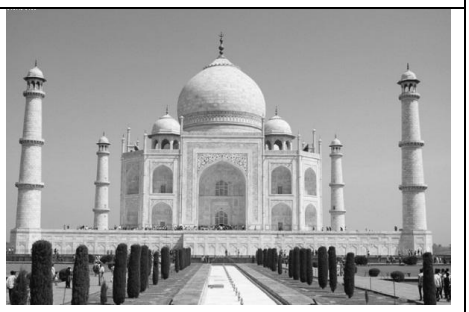
3.2.5 Simulasi Metode LSB Python

Proses penyisipan metode LSB dengan menggunakan python seperti berikut:

```
        PROSES STEGANOGRAFI LSB

Pilih satu :
 1.Encode
 2.Decode
  3.Exit
1
Input cover image dan ekstensinya: E:/COVER/PNG/B/cover.png
Input Kode Ciphertext : 43, 0, 1, 102, 62, 400, 12, 75, 52, 126, 0, 43, 138
Simpan Stego image dan ekstensinya: E:/COVER/PNG/B/slsb.png

 Encode sukses.

stego image tersimpan di :  E:/COVER/PNG/B/slsb.png
```

Gambar 3.10 Simulasi penyisipan metode LSB

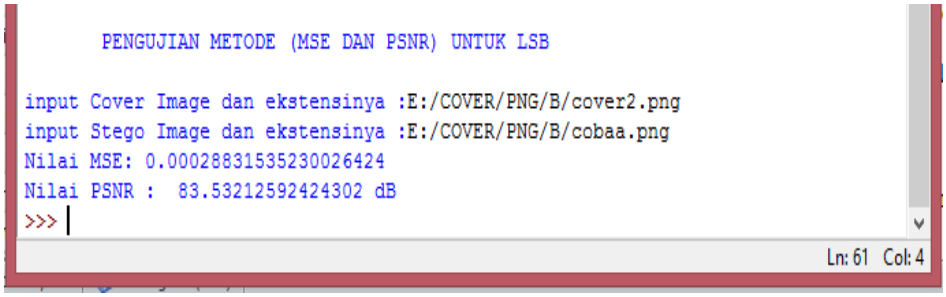
Menghasilkan stego image seperti berikut

Cover Image	*Stego Image*
	

Tabel 3.6 Hasil Simulasi metode LSB

3.2.6 Pengujian Metode LSB

Pengujian cover image untuk mendapatkan kualitas citra yang baik dengan metode LSB seperti berikut:



Gambar 3.11 Pengujian Metode LSB

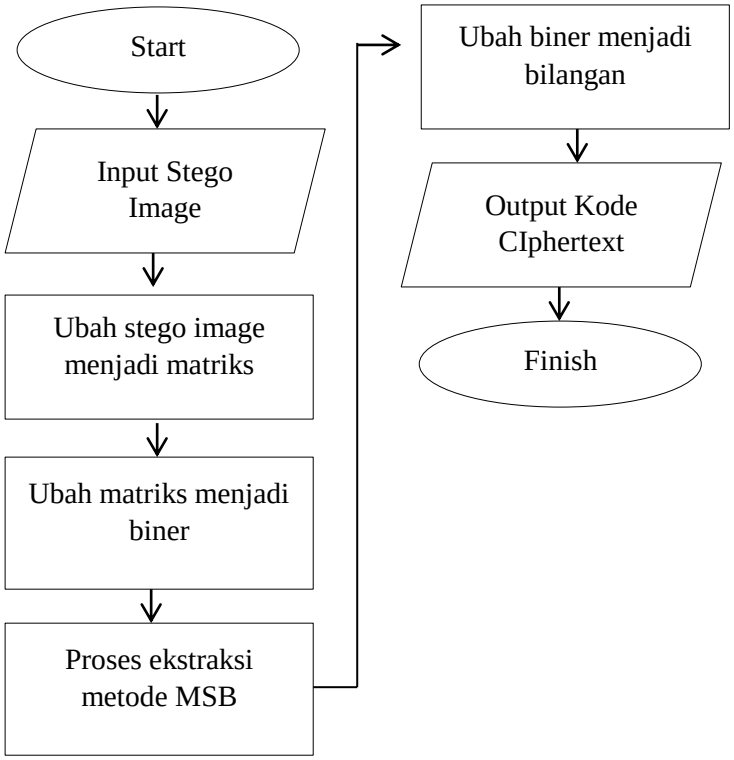
Berdasarkan Gambar 3.11, diperoleh nilai $MSE = 0{,}00028$ dan nilai $PSNR = 83{,}5321$ db. Menurut tabel kualitas citra dengan *cover image* tersebut sangat baik, karena tidak tedapat banyak noise dan menghasilkan *stego image* yang tidak megalami banyak perubahan dengan *cover image*.

3.2.7 Proses Pengiriman

Setelah pesan terenkrip dengan kriptografi RSA dan kode *ciphertext* tersisipkan ke dalam citra dengan steganografi MSB dan LSB, maka *stego image* dikirimkan kepada pihak yang dituju. Pengiriman dapat melalui media sosial ataupun bertemu secara langsung. Setelah *stego image* terkirim, maka pihak tersebut akan melakukan proses ekstraksi dari metode MSB dan LSB.

3.2.8 Proses Ekstraksi Metode MSB

Proses ekstraksi metode MSB dapat diketahui dengan flowchart dibawah ini:



Gambar 3.12 Flowchart proses ekstraksi metode MSB

Berdasarkan Gambar 3.12, diketahui proses ekstraksi metode MSB dengan langkah menginputkan *stego image*, kemudian ubah *stego image* tersebut menjadi matriks, setelah diubah menjadi matriks ubah kembali menjadi bilangan biner, setelah itu ambil nilai dari setiap bit pertama dan kumpulkan menjadi 8 bit, kemudian konversi bilangan biner menjadi bilangan, dan outputnya adalah kode *ciphertext*.

Berikut adalah contoh proses ekstraksi kode *ciphertext* dari *stego image*, dengan langkah-langkah sebagai berikut.

1. Inputkan *stego image*.

Gambar 3.13 *Stego image* metode MSB

2. Kemudian *stego image* dikonversi ke dalam bentuk matriks.

$$\begin{bmatrix} 42 & 42 & 42 & 162 & 34 & 162 & 34 & 162 & 34 & 34 & 34 & 35 \\ 35 & 35 & 34 & 34 & 34 & 34 & 34 & 36 & 36 & 36 & 164 & 36 \\ 164 & 164 & 36 & 36 & 164 & 164 & 36 & 36 & 36 & 163 & 163 & 163 \\ 163 & 163 & 163 & 163 & 163 & 163 & 163 & 162 & 162 & 163 & 163 & 163 \\ 163 & 164 & 164 & 164 & 164 & 164 & 164 & 164 & 164 & 163 & 163 & 163 \\ 163 & 163 & 164 & 162 & 162 & 164 & 163 & 162 & 162 & 163 & 164 & 164 \\ 163 & 164 & 164 & 162 & 162 & 163 & 164 & 164 & 164 & 163 & 163 & 163 \\ 162 & 163 & 162 & 163 & 163 & 164 & 164 & 164 & 164 & 164 & \dots & \dots \end{bmatrix}$$

3. Setelah itu, matriks dari *stego image* dikonversi menjadi bilangan biner seperti berikut.

$$\begin{bmatrix} 00100011 & 00100011 & 00100011 & 10100010 & 00100010 & 10100010 \\ 00100010 & 10100010 & 00100010 & 00100010 & 00100010 & 00100011 \\ 00100011 & 00100011 & 00100011 & 00100010 & 00100010 & 00100010 \\ 00100010 & 00100010 & 00100100 & 00100100 & 00100100 & 10100100 \\ 00100100 & 10100100 & 10100100 & 00100100 & 00100100 & 10100100 \\ 10100100 & 00100100 & 00100100 & 00100100 & \dots & \dots \end{bmatrix}$$

4. Kemudian ambil setiap bit pertama dari *stego image* yang telah dikonversi menjadi bilangan biner.

00010101	00000000	00000001	01100110
00111110	00001100	01001011	00110100
01111110	00000000	00101011	10001010

Tabel 3. 7 Biner dari kode *ciphertext* (1)

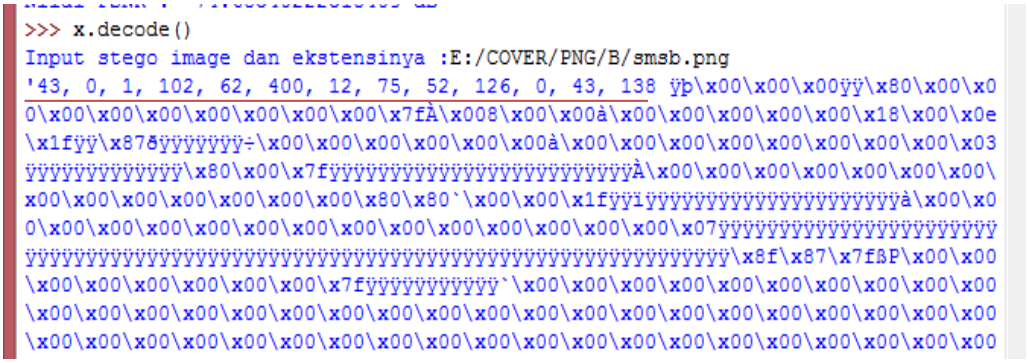
5. Setelah itu, bit dikonversi ke dalam bentuk desimal.

43	0	1	102
62	12	75	52
126	0	43	138

Tabel 3. 8 Kode *Ciphertext* (1)

3.2.9 Simulasi Ekstraksi metode MSB dengan Python

Proses ekstraksi metode MSB dengan python dibawah ini.

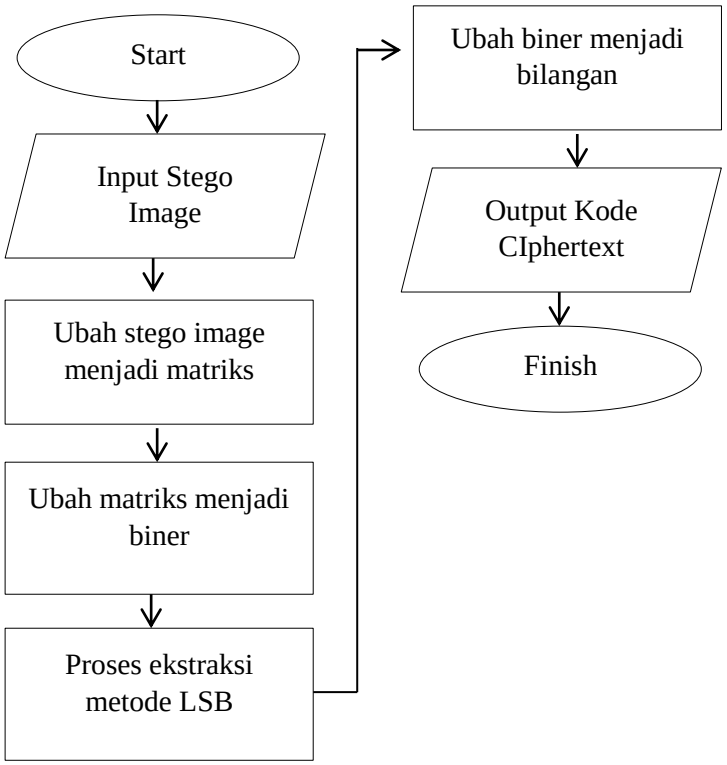


Gambar 3.14 Simulasi Decode (Ekstraksi) metode MSB

Berdasarkan Gambar 3.14, proses ekstraksi dengan metode MSB menghasilkan kode *ciphertext* yang sama yaitu, 43, 0, 1, 102, 62, 400, 12, 75, 52, 126, 0, 43, 138. Artinya dengan citra tersebut, kode rahasia yang disisipkan dapat diungkap kembali.

3.2.10 Proses Ekstraksi Metode LSB

Proses ekstraksi metode LSB dapat diketahui dengan flowchart dibawah ini,

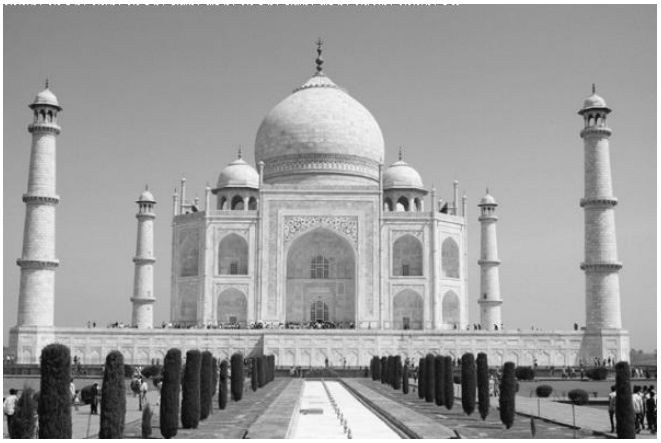


Gambar 3.15 Flowchart Proses ekstraksi Metode LSB

Berdasarkan Gambar 3.15, proses ekstraksi metode LSB dengan menginputkan *stego image*, kemudian ubah *stego image* menjadi matriks, setelah itu konversi matriks dari *stego image* menjadi bilangan biner, kemudian dilakukan proses ekstraksi metode LSB dengan cara mengambil setiap bit terakhir dari *stego image* dan rangkai menjadi 8 bit, setelah itu konversi bilangan biner menjadi desimal dan outputnya merupakan kode *ciphertext*.

Contoh proses ekstraksi metode LSB dilakukan untuk mengungkap kembali kode *ciphertext* yang telah disembunyikan dengan cara seperti berikut :

1. Inputkan *stego image*



Gambar 3.15 *Stego image* metode LSB

Gambar 3.15, dikonversi ke bentuk matriks seperti berikut

$$\begin{bmatrix} 162 & 162 & 162 & 163 & 162 & 163 & 162 & 163 & 162 & 162 & 162 & 162 \\ 162 & 162 & 162 & 162 & 162 & 162 & 162 & 164 & 164 & 164 & 165 & 164 \\ 165 & 165 & 165 & 164 & 164 & 164 & 164 & 164 & 164 & 163 & 163 & 163 \\ 163 & 163 & 163 & 163 & 163 & 163 & 163 & 162 & 162 & 163 & 163 & 163 \\ 163 & 164 & 164 & 164 & 164 & 164 & 164 & 164 & 164 & 163 & 163 & 163 \\ 163 & 163 & 164 & 162 & 162 & 164 & 163 & 162 & 162 & 163 & 164 & 164 \\ 163 & 164 & 164 & 162 & 162 & 163 & 164 & 164 & 164 & 163 & 163 & 163 \\ 162 & 163 & 162 & 163 & 163 & 164 & 164 & 164 & 164 & 164 & \dots & \dots \end{bmatrix}$$

2. Kemudian matriks dari *stego image* dikonversi menjadi bilangan biner.

$$\begin{bmatrix} 10100010 & 10100010 & 10100010 & 10100011 & 10100010 & 10100011 \\ 10100010 & 10100011 & 10100010 & 10100010 & 10100010 & 10100010 \\ 10100010 & 10100010 & 10100010 & 10100010 & 10100010 & 10100010 \\ 10100010 & 10100010 & 10100100 & 10100100 & 10100100 & 10100101 \\ 10100100 & 10100101 & 10100101 & 10100101 & 10100100 & 10100100 \\ 10100101 & 10100101 & 10100100 & 10100100 & \dots & \dots \end{bmatrix}$$

3. Setelah itu, ambil setiap bit terakhir dari *stego image*

00010101	00000000	00000001	01100110
00111110	00001100	01001011	00110100
01111110	00000000	00101011	10001010

Tabel 3.9 Biner dari kode *ciphertext* (2)

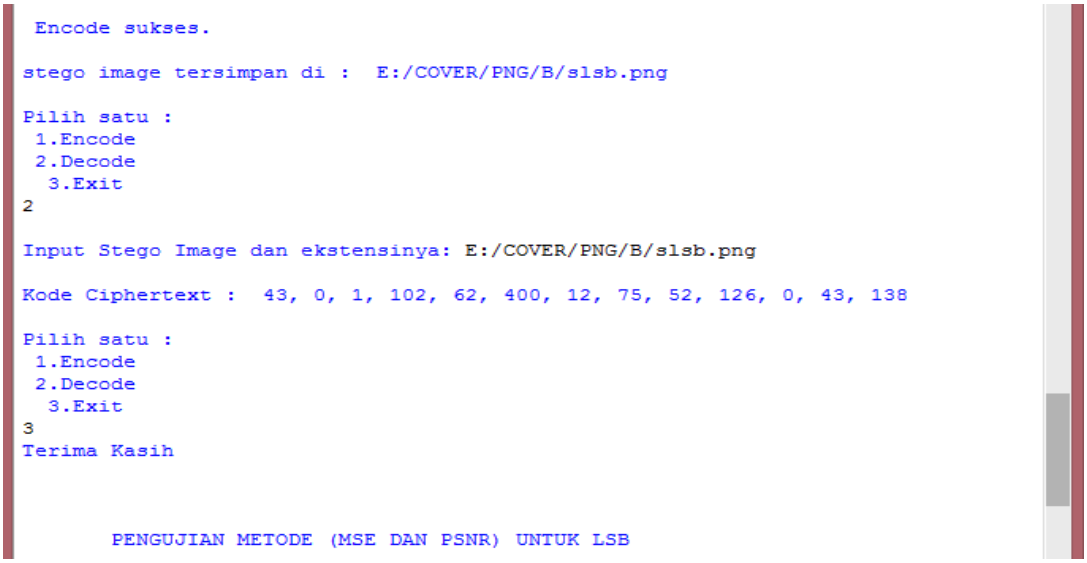
4. Setelah itu, bit dikonversi ke dalam bentuk desimal

43	0	1	102
62	12	75	52
126	0	43	138

Tabel 3.10 Kode *Ciphertext* (2)

3.2.11 Simulasi Ekstraksi metode LSB dengan Python

Proses ekstraksi metode LSB dengan menggunakan python seperti berikut

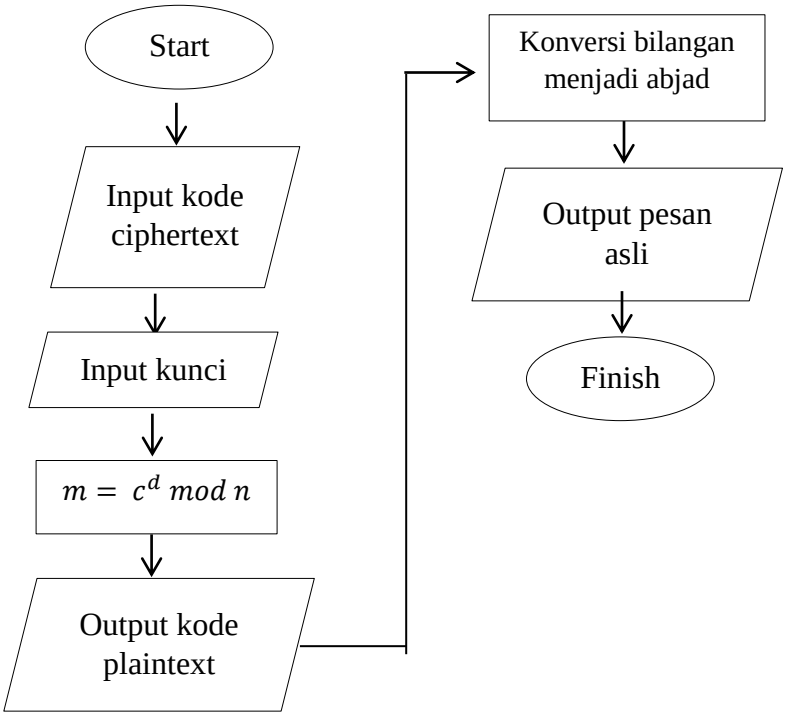


Gambar 3.17 Simulasi Decode(ekstraksi) Metode LSB

Berdasarkan Gambar 3.17 *stego image* dengan metod LSB dapat di ungkap kode rahasia yang disisipkan dan kode rahasia tersebut tidak rusak. Maka dari itu cover image tersebut juga baik digunkan untuk metode LSB.

3.2.12 Proses Dekripsi Metode RSA

Setelah kode pesan terenkrip, kemudian dilakukan proses dekripsi seperti flowchart dibawah ini:



Gambar 3.17 Flowchart proses dekripsi

Berdasarkan gambar 3.17, diketahui bahwa proses dekripsi melalui beberapa tahap seperti menginputkan kode *ciphertext* dan kunci privat, kemudian dilakukan proses dekripsi dengan kunci privat tersebut dan persamaan $m = c^d \ mod \ n$, setelah itu didapatkan hasil kode plaintextnya, dari kode plaintext tersebut dikonversi kembali menjadi abjad dan didapatkan pesan aslinya, seperti pada program python di bawah ini.

```
Input kunci privat : 77
Input kode Ciphertext :43, 0, 1, 102, 62, 400, 12, 75, 52, 126, 0, 43, 138
Plaintextnya : KABURMENDAKI
```

Gambar 3.19 Simulasi Proses dekripsi

Contoh proses dekripsi dengan kriptografi RSA dan menggunakan kunci privat (77,143) serta persamaan $M_i = C^d mod\ n$ seperti berikut:

$$M_1 = 43^{77} mod\ 143 = 10$$

$$M_2 = 0^{77} mod\ 143 = 0$$

$$M_3 = 1^{77} mod\ 143 = 1$$

$$M_4 = 102^{77} mod\ 143 = 20$$

$$M_5 = 62^{77} mod\ 143 = 17$$

$$M_6 = 12^{77} mod\ 143 = 12$$

$$M_7 = 75^{77} mod\ 143 = 4$$

$$M_8 = 52^{77} mod\ 143 = 13$$

$$M_9 = 126^{77} mod\ 143 = 3$$

$$M = 0^{77} mod\ 143 = 0$$

$$M_{11} = 43^{77} mod\ 143 = 10$$

$$M_{12} = 138^{77} mod\ 143 = 8$$

Di dapatkan kode *plaintext* : 10, 0, 1, 20, 17, 12, 4, 13, 3, 0, 10, 8 jika di konversi ke huruf abjad artinya" Kabur Mendaki".

3.2.13 Tabel Hasil dengan Python

Proses steganografi metode MSB dan LSB menggunakan python mendapatkan hasil seperti berikut:

1. Tabel hasil metode MSB (*Most Significant Bit*)

No	***Cover Image***	***Stego Image***
1		
2		
3		
4		

5		
6		

Tabel 3.11 Tabel Hasil Metode MSB

2. Tabel hasil metode LSB (*Least Significant Bit*)

No	***Cover Image***	***Stego Image***
1		
2		

3		
4		
5		
6		

Tabel 3.12 Tabel Hasil Metode LSB

3.2.14 Tabel nilai MSE dan PSNR dari Metode MSB dan LSB

Proses perhitungan nilai MSE dan PSNR dengan python didapatkan hasil seperti berikut:

No	Metode MSB			Metode LSB		
	MSE	**PSNR**	**Status**	**MSE**	**PSNR**	**Status**
1	0.00266	73.8736	Baik	4.5839	101.5184	Baik
2	0.00119	77.3737	Baik	3.7466	102.3943	Baik
3	0.00121	77.2716	Baik	4.7722	101.3435	Baik
4	0.00319	73.0859	Baik	5.2905	100.8958	Baik
5	0.00176	75.6676	Baik	4.6257	101.4789	Baik
6.	0.00550	70.7207	Baik	1.1394	97.5638	Baik

Tabel 3.13 Hasil Nilai MSE dan PSNR

Dari tabel 3.13 didapatkan *stego image* terbaik dengan metode MSB yaitu pada gambar nomor 2, menurut tabel kualitas citra, nilai PSNR-nya 77.3737dB menghasilkan hasil yang baik karena *stego image* tersebut tidak terdapat banyak *noise* dan didapat pula nilai eror cukup rendah daripada *stego image* lainnya sebesar 0.00119, yang artinya kulitas *cover image* dan *stego image* tidak mengalami banyak perubahan.

Kemudian *stego image* yang terbaik dari metode LSB berdasarkan tabel 3.13 yaitu gambar nomor 6 dengan nilai eror paling rendah yaitu 1.1394 dan nilai PSNR sebesar 97.5638, menurut tabel kualitas citra dengan nilai PSNR lebih dari 60 maka *stego image* sangat baik karena tidak mengandung banyak noise dan tidak terjadi banyak perubahan dari *cover image*, yang artinya untuk kulitas *cover image* dan *stego image* dengan metode LSB tidak mengalami banyak perubahan.

3.3 Integrasi Kriptografi Menurut Kajian dalam Al-Qur'an

Menjaga keamanan pesan adalah hal yang wajib dilakukan sehingga pesan tetap terjaga kerahasiaannya dari pihak yang tidak berwenang. Dalam Al-Qur'an surah Al-Waqi'ah ayat 77-80 sudah dijelaskan sebagai berikut:

اِنَّهٗ لَقُرْءَانٌ كَرِيْم، فِيْ كِتبٍ مَكْنُوْن، لاَّيَمَسُهٗ,إِلاَّالْمُطَهَّرُوْنَ، تَنْزِيْلٌ مِنْ رَبِّ الْعَلَمِيْنَ،

Artinya : *"bahwa sesungguhnya (yang dibacakan kepada kamu) itu ialah Al-Qur'an yang mulia. Yang tersimpan dalam kitab yang cukup terpelihara, yang tidak disentuh melainkan oleh makhluk-makhluk yang disucikan, Al-Qur'an itu diturunkan dari Allah, Tuhan sekalian alam."* (Q.S Al-Waqi'ah:77-80)

Dari ayat tersebut dijelaskan tentang jaminan Allah kepada Al-Qur'an. Allah memelihara dan menjaga isi Al-Qur'an dari upaya syetan yang ingin merubah isi Al-Qur'an, sehingga tetap terjaga kemurniannya. Sama halnya dengan menjaga keamanan pesan dari pihak yang tidak berwenang, yang ingin mengetahui isi pesan yang telah disandikan dan disembunyikan. Berbagai cara dapat dilakukan untuk menjaga kerahasiaan pesan, seperti teknik kriptografi. Kriptografi adalah ilmu dan seni yang digunakan untuk menjaga keamanan pesan, hasil dari kriptografi merupakan pesan yang tidak dapat dibaca, dalam hal ini dapat membuat orang lain menjadi curiga, maka dari itu untuk mengurangi kecurigaan tersebut, ditambahkan teknik steganografi. Steganografi adalah ilmu dan seni menyembunyikan pesan ke dalam media lain. Dalam penelitian ini digunakan media citra, dengan tujuan, agar orang lain terkecoh dengan tampilan citra yang telah disisipkan pesan rahasia. Dengan menggunakan dua teknik tersebut, agar pesan dapat terjaga kerahasiaannya dari pihak yang tidak berwenang.

BAB IV
PENUTUP

4.1 Kesimpulan

Berdasarkan pembahasan dari tabel 3.13 dapat disimpulkan bahwa *cover image* yang digunakan dengan metode MSB diperoleh *stego image* yang baik yaitu pada gambar 2, dengan nilai eror terkecil daripada *stego image* yang lain yaitu 0,00119 dan nilai PSNR 77,3737 dB, menurut tabel 2.2, nilai PSNR yang didapat menyebabkan *stego image* tidak terdapat banyak *noise*, sehingga tidak mengalami banyak perubahan dari *cover image*. Sedangkan untuk metode LSB diperoleh *stego image* yang baik yaitu gambar 6, dengan nilai eror terkecil yaitu 1,1394 dan nilai PSNR-nya yaitu 97,5638 dB. Menurut tabel 2.2, nilai PSNR tersebut menghasilkan *stego image* yang sangat baik karena tidak ada *noise* dan tidak banyak mengalami perubahan, serta tidak menimbulkan kecurigaan bagi pihak lain. Dari kedua metode steganografi tersebut, dapat disimpulkan bahwa metode LSB adalah metode yang baik untuk menyembunyikan pesan tersebut, karena penyisipan dilakukan pada bit yang tidak terlalu berpengaruh dalam citra sehingga *cover image* tidak mengalami banyak perubahan.

4.2 Saran

Pada penelitian ini, penulis telah melakukan proses kriptografi RSA, dilanjutkan proses steganografi MSB dan LSB dengan media citra. Penulis menyarankan untuk penelitian selanjutnya, menggunakan metode kriptografi dan steganografi metode yang lain, dengan media seperti video maupun audio.

DAFTAR PUSTAKA

Ariyus, Dony. 2006. *Kriptografi: Keamanan Data dan Komunikasi*. Yogyakarta: Graha Ilmu.

Ariyus, Dony. 2006. *Pengantar Ilmu Kriptografi, Teori, Analisis dan Implementasi*. Yogyakarta: Andi.

Azlansyah, Muhammad, dkk. 2019. *Penyisipan Pesan pada Citra Digital menggunakan Metode Least Significant Bit*. ITS. Jurnal Sains dan Seni. Vol.8, No.1.

Benny, Leo. 2017. *Analisis dan Perancangan Aplikasi Kriptografi Keamanan File Berbasis Teks dengan Menggunakan Metode RSA*. Medan. Riset dan E-Jurnal Manajemen Informatika Komputer. Vol.1 No 2.

Handoyo, Antonius Erick, dkk. 2018. *Teknik Penyembunyian dan Enkripsi Pesan pada Citra Digital dengan Kombinasi Metode LSB dan RSA*. Universitas Dian Nuswantoro. Jurnal Teknologi dan Sistem Komputer. 6(1).

Ibrahim, Rohmat Nur, Ilham M.S. 2017. *Perancangan Aplikasi Stegakrip dengan Metode LSB dan Algoritma RSA Berbasis WEB*. STMIK Mardira Indonesia. Jurnal Computech & Bisnis. Vol.11 No 1.

Jatmoko, dkk. 2018. *Uji Performa Penyisipan Pesan dengan Metode LSB dan MSB pada Citra Digital untuk Keamanan Komunikasi.* Universitas Dian Nuswantoro. Semarang. Jurnal Dinamika Rekayasa. Vol. 14 No. 1.

Kromodimoeljo, Sentot. 2009. *Teori dan aplikasi Kriptografi.* SPK IT Consulting.

Kusumanto, RD, dkk. 2011. *Pengolahan Citra Digital untuk Mendeteksi Obyek menggunakan Pengolahan Warna Model Normalisasi RGB.* Politeknik Negeri Sriwijaya, Palembang. Seminar Nasional Teknologi Informasi & Komunikasi Terapan.

Manalu, Harry S. 2013. *Penerapan Metode Most Significant Bit untuk Penyisispan Pesan Teks Pada Citra Digital*. STMIK Budidarma. Medan. Pelita Informatika Budi Darma. Vol IV, No 1.

Munir, Rinaldi. 2004. *Kriptografi.* Bandung: Institut Teknologi Bandung.

Munir, Rinaldi. 2019. *Kriptografi Edisi 2* .Bandung: Institut Teknologi Bandung.

Munir, Rinaldi. 2004. *Steganografi dan Watermarking.* Bandung: Institut Teknologi Bandung.

Munir, Rinaldi. 2004. *Pengantar Pengolahan Citra.* Bandung: Institut Teknologi Bandung.

Sadikin, Rifki. 2012. *Kriptografi untuk Keamanan dan Jaringan dan Implementasinya dalam Bahasa Java*. Yogykarta: Andi.

Saputra, Ragil, dkk. 2016. *Implementasi Kriptografi Kunci Publik dengan Algoritma RSA-CRT pada Aplikasi Instant Messaging.* Semarang. Universitas Diponegoro. *Scientific journal of informatics*. Vol.3 No.1.

Syawal, Muhamad Fitra, dkk. 2016. *Implementasi Teknik Steganografi menggunakan Algoritma Vignere Cipher dan Metode LSB.* Universitas Budi Luhur. Jurnal TICOM Vol.4 No.3.

Utomo, Tri Prasetyo. 2017. *Steganografi Gambar dengan Metode Least Significant Bit untuk Proteksi Komunikasi Media Online.* UIN Sunan Gunung Djati Bandung.

RIWAYAT HIDUP

Khilmi Hani, Lahir di Kota Sidoarjo, tanggal 21 Mei 1998, biasa di panggil Hani, Khilmi, Jodha, tinggal di Desa Kedungkendo RT 13/05, Kecamatan Candi, Kabupaten Sidoarjo. Anak kedua dari dua bersaudara. Putri Bapak Karis dan Ibu Turah serta Adik dari M.Uman.

Pendidikan Taman Kanak-Kanak ditempuh di RA Al-Hikmah lulus tahun 2004, Pendidikan dasarnya ditempuh di MI-Ma'arif Nu KedungKendo lulus tahun 2010. Selanjutnya melanjutkan ke sekolah menengah pertama di MTS-Ma'arif Nu, lulus tahun 2013. Kemudian melanjutkan ke sekolah menengah kejuruan di SMK Negeri 2 Buduran Sidoarjo, lulus tahun 2016. Setelah itu melanjutkan ke perguruan tinggi di Universitas Islam Negeri Maulana Malik Ibrahim Malang. Mengambil jurusan Matematika, di Fakultas Sains dan Teknologi. Penulis dapat dihubungi melalui email : khilmihani21@gmail.com.

KEMENTRIAN AGAMA RI
UNIVERSITAS ISLAM NEGERI
MAULANA MALIK IBRAHIM MALANG
FAKULTAS SAINS DAN TEKNOLOGI
Jl. Gajayana No.50 Dinoyo Malang Telp./Fax.(0341)558933

BUKTI KONSULTASI SKRIPSI

Nama : Khilmi Hani
NIM : 16610093
Fakultas/Jurusan : Sains dan Teknologi/ Matematika
Judul skripsi : Algoritma Kriptografi dan Steganografi untuk Pengamanan Pesan ke dalam Citra
Pembimbing I : Muhammad Khudzaifah M.Si
Pembimbing II : Juhari M.Si

No	Tanggal	Hal	Tanda Tangan
1	3-01-2020	Konsultasi Bab I dan Bab II	1
2	3-02-2020	Konsultasi Kajian Keagamaan	2
3	12-03-2020	Revisi Bab I, bab II dan Konsultasi Bab III	3
4	3-02-2020	Revisi Kajian Keagamaan	4
5	24-03-2020	Revisi Bab III	5
6	27-03-2020	ACC Bab I dan Bab II	6
7	5-04-2020	ACC Kajian Keagamaan	7
8	5-04-2020	ACC Bab III	8
9	22-04-2020	Konsultasi Bab IV	9
10	24-04-2020	ACC Bab IV	10
11	27-04-2020	ACC Keseluruhan Kajian Keagamaan	11
12	05-05-2020	ACC Keseluruhan	12

Malang, 12 Agustus 2020
Mengetahui,
Ketua Jurusan Matematika

Dr. Usman Pagalay, M.Si
NIP.19650414 200312 1 001

LAMPIRAN

SCRIPT KRIPTOGRAFI RSA

```
import math
from math import *
k="\nKRIPTOGRAFI RSA\n"
print('\n'.join('{:^40}'.format(s) for s in k.split('\n')))
#pilih p dan q
p = int(input("Input bilangan prima p : "))
q = int(input("Input bilangan prima q : "))
#Cek p dan q prima
def cekprima(a):
    if(a==2):
        return True
    elif a<2 and a%2==0:
        return False
    elif(a>2):
        for i in range(2,a):
            if not(a%i):
                return false
    return True
cek_p = cekprima(p)
cek_q = cekprima(q)
while(((cek_p==False)or(cek_q==False))):
    p = int(input("Input bilangan prima p : "))
    q = int(input("Input bilangan prima q : "))
    cek_p = cekprima(p)
    cek_q = cekprima(q)
#nilai N=pxq
n = p * q
```

```
print("Nilai n = ",n)
#Phi(n)= (p-1)*(q-1)
r= (p-1)*(q-1)
print("Phi_n = ",r)
#GCD
def gcd(a,b):
    while b != 0:
        c = a % b
        a = b
        b = c
    return a
#Algoritma Euclid
def eugcd(a,m):
    for x in range(1, m):
        if (a * x) % m == 1:
            return x
    return None
#Extend algoritma euclid
def eea(a,b):
    if(a%b==0):
        return(b,0,1)
    else:
        gcd,s,t = eea(b,a%b)
        s = s-((a//b) * t)
        return(gcd,t,s)
#multple inverse
def multi(e,r):
    gcd,s,_=eea(e,r)
    if(gcd!=1):
```

```
        return None
    else:
        return s%r
#mencari nilai yang membuat e,r relative prima
def coprimes(a):
    l = []
    for x in range(2, a):
        if gcd(a, x) == 1 and eugcd(x,r) != None:
            l.append(x)
    for x in l:
        if x == eugcd(x,r):
            l.remove(x)
    return l
print("\npilih nilai e yang relatif prima dengan Phi n:")
print(str(coprimes(r)) + "\n")
e=int(input())
d=eugcd(e,r)
publik = (e,n)
privat = (d,n)
print("Kunci Publik: ",publik)
print("Kunci privat: ",privat)
#Enkripsi
def enkripsi(pubkey,pesan):
    pubkey=e,n
    x=[]
    m=0
    for i in pesan:
        if(i.isupper()):
            m = ord(i)-65
```

```
            c=(m**e)%n
            x.append(c)
        elif(i.islower()):
            m= ord(i)-97
            c=(m**e)%n
            x.append(c)
        elif(i.isspace()):
            spc=400
            x.append(400)
    return x
 #Dekripsi
def dekripsi(privkey,cipher):
    d,n=privkey
    txt=cipher.split(',')
    x=''
    m=0
    for i in txt:
        if(i=='400'):
            x+=' '
        else:
            m=(int(i)**d)%n
            m+=65
            c=chr(m)
            x+=c
    return x
#Message
message = input("Tulis pesan :")
print("Pesan yang telah diinput :",message)
#Proses enkripsi
```

```
enc=enkripsi(publik,message)
print("Kode Ciphertext : ",enc)
kunci = int (input("Input kunci privat : "))
cipher = input ("Input kode Ciphertext :")
dec =  dekripsi(privat,cipher)
print("Plaintextnya :",dec,("\n"))
```

SCRIPT STEGANOGRAFI MSB

```
raw_input=input
from PIL import Image
class MSB():
    def __init__(self, img):
        self.image = img
        self.width, self.height = img.size
        self.size = self.width * self.height
        self.maskone = 128
        self.maskzero = 127
#Membaca Bitdata
    def AmbilBit(self, data):
        new = []
        for i in data:
            new.append(format(ord(i), '08b'))
        return new
    #get pixel dari gambar
    def EncodeGambar(self, img, data):
        datalist = self.AmbilBit(data)
        datalen = len(datalist)
        if 8 *datalen > self.size:
            raise ValueError('data terlalu besar')
        imdata = iter(img.getdata())
```

```
        pix = []
        for i in range(8*datalen):
            pix.append(imdata.__next__())
        pix1 = []
        for i in range(datalen):
            for j in range(0, 8):
                if(datalist[i][j] == '0'):
                    pix1.append(pix[j] & self.maskzero)
                else:
                    pix1.append(pix[j] | self.maskone)
        return pix1
```

#konversi pesan

```
    def encodePesan(self, img, data):
        (x, y) = (0, 0)
        for Pix_val in self.EncodeGambar(img, data):
            img.putpixel((x,y), Pix_val)
            if( x == self.width - 1):
                x = 0
                y += 1
            else:
                x += 1
```

#Proses Encode(Penyisipan)

```
    def encode(self, img):
        data = raw_input("Input Kode Ciphertext : ")
        if(len(data) == 0):
            raise ValueError('data kosong')
                newimg = img.copy()
        self.encodePesan(newimg, data)
        new_img_name = input("Simpan stego image dengan ekstensi :")
```

```
        newimg.save(new_img_name, str(new_img_name.split(".")[1]))
```

#Proses Decode(Ekstraksi)

```
    def decode(self):
        img = input("Input stego image dan ekstensinya :")
        image = Image.open(img, 'r')
        data = ''
        imgdata = iter(image.getdata())
        pixel = []
        while True:
            try:
                pixel.append(imgdata.__next__())
            except StopIteration:
                break
            binstr = ''
        for i in pixel:
            if(i & 128 == 0):
                binstr += '0'
            else:
                binstr += '1'
        data2 = []
        for i in range(1000):
            data2.append(binstr[i * 8:(8 + i * 8)])
            data += chr(int(data2[i], 2))
        return data
```

#IMPORT FROM MSB

```
    from MSB import MSB
   from PIL import Image
w="\nPROSES STEGANOGRAFI MSB\n"
print('\n'.join('{:^40}'.format(s) for s in w.split('\n')))
```

```
imgg = input("input Cover Image dan ekstensinya : ")
img1 = Image.open(imgg, 'r').convert('L')
x = MSB(img1)
x.encode(img1)
#img2 = input("Input stego image dan ekstensinya :")
#imag = Image.open(img2, 'r')
x.decode()
("\n")
```

SCRIPT PENGUJIAN METODE MSB

```
from math import log10, sqrt
import cv2
import numpy as np
K="\nPENGUJIAN METODE (MSE DAN PSNR) UNTUK MSB\n"
print('\n'.join('{:^55}'.format(s) for s in K.split('\n')))
def PSNR(cover, stego):
mse = np.square(np.subtract(cover, stego)).mean()
if(mse == 0):
return 100
max_pixel = 255.0
psnr = 20 * log10(max_pixel/sqrt(mse))
return psnr
def MSE(cover, stego):
    Y = np.square(np.subtract(cover, stego)).mean()
    print("Nilai MSE:", Y)
img = input ("input Cover Image dan ekstensinya :")
cover = cv2.imread(img)
img1 = input ("input Stego Image dan ekstensinya :")
stego = cv2.imread(img1)
value = PSNR(cover, stego)
```

```
v= MSE(cover, stego)
print(f"Nilai PSNR :  {value} dB")
```

SCRIPT STEGANOGRAFI LSB

```
import cv2
import docopt
import numpy as np
class LSB():
    def __init__(self, im):
    self.image = im
    self.height, self.width, self.nbchannels = im.shape
    self.size = self.width*self.height
    self.maskoneValues = [1<<0, 1<<1, 1<<2, 1<<3, 1<<4, 1<<5, 1<<6, 1<<7]
    self.maskone = self.maskoneValues.pop(0) #remove nilai pertama saat sedang digunakan
    self.maskzerovalues = [255-(1<<i) for i in range(8)]
    self.maskzero = self.maskzerovalues.pop(0)
        self.width = 0  # Posisi lebar
    self.height = 0 # Posisi ketinggian
    self.curchan = 0   # Posisi saat ini
  # Berfungsi untuk menyisipkan bit ke dalam gambar
   parameter: bit nilai biner yang akan disisipkan secara sekunsial
   def put_binary_value(self, bits):
        for c in bits:  #Iterasi semua bit
      val = list(self.image[self.height,self.width]) #Dapatkan nilai piksel sebagai daftar (val sekarang menjadi array 3D)
        if int(c):
          val[self.curchan] = int(val[self.curchan])|self.maskone
        else:   #Jika bit tidak reset, reset ulang dalam gambar
          val[self.curchan] = int(val[self.curchan])&self.maskzero
        #Update image
```

```
            self.image[self.height,self.width] = tuple(val)
            self.next_slot()
#Berfungsi untuk memindahkan pointer ke pix berikutnya
        def next_slot(self):
            if self.curchan == self.nbchannels-1: #If looped over all channels
                self.curchan = 0
                if self.width == self.width-1:
                    self.width = 0
                    if self.height == self.height-1:
                        self.height = 0
                        if self.maskone == 128:
                            raise SteganographyException("gambar terisi")
                        else:
                            self.maskone = self.maskoneValues.pop(0)
                            self.maskzero = self.maskzerovalues.pop(0)
                    else:
                        self.height +=1
                else:
                    self.width +=1
            else:
                self.curchan +=1
#Berfungsi untuk membaca sedikit dari gambar, pada [tinggi, lebar] [chanael] tertentu
        def read_bit(self): #Read bit int the image
            val = self.image[self.height,self.width][self.curchan]
            val = int(val) & self.maskone
            self.next_slot()
            if val > 0:
                return "1"
            else:
```

```
            return "0"


    def read_byte(self):
        return self.read_bits(8)
    # Function read jumlah bit
    def read_bits(self, nb):
        bits = ""
        for i in range(nb):
            bits += self.read_bit()
        return bits
    #Function untuk menghasilkan nilai byte int dan mengembalikannya
    def byteValue(self, val):
        return self.binary_value(val, 8)
    #Function mengembalikan nilai biner int sebagai byte
    def binary_value(self, val, bitsize):
        #Extract binary
        binval = bin(val)[2:]
        if len(binval)>bitsize:
            raise SteganographyException("Nilai biner lebih besar dari ukuran ")
        while len(binval) < bitsize:
            binval = "0"+binval
        return binval
    def encode_text(self, txt):
        l = len(txt)
        binl = self.binary_value(l, 16)
        self.put_binary_value(binl)
        for char in txt:
            c = ord(char)
            self.put_binary_value(self.byteValue(c))
```

```
        return self.image
    def decode_text(self):
        ls = self.read_bits(16) #Read ukuran teks dalam byte
        l = int(ls,2)   #Returns decimal value
        i = 0
        unhideTxt = ""
        while i < l: #Read all bytes of the text
            tmp = self.read_byte() #So one byte
            i += 1
            unhideTxt += chr(int(tmp,2)) #Every chars concatenated to str
        return unhideTxt
ch=0
j="\nPROSES STEGANOGRAFI LSB\n"
print('\n'.join('{:^40}'.format(s) for s in j.split('\n')))
while ch!=3:
    Message = "Pilih satu : \n 1.Encode \n 2.Decode \n 3.Exit"
    print('\n'.join('{:^10}'.format(s) for s in Message.split('\n')))
    ch=int(input())
    if ch==3:
        break

    if ch==1:
        #print("Input cover image dan ekstensinya: ")
        wd=input("Input cover image dan ekstensinya: ")
        #Create object of class
        obj=LSB(cv2.imread(wd))
        #print("\n Input Kode Ciphertext : ")
        msg=input("Input Kode Ciphertext : ")
        encrypted_img=obj.encode_text(msg)
```

```
        #print("Save Stego image dan ekstensinya: ")
        dest=input("Simpan Stego image dan ekstensinya: ")
        #print("\nSaving image in destination.")
        cv2.imwrite(dest,encrypted_img)
        print("\n Encode sukses.\n")
        print("stego image tersimpan di : ",dest,("\n"))
    elif ch==2:
        #print("Input Stego Image dan ekstensinya: ")
        wd=input("\nInput Stego Image dan ekstensinya: ")
        img=cv2.imread(wd)
        obj=LSB(img)
        print("\nKode Ciphertext : ",obj.decode_text(),"\n")
print("Terima Kasih")
print ("\n")
```

SCRIPT PENGUJIAN METODE LSB

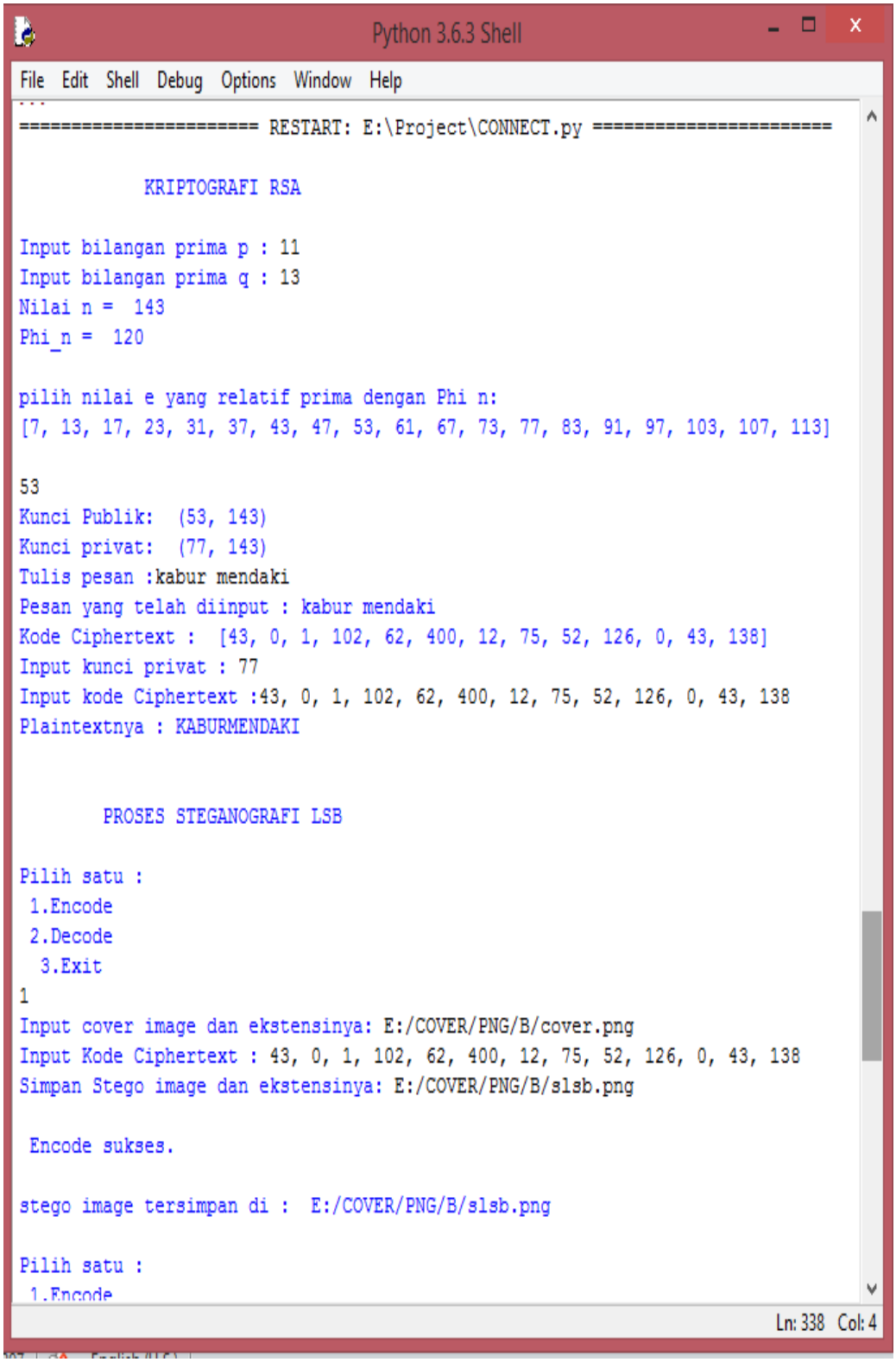
```
from math import log10, sqrt
import cv2
import numpy as np
K="\nPENGUJIAN METODE (MSE DAN PSNR) UNTUK LSB\n"
print('\n'.join('{:^55}'.format(s) for s in K.split('\n')))
def PSNR(cover, stego):
mse = np.square(np.subtract(cover, stego)).mean()
if(mse == 0):
return 100
max_pixel = 255.0
psnr = 20 * log10(max_pixel/sqrt(mse))
return psnr
def MSE(cover, stego):
    Y = np.square(np.subtract(cover, stego)).mean()
```

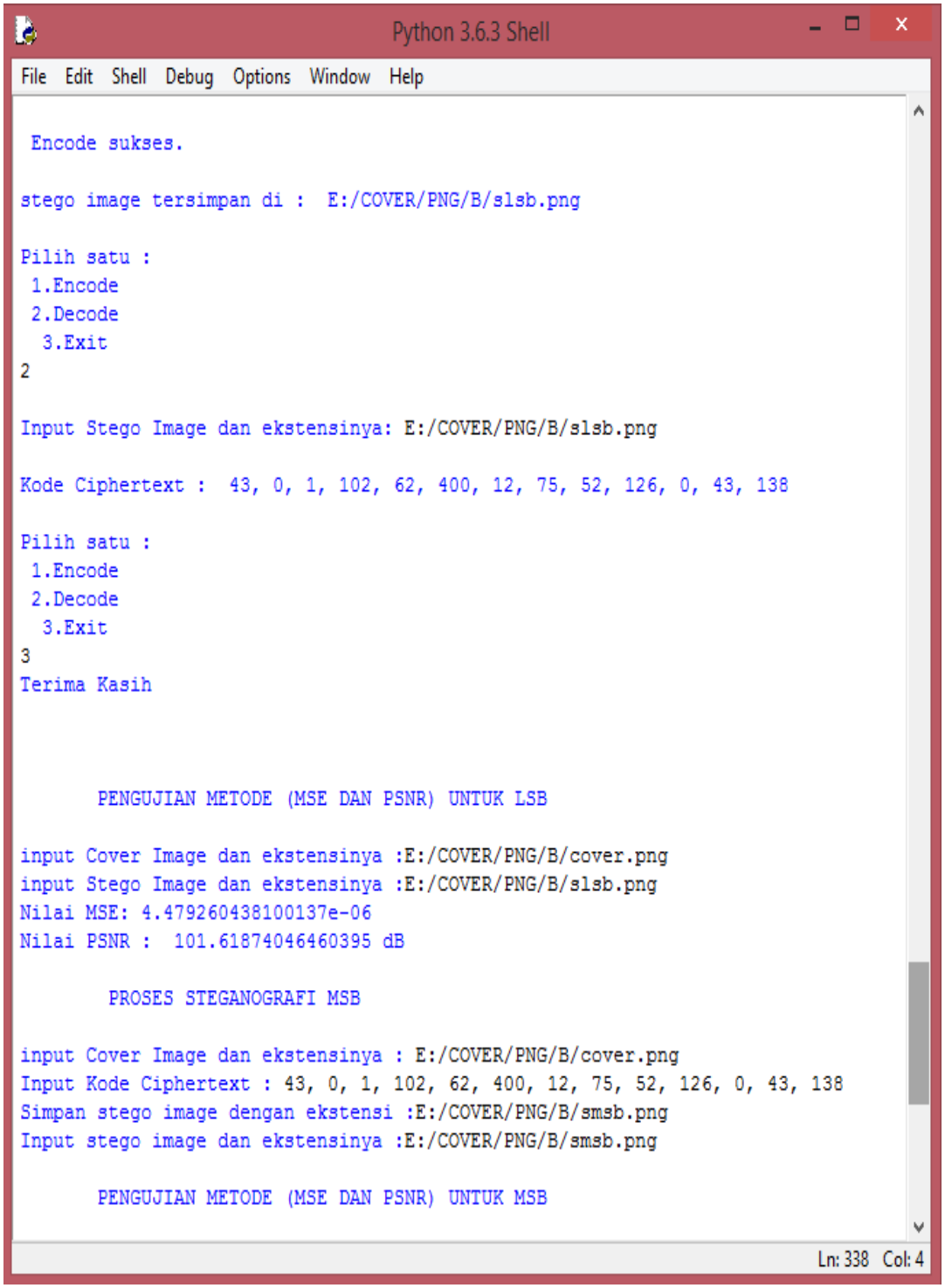
```
    print("Nilai MSE:", Y)
img = input ("input Cover Image dan ekstensinya :")
cover = cv2.imread(img)
img1 = input ("input Stego Image dan ekstensinya :")
stego = cv2.imread(img1)
value = PSNR(cover, stego)
v= MSE(cover, stego)
print(f"Nilai PSNR :  {value} dB")
```

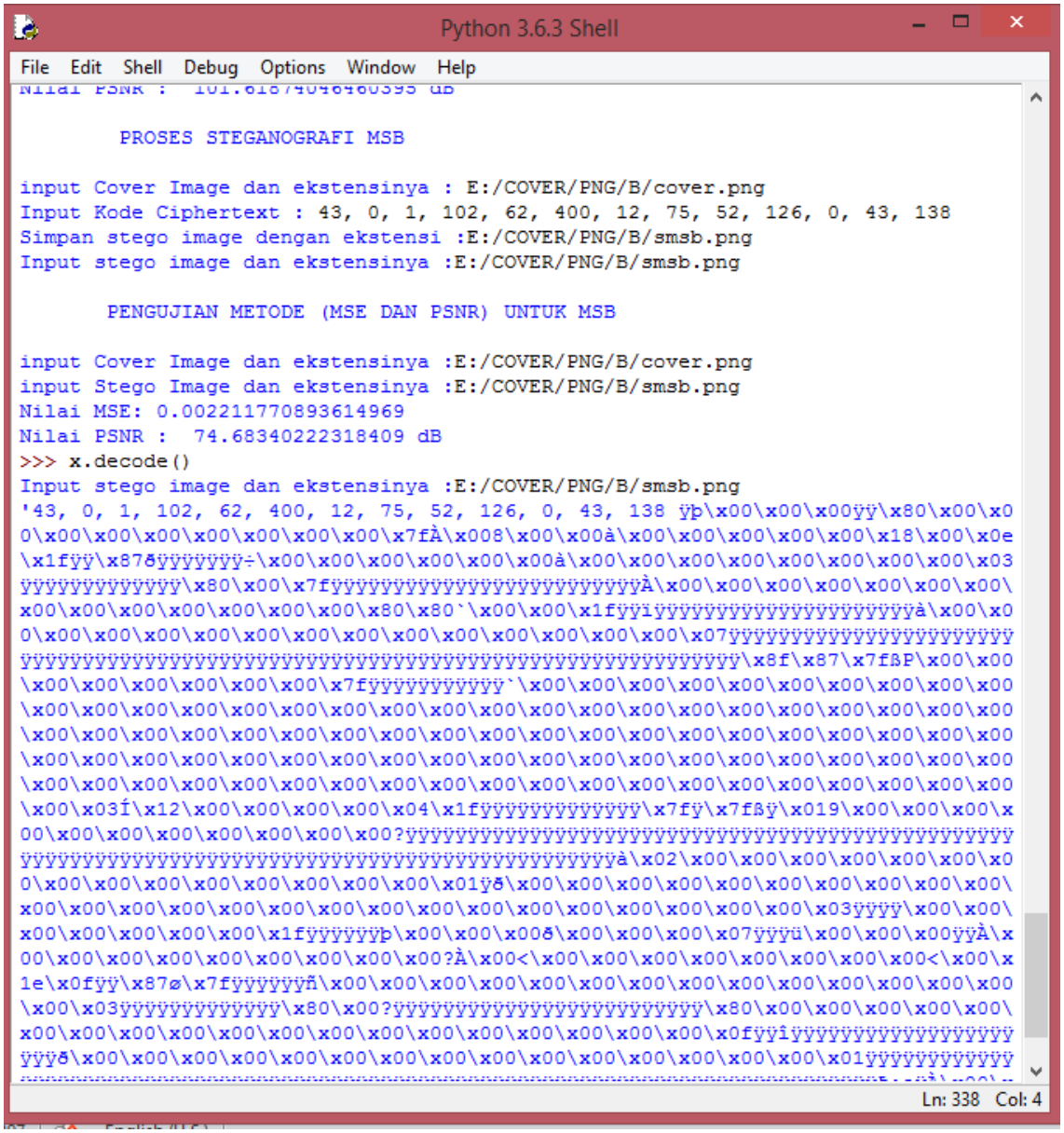
SCRIPT CONNECT ANTAR SCRIPT

```
from RSA import *
from LSB import *
from MSE import *
from main import *
from MSE1 import *
```

HASIL SCRIPT CONNECT


Python 3.6.3 Shell
File Edit Shell Debug Options Window Help
...
========== RESTART: E:\Project\CONNECT.py ==========
KRIPTOGRAFI RSA
Input bilangan prima p : 11
Input bilangan prima q : 13
Nilai n = 143
Phi_n = 120
pilih nilai e yang relatif prima dengan Phi n:
[7, 13, 17, 23, 31, 37, 43, 47, 53, 61, 67, 73, 77, 83, 91, 97, 103, 107, 113]
53
Kunci Publik: (53, 143)
Kunci privat: (77, 143)
Tulis pesan :kabur mendaki
Pesan yang telah diinput : kabur mendaki
Kode Ciphertext : [43, 0, 1, 102, 62, 400, 12, 75, 52, 126, 0, 43, 138]
Input kunci privat : 77
Input kode Ciphertext :43, 0, 1, 102, 62, 400, 12, 75, 52, 126, 0, 43, 138
Plaintextnya : KABURMENDAKI
PROSES STEGANOGRAFI LSB
Pilih satu :
1.Encode
2.Decode
3.Exit
1
Input cover image dan ekstensinya: E:/COVER/PNG/B/cover.png
Input Kode Ciphertext : 43, 0, 1, 102, 62, 400, 12, 75, 52, 126, 0, 43, 138
Simpan Stego image dan ekstensinya: E:/COVER/PNG/B/slsb.png
Encode sukses.
stego image tersimpan di : E:/COVER/PNG/B/slsb.png
Pilih satu :
1.Encode
Ln: 338 Col: 4


Python 3.6.3 Shell
File Edit Shell Debug Options Window Help
Encode sukses.
stego image tersimpan di : E:/COVER/PNG/B/slsb.png
Pilih satu :
1.Encode
2.Decode
3.Exit
2
Input Stego Image dan ekstensinya: E:/COVER/PNG/B/slsb.png
Kode Ciphertext : 43, 0, 1, 102, 62, 400, 12, 75, 52, 126, 0, 43, 138
Pilih satu :
1.Encode
2.Decode
3.Exit
3
Terima Kasih
PENGUJIAN METODE (MSE DAN PSNR) UNTUK LSB
input Cover Image dan ekstensinya :E:/COVER/PNG/B/cover.png
input Stego Image dan ekstensinya :E:/COVER/PNG/B/slsb.png
Nilai MSE: 4.479260438100137e-06
Nilai PSNR : 101.61874046460395 dB
PROSES STEGANOGRAFI MSB
input Cover Image dan ekstensinya : E:/COVER/PNG/B/cover.png
Input Kode Ciphertext : 43, 0, 1, 102, 62, 400, 12, 75, 52, 126, 0, 43, 138
Simpan stego image dengan ekstensi :E:/COVER/PNG/B/smsb.png
Input stego image dan ekstensinya :E:/COVER/PNG/B/smsb.png
PENGUJIAN METODE (MSE DAN PSNR) UNTUK MSB
Ln: 338 Col: 4


Python 3.6.3 Shell
File Edit Shell Debug Options Window Help
Nilai PSNR : 101.6187404646039 dB
PROSES STEGANOGRAFI MSB
input Cover Image dan ekstensinya : E:/COVER/PNG/B/cover.png
Input Kode Ciphertext : 43, 0, 1, 102, 62, 400, 12, 75, 52, 126, 0, 43, 138
Simpan stego image dengan ekstensi :E:/COVER/PNG/B/smsb.png
Input stego image dan ekstensinya :E:/COVER/PNG/B/smsb.png
PENGUJIAN METODE (MSE DAN PSNR) UNTUK MSB
input Cover Image dan ekstensinya :E:/COVER/PNG/B/cover.png
input Stego Image dan ekstensinya :E:/COVER/PNG/B/smsb.png
Nilai MSE: 0.002211770893614969
Nilai PSNR : 74.68340222318409 dB
>>> x.decode()
Input stego image dan ekstensinya :E:/COVER/PNG/B/smsb.png
'43, 0, 1, 102, 62, 400, 12, 75, 52, 126, 0, 43, 138
Ln: 338 Col: 4