ENCRYPTION IMAGE WITH RSA METHOD

Skripsi

Diajukan untuk Memenuhi Salah Satu Syarat

Memperoleh Gelar Sarjana Teknik

Jurusan Teknik Informatika

Di susun oleh :

Rosalia Stevania Karo

NIM : 99 5314 063

JURUSAN TEKNIK INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS SANATA DHARMA

YOGYAKARTA

2004

ENCRYPTION IMAGE WITH RSA METHOD

Skripsi

Diajukan untuk Memenuhi Salah Satu Syarat

Memperoleh Gelar Sarjana Teknik

Jurusan Teknik Informatika

Di susun oleh :

Rosalia Stevania Karo

NIM : 99 5314 063

JURUSAN TEKNIK INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS SANATA DHARMA

YOGYAKARTA

2004

PENYANDIAN CITRA DENGAN METODE RSA

Skripsi

Di susun oleh :

Rosalia Stevania Karo

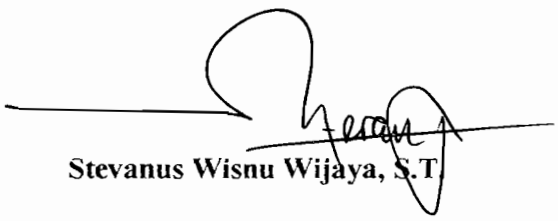
NIM : 99 5314 063

Telah disetujui oleh :

Pembimbing I

Drs J.J Siang, M.Sc. **tanggal, 10 Mei 2004**

Pembimbing II



Stevanus Wisnu Wijaya, S.T. **tanggal, 11 Mei 2004**

ENCRYPTION IMAGE WITH RSA METHOD

Skripsi

Dipersiapkan dan ditulis oleh :

Rosalia Stevania Karo

NIM : 99 5314 063

Telah dipertahankan di depan Panitia Penguji

pada tanggal, 24 Mei 2004 dan dinyatakan memenuhi syarat

Susunan Panitia Penguji

	Nama Lengkap	Tanda tangan
Ketua	Drs J.J Siang, M.Sc.	.....
Sekretaris	Stevanus Wisnu Wijaya, S.T.	.....
Anggota	JB. Budi Darmawan, S.T., M.Sc	.....
Anggota	D. S. Bambang Soelistijanto, S.T	.....

Yogyakarta, Mei 2004

Fakultas Teknik

Universitas Sanata Dharma

Yogyakarta

Dekan

(Ir. Gregorius Harjanto)

PERSEMBAHAN

Teruntuk :

Yesus sahabat setiaku dan Bunda Maria yang selalu membimbing dan menemani setiap langkah hidupku.

Bapak, ema, ka' diding, ka' idang, ka'edik, dan ka'ency yang setia mendoakan dan memberi dukungan.

And To All my Family.

Universitas Sanata Dharma, Yogyakarta.

KATA MUTIARA

- *Tak ada mawar yang tak berduri*

- *Tak ada kebahagiaan yang didapat tanpa suatu jerih payah*

KATA PENGANTAR

Puji Syukur penulis haturkan kepada Tuhan Yang Maha Baik, karena berkat dan rahmatnya penulis dapat menyelesaikan penulisan skripsi ini. Skripsi dengan judul **"Penyandian Citra Dengan Metode RSA"** disusun sebagai salah satu syarat memperoleh gelar Sarjana Teknik pada Fakultas Teknik Universitas Sanata Dharma Yogyakarta.

Banyak pihak yang telah memberi dukungan dan bantuan kepada penulis dalam menyelesaikan penulisan skripsi ini. Untuk itu pada kesempatan ini, penulis ingin menyampaikan terima kasih kepada :

1. Bapak Ir. Gregorius Harjanto sebagai Dekan Fakultas Teknik, Universitas Sanata Dharma Yogyakarta;
2. Bapak Drs J.J Siang, M.Sc, selaku dosen pembimbing I, yang telah meluangkan waktu, tenaga dan dengan penuh kesabaran serta ketelitian membimbing penulis dalam penyusunan skripsi ini.
3. Bapak Stevanus Wisnu Wijaya, S.T., selaku dosen pembimbing II, yang telah membimbing dan memberikan masukan bagi penulis dalam penyusunan skripsi ini.
4. Dewan Penguji yang telah bersedia menguji skripsi ini serta telah memberikan kemudahan dan semangat pada saat ujian berlangsung;
5. Pak Bele, Mas Danang, Mas Catur, dan Pak Dar, selaku staf laboratorium yang telah memberikan pelayanan dalam penyelesaian skripsi;

6. Dosen – dosen jurusan teknik informatika Sanata Dharma beserta semua staff, yang telah memberikan penulis pengetahuan, bimbingan, fasilitas, sehingga penulis bisa menyelesaikan penyusunan skripsi ini.
7. Bapak, ema, ka' sis, ka' ita, ka' evi, ka' rensi yang selalu setia mendoakan dan memberi dukungan baik secara moril maupun material.
8. Ka' Sius sekeluarga, Ka' Gabriel sekeluarga, Om' Thomas sekeluarga dan semua keluarga yang ada di Riung, Mbay, Poma, Marunggela terima kasih atas dukungan dan doanya.
9. Sobat – sobatku yang baik hati dan tidak sombong, Silvia, Iin, Agung, Endah, Noni, Yayuk, Retno, Rini terima kasih atas dukungan dan bantuannya.
10. Ade – adeku yang mentel Byby, Maya, Stin, Merry, Wati, Evy yang sudah banyak membantu baik material maupun moril dalam penyusunan skripsi ini.
11. Keluarga Prayan Wetan, Mas Bagas, Mba Nita, Mba Dona, Mas Gendon, Mba Ati, Mba Tanti juga buat Mas Uga kecil yang selalu setia menemani dan memberi perhatian serta dukungannya.
12. Keluarga Besar Riung yang ada di Jogya, terima kasih atas semua kerja sama dan dukungan serta kebersamaannya.
13. Semua pihak yang telah membantu yang tidak dapat disebutkan namanya satu persatu terutama teman – teman angkatan ' 99 yang telah memberi semangat dan membantu penulis dalam menyelesaikan skripsi ini.

Penulis menyadari masih ada kekurangan - kekurangan dalam penulisan skripsi ini. Untuk itu penulis mohon maaf dan siap menerima masukan, kritik dan

saran dari para pembaca. Namun, penulis juga berharap semoga skripsi ini bisa memberi manfaat bagi siapa saja yang membaca dan membutuhkan.

Yogyakarta, Mei 2004

Penulis

Ross

ABSTRAKSI

Salah satu metode yang dipakai dalam penyandian adalah metode RSA. Penggunaan metode ini merupakan salah satu alternatif untuk menjaga keamanan dan kerahasiaan suatu pesan karena ada kunci publik. Dalam tugas akhir ini, metode RSA diaplikasikan untuk mengenkripsi suatu citra yang bertipe JPEG dan BMP.

Proses penyandian citra dengan metode RSA membutuhkan beberapa masukan berupa file citra yang akan disandikan dan dua buah bilangan prima serta kunci umum e. Proses penyandian diawali dengan mengambil nilai RGB dari tiap piksel dan kemudian mengenkripsi nilai RGB dengan metode RSA. Keluarannya adalah sebuah citra yang sudah tersandikan. Citra hasil penyandian akan didekripsi atau ditransformasikan kembali ke citra asli.

Hasil yang didapat menunjukan bahwa metode RSA cocok untuk enkripsi citra. Ini terbukti karena citra hasil enkripsi dapat teracak dan citra yang sudah teracak ini dapat didekripsi atau ditransformasikan kembali ke citra asli. Namun metode RSA juga memiliki kelemahan dimana jika suatu citra dienkripsi dua kali maka citra hasil enkripsi tidak selalu dapat ditransformasikan kembali ke citra asli. Hal ini disebabkan karena jika nilai RGB dari citra asli yang akan dikenai proses lebih besar sama dengan N (hasil kali dua bilangan prima) maka nilai RGB asli harus dibagi dengan N. Data hasil pembagian ini tidak dipakai dalam proses enkripsi kedua, sehingga menyebabkan citra hasil enkripsi tidak dapat didekripsi kembali ke citra asli.

ABSTRACT

One of encryption method is RSA. This method is an alternative to security watch over and to keep a secret message because there public's key exists. In this thesis, RSA method applied to encrypt a JPEG and BMP image type.

Image coding process with RSA method want several input shaped image's file, two prime number, and public key "e". Process coding started with RGB value at every pixel and then encrypts the RGB value by RSA method. The output is an already coded image. Image coding result will decrypted or transformed to original image.

The result shows that RSA method suitable to encrypt image. Proved, image encrypt result can be precipitate or randomized and this already randomized image can decrypted or transformed to the original image. But RSA method also has weakness. If an image encrypted twice, the result of encrypt image sometimes not transformed to original image. This matter will be caused if RGB value from original image bitten the bigger or equal with N process (multiplying result two prime numbers), so the original RGB value must divided by N. This result is not used at second encrypt process. Hence this image encrypt result can not decrypted to original image.

DAFTAR ISI

HALAMAN JUDUL i

HALAMAN PERSETUJUAN ii

HALAMAN PENGESAHAN iii

HALAMAN PERSEMBAHAN iv

KATA MUTIARA v

KATA PENGANTAR vi

ABSTRAKSI ix

ABSTRACT x

DAFTAR ISI xi

DAFTAR GAMBAR xiv

BAB I PENDAHULUAN

1.1 Latar Belakang Masalah 1

1.2 Batasan Masalah 2

1.3 Tujuan dan Manfaat 2

1.4 Rumusan Masalah 3

1.5 Sistematika Penulisan 3

BAB II LANDASAN TEORI

2.1 Teknik Penyandian (Cryptography) 5

2.2 Metode RSA 7

2.2.1 Algoritma Pembangkitan Kunci 7

2.2.2 Algoritma Enkripsi 8

2.2.3 Algoritma Dekripsi 8

2.3 Konsep Citra 9

2.4 Contoh Aplikasi Penyandian Citra Secara Manual 9

BAB III PERANCANGAN

3.1 Perancangan Struktur Menu 15

3.2 Algoritma Umum Program Penyandian Citra 17

3.2.1 Proses Enkripsi 18

3.2.2 Proses Dekripsi 21

3.2.3 Algoritma Pengujian Bilangan Prima 22

3.2.4 Algoritma Euclid 24

3.2.5 Algoritma Untuk Menghitung Kunci d 25

3.2.6 Algoritma Eksponensial 26

3.3 Perancangan Antar Muka 28

3.3.1 Rancangan Tampilan Program Utama 28

3.3.2 Rancangan Input dan Output 29

3.4 Perancangan Struktur Data 32

BAB IV IMPLEMENTASI DAN ANALISA

4.1 Proses Kerja 35

4.1.1 Program Enkripsi 35

4.1.2 Program Dekripsi 40

4.1.3 Program Pengujian Kesamaan Dua Citra 44

4.2 Hasil Implementasi 45

4.2.1 Form Utama 45
4.2.2 Form Enkripsi 46
4.2.3 Form Dekripsi 48
4.2.4 Form Cek File Gambar 51
4.2.5 Form Contents 55
4.2.6 Form About Us 56
4.3 Pembahasan 57
BAB V PENUTUP
5.1 Kesimpulan 62
5.2 Saran 63
DAFTAR PUSTAKA 64
LAMPIRAN

DAFTAR GAMBAR

Gambar 2.1 Skema Teknik Penyandian Asimetris 6

Gambar 3.1 Rancangan Struktur Menu Program 16

Gambar 3.2 Flowchart Program Enkripsi 19

Gambar 3.3 Flowchart Program Dekripsi 21

Gambar 3.4 Flowchart Program Pengecekan Bilangan Prima 23

Gambar 3.5 Flowchart Program Algoritma Euclid 24

Gambar 3.6 Flowchart Program Menghitung Kunci d 25

Gambar 3.7 Flowchart Program Algoritma Eksponensial 27

Gambar 3.8 Rancangan Tampilan Program Utama 28

Gambar 3.9 Rancangan Tampilan Input dan Output Proses Enkripsi 29

Gambar 3.10 Rancangan Tampilan Input dan Output Proses Dekripsi 31

Gambar 4.1 Tampilan Form Utama 45

Gambar 4.2 Tampilan Form Enkripsi 46

Gambar 4.3 Tampilan Form Enkripsi Setelah File Gambar Dibuka 47

Gambar 4.4 Tampilan Form Hasil Proses Enkripsi 48

Gambar 4.5 Tampilan Form Dekripsi 49

Gambar 4.6 Tampilan Form Dekripsi Setelah File Gambar Dibuka 50

Gambar 4.7 Tampilan Form Hasil Proses Dekripsi 51

Gambar 4.8 Tampilan Form Cek File Citra 52

Gambar 4.9 Tampilan Form Cek File Setelah File Gambar Dibuka 53

Gambar 4.10 Tampilan Form Cek File Untuk Gambar Yang Sama 54

Gambar 4.11 Tampilan Form Cek File Untuk Gambar Yang Tidak Sama 55

Gambar 4.12 Tampilan Form Contents 56
Gambar 4.13 Tampilan Form About Us 56
Gambar 4.14 Gambar Asli 58
Gambar 4.15 Gambar Hasil Enkripsi I 58
Gambar 4.16 Gambar Hasil Enkripsi II 58
Gambar 4.17 Gambar Hasil Dekripsi I 58
Gambar 4.18 Gambar Hasil Dekripsi II 58
Gambar 4.19 Gambar Asli 60
Gambar 4.20 Gambar Hasil Enkripsi I 60
Gambar 4.21 Gambar Hasil Enkripsi II 60
Gambar 4.22 Gambar Hasil Dekripsi I 60
Gambar 4.23 Gambar Hasil Dekripsi II 60

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Dengan semakin berkembangnya teknologi komputer, memungkinkan semua pihak untuk bisa mendapatkan informasi, mendistribusikan informasi dan bahkan melakukan akses terhadap informasi. Namun masalah keamanan merupakan aspek penting dalam melindungi dan menjaga kerahasiaan informasi atau data dari pihak-pihak yang tidak bertanggung jawab baik terhadap pemalsuan, pencurian maupun pengubahan terhadap data. Ada banyak cara yang digunakan untuk melindungi dan menjaga kerahasiaan informasi atau data, salah satu cara adalah dengan teknik penyandian (*cryptography*).

Penyandian (*cryptography*) adalah teknik untuk menyamarkan atau menyandikan suatu pesan agar tidak mudah dibaca oleh orang lain. Penyandian (*cryptography*) terdiri dari dua bagian penting yaitu enkripsi dan dekripsi. Enkripsi merupakan transformasi data ke bentuk yang tidak mungkin dibaca pihak lain, sedangkan dekripsi merupakan kebalikan dari enkripsi, yaitu mengembalikan data yang ditransformasi ke bentuk semula.

Ada berbagai macam metode penyandian yang dapat digunakan untuk melindungi dan menjaga kerahasiaan serta keamanan suatu pesan. Salah satu metode yang digunakan adalah metode RSA. Penggunaan metode ini merupakan salah satu alternatif untuk menjaga keamanan dan kerahasiaan suatu pesan karena ada kunci publik. Dalam tugas akhir ini, metode RSA akan diaplikasikan untuk

mengenkripsi suatu citra atau gambar. Caranya adalah dengan mengambil nilai RGB dari tiap pixel dan kemudian mengenkripsi nilai RGB dengan metode RSA.

1.2 Batasan Masalah

Untuk membangun aplikasi penyandian citra dengan metode RSA, maka dalam proses pembuatannya terdapat beberapa batasan untuk masalah ini, yaitu :

1. Metode penyandian yang digunakan adalah metode RSA.
2. File citra yang diinputkan oleh user mempunyai format JPEG dan Bitmap (24 bit warna).
3. Nilai n (modulus) sebagai hasil kali sembarang dua bilangan prima p dan q yang juga dipakai sebagai kunci umum maupun kunci rahasia (kunci pribadi) dibatasi lebih kecil sama dengan 255.
4. Bahasa pemrograman yang digunakan adalah Borland Delphi 6.0.

1.3 Tujuan dan Manfaat

Penelitian yang dilakukan bertujuan untuk merancang dan mengimplementasikan program yang mampu menyandikan (*mengenkrip dan mendekrip*) sebuah gambar atau citra dengan metode RSA.

Beberapa manfaat yang ingin dicapai adalah :

1. Memahami dan mendalami tentang teknik penyandian citra dengan metode RSA
2. Mengetahui kekurangan dan kelebihan dari teknik penyandian citra dengan metode RSA.

1.4 Rumusan Masalah

1. Bagaimana membangun aplikasi penyandian citra dengan metode RSA ?
2. Bagaimana melihat kekurangan dan kelebihan dari aplikasi penyandian citra dengan metode RSA ?

1.5 Sistematika Penulisan

BAB I PENDAHULUAN

Membahas tentang latar belakang masalah, batasan masalah, tujuan dan manfaat, rumusan masalah, sistematika penulisan dan metodologi penelitian.

BAB II LANDASAN TEORI

Menjelaskan tentang teori-teori yang mendukung antara lain teknik penyandian (*cryptography*), metode RSA, konsep citra, dan contoh penerapan aplikasi secara manual.

BAB III PERANCANGAN

Membahas tentang perancangan struktur menu, algoritma umum program penyandian citra, perancangan antar muka (program utama, input dan output untuk proses enkripsi dan proses dekripsi), dan perancangan struktur data.

BAB IV IMPLEMENTASI DAN ANALISA

Membahas tentang proses kerja program, hasil implementasi program dan analisa dari hasil implementasi.

BAB V PENUTUP

Berisikan kesimpulan dari hasil pembuatan program dan saran.

BAB II

LANDASAN TEORI

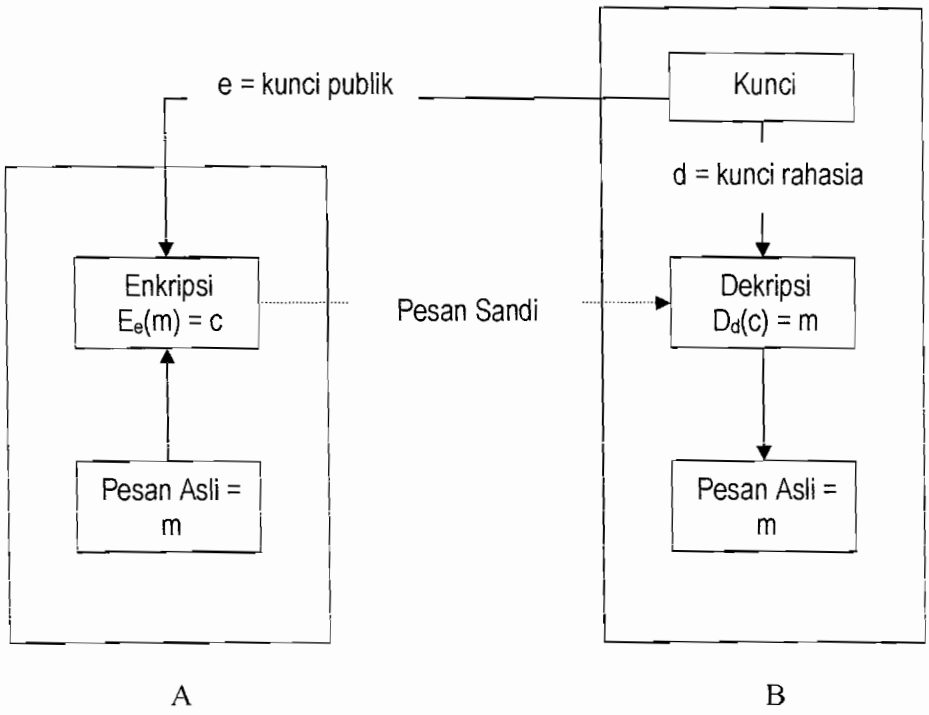
2.1 Teknik Penyandian (*Cryptography*)

Teknik penyandian (*cryptography*) adalah teknik untuk menyamarkan atau menyandikan suatu pesan agar tidak mudah dibaca oleh orang lain. Teknik penyandian terdiri dari dua bagian penting yaitu enkripsi dan dekripsi. Enkripsi adalah transformasi data ke bentuk yang tidak mungkin dibaca oleh pihak lain tanpa mengetahui kuncinya, sedangkan dekripsi yang merupakan kebalikan dari enkripsi adalah mengubah kembali data ke bentuk semula. Baik enkripsi maupun dekripsi selalu membutuhkan suatu informasi rahasia yang disebut dengan kunci.

Berdasarkan sifat kuncinya, terdapat dua jenis teknik penyandian (*cryptography*) yaitu teknik penyandian simetris (dengan kunci rahasia) dan teknik penyandian asimetris (dengan kunci publik). Dalam teknik penyandian simetris, kunci yang sama dipakai untuk melakukan enkripsi dan dekripsi sehingga baik pengirim maupun penerima informasi harus memiliki kunci yang sama untuk mengolahnya. Kelemahan dari teknik penyandian simetris (dengan kunci rahasia) adalah jika pengirim dan penerima informasi tinggal di tempat yang berjauhan maka kunci tersebut harus dikomunikasikan lewat media komunikasi seperti melalui surat, telpon, internet dan lain sebagainya yang kemungkinan akan diketahui oleh pihak lain sehingga keamanan menjadi tidak terjamin.

Untuk teknik penyandian asimetris (dengan kunci publik), penerima memiliki dua buah kunci yaitu kunci publik dan kunci rahasia. Kunci publik bisa

diketahui oleh banyak orang, tetapi kunci rahasia hanya diketahui oleh penerima saja. Bahkan pengirimpun tidak mengetahui kunci rahasia sehingga tidak bisa mendekrip kembali pesan yang telah dienkripnya. Misalkan A akan mengirim pesan m pada B. B memiliki dua buah kunci yaitu kunci publik (e) dan kunci rahasia (d). Kunci publik (e) akan dikirim pada A. Kemudian A mengenkrip pesan asli m menjadi pesan sandi c dengan kunci publik (e) dan mengirimkannya pada B. Selanjutnya B akan mendekrip pesan sandi c yang diterimanya dengan kunci rahasia (d) yang hanya diketahuinya sendiri menjadi pesan asli m. Keadaan ini dapat digambarkan dalam gambar 2.1



Gambar 2.1 : Skema teknik penyandian asimetris (dengan kunci publik)

Keuntungan dari teknik penyandian asimetris (dengan kunci publik) adalah tidak diperlukannya media komunikasi antara pengirim dan penerima informasi (misalkan, A dan B) untuk menentukan kunci rahasia sehingga keamanannya dapat lebih terjamin.

2.2 Metode RSA

Teknik penyandian asimetris (dengan kunci publik) yang paling terkenal adalah metode RSA (*Rivest, Shamir dan Adleman*). Metode ini mulai diperkenalkan pada tahun 1978 oleh ketiga penemu utamanya yaitu Ronald L. Rivest, Adi Shamir, dan Leonard Adleman (*Rivest Shamir Adleman*).

Metode RSA menggunakan dua buah kunci yaitu kunci publik dan kunci rahasia (kunci pribadi). Untuk menghasilkan kunci publik dan kunci rahasia (kunci pribadi) dibuat algoritma yang berfungsi untuk membangkitkan kunci. Kunci publik digunakan untuk melakukan proses enkripsi dan kunci rahasia (kunci pribadi) digunakan untuk melakukan proses dekripsi. Metode RSA pada dasarnya dapat dibedakan menjadi tiga bagian besar yakni algoritma pembangkitan kunci (*key generation algorithm*), algoritma enkripsi (*encryption algorithm*) dan algoritma dekripsi (*decryption algorithm*).

2.2.1 Algoritma Pembangkitan Kunci

Algoritma pembangkitan kunci berfungsi untuk menghasilkan kunci publik dan kunci rahasia yang merupakan sepasang kunci untuk satu pelaku sistem informasi. Algoritma pembangkitan kunci adalah sebagai berikut :

1. Menentukan dua buah bilangan prima yang acak dan berbeda p dan q, masing-masing berukuran sama.
2. Menghitung $n = pq$ dan $\emptyset = (p-1)(q-1)$.
3. Memilih bilangan bulat acak e yang relatif prima dengan $\emptyset$ dimana $1<e<\emptyset$, sehingga $gcd\ (e,\emptyset) = 1$.
4. Menghitung bilangan bulat unik d, sehingga $e.d = 1 \pmod{\emptyset}$.
5. Dengan demikian kunci publik adalah (n, e) dan kunci rahasia adalah (n, d).

Dalam metode RSA, bilangan e dinamakan sebagai eksponen kunci publik (*public key exponent*), d dinamakan sebagai eksponen kunci rahasia atau kunci pribadi (*private key exponent*), sedangkan n dinamakan modulus.

2.2.2 Algoritma Enkripsi

Proses enkripsi dilakukan dengan menggunakan kunci publik (n, e). Algoritma enkripsi secara matematis adalah sebagai berikut :

$$c = m^e (mod\ n)$$

dengan

c = pesan hasil enkripsi

m = pesan asli

2.2.3 Algoritma Dekripsi

Proses dekripsi dilakukan dengan menggunakan kunci rahasia (n, d).

Algoritma dekripsi secara matematis adalah sebagai berikut :

$$m = c^d (mod\ n)$$

2.3 Konsep Citra

Pada dasarnya untuk membangun sebuah gambar atau citra dibutuhkan banyak sekali titik sebagai elemen dasar penyusunnya. Titik-titik kecil yang tampak dilayar komputer disebut piksel. Piksel ini merupakan elemen terkecil dari tampilan layar yang dapat dikendalikan.

Setiap piksel yang terletak pada suatu koordinat, mempunyai nilai yang menunjukkan intensitas warna dan merupakan kombinasi dari tiga buah angka yaitu R, G, dan B yang menentukan proporsi warna merah (*Red*), hijau (*Green*) dan biru (*Blue*). R, G, dan B masing-masing memiliki range antara 0 hingga 255 sehingga jumlah warna yang dapat dipilih untuk sebuah piksel adalah 256 x 256 x 256 = 16,7 juta warna.

2.4 Contoh Aplikasi Penyandian Citra Secara Manual

Misalkan diperoleh nilai R, G, dan B sebagai berikut :

Pixel	R	G	B
(1,1)	70	215	255
(1,2)	180	25	220
(2,1)	10	250	145
(2,2)	249	55	47

Untuk menyandikan suatu citra (dengan mengandaikan nilai RGB seperti pada sampel diatas), dapat dilakukan beberapa proses berikut :

1. Proses membangkitkan atau menghasilkan kunci

- Misalkan dipilih bilangan prima $p = 13$ dan $q = 19$, maka

  $n = pq = 247$ dan

  $\emptyset = (p-1)(q-1) = 216$

- Misalkan secara acak memilih $e = 23$ (dimana e tidak memiliki faktor yang sama dengan $\emptyset = 216$).
- Menghitumg d dengan rumus :

  $e.d = 1 \pmod{\emptyset}$

  $23\ d = 1 \pmod{216}$

  didapat $d = 47$

Dengan demikian diperoleh kunci publiknya adalah $(n, e) = (247, 23)$. Sedangkan kunci rahasia atau kunci pribadi $d = 47$.

2. Proses enkripsi

   Proses enkripsi suatu citra dilakukan dengan menggunakan kunci publik $(n, e) = (247, 23)$. Selanjutnya mengambil nilai RGB dari tiap pixel (sebagai pesan m) dan disandikan dengan menggunakan rumus :

   $$c = m^e \pmod{n}$$

   Jika nilai RGB asli (pesan m) lebih besar sama dengan n maka nilai RGB asli (pesan m) akan dibagi dengan n dan hasil bagi akan disimpan dalam variable misalkan xr, xg, dan xb. Sehingga pada proses enkripsi ini akan dilakukan pengecekan apakah nilai RGB asli (pesan m) ada yang lebih besar sama dengan n (m >= n). Ternyata ada nilai RGB asli (pesan m) yang lebih besar dari n yaitu pada piksel (1,1) nilai B = 255, piksel (2,1) nilai G = 250, dan

pada piksel (2,2) nilai R = 249. Nilai RGB ini akan dibagi dengan n, sehingga didapat hasil pembagian adalah : untuk piksel (1,1) xb = 255 / 247 = 1, piksel (2,1) nilai xg = 250 / 247 = 1, dan piksel (2,2) nilai xr = 249 / 247 = 1. Data hasil pembagian ini disimpan yang nantinya akan digunakan pada proses dekripsi.

Proses enkripsi adalah sebagai berikut :

- Pixel (1,1)

  70 disandikan menjadi $(70)^{23} \pmod{247} = 242$

  215 disandikan menjadi $(215)^{23} \pmod{247} = 119$

  255 disandikan menjadi $(255)^{23} \pmod{247} = 31$

- Pixel (1,2)

  180 disandikan menjadi $(180)^{23} \pmod{247} = 149$

  25 disandikan menjadi $(25)^{23} \pmod{247} = 233$

  220 disandikan menjadi $(220)^{23} \pmod{247} = 64$

- Pixel (2,1)

  10 disandikan menjadi $(10)^{23} \pmod{247} = 212$

  250 disandikan menjadi $(250)^{23} \pmod{247} = 243$

  145 disandikan menjadi $(145)^{23} \pmod{247} = 46$

- Pixel (2,2)

  249 disandikan menjadi $(249)^{23} \pmod{247} = 241$

  55 disandikan menjadi $(55)^{23} \pmod{247} = 139$

  47 disandikan menjadi $(47)^{23} \pmod{247} = 187$

Hasil dari proses enkripsi berupa sebuah citra hasil penyandian dengan nilai RGB yang diperoleh dinyatakan dengan nilai R′G′B′ (sebagai pesan sandi c). Hasilnya adalah sebagai berikut :

Pixel	R′	G′	B′
(1,1)	242	119	31
(1,2)	149	233	64
(2,1)	212	243	46
(2,2)	241	139	187

3. Proses dekripsi

Setelah melakukan proses enkripsi maka citra hasil penyandian akan diubah kembali menjadi citra yang asli atau gambar asli. Proses dekripsi dilakukan dengan menggunakan kunci rahasia atau kunci pribadi $(n, d) = (247, 47)$. Selanjutnya mengubah kembali nilai R′G′B′ (sebagai pesan c) dari tiap pixel menjadi nilai RGB semula (sebagai pesan m) dengan menggunakan rumus :

$$m1 = c^d (mod\ n)$$

Proses dekripsi adalah sebagai berikut :

- Pixel (1,1)

  242 didekrip menjadi $(242)^{47} \pmod{247} = 70$

  119 didekrip menjadi $(119)^{47} \pmod{247} = 215$

  31 didekrip menjadi $(31)^{47} \pmod{247} = 8$

- Pixel (1,2)

  149 didekrip menjadi $(149)^{47} \pmod{247} = 180$

233 didekrip menjadi $(233)^{47} \pmod{247} = 25$

64 didekrip menjadi $(64)^{47} \pmod{247} = 220$

- Pixel (2,1)

212 didekrip menjadi $(212)^{47} \pmod{247} = 10$

243 didekrip menjadi $(243)^{47} \pmod{247} = 3$

46 didekrip menjadi $(46)^{47} \pmod{247} = 145$

- Pixel (2,2)

241 didekrip menjadi $(241)^{47} \pmod{247} = 2$

139 didekrip menjadi $(139)^{47} \pmod{247} = 55$

187 didekrip menjadi $(187)^{47} \pmod{247} = 47$

Hasil dari proses dekripsi adalah sebagai berikut :

Pixel	R1	G1	B1
(1,1)	70	215	8
(1,2)	180	25	220
(2,1)	10	3	145
(2,2)	2	55	47

Nilai RGB hasil dekripsi ada yang tidak sama dengan nilai RGB asli. Dari table hasil proses dekripsi terlihat bahwa pada piksel (1,1) nilai B1 = 8 tidak sama dengan B = 255, piksel (2,1) nilai G1 = 3 tidak sama dengan G = 250, dan piksel (2,2) nilai R1 = 2 tidak sama dengan R = 249. Agar nilai RGB hasil dekripsi sama dengan nilai RGB asli maka berlaku rumus $m = m1 + x * n$ dengan x adalah xr atau xg atau xb.

Untuk piksel (1,1) nilai B = B1 + xb * n = 8 + 1 * 247 = 255

piksel (2,1) nilai G = G1 + xg * n = 3 + 1 * 247 = 250

piksel (2,2) nilai R = R1 + xr * n = 2 + 1 * 247 = 249

Sehingga diperoleh nilai RGB hasil proses dekripsi sama dengan nilai RGB asli. Hasil proses dekripsi setelah diberlakukan rumus $m = m1 + x * n$ adalah :

Pixel	R1	G1	B1
(1,1)	70	215	255
(1,2)	180	25	220
(2,1)	10	250	145
(2,2)	249	55	47

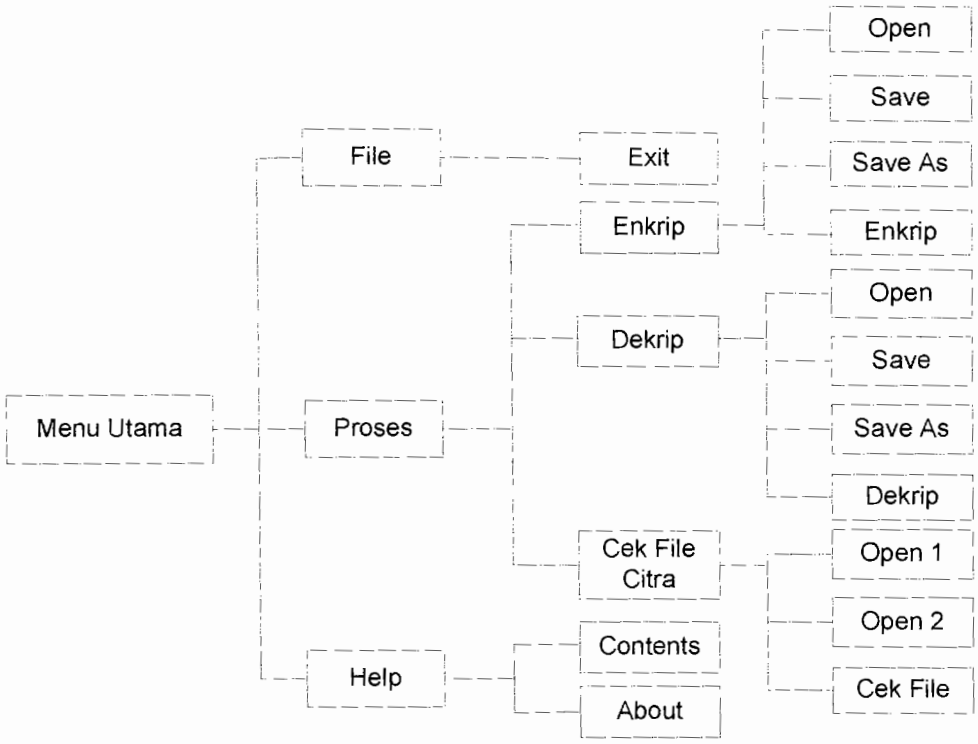
BAB III

PERANCANGAN

Pada bab ini akan dijelaskan mengenai perancangan yang meliputi perancangan struktur menu, algoritma umum program penyandian citra, perancangan antar muka (program utama, input dan output untuk proses enkripsi atau dekripsi), perancangan struktur data yang diperlukan untuk pembuatan program.

3.1 Perancangan Struktur Menu

Dalam program penyandian citra terdapat tiga menu utama yaitu menu File, menu Proses dan menu Help. Menu File terdiri dari satu buah submenu yaitu menu Exit. Menu Proses terdiri dari tiga buah submenu yaitu menu Enkripsi, Dekripsi dan Cek File Citra. Pada submenu enkripsi dan dekripsi terdapat lagi empat buah menu pilihan yaitu menu Open, Enkrip / Dekrip, Save dan Save As. Submenu Cek File Citra terdapat tiga buah menu pilihan yaitu menu Open1, Open 2 dan Cek File. Sedangkan menu Help berisi submenu Contents dan About. Bentuk perancangan struktur menu adalah sebagai berikut :



Gambar 3.1 : Rancangan Struktur Menu Program

Keterangan :

1. File = Main Menu File, terdiri dari satu buah submenu yaitu :
   - Exit, untuk keluar dari program penyandian citra.
2. Proses = Main Menu Proses, untuk melakukan proses penyandian citra. Pada main menu ini terdapat tiga buah submenu yaitu :
   - Enkripsi, untuk melakukan proses enkripsi citra.
   - Dekripsi, untuk melakukan proses dekripsi citra.
   - Cek file citra, untuk menguji kesamaan dua citra atau gambar.

   Pada menu enkripsi dan dekripsi masih terdapat empat buah submenu yaitu :

   1. Open, untuk membuka file citra yang akan dikenakan proses dekripsi.

2. Save, untuk menyimpan file citra yang telah dibuka atau yang telah di proses dengan nama yang sama.
3. Save As, untuk menyimpan file citra yang telah dibuka atau yang telah di proses dengan nama yang sama atau dengan nama yang lain.
4. Enkrip / Dekrip, untuk melakukan proses enkripsi atau dekripsi citra.

Pada menu cek file citra terdapat tiga buah menu pilihan yaitu :

1. Open 1, untuk membuka file citra atau gambar pertama.
2. Open 2, untuk membuka file citra atau gambar kedua.
3. Cek File, untuk melakukan proses pengujian kesamaan dua file citra atau gambar.

3. Help = Main Menu Help, untuk membantu pemakai dalam menjalankan program penyandian citra. Main menu Help berisi dua buah submenu yaitu :
   - Contents, menjelaskan sekilas tentang program penyandian citra dan bagaimana cara menjalankannya.
   - About, berisi keterangan tentang nama pembuat program.

3.2 Algoritma Umum Program Penyandian Citra

Langkah 1 : Tentukan proses enkripsi atau dekripsi yang akan dikerjakan.

Langkah 2 : Baca file citra sesuai dengan pilihan proses pada langkah 1.

Langkah 3 : Inputkan dua buah bilangan prima yang berbeda secara acak, juga data kunci sesuai dengan pilihan proses pada langkah 1.

Langkah 4 : Lakukan proses sesuai dengan pilihan proses pada langkah 1.

Langkah 5 : Tampil hasil proses dan simpan hasilnya.

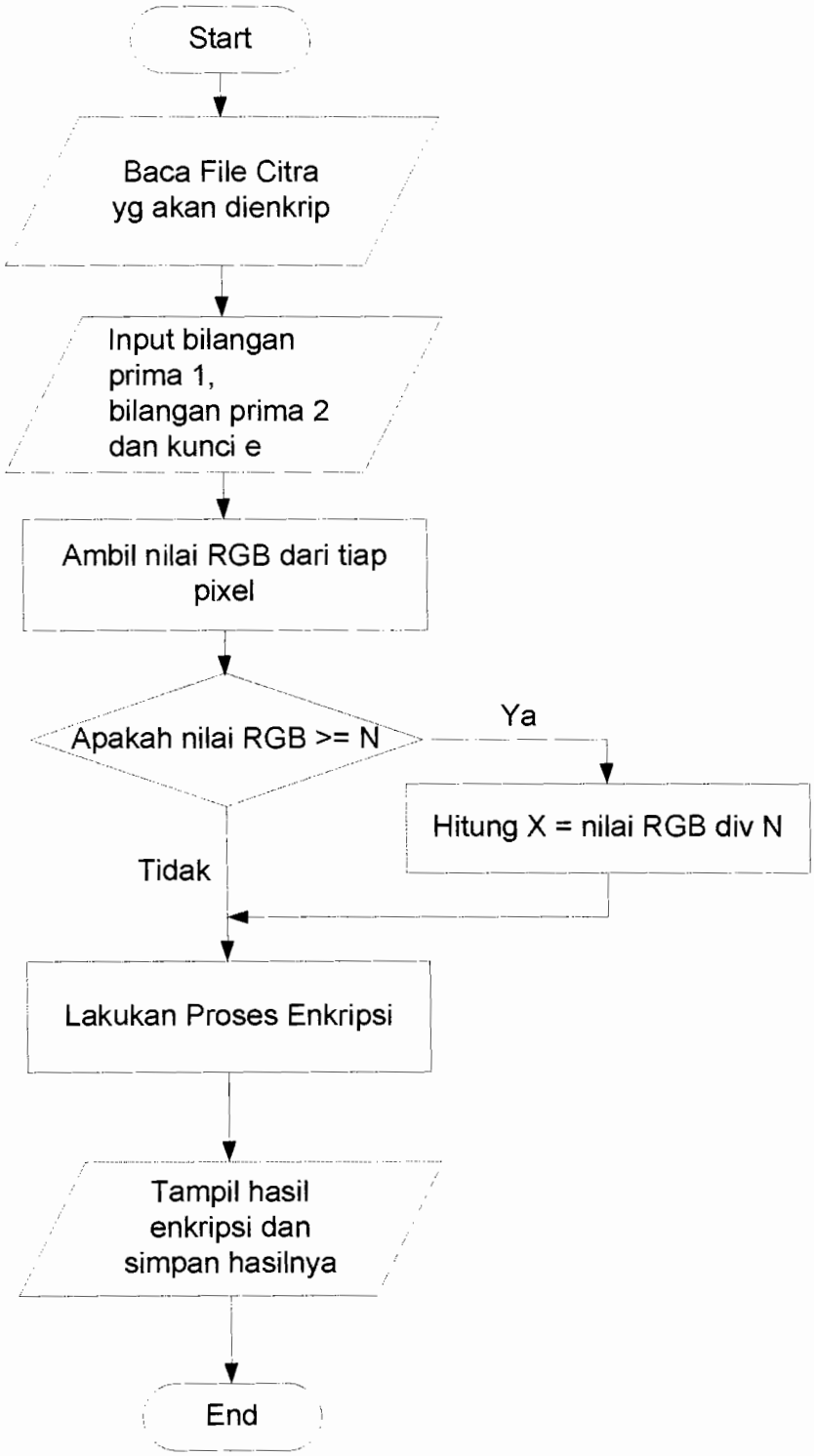
Langkah 6 : Selesai.

Berdasarkan algoritma umum program yang merupakan urutan kerja dari proses penyandian citra secara menyeluruh maka untuk langkah 2 sampai dengan langkah 6 proses kerjanya akan dijalankan sesuai dengan pilihan proses pada langkah 1. Dan pada langkah 2, baik proses enkripsi maupun dekripsi membutuhkan algoritma untuk membangkitkan kunci, seperti yang telah disebutkan pada bab sebelumnya (subbab 2.2.1 : Algoritma pembangkitan kunci). Untuk menghitung kunci ada beberapa algoritma yang digunakan yaitu algoritma pengujian bilangan prima, algoritma euclid, dan algoritma untuk menghitung kunci d. Dan pada langkah 4 untuk melakukan proses enkripsi dan proses dekripsi dibutuhkan algoritma eksponensial.

Proses kerja program penyandian citra akan diawali dengan menentukan proses enkripsi atau dekripsi yang akan dikerjakan.

3.2.1 Proses Enkripsi

Jika yang dipilih proses enkripsi maka flowchart program proses enkripsi citra atau gambar adalah sebagai berikut :



Gambar 3.2 : Flowchart Program Enkripsi

Langkah awal dari proses enkripsi adalah baca file citra yang akan dienkripsi. Selanjutnya menginputkan dua buah bilangan prima serta kunci umum e. Setelah semua data diinputkan, proses selanjutnya adalah mengambil nilai RGB dari tiap pixel. Dalam metode RSA, jika pesan asli lebih besar sama dengan kunci umum N maka hasil dekripsi tidak dapat ditransformasikan ke bentuk semula atau pesan hasil dekripsi tidak sama dengan pesan aslinya. Dan ini berlaku juga dalam proses penyandian citra atau gambar. Agar citra atau gambar hasil dekripsi sama dengan citra atau gambar asli maka dibuat rumus :

$$M = M1 + X * N$$

dengan M = pesan asli

M1= pesan hasil dekripsi

X = data hasil pembagian = nilai RGB div N

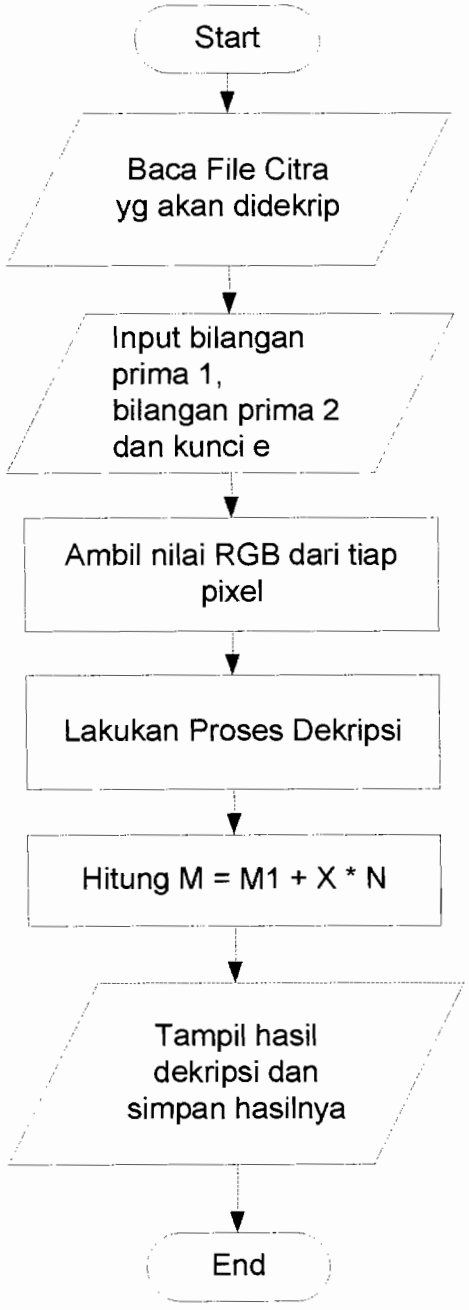
N = nilai kunci N (hasil kali prima1 dan prima2)

Sehingga dalam proses enkripsi ini akan dilakukan pengecekan, apakah nilai RGB dari gambar asli lebih besar sama dengan kunci N (hasil kali dua bilangan prima). Jika ya maka nilai RGB akan dibagi dengan N dan data hasil pembagian akan disimpan dalam suatu file yang nantinya akan digunakan dalam proses dekripsi. Jika tidak maka nilai RGB dari gambar asli akan langsung dikenai proses enkripsi. Dalam hal ini misalkan X berisi data hasil pembagian.

Output dari proses enkripsi berupa sebuah citra (gambar) yang sudah tersandikan dengan nilai RGB yang diperoleh berbeda dengan nilai RGB awal. Citra (gambar) hasil proses enkripsi ini kemudian disimpan dalam bentuk file. Selain menyimpan file gambar juga akan disimpan data hasil pembagian jika nilai RGB lebih besar sama dengan N.

3.2.2 Proses Dekripsi

Jika yang dipilih proses dekripsi maka flowchart program proses dekripsi citra adalah sebagai berikut :



dengan
M = RGB Asli
M1= RGB hasil Dekripsi
X = data hasil pembagian
N = hasil kali dua bil. prima

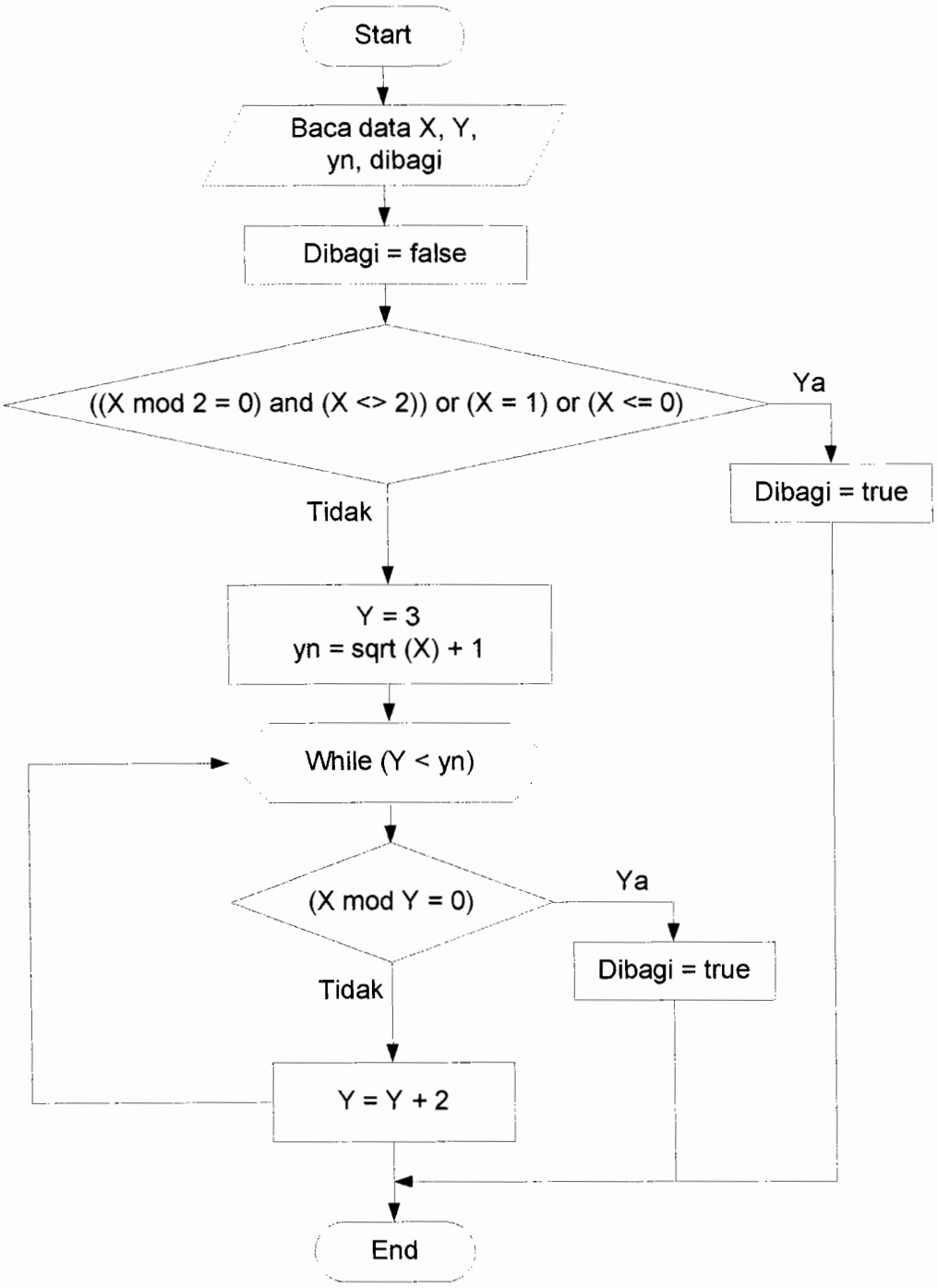
Gambar 3.3 : Flowchart Program Dekripsi

Langkah awal dari proses dekripsi adalah baca file citra (file citra sebagai hasil proses enkripsi) yang akan didekripsi. Selanjutnya menginputkan dua buah bilangan prima serta kunci umum e. Setelah semua data diinputkan, proses selanjutnya adalah mengambil nilai RGB dari tiap pixel pada gambar hasil enkripsi dan kemudian didekrip (ditransformasikan ke bentuk semula). Proses selanjutnya mengecek apakah ada data hasil pembagian yang diperoleh saat proses enkripsi. Jika ada maka rumus M = M1 + X * N digunakan pada proses dekripsi ini. Sehingga nilai R, G, dan B hasil proses dekripsi sama dengan nilai R, G, dan B asli atau citra hasil proses dekripsi sama dengan citra asli.

Output dari proses dekripsi berupa sebuah citra (gambar) asli dengan nilai RGB yang diperoleh adalah nilai RGB awal. Citra (gambar) hasil proses dekripsi ini kemudian disimpan dalam bentuk file.

3.2.3 Algoritma Pengujian Bilangan Prima

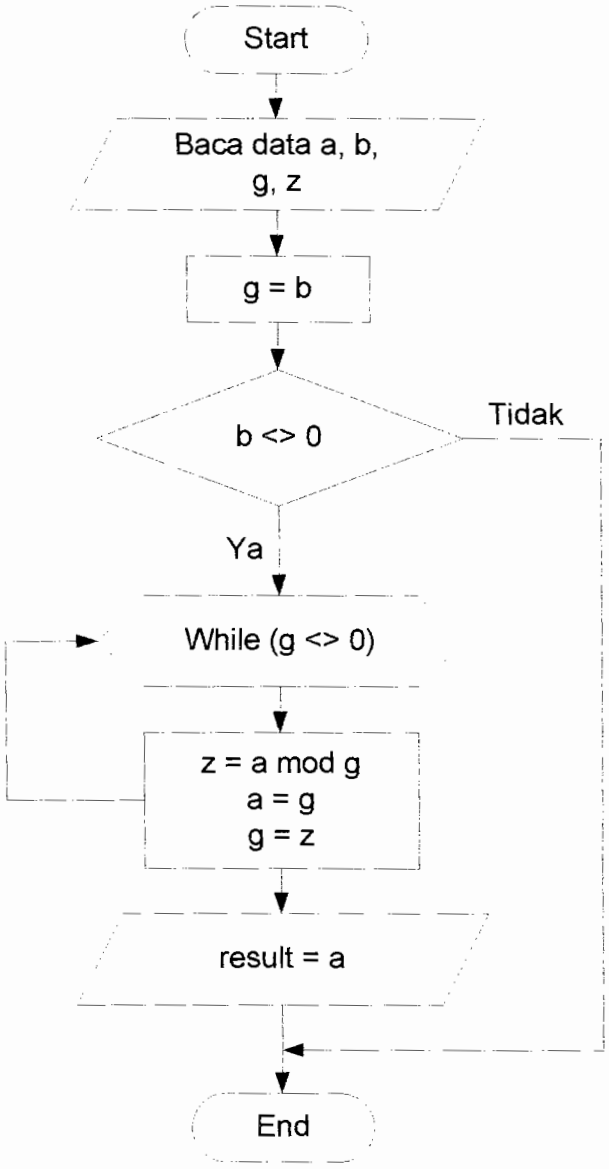
Algoritma ini berfungsi untuk menguji apakah suatu bilangan merupakan bilangan prima. Inputannya berupa sebuah bilangan bulat yang dimisalkan dengan X. Sedangkan outputnya bertipe boolean dimana jika dibagi = false berarti X adalah bilangan prima dan jika dibagi = true berarti X adalah bukan bilangan prima. Flowchart programnya adalah sebagai berikut :



Gambar 3.4 : Flowchart Program Pengecekan Bilangan Prima

3.2.4 Algoritma Euclid

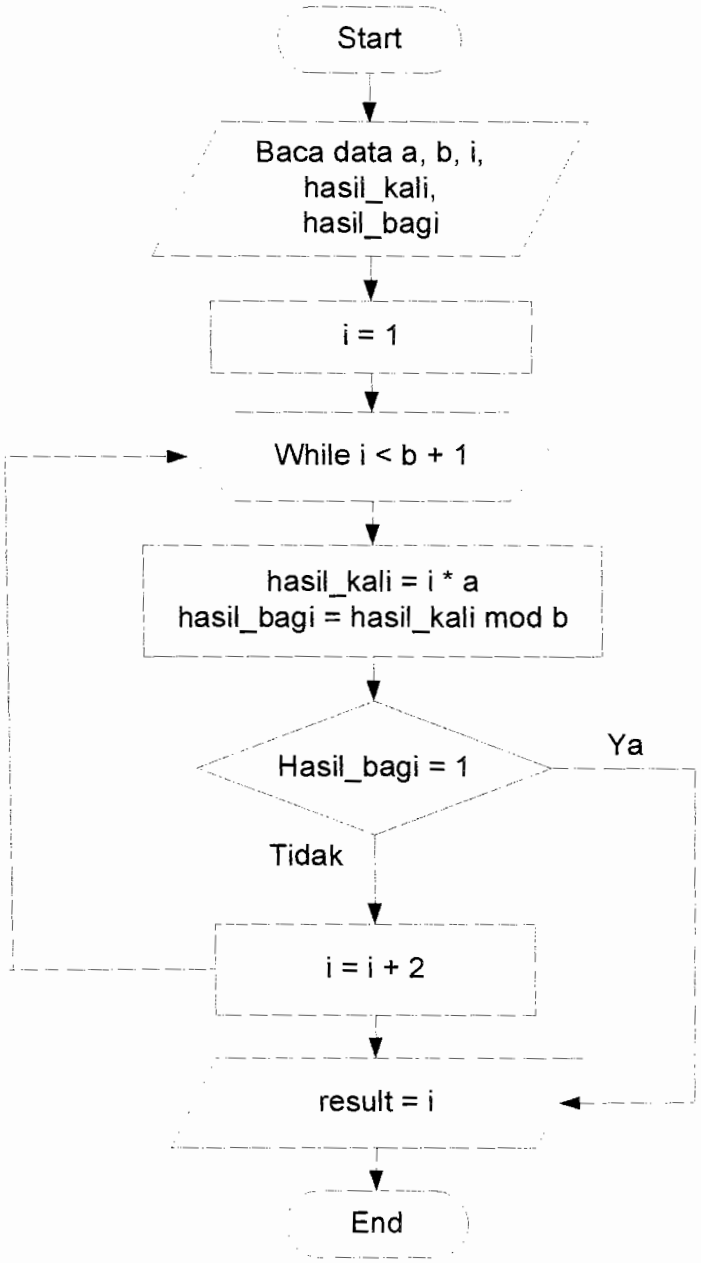
Algoritma ini dipakai untuk menguji apakah bilangan a relatif prima terhadap b. Inputannya berupa dua buah bilangan bulat positif misalkan a dan b. Jika hasilnya 1, berarti a relatif prima terhadap b. Flowchart programnya adalah sebagai berikut :



Gambar 3.5 : Flowchart Program Algoritma Euclid

3.2.5 Algoritma Untuk Menghitung Kunci d

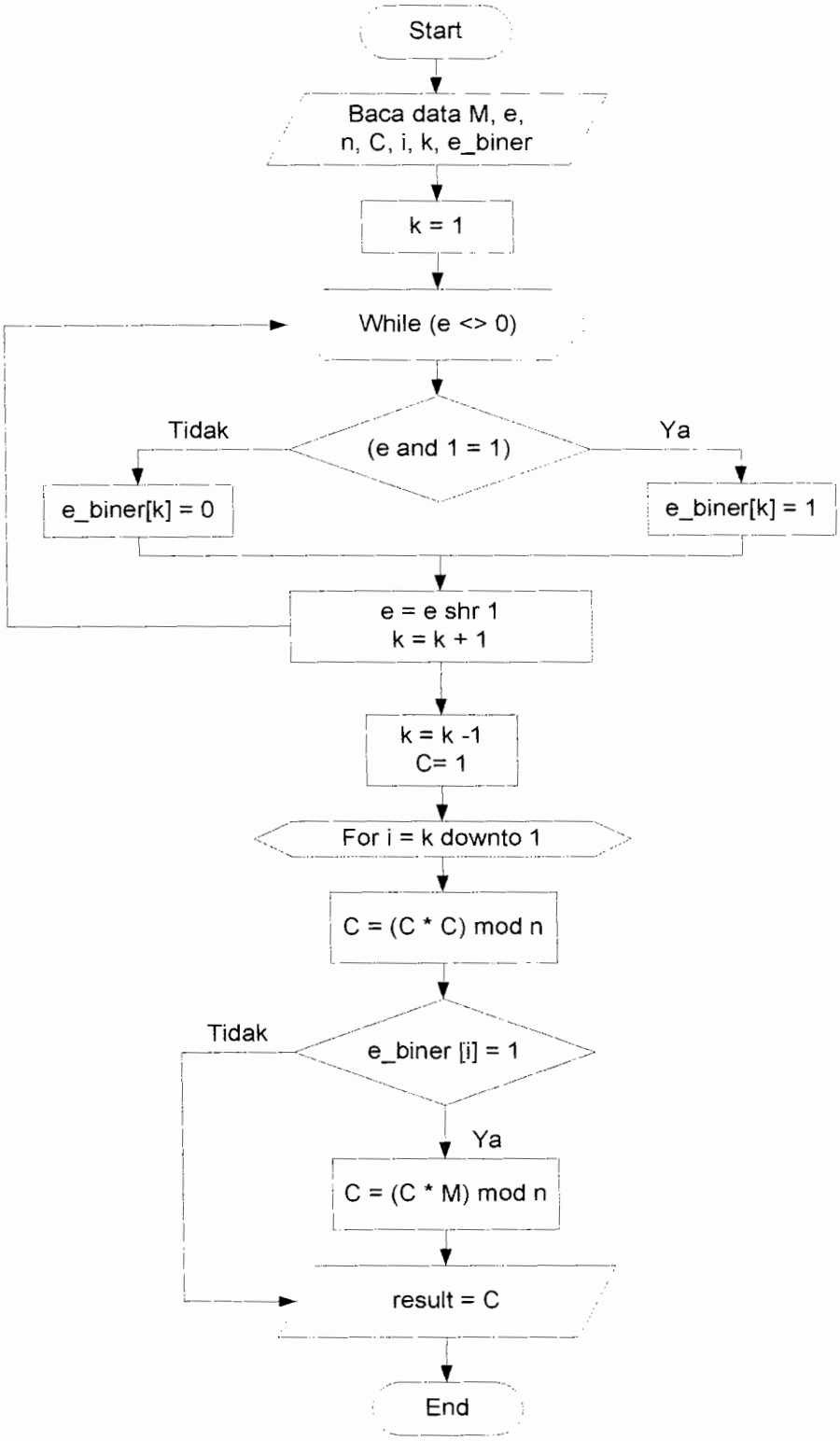
Algoritma ini digunakan untuk mengitung kunci d. Kunci d dihitung berdasarkan kunci e yang diinputkan serta hasil kali bilangan prima1 dikurangi 1 dan bilangan prima 2 dikurangi 1. Flowchart programnya adalah sebagai berikut :



Gambar 3.6 : Flowchart Program Menghitung Kunci d

3.2.6 Algoritma Eksponensial

Dalam metode RSA untuk proses kerja pada langkah 4, baik proses enkripsi maupun proses dekripsi untuk perhitungan $c = m^e \bmod n$ dan $m = c^d \bmod n$ tidak perlu menghitung hasil pemangkatan yang sesungguhnya karena mungkin akan mendapatkan hasil yang sangat besar. Maka dalam proses perhitungan dapat menggunakan algoritma eksponensial. Terdapat tiga buah inputan data misalkan M, e, n. Nilai e akan dipresentasikan dalam notasi biner. Output dari algoritma ini berupa data yang sudah terenkripsi atau terdekripsi yang disimpan dalam variable C. Flowchart programnya adalah sebagai berikut :



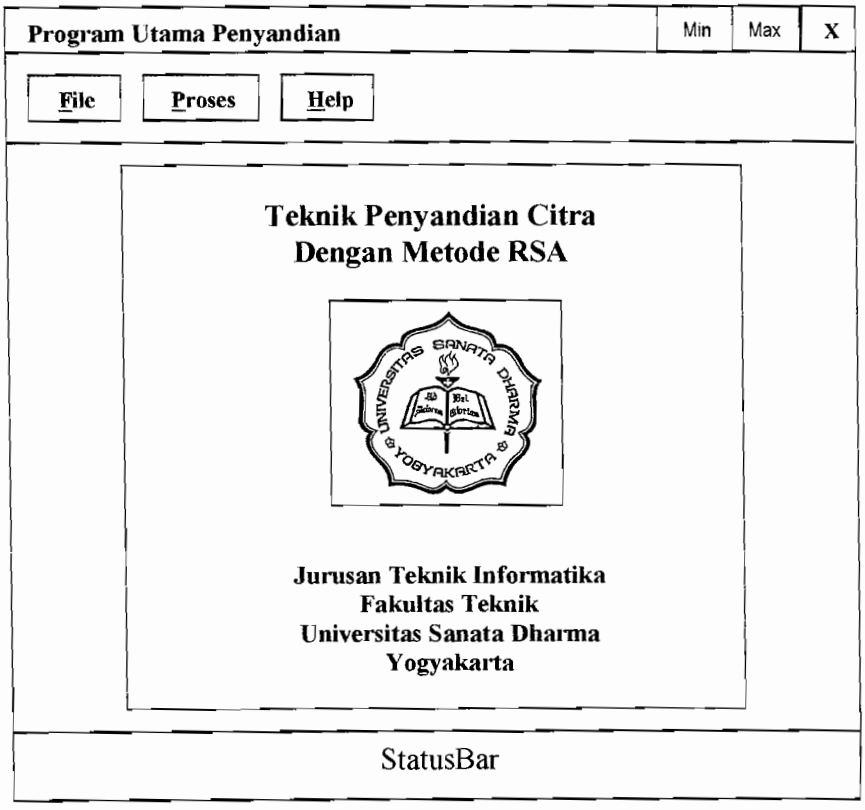
Gambar 3.7 : Flowchart Program Algoritma Eksponensial

3.3 Perancangan Antar Muka

Bagian ini akan menjelaskan tentang perancangan antar muka dari program penyandian citra dengan metode RSA yang meliputi perancangan tampilan utama program, perancangan input dan output dari proses enkripsi dan proses dekripsi.

3.3.1 Rancangan Tampilan Program Utama

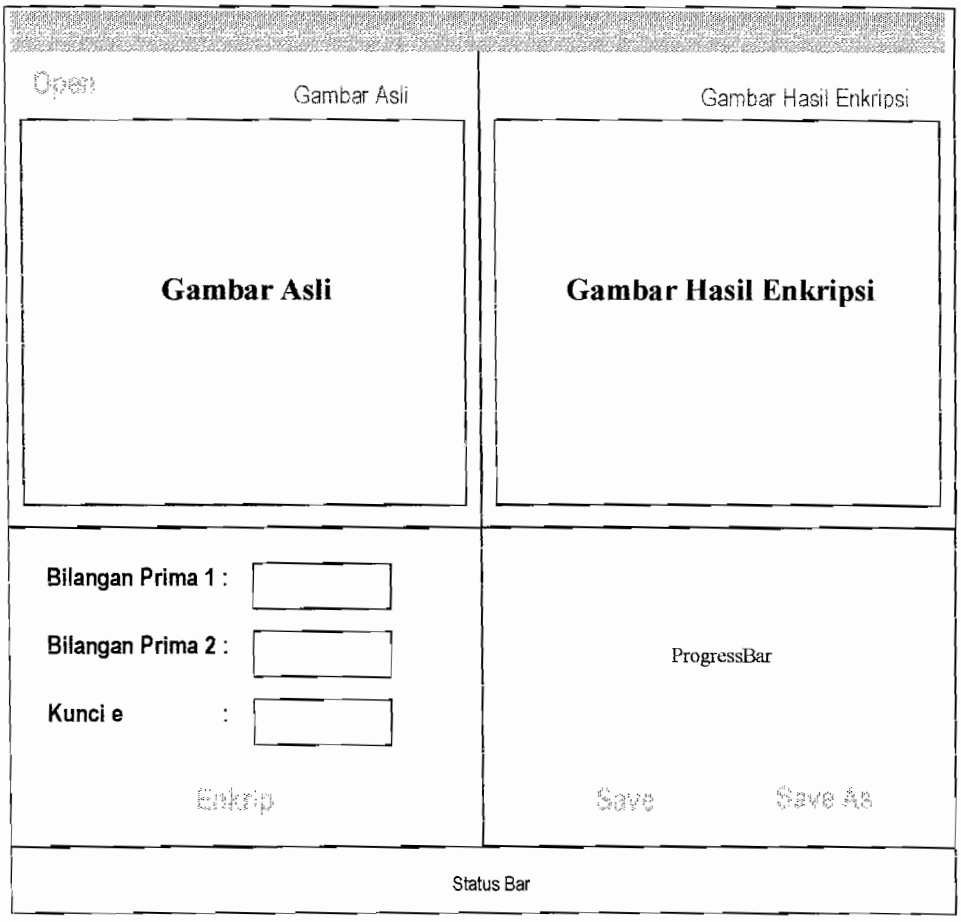
Saat program dijalankan maka akan muncul tampilan form utama program. Bentuk rancangan tampilannya adalah sebagai berikut :



Gambar 3.8 : Rancangan Tampilan Program Utama

3.3.2 Rancangan Input dan Output

Proses penyandian citra dengan metode RSA terdiri dari dua proses yaitu proses enkripsi dan proses dekripsi. Proses enkripsi membutuhkan masukkan berupa sebuah file citra yang akan dienkrip dan dua buah bilangan prima serta kunci umum e. Dua buah bilangan prima dan kunci e akan diinputkan oleh user melalui keyboard. Bentuk rancangan tampilan input dan output pada proses enkripsi adalah sebagai berikut :



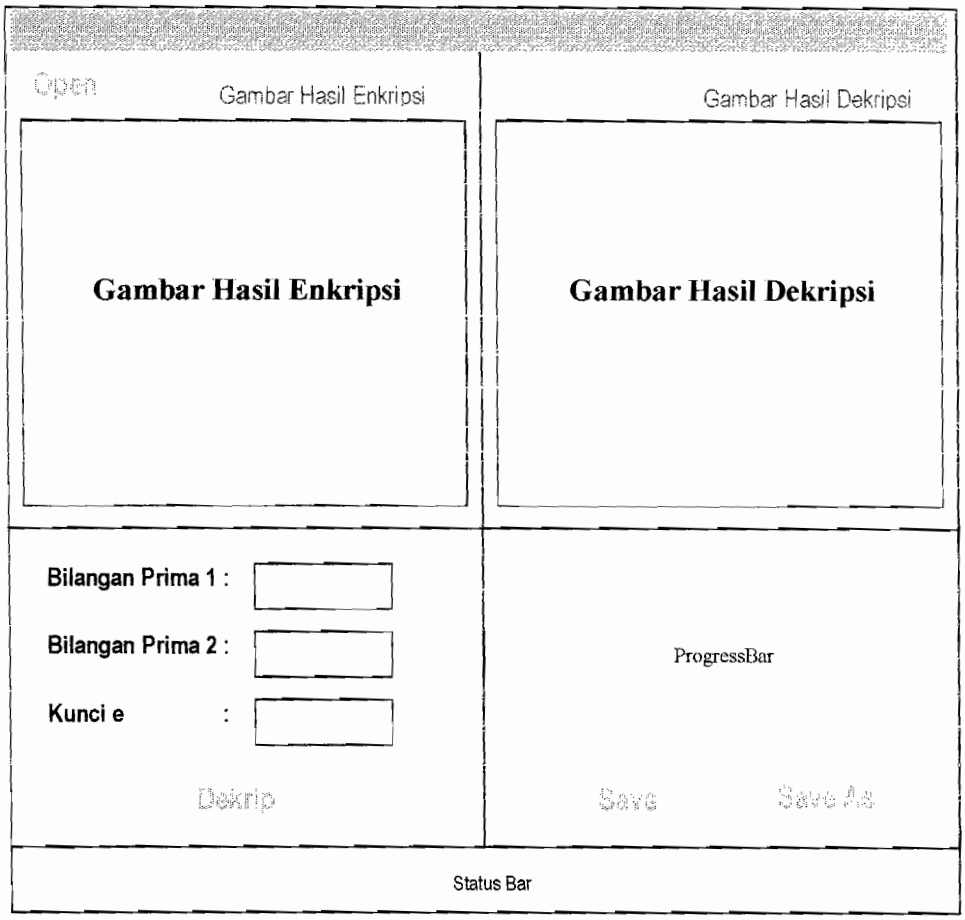
Gambar 3.9 : Rancangan Tampilan Input dan Output Proses Enkripsi

Keterangan :

Gambar asli	:	gambar yang akan dienkripsi (sebagai inputan data pada proses enkripsi).
Gambar Hasil Enkripsi	:	gambar hasil enkripsi (sebagai output pada proses enkripsi).
Bil. Prima 1 dan 2	:	sebagai inputan data pada proses enkripsi.
Kunci e	:	inputan data kunci umum untuk proses enkripsi.

Keluaran (output) dari proses enkripsi adalah sebuah gambar yang sudah tersandikan. Gambar hasil enkripsi ini akan disimpan dalam bentuk file yang nantinya akan digunakan untuk proses dekripsi.

Sedangkan untuk proses dekripsi membutuhkan masukkan berupa sebuah file citra yang akan didekrip (file citra hasil enkripsi) dan dua buah bilangan prima serta kunci umum e. Namun untuk melakukan proses dekripsi menggunakan kunci pribadi d. Kunci pribadi d dapat dihitung berdasarkan kunci umum e yang telah diinputkan. Dua buah bilangan prima dan kunci e akan diinputkan oleh user melalui keyboard. Bentuk rancangan tampilan input dan output pada proses dekripsi adalah sebagai berikut :



Gambar 3.10 : Rancangan Tampilan Input dan Output Proses Dekripsi

Keterangan :

Gambar Hasil Enkripsi: gambar yang akan didekripsi (sebagai inputan data pada proses dekripsi).

Gambar Hasil Dekripsi: gambar hasil dekripsi (sebagai output pada proses dekripsi).

Bil. Prima 1 dan 2 : sebagai inputan data pada proses dekripsi.

Kunci e : inputan data kunci umum untuk proses dekripsi.

Keluaran (output) dari proses dekripsi adalah berupa sebuah gambar yang asli. Gambar hasil proses dekripsi akan disimpan dalam bentuk file. File citra hasil proses dekripsi ini berisi informasi yang sama dengan file sumbernya.

3.4 Perancangan Struktur Data

Untuk membuat program penyandian gambar atau citra dibutuhkan struktur data yaitu :

1. Tipe Data Array

Array adalah suatu tipe data terstruktur yang terdapat dalam memori yang terdiri dari sejumlah elemen (tempat) yang mempunyai tipe data yang sama dan merupakan gabungan dari beberapa variable serta memiliki jumlah komponen yang jumlahnya tetap. Elemen-elemen array tersusun secara sequential dalam memori komputer. Array yang digunakan dalam program ini adalah :

- Datawarna = Array [1..max] of byte.

  Array ini digunakan untuk menampung nilai RGB dari gambar yang telah dibuka. Nilai indeksnya dimulai dari 1 sampai dengan max (max = 10000000). Dalam program array ini digunakan pada proses enkripsi dan dekripsi gambar.

- Nilai : Array [1..max] of mydata.

  Array yang digunakan untuk menampung data hasil pembagian nilai RGB dengan kunci N jika pesan asli atau RGB dari gambar asli lebih besar sama

dengan kunci N (hasil kali dua bilangan prima) dan untuk menampung posisi pikselnya. Data – data tesebut antara lain px, py, xr, xg, dan xb.

2. Tipe Data Record

Record adalah sebuah tipe data yang digunakan untuk menyimpan sekumpulan data yang saling berhubungan. Elemen – elemen dalam record dapat mempunyai tipe data yang sama atau tipe data yang berbeda. Dalam penggunaannya tipe data record dapat digabungkan dengan tipe data array. Dalam program ini tipe record digunakan untuk menyimpan data hasil pembagian nilai RGB dari gambar yang dibuka dengan kunci N jika nilai RGB lebih besar sama dengan N dan juga untuk menyimpan posisi pikselnya. Tipe record dalam program dideklarasikan sebagai berikut :

```
mydata = record
    px, py : Word;
    xr, xg, xb : Byte;
  end;
var nilai: array [1..max] of mydata;
```

Terdapat suatu tipe data baru yaitu mydata. Pendeklarasian variable bertipe mydata sama dengan pendeklarasian variable biasa. Elemen – elemennya terdiri dari px, py yang bertipe word dan xr, xg, xb yang bertipe byte. Nilai adalah array dengan nilai indeks yang dimulai dari 1 sampai max (dengan max = 10000000) yang bertipe mydata. Untuk mengacu pada sebuah elemen record digunakan operator titik (.). Karena nilai bertipe array maka dalam pendeklarasiannya harus menyatakan nilai indeksnya dahulu yang kemudian diikuti dengan operator titik. Setelah itu menyatakan elemen yang akan

dideklarasikan. Dalam program ini variable yang menyatakan nilai index dari variable nilai adalah jlh. Contoh penulisan dalam program adalah sebagai berikut :

```
begin
  nilai[jlh].xr := datawarna[x] div N;
  nilai[jlh].xg := datawarna[x+1] div N;
  nilai[jlh].xb := datawarna[x+2] div N;
  nilai[jlh].px := i;
  nilai[jlh].py := j;
end
```

Data – data ini disimpan dalam suatu file text dan akan dibaca secara sekuensial atau berurut. Tipe record dalam program digunakan pada proses enkripsi dan dekripsi gambar.

BAB IV

IMPLEMENTASI DAN ANALISA

4.1 Proses Kerja

Program penyandian ini pada dasarnya bertujuan untuk mengenkrip dan mendekrip sebuah citra dengan metode RSA. Berdasarkan perancangan program yang telah dibuat, maka akan dibahas proses kerja dari program penyandian citra yang terdiri dari program enkripsi, program dekripsi dan program pengujian kesamaan dua citra.

4.1.1 Program Enkripsi

Proses kerja dari program enkripsi citra atau gambar adalah sebagai berikut :

1. Baca atau buka file gambar yang akan dienkrip. File gambar yang digunakan adalah file JPEG dan file Bitmap. Namun proses enkrip tidak dapat dilakukan pada file JPEG. Agar bisa dienkrip maka file JPEG akan dikonversikan ke file Bitmap. Adapun fungsi yang digunakan untuk meng-convert file JPEG ke file Bitmap. Listing programnya adalah sebagai berikut :

```
function JPEG2Bitmap(JPEGFile : String) : TBitmap;
var
   myJPEG:TJPEGImage;
begin
  myJPEG := TJPEGImage.Create ;
  try
    myJPEG.LoadFromFile(JPEGFile);
    Result := TBitmap.Create;
    myJPEG.PixelFormat:=jf24bit;
  finally
    myJPEG.Free;
  end;
end;
```

2. Input dua bilangan prima 1 dan prima 2 serta kunci e.

   Mekanisme untuk menciptakan kunci diawali dengan menginputkan dua buah bilangan. Bilangan yang diinputkan akan diuji apakah merupakan bilangan prima. Adapun fungsi yang digunakan untuk menguji bilangan prima. Fungsi ini dirancang berdasarkan algoritma yang telah disebutkan pada bab sebelumnya (subbab 3.2.3). Keluaran dari fungsi ini berupa nilai false atau true (bertipe boolean). Nilai false berarti bilangan yang diinputkan adalah bilangan prima dan nilai true berarti bilangan yang diinputkan bukan bilangan prima. Listing programnya adalah sebagai berikut :

```
Function Prima(X: longint): Boolean;
var Y, yn : longint;
    dibagi : boolean;
begin
  dibagi := false;
if ((X mod 2 = 0) And (X <> 2)) or (X = 1) or (X <= 0) then
     dibagi := true
else
  begin
    Y := 3;
    yn := trunc(sqrt(X)) + 1;
    while (Y < yn) do
    begin
      if (X mod Y = 0) then
         begin
           dibagi := true;
           break;
         end;
         inc(Y,2);
    end;
   end;
   Prima := Not dibagi;
end;
```

   Selanjutnya menginput kunci e, dimana kunci e harus relatif prima terhadap Q (dengan Q adalah hasil kali dari bilangan prima 1 dikurangi 1 dan bilangan prima 2 dikurangi 1). Adapun fungsi program yang dirancang berdasarkan algoritma yang telah disebutkan pada bab sebelumnya (subbab 3.2.4) yang

digunakan untuk mengetahui apakah kunci e relatif prima terhadap Q. Keluaran dari fungsi ini adalah gcd (*greatest common divisor atau faktor persekutuan terbesar*) dari dua bilangan yang diinputkan. Jika nilai keluaran fungsi adalah 1 maka kunci e relatif prima terhadap Q. Listing programnya adalah sebagai berikut :

```
Function gcd(a,b:integer):integer;
//  return greatest common denominator of a and b
var g,z:integer;
begin
        g:=b;
        If b<>0 then
        while g<>0 do
        begin
            z:=a mod g;
            a:=g;
            g:=z;
        end;
        result:=a;
end;
```

3. Melakukan proses enkripsi.

   Enkripsi gambar akan diawali dengan mengambil nilai R, G, dan B dari tiap piksel pada gambar yang telah dibuka. Pengambilan nilai R, G, dan B dilakukan melalui proses looping (perulangan) yang kemudian datanya disimpan dalam variabel datawarna bertipe array. Selanjutnya akan dilakukan proses pengecekan apakah nilai R, G, dan B yang diambil dari tiap piksel pada gambar asli lebih besar sama dengan N (dimana N adalah hasil kali bilangan prima 1 dan prima 2). Jika ya, maka nilai R, G, dan B yang disimpan dalam variabel datawarna akan dibagi dengan N. Hasil bagi akan disimpan dalam variabel xr, xg, dan xb sedangkan px dan py menunjukan posisi pikselnya. Listing programnya adalah sebagai berikut :

```
for i := 0 to lebar - 1 do
begin
  for j := 0 to panjang - 1 do
  begin
    gauge1.Progress := round(m1/luas * 50);
    datawarna[x]:=getRvalue(image2.Canvas.Pixels[i,j]);
    datawarna[x+1]:=getGvalue(image2.Canvas.Pixels[i,j]);
    datawarna[x+2]:=getBvalue(image2.Canvas.Pixels[i,j]);
    if (datawarna[x] >= N) or (datawarna[x+1] >= N)
       or (datawarna[x+2] >= N) then
       begin
          nilai[jlh].xr := datawarna[x] div N;
          nilai[jlh].xg := datawarna[x+1] div N;
          nilai[jlh].xb := datawarna[x+2] div N;
          nilai[jlh].px := i;
          nilai[jlh].py := j;
          Inc(jlh);
       end;
       x:=x+3;
       inc(m1);
  end;
end;
```

Setelah melakukan proses pengambilan nilai R, G, dan B dari tiap piksel yang disimpan pada variabel datawarna, maka proses selanjutnya adalah melakukan proses enkripsi. Listing programnya adalah sebagai berikut :

```
  sel := panjang * lebar * 3;
    x := 1;
    while x <= sel+1 do
    begin
      C := exp_encrypt(datawarna[x],Kunci_e,N);
      datawarna[x]:=C;
      x:=x+1;
    end;
```

Pada proses enkripsi ini akan dipanggil fungsi exp_encrypt untuk melakukan proses enkripsi. Fungsi ini membutuhkan tiga buah inputan data yaitu datawarna (x) yang berisi nilai RGB, kunci e, dan N sebagai kunci umum. Fungsi exp_encrypt dirancang berdasarkan algoritma yang telah disebutkan pada bab sebelumnya (subbab 3.2.6). Keluaran atau output dari fungsi ini

adalah nilai RGB yang sudah tersandikan atau terenkripsi. Listing programnya adalah sebagai berikut:

```
function exp_encrypt(M,e,n:longint):longint;
var C1 : longint;
    i,k:integer;
    e_biner : array[1..32] of integer;
begin
  k:=1;
  while (e <> 0) do
  begin
    if (e and 1 = 1) then
    begin
       e_biner[k] := 1;
    end
    else
    begin
       e_biner[k] := 0;
    end;
   e := e shr 1;
   inc(k);
   end;
   dec(k);
   C1:=1;
   for i:=k downto 1 do
   begin
     C1 := (C1 * C1) mod n;
     if (e_biner[i] = 1) then
         C1 := (C1 * M) mod n;
   end;
   result := C1;
end;
```

Setelah melakukan proses enkripsi pada nilai R, G, dan B yang diambil dari tiap piksel pada gambar asli maka selanjutnya akan dilakukan proses looping (perulangan) untuk membentuk kembali sebuah gambar hasil enkripsi. Listing programnya adalah sebagai berikut :

```
for i := 0 to lebar - 1 do
  begin
  for j := 0 to panjang - 1 do
    begin
      gauge1.Progress := round(m1/luas * 50);
      image2.Canvas.Pixels[i,j]:=rgb(datawarna[x-
      2],datawarna[x-1],datawarna[x]);
      x:=x+3;
      inc(m1);
    end;
  end;
```

4. Hasil dari proses enkripsi adalah sebuah gambar atau citra yang sudah terenkripsi. Selanjutnya gambar hasil enkripsi ini akan disimpan dengan format bmp. Pada proses penyimpanan, selain menyimpan file gambar juga akan disimpan data hasil pembagian yakni xr, xg, dan xb serta posisi pikselnya yaitu px dan py dalam file text yang nantinya akan digunakan pada proses dekripsi. Nama file teksnya sama dengan nama file gambar yang disimpan. Misalkan file gambar hasil enkripsi disimpan dengan nama en1.bmp maka file teks akan tersimpan secara otomatis dengan nama en1.bmp.

4.1.2 Program Dekripsi

Proses kerja dari program dekripsi citra atau gambar adalah sebagai berikut :

1. Baca atau buka file gambar yang akan didekrip. File gambar yang dibuka adalah file Bitmap (bmp). Saat file gambar dibuka maka secara otomatis file teks yang berisi data hasil pembagian nilai R, G, B dari gambar asli akan dibuka.
2. Input dua bilangan prima 1 dan prima 2 serta kunci e.

   Mekanisme penciptaan kunci untuk proses enkripsi dan proses dekripsi pada dasarnya sama. Dua bilangan prima dan kunci e yang diinputkan pada proses dekripsi ini harus sama dengan yang diinputkan pada proses enkripsi. Hanya saja untuk melakukan proses dekripsi menggunakan kunci d. Kunci d dapat dihitung berdasarkan kunci e yang diinputkan dan Q sebagai hasil kali bilangan prima 1 dikurangi 1 dan bilangan prima 2 dikurangi 1. Adapun

fungsi yang digunakan untuk menghitung kunci d. Fungsi ini dirancang berdasarkan algoritma yang telah dijelaskan pada bab sebelumnya (subbab 3.2.5). Keluaran dari fungi ini adalah hasil invers dari dua bilangan yang diinputkan. Dalam program dua bilangan yang dimaksud adalah kunci e dan Q. Hasil invers ini diperoleh dengan rumus e * d = 1 (mod Q) sehingga $d = e^{-1}$ mod (Q) dimana Q adalah (bilangan prima1 - 1) x (bilangan prima2 - 1).

Listing programnya adalah sebagai berikut :

```
function invers_d(a,b: longint):longint;
var i,hasil_kali,hasil_bagi:longint;
begin
  i:=1;
  while (i < b+1) do
    begin
      hasil_kali:=i * a;
      hasil_bagi:=hasil_kali mod b;
      if hasil_bagi = 1 then
      break;
      inc(i,2);
    end;
    result:=i;
end;
```

3. Dekripsi gambar akan diawali dengan pemanggilan fungsi invers_d untuk mendapatkan nilai kunci d. Kemudian mengambil nilai R, G, dan B dari tiap piksel pada gambar hasil enkripsi yang telah dibuka. Pengambilan nilai R, G, dan B dilakukan melalui proses looping (perulangan) yang kemudian datanya disimpan dalam variabel datawarna bertipe array. Listing programnya adalah sebagai berikut :

```
for i := 0 to lebar - 1 do
  begin
  for j := 0 to panjang - 1 do
    begin
      gauge1.Progress:=round(m1/luas * 50);
      datawarna[x]:=getRvalue(image2.Canvas.Pixels[i,j]);
      datawarna[x+1]:=getGvalue(image2.Canvas.Pixels[i,j]);
      datawarna[x+2]:=getBvalue(image2.Canvas.Pixels[i,j]);
      x:=x+3;
      inc(m1);
    end;
  end;
```

Setelah melakukan proses pengambilan nilai R, G, dan B dari tiap piksel yang disimpan pada variabel datawarna, maka proses selanjutnya adalah melakukan proses dekripsi. Listing programnya adalah sebagai berikut :

```
sel := panjang * lebar * 3;
x := 1;
while x <= sel+1 do
begin
  M:=exp_decrypt(datawarna[x],Kunci_d,N);
  datawarna[x]:=M;
  x:=x+1;
end;
```

Pada proses dekripsi ini akan dipanggil fungsi exp_decrypt untuk melakukan proses dekripsi. Fungsi ini membutuhkan tiga buah inputan data yaitu datawarna (x) yang berisi nilai RGB, kunci d, dan N sebagai kunci umum. Keluaran atau output dari fungsi ini adalah nilai RGB yang sudah terdekripsi. Listing programnya adalah sebagai berikut:

```
function exp_decrypt(C,e,n:longint):longint;
var M1 : longint;   i,k : integer;
      e_biner : array[1..32] of integer;
begin
  k:=1;
  while (e <> 0) do
  begin
    if (e and 1 = 1) then
    begin
       e_biner[k] := 1;
    end
    else
    begin
       e_biner[k] := 0;
    end;
  e := e shr 1;
  inc(k);
  end;
  dec(k);
  M1:=1;
  for i:=k downto 1 do
  begin
    M1 := (M1 * M1) mod n;
    if (e_biner[i] = 1) then
        M1 := (M1 * C) mod n;
  end;
  result := M1;
end;
```

Setelah melakukan proses dekripsi pada nilai R, G, dan B yang diambil dari tiap piksel pada gambar hasil enkripsi maka selanjutnya akan dilakukan proses looping (perulangan) untuk membentuk sebuah citra hasil dekripsi. Pada saat melakukan proses looping akan dicek apakah terdapat data hasil pembagian yang tersimpan dalam file text. Jika ada maka akan dilakukan proses perhitungan dimana nilai R, G, dan B hasil proses dekripsi yang disimpan dalam variabel datawarna akan ditambah dengan data hasil pembagian kemudian dikalikan dengan N. Selanjutnya terbentuklah gambar hasil dekripsi. Listing programnya adalah sebagai berikut :

```
for i := 0 to lebar - 1 do
begin
  for j := 0 to panjang - 1 do
  begin
    if (Ftext) and (nilai[k].py = j) and (nilai[k].px = i)
  then
     begin
       datawarna[x-2] := datawarna[x-2]+nilai[k].xr*N;
       datawarna[x-1] := datawarna[x-1]+nilai[k].xg*N;
       datawarna[x] := datawarna[x]+nilai[k].xb*N;
       Inc(k);
       if (k = jlh) then
         Ftext := False;
     end;
     image2.Canvas.Pixels[i,j]:=rgb(datawarna[x-
  2],datawarna[x-1],datawarna[x]);
     gauge1.Progress:=round(m1/luas * 50);
     x:=x+3;
     inc(m1);
  end;
end;
```

4. Hasil dari proses dekripsi adalah sebuah gambar atau citra yang sudah terdekripsi. Gambar hasil dekripsi ini akan disimpan dengan format bmp.

4.1.3 Program Pengujian Kesamaan Dua Citra

Proses ini dilakukan untuk menguji apakah gambar 1 dan gambar 2 sama. Pengujian ini berlaku juga dalam program penyandian citra dimana untuk mengetahui kesamaan dari gambar hasil dekripsi dan gambar asli. Terdapat dua inputan data yaitu gambar 1 dan gambar 2. Proses pengujian diawali dengan mengambil warna dari tiap piksel pada gambar 1 dan gambar 2 melalui proses perulangan (looping) dan disimpan dalam variable datawarna_gambar1 dan variable datawarna_gambar2 yang bertipe data array. Selanjutnya melakukan proses pengecekan apakah nilai RGB yang diambil dari gambar 1 tidak sama dengan nilai RGB yang diambil dari gambar 2. Jika tidak sama maka akan tampil pesan bahwa gambar 1 tidak sama dengan gambar 2. Jika sama maka akan tampil pesan gambar 1 sama dengan gambar 2. Listing programnya adalah sebagai berikut :

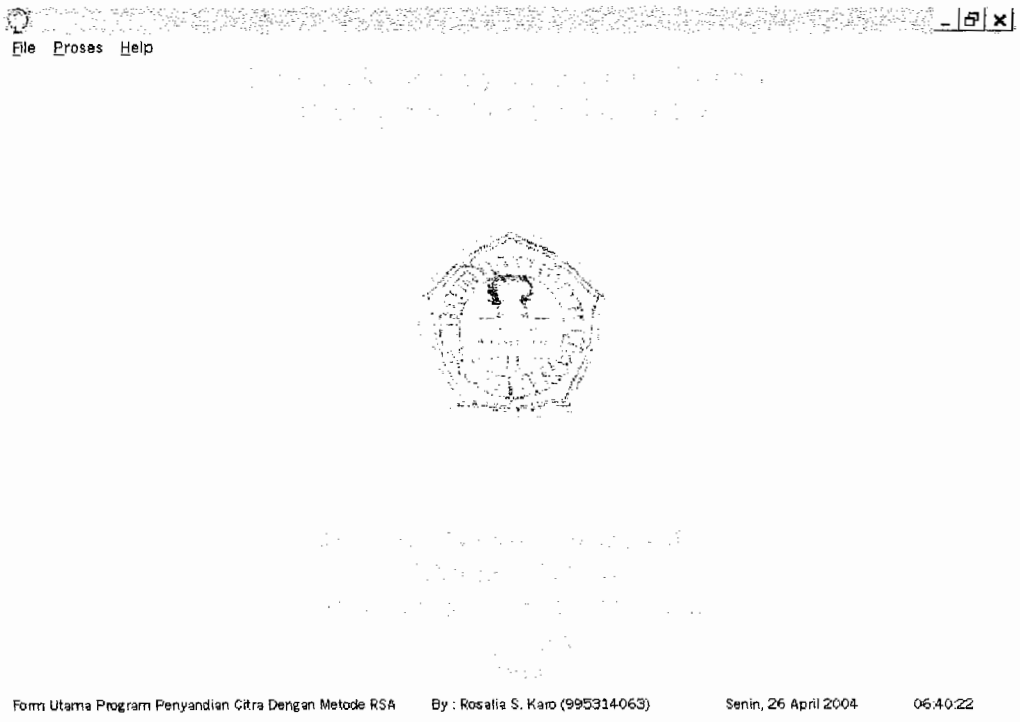
```
for i := 0 to lebar1 - 1 do
begin
  for j := 0 to tinggi1 - 1 do
  begin
    datawarna_gambar1[jlh_p] := gambar_1.Canvas.Pixels[i,j];
    datawarna_gambar2[jlh_p] := gambar_2.Canvas.Pixels[i,j];
    if ((getRvalue(datawarna_gambar1[jlh_p])) <>
  (getRvalue(datawarna_gambar2[jlh_p]))) or
           ((getGvalue(datawarna_gambar1[jlh_p])) <>
  (getGvalue(datawarna_gambar2[jlh_p]))) or
           ((getBvalue(datawarna_gambar1[jlh_p])) <>
  (getBvalue(datawarna_gambar2[jlh_p]))) then
    begin
      MessageDlg('Gambar I tidak sama dengan Gambar
II',mtInformation,[mbOk], 0);
      exit;
    end
    else
    begin
      MessageDlg('Gambar I sama dengan Gambar
  II',mtInformation,[mbOk], 0);
      exit;
    end;
    inc(jlh_p);
  end;
end;
```

4.2 Hasil Implementasi

Program penyandian citra atau gambar terdiri dari beberapa form yaitu form utama, form enkripsi, form dekripsi, form cek file citra, form contents, dan form about us.

4.2.1 Form Utama

Saat program Pencryptcitra.Exe dijalankan maka akan muncul tampilan form utama penyandian. Tampilan dari form utama adalah sebagai berikut :

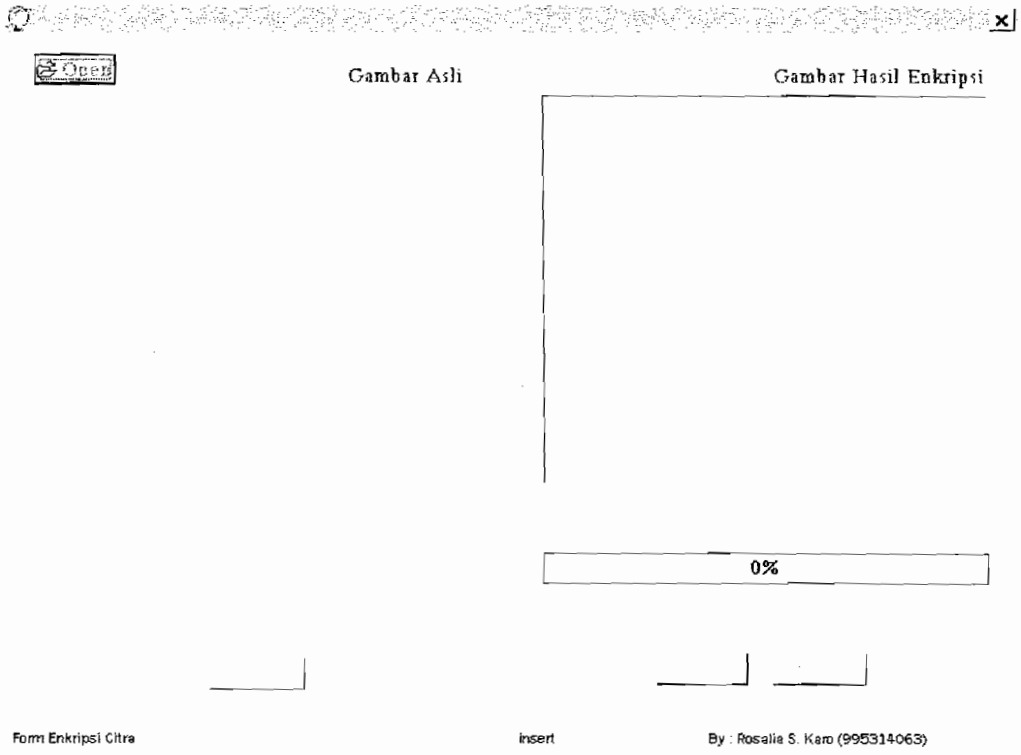


Gambar 4.1 : Tampilan Form Utama

Menu pilihan yang ada pada form utama adalah menu file yang terdiri dari submenu exit (untuk keluar dari program penyandian), menu proses yang terdiri dari submenu enkripsi, dekripsi, cek file citra, dan menu help terdiri dari submenu contents dan about us.

4.2.2 Form Enkripsi

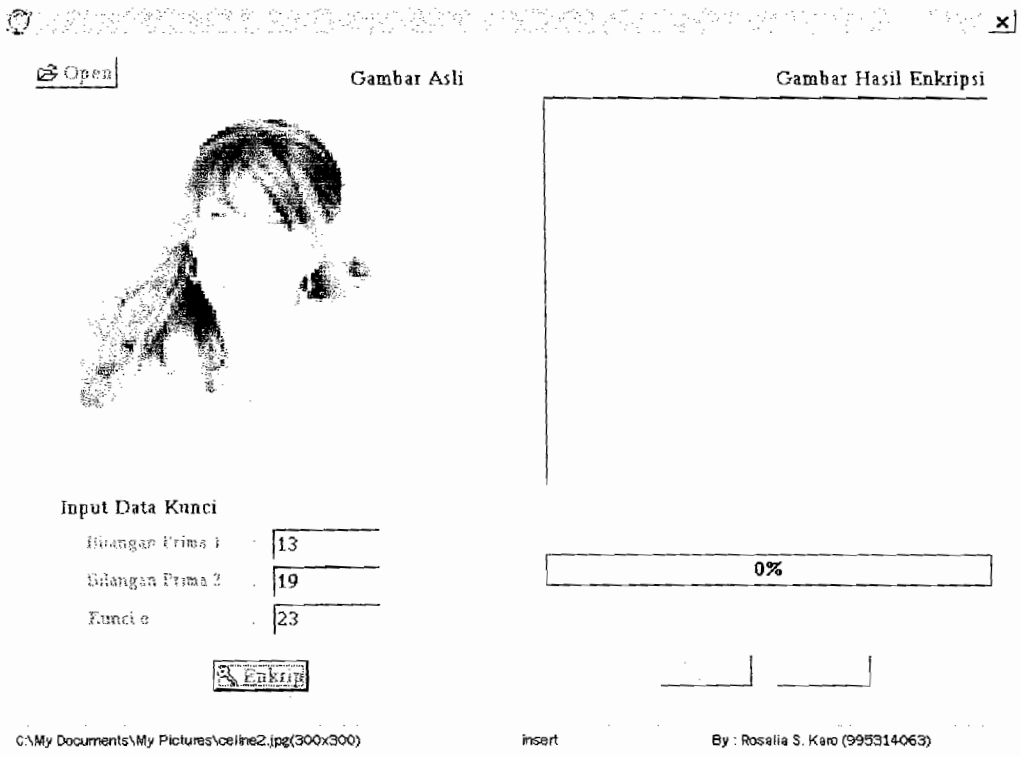
Jika yang dipilih menu enkripsi maka akan tampil form enkripsi. Menu pilihan yang ada dalam form ini adalah tombol open, enkrip, save dan save as. Menu pilihan yang aktif saat form ini dipilih adalah tombol open sedangkan menu pilihan yang lain belum berfungsi. Tampilan form enkripsi adalah :



Gambar 4.2 : Tampilan Form Enkripsi

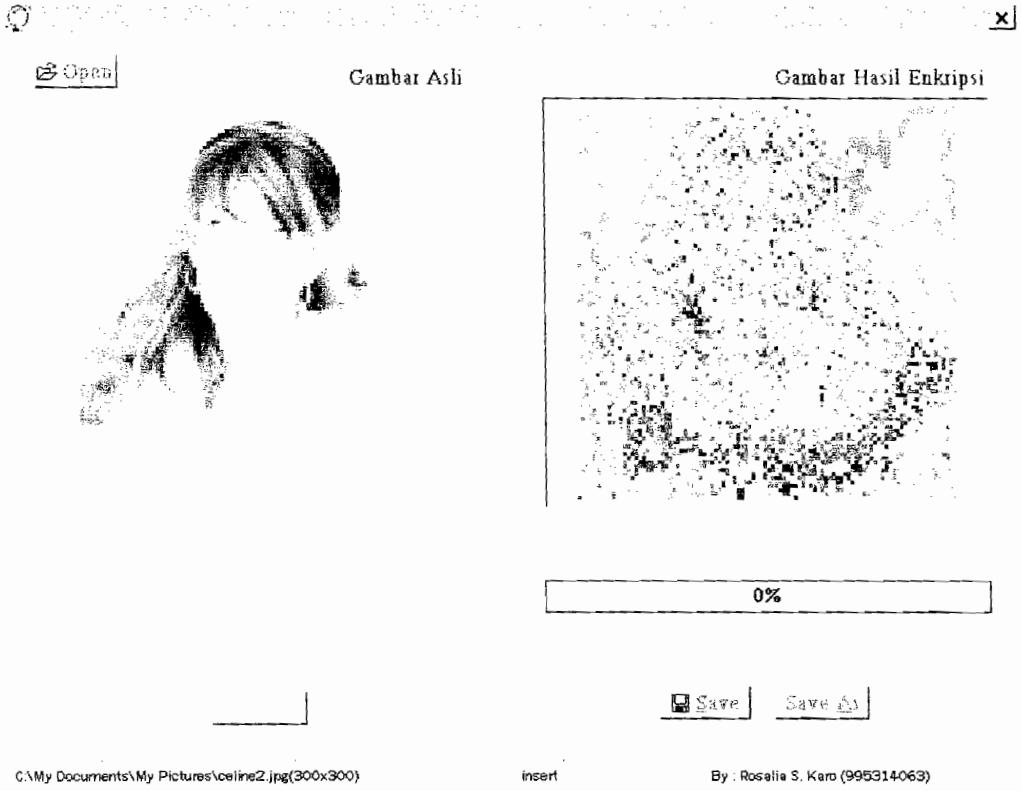
Jika file gambar sudah dibuka maka kotak inputan data kunci baru aktif. Kemudian menginput bilangan prima 1 dan bilangan prima 2 serta kunci e. User tidak dapat menginputkan bilangan kedua jika bilangan pertama belum diinputkan. Kedua bilangan yang diinputkan harus bilangan prima dan berbeda serta hasil kali dua bilangan prima tersebut harus lebih kecil dari 256. Setelah

semua data kunci diinputkan maka tombol enkrip akan diaktifkan dan siap untuk melakukan proses enkripsi gambar. Contoh tampilannya adalah :



Gambar 4.3 : Tampilan Form Enkripsi Setelah File Gambar Dibuka

Proses selanjutnya adalah melakukan proses enkripsi gambar dengan menekan tombol enkrip. Untuk mengetahui kemajuan proses enkripsi gambar ditentukan oleh progressbar yang berjalan. Hasil dari proses enkripsi adalah sebuah gambar yang sudah tersandikan atau terenkripsi. Contoh tampilannya adalah sebagai berikut :

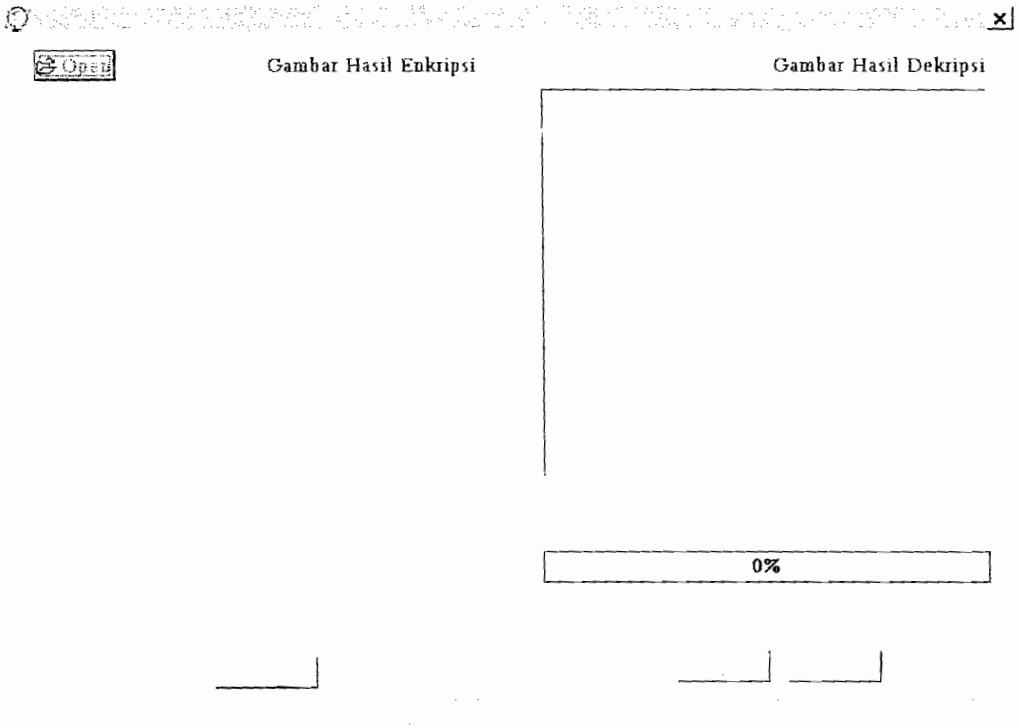


Gambar 4.4 : Tampilan Form Hasil Proses Enkripsi

Setelah proses enkripsi gambar selesai maka tombol save dan save as baru aktif. Gambar hasil enkripsi ini akan disimpan dengan format bmp. Fasilitas untuk penyimpanan gambar menggunakan menu save dan save as.

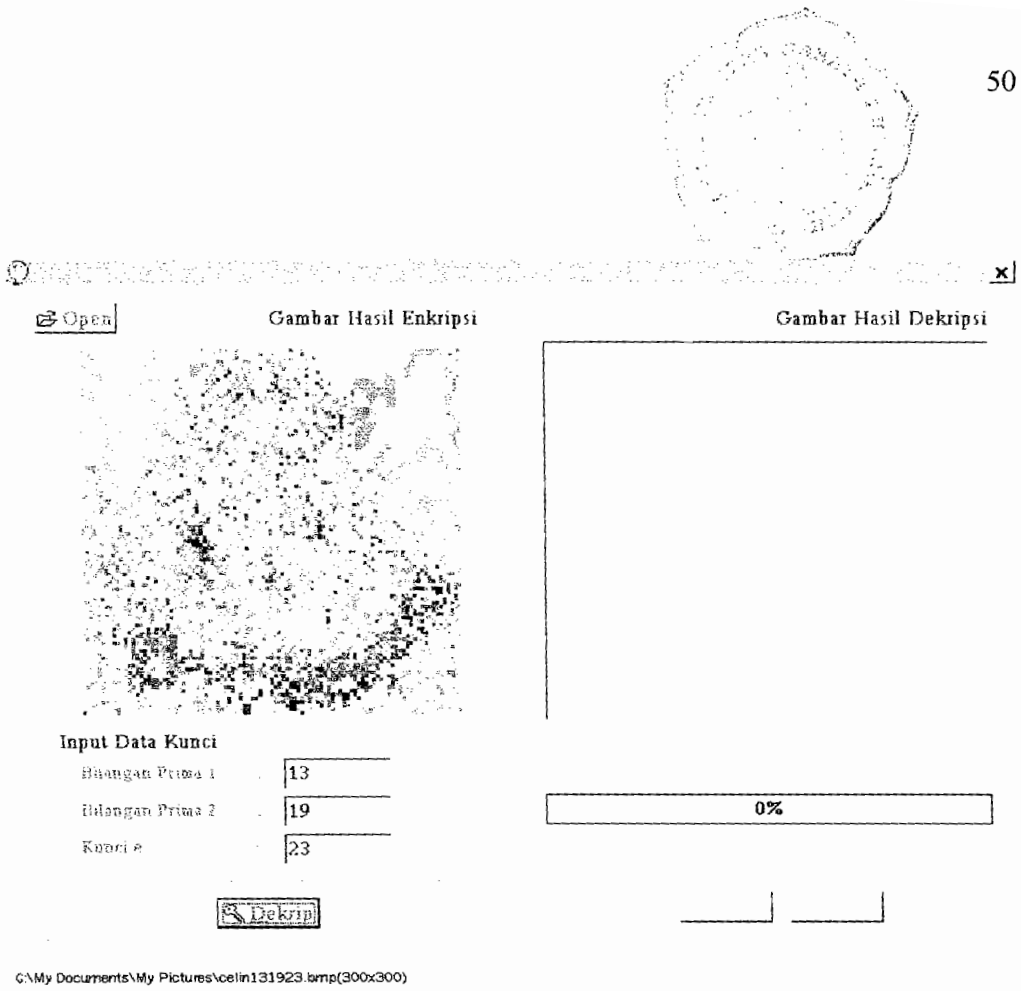
4.2.3 Form Dekripsi

Jika yang dipilih menu dekripsi maka akan tampil form dekripsi. Menu pilihan yang ada dalam form ini adalah tombol open, dekrip, save dan save as. Menu pilihan yang aktif saat form ini dipilih adalah tombol open sedangkan menu pilihan yang lain belum berfungsi. Tampilan form dekripsi adalah sebagai berikut :



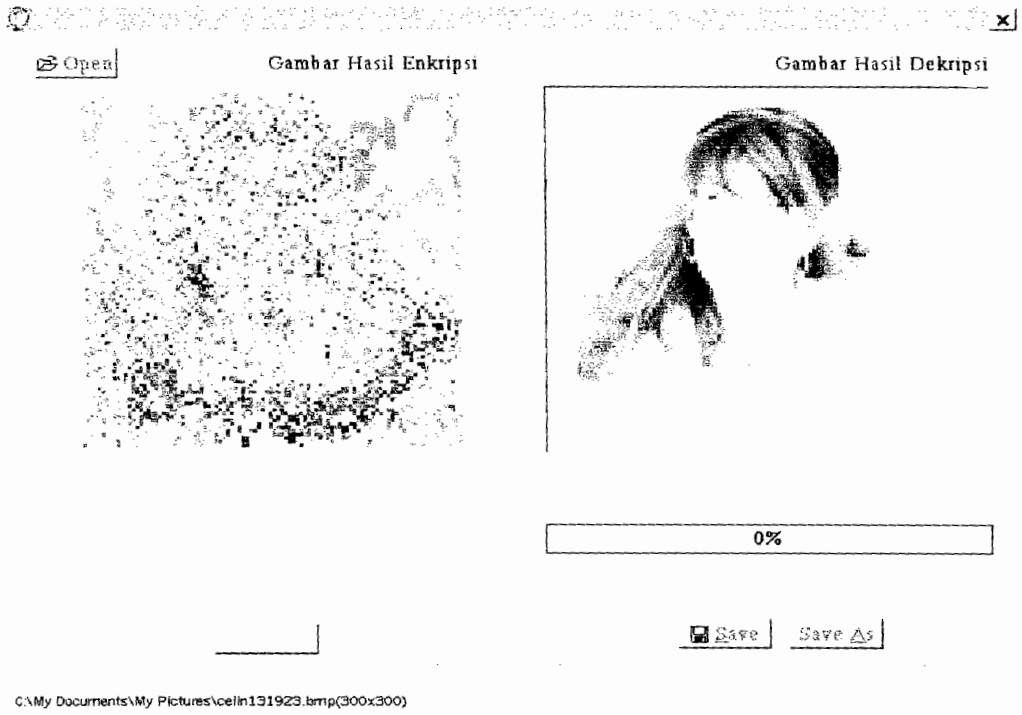
Gambar 4.5 : Tampilan Form Dekripsi

Jika file gambar sudah dibuka maka kotak inputan data kunci baru aktif. Kemudian menginput bilangan prima 1 dan bilangan prima 2 serta kunci e. User tidak dapat menginputkan bilangan kedua jika bilangan pertama belum diinputkan. Kedua bilangan yang diinputkan harus bilangan prima dan berbeda serta hasil kali dua bilangan prima tersebut harus lebih kecil dari 256. Pada proses dekripsi bilangan prima 1 dan bilangan prima 2 serta kunci e yang diinputkan harus sama dengan yang diinputkan pada proses enkripsi. Setelah semua data kunci diinputkan maka tombol dekrip akan diaktifkan dan siap untuk melakukan proses dekripsi gambar. Contoh tampilannya adalah sebagai berikut :



Gambar 4.6 : Tampilan Form Dekripsi Setelah File Gambar Dibuka

Proses selanjutnya adalah melakukan proses dekripsi gambar dengan menekan tombol dekrip. Untuk mengetahui kemajuan proses dekripsi gambar ditentukan oleh progressbar yang berjalan. Hasil dari proses dekripsi adalah sebuah gambar yang sudah terdekripsi atau sebuah gambar yang sudah ditransformasikan ke gambar aslinya. Contoh tampilannya adalah sebagai berikut :

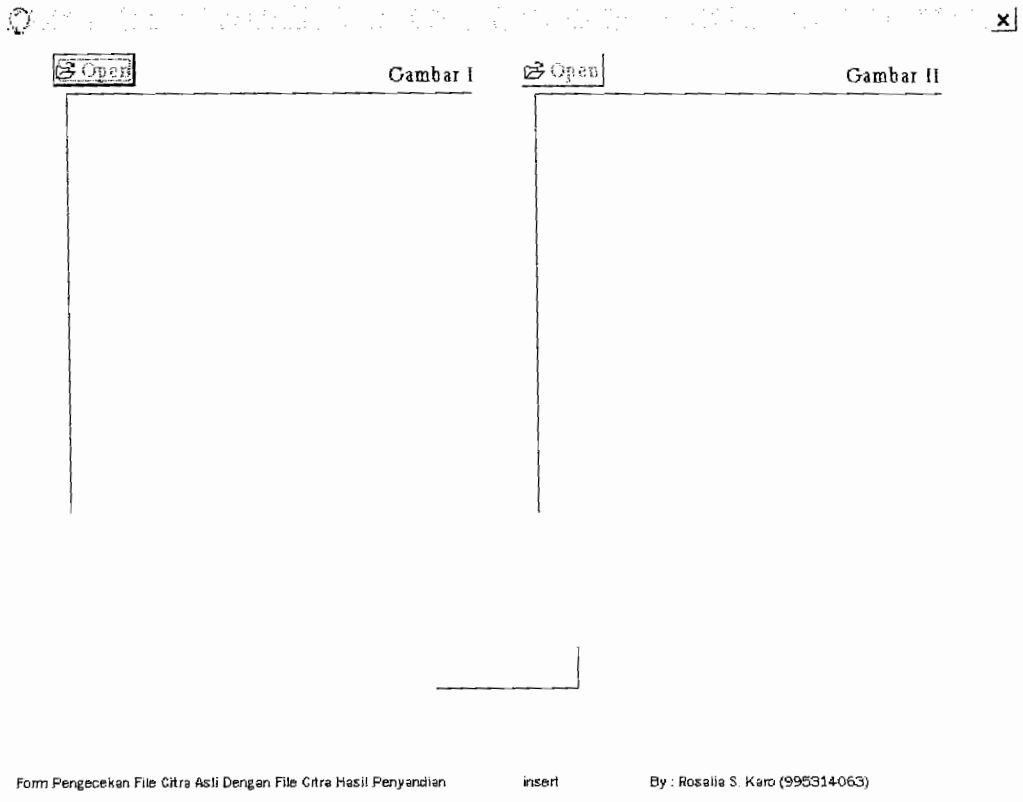


Gambar 4.7 : Tampilan Form Hasil Proses Dekripsi

Setelah proses dekripsi gambar selesai maka tombol save dan save as baru aktif. Gambar hasil dekripsi ini akan disimpan dengan format bmp. Fasilitas untuk penyimpanan gambar menggunakan menu save dan save as.

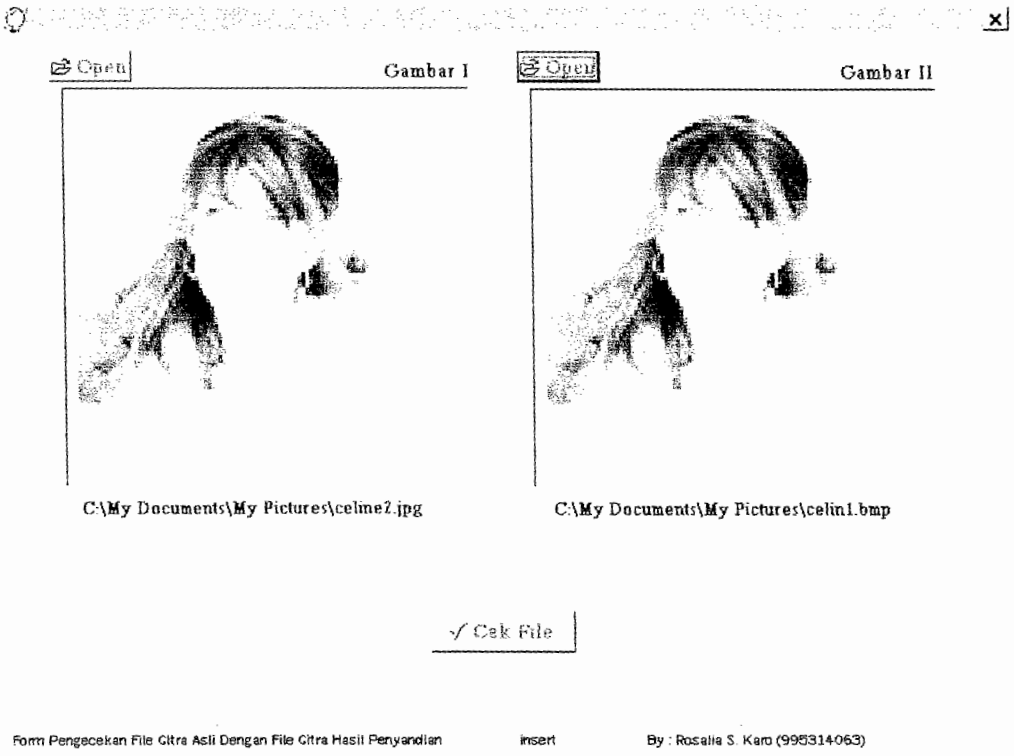
4.2.4 Form Cek File Gambar

Jika yang dipilih menu cek file citra maka tampil form cek file citra. Form ini digunakan untuk mengecek apakah gambar I sama dengan gambar II. Jadi terdapat dua inputan gambar. Menu pilihan yang ada dalam form ini adalah tombol open gambar 1, open gambar 2 dan cek file. Menu pilihan yang aktif saat form ini dipilih adalah tombol open gambar 1 dan open gambar 2, sedangkan tombol cek file belum berfungsi. Tampilan formnya adalah sebagai berikut :



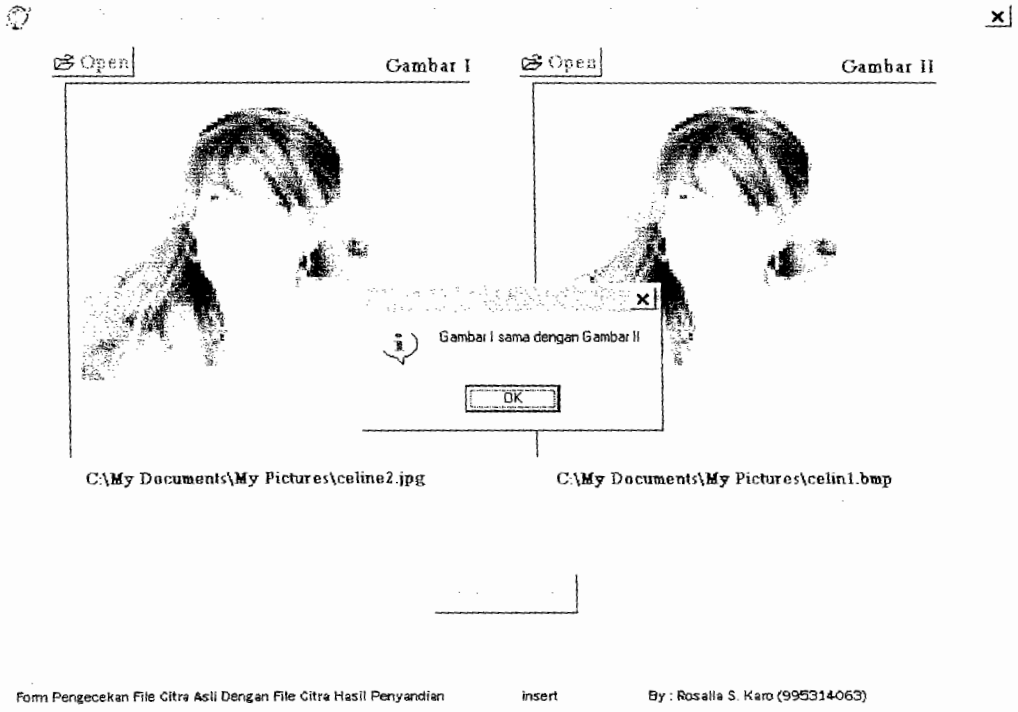
Gambar 4.8 : Tampilan Form Cek File Citra

File gambar yang digunakan adalah file JPEG dan file Bitmap. Jika file gambar 1 dan gambar 2 sudah dibuka maka tombol cek file baru aktif. Contoh tampilannya adalah sebagai berikut :



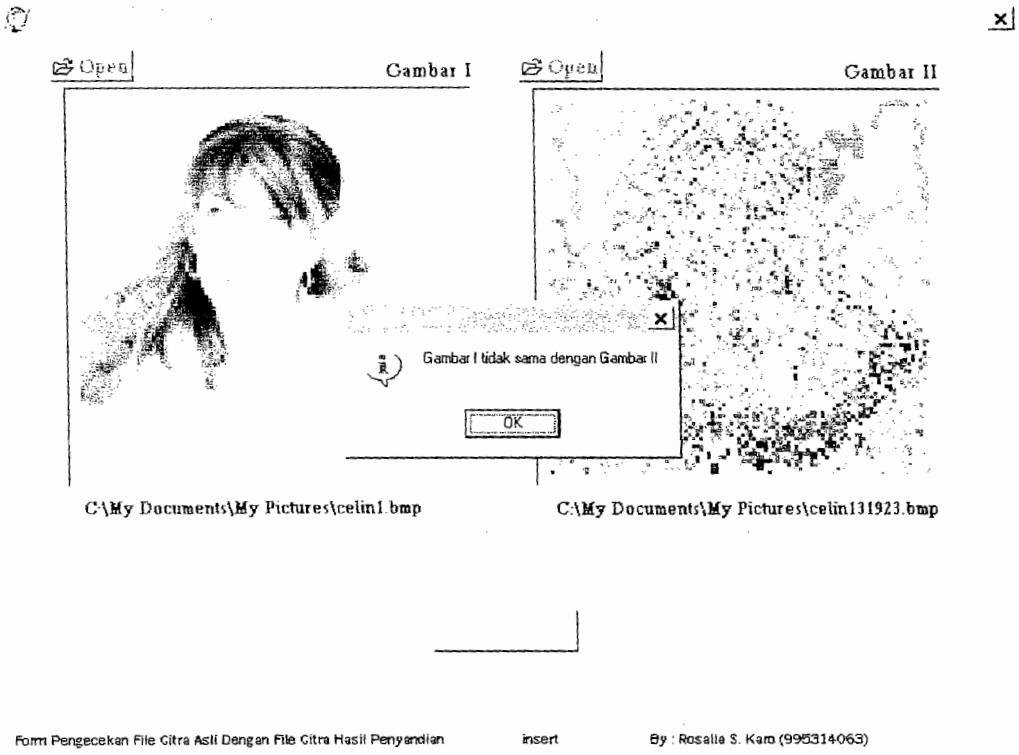
Gambar 4.9 : Tampilan Form Cek File Setelah File Gambar Dibuka

Selanjutnya adalah melakukan proses pengujian kesamaan dari dua gambar yang diinputkan dengan menekan tombol cek file. Jika hasil pengujian dua gambar yang diinputkan sama maka akan tampil pesan "Gambar I sama dengan Gambar II". Contoh tampilannya adalah sebagai berikut :



Gambar 4. 10 : Tampilan Form Cek File Untuk Gambar Yang Sama

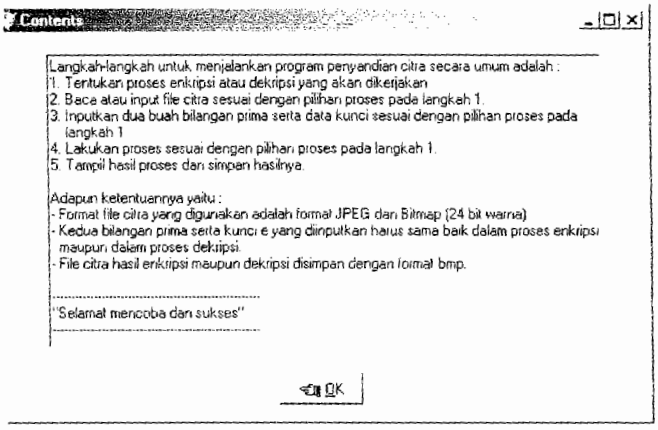
Jika hasil pengujian dua gambar yang diinputkan tidak sama maka akan tampil pesan "Gambar I tidak sama dengan Gambar II". Contoh tampilannya adalah sebagai berikut :



Gambar 4. 11 : Tampilan Form Cek File Untuk Gambar Yang Tidak Sama

4.2.5 Form Contents

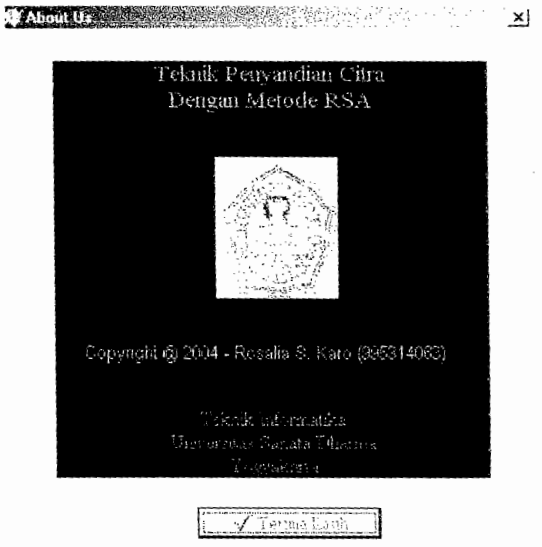
Form ini berisi langkah-langkah dalam menjalankan program, yang membantu pengguna atau user untuk melakukan proses kerja enkripsi dan dekripsi sebuah citra dengan metode RSA. Contoh tampilan formnya adalah sebagai berikut :



Gambar 4.12 : Tampilan Form Contents

4.2.6 About Us

Form ini berisi keterangan pembuat program. Tampilan formnya adalah :



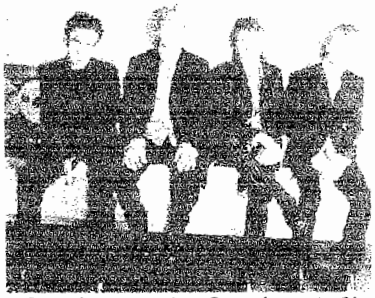
Gambar 4.13 : Tampilan Form About Us

4.3 Pembahasan

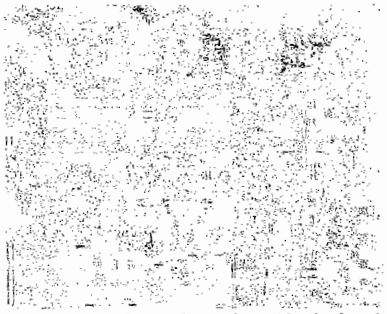
Setelah program dicoba maka diperoleh hasil pembahasan adalah sebagai berikut :

1. Secara umum program penyandian citra dengan metode RSA dapat berjalan dengan baik. Ini terbukti karena program ini mampu melakukan proses enkripsi dan proses dekripsi dengan benar. Contohnya dapat dilihat pada hasil implementasi (subbab 4.2). Tampilan hasil proses enkripsi tampak dalam gambar 4.4 dan tampilan hasil proses dekripsi tampak dalam gambar 4.7. Setelah melakukan pengujian, ternyata gambar hasil dekripsi sama dengan gambar aslinya. Tampilan hasil pengujian tampak dalam gambar 4.10.
2. Dalam program penyandian gambar jika suatu gambar dikenai proses enkripsi dan proses dekripsi sebanyak dua kali maka ada dua hasil yang diperoleh yakni gambar bisa ditransformasikan ke bentuk asli atau tidak dapat ditransformasikan kebentuk asli. Semuanya ini tergantung nilai RGB dari gambar yang akan dikenai proses. Jika gambar yang akan dikenai proses enkripsi dan dekripsi sebanyak dua kali, nilai RGB yang diambil untuk diproses lebih kecil dari kunci N maka gambar yang sudah dienkrip dua kali bisa langsung ditransformasikan ke bentuk semula. Contoh :
   - Misalkan dipilih gambar **BonJovi.jpg** dan bilangan prima 1 = 13, bilangan prima 2 = 19, kunci e = 23 maka tampilan hasil proses enkripsi pertama tampak dalam gambar 4.15.
   - Gambar hasil proses enkripsi pertama akan dikenai proses enkripsi kedua. Tampilan hasil proses enkripsi kedua tampak dalam gambar 4.16.

- Gambar hasil proses enkripsi kedua akan dikenai proses dekripsi pertama. Tampilan hasil proses dekripsi pertama tampak dalam gambar 4.17.
- Gambar hasil proses dekripsi pertama akan dikenai proses dekripsi kedua. Tampilan hasil proses dekripsi kedua tampak dalam gambar 4.18.
- Setelah melakukan pengujian ternyata gambar yang dikenai proses enkripsi dan proses dekripsi dua kali bisa ditransformasikan lagi ke gambar asli jika nilai RGB yang terkandung dalam gambar yang akan dikenai proses lebih kecil dari kunci N.



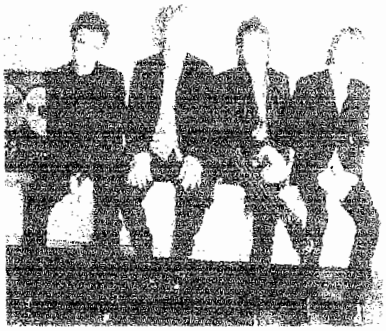
Gambar 4.14 : Gambar Asli



Gambar 4.15 : Gambar Enkripsi I

Gambar 4. 16 : Gambar Enkripsi II

Gambar 4.17 : Gambar Dekripsi I



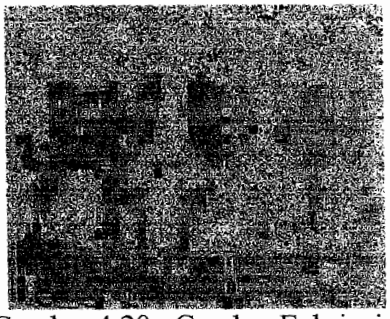
Gambar 4.18 : Gambar Dekripsi II

Jika gambar yang akan dikenai proses enkripsi dan dekripsi sebanyak dua kali, nilai RGB yang diambil untuk diproses lebih besar sama dengan kunci N maka gambar yang sudah dienkrip dua kali tidak dapat ditransformasikan ke bentuk semula. Dalam metode RSA, jika nilai RGB yang diambil dari suatu gambar pada saat akan melakukan proses enkripsi lebih besar sama dengan kunci N maka nilai RGB akan dibagi dengan kunci N dan data hasil pembagian akan disimpan. Data ini akan digunakan saat melakukan proses dekripsi sehingga nilai RGB yang sudah tersandikan dapat ditransformasikan ke RGB asli. Namun dalam proses enkripsi dan dekripsi dua kali ini, data hasil pembagian yang sudah disimpan saat proses enkripsi pertama dilakukan, tidak dipakai lagi pada proses enkripsi kedua. Karena pada proses enkripsi kedua yang diambil hanya nilai RGB dari gambar hasil enkripsi pertama. Hal ini yang menyebabkan gambar hasil dekripsi tidak sama dengan gambar asli karena ada data yang tidak digunakan dalam proses ini atau terjadi kehilangan data. Contoh :

- Misalkan dipilih gambar **BonJovi.jpg** dan bilangan prima 1 = 7, bilangan prima 2 = 13, kunci e = 17. Tampilan hasil proses enkripsi pertama tampak dalam gambar 4.20.
- Gambar hasil proses enkripsi pertama akan disimpan dan secara otomatis akan tersimpan juga data hasil pembagian dalam file text. Selanjutnya akan dilakukan proses enkripsi kedua Namun data hasil pembagian yang tersimpan dalam file text tidak digunakan dalam proses enkripsi kedua. Tampilan hasil proses enkripsi kedua tampak dalam gambar 4.21.

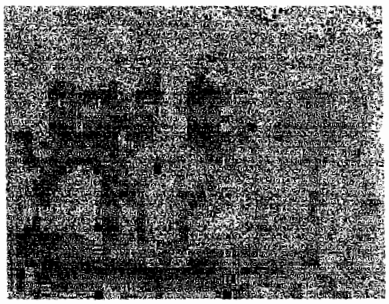
- Gambar hasil proses enkripsi kedua akan dikenai proses dekripsi pertama. Tampilan hasil proses dekripsi pertama tampak dalam gambar 4.22.
- Gambar hasil proses dekripsi pertama akan dikenai proses dekripsi kedua. Tampilan hasil proses dekripsi kedua tampak dalam gambar 4.23.
- Setelah melakukan pengujian ternyata gambar yang dikenai proses enkripsi dan proses dekripsi dua kali tidak bisa ditransformasikan lagi ke gambar asli jika nilai RGB yang terkandung dalam gambar yang akan dikenai proses lebih besar sama dengan kunci N.

Gambar 4.19 : Gambar Asli



Gambar 4.20 : Gambar Enkripsi I

Gambar 4.21 : Gambar Enkripsi II



Gambar 4.22 : Gambar Dekripsi I

Gambar 4.23 : Gambar Dekripsi II

3. Dalam program penyandian gambar jika bilangan prima dan nilai kunci yang diinputkan kecil maka gambar hasil enkripsi cendrung berwarna gelap (kehitam - hitaman). Hal ini disebabkan karena jika nilai kunci N yang merupakan hasil kali dua bilangan prima kecil misalkan 15 maka nilai RGB hasil proses enkripsi berkisar antara 0 – 14 dan berlaku baik untuk nilai RGB lebih kecil dari kunci N atau nilai RGB lebih besar sama dengan kunci N sehingga warnanya cendrung hitam . Dan jika bilangan prima dan nilai kunci e yang diinputkan besar maka gambar hasil proses enkripsi terlihat jelas pola gambarnya dengan warna yang teracak. Hal ini disebabkan karena jika nilai kunci N yang merupakan hasil kali dua bilangan prima besar misalkan 247 maka nilai RGB hasil proses enkripsi berkisar antara 0 – 246 dan nilai RGB yang dihasilkan lebih bervariasi sehingga warnanya tidak gelap.
4. Dalam program ini jika dua bilangan prima dan kunci e yang diinputkan pada proses dekripsi tidak sama dengan bilangan prima dan kunci e yang diinputkan pada proses enkripsi maka gambar hasil enkripsi tidak dapat ditransformasikan ke gambar asli. Hal ini disebabkan karena pada proses dekripsi nilai kunci d dihitung berdasarkan nilai kunci e dan bilangan prima yang diinputkan. Jika nilai kunci yang diinputkan tidak sama maka kunci d yang didapat dari kunci e dan bilangan prima yang diinputkan pada proses enkripsi tidak sama dengan kunci d yang didapat dari kunci e dan bilangan prima yang diinputkan pada proses dekripsi sehingga mengakibatkan gambar hasil enkripsi tidak dapat ditransformasikan ke gambar asli.

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil implementasi dan analisa program dapat diambil beberapa kesimpulan antara lain :

1. Program penyandian citra dengan metode RSA dapat berjalan dengan baik. Program ini mampu melakukan proses enkripsi dan proses dekripsi dengan benar karena terbukti bahwa gambar hasil proses dekripsi dengan metode RSA sama dengan gambar aslinya.
2. Apabila suatu gambar dikenai proses enkripsi dan dekripsi sebanyak dua kali maka ada dua hasil yang diperoleh yaitu pertama gambar hasil enkripsi bisa di transformasikan ke gambar asli dan kedua gambar hasil enkripsi tidak selalu dapat ditransformasikan ke gambar asli. Hal ini tergantung nilai RGB dari gambar yang akan dikenai proses.
3. Enkripsi dengan nilai kunci yang kecil akan menghasilkan gambar dengan warna yang cendrung gelap (kehitam-hitaman).
4. Dalam program penyandian dengan metode RSA, karena nilai kunci N (hasil kali bilangan prima 1 dan bilangan prima 2) dibatasi lebih kecil sama dengan 255 maka bilangan prima yang dipilih juga terbatas.

5.2 Saran

1. Untuk pengembangan program selanjutnya, keterbatasan saat ini perlu diatasi sehingga tuntutan dari algoritma RSA yang menghendaki kunci – kunci mempunyai digit besar dapat terpenuhi. Dalam hal ini, nilai kunci N (hasil kali bilangan prima 1 dan bilangan prima 2) bisa lebih besar dari 255 sehingga bilangan prima yang dipilih tidak terbatas.
2. Karena dalam program penyandian citra dengan metode RSA, file citra yang diinputkan adalah kombinasi dari file JPEG dan Bitmap maka dalam pengembangan selanjutnya dapat menggunakan file JPEG murni.

DAFTAR PUSTAKA

A. J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone. CRC Press (1996). *Handbook of Applied Cryptography*. http://www.cacr.math.uwaterloo.ca/hac.

Schneier, Bruce. (1996). *Applied Cryptography Protocols, Algorithms, and Source Code in C*, Second Edition, John Wiley & Sons, Inc.

J.J.Siang (2002). *Bilangan Prima : Perkembangan dan Aplikasinya*. home.unpar.ac.id/~integral/Volume%207/Volume%207.htm cached.

Pranata, Antony. (2000). *Pemrograman Borland Delphi*. (Edisi tiga). Penerbit Andi Yogyakarta.

Pranata, Antony. (2002). *Pemrograman Borland Delphi 6*. (Edisi empat). Penerbit Andi Yogyakarta.

Martina, Ir. Inge. (1999). *36 Jam Belajar Komputer Delphi 4.0*. Penerbit PT Elex Media Komputindo, Kelompok Gramedia, Anggota IKAPI, Jakarta.

Pranata, Antony. (2000). *Algoritma dan Pemrograman*. (Edisi pertama). J&J Learning, Yogyakarta.

Santosa, Ir. P. Insap M.Sc (1994). *Grafika Komputer dan Antarmuka Grafis*. Yogyakarta : Andi Offset.

LAMPIRAN

Program Aplikasi Penyandian Citra

```
program Pencryptcitra;

uses
  Forms,
  UnitMenuUtama in 'UnitMenuUtama.pas' {Form1},
  UnitEnkripsi in 'UnitEnkripsi.pas' {Form2},
  UnitDekripsi in 'UnitDekripsi.pas' {Form3},
  UnitAbout in 'UnitAbout.pas' {Form4},
  Unit5 in 'C:\Program Files\Common Files\Borland
  Shared\Images\Default\Unit5.pas' {Form5},
  UnitAboutUs in 'C:\Program Files\Common Files\Borland
  Shared\Images\Default\UnitAboutUs.pas' {AboutBox},
  UnitContents in 'UnitContents.pas' {Frame1: TFrame},
  UnitCekFileCitra in 'UnitCekFileCitra.pas' {Form6};

{$R *.res}

begin
  Application.Initialize;
  Application.CreateForm(TForm1, Form1);
  Application.CreateForm(TForm2, Form2);
  Application.CreateForm(TForm3, Form3);
  Application.CreateForm(TForm4, Form4);
  Application.CreateForm(TForm5, Form5);
  Application.CreateForm(TAboutBox, AboutBox);
  Application.CreateForm(TForm6, Form6);
  Application.Run;
end.
```

Program Menu Utama

```
unit UnitMenuUtama;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics,
  Controls, Forms, Dialogs, Menus, jpeg, ExtCtrls, StdCtrls,
  ComCtrls, ImgList;

type
  TForm1 = class(TForm)
    MainMenuUtama: TMainMenu;
    File1: TMenuItem;
    Exit1: TMenuItem;
    Proses1: TMenuItem;
    Enkripsi1: TMenuItem;
    N1: TMenuItem;
    Dekripsi1: TMenuItem;
    Help1: TMenuItem;
    Content1: TMenuItem;
    N2: TMenuItem;
    About1: TMenuItem;
```

```
    StatusBar1: TStatusBar;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    Label6: TLabel;
    Label7: TLabel;
    Bevel1: TBevel;
    Image1: TImage;
    ImageList1: TImageList;
    N3: TMenuItem;
    CekFileCitra1: TMenuItem;
    procedure Exit1Click(Sender: TObject);
    procedure Enkripsi1Click(Sender: TObject);
    procedure Dekripsi1Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure About1Click(Sender: TObject);
    procedure Content1Click(Sender: TObject);
    procedure CekFileCitra1Click(Sender: TObject);
  private
    { Private declarations }

  public

    { Public declarations }
  end;

var
  Form1: TForm1;

implementation

uses UnitEnkripsi, UnitDekripsi, Unit5, UnitAboutUs, UnitAbout,
  UnitCekFileCitra;

{$R *.dfm}

procedure TForm1.Exit1Click(Sender: TObject);
begin
if  (messagebox(0,'Apakah  Anda  Yakin  Untuk  Keluar  Dari  Program
    Ini...?','Perhatian',
    MB_YESNO+MB_DEFBUTTON1))<>IDNO then
    postQuitmessage(0);
    exit;
end;

procedure TForm1.Enkripsi1Click(Sender: TObject);
begin
  form2.show;
end;

procedure TForm1.Dekripsi1Click(Sender: TObject);
begin
  form3.show;
end;
```

```
procedure TForm1.FormCreate(Sender: TObject);
begin
  statusbar1.Panels[0].Text:='Form Utama Program Penyandian Citra
                              Dengan Metode RSA';
  statusbar1.Panels[1].Text:='By : Rosalia S. Karo (995314063)';
  statusbar1.Panels[2].Text:='   '+FormatDateTime('dddd,  d  mmmm
                              yyyy', Date);
  statusbar1.Panels[3].Text:=' '+FormatDateTime('hh:nn:ss', time);
end;

procedure TForm1.About1Click(Sender: TObject);
begin
  form4.Show;
end;

procedure TForm1.Content1Click(Sender: TObject);
begin
  form5.Show;
end;

procedure TForm1.CekFileCitra1Click(Sender: TObject);
begin
  form6.show;
end;
end.
```

Program Enkripsi Citra

```
unit UnitEnkripsi;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics,
  Controls, Forms, Dialogs, ComCtrls, StdCtrls, Buttons, ExtCtrls,
  ExtDlgs, Menus, Gauges, JPEG;

const max=10000000;

type
  TForm2 = class(TForm)
    GroupBox1: TGroupBox;
    GroupBox2: TGroupBox;
    OpenPictureDialog1: TOpenPictureDialog;
    SavePictureDialog1: TSavePictureDialog;
    StatusBar1: TStatusBar;
    Open: TBitBtn;
    Label1: TLabel;
    Bevel1: TBevel;
    gambar_asli: TImage;
    input_kunci: TGroupBox;
    Label2: TLabel;
    ScrollBox1: TScrollBox;
    gambar_enkrip: TImage;
    prima_1: TLabel;
```

```
    prima_2: TLabel;
    kunciumum_e: TLabel;
    Edit1: TEdit;
    Edit2: TEdit;
    Edit3: TEdit;
    Panel1: TPanel;
    proses_enkrip: TBitBtn;
    Gauge1: TGauge;
    Panel2: TPanel;
    Save: TBitBtn;
    SaveAs: TBitBtn;
    procedure Close1Click(Sender: TObject);
    procedure Edit1Exit(Sender: TObject);
    procedure Edit2Exit(Sender: TObject);
    procedure Edit3Exit(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure OpenClick(Sender: TObject);
    procedure gambar_asliClick(Sender: TObject);
    procedure gambar_enkripClick(Sender: TObject);
    procedure FormShow(Sender: TObject);
    procedure    FormClose(Sender:    TObject;    var    Action:
    TCloseAction);
    procedure SaveClick(Sender: TObject);
    procedure SaveAsClick(Sender: TObject);
    procedure proses_enkripClick(Sender: TObject);
  private
    { Private declarations}
  public
    { Public declarations  }
    CurrentFile:string;
  end;
  mydata = record
    px, py: Word;
    xr, xg, xb: Byte;
  end;
var datawarna:array[1..max] of Byte;
    nilai: array [1..max] of mydata;
    x,sel,luas,m1:longint;
    Kunci_e,N,Q:longint;
    Prima1, Prima2: boolean;
    C:Longint;
    jlh: LongWord;

  Form2: TForm2;

implementation

uses UnitMenuUtama;

{$R *.dfm}

function JPEG2Bitmap(JPEGFile : String) : TBitmap;
var
   myJPEG:TJPEGImage;
begin
  myJPEG := TJPEGImage.Create ;
```

```
  try
    myJPEG.LoadFromFile(JPEGFile);
    Result := TBitmap.Create;

    myJPEG.PixelFormat:=pf24bit;
  finally
    myJPEG.Free;
  end;
end;

procedure TForm2.Close1Click(Sender: TObject);
begin
  form2.Hide;
  form1.show;
end;

Function gcd(a,b:integer):integer;
var g,z:integer;
begin
  g:=b;
  If b<>0 then
  while g<>0 do
  begin
    z:=a mod g;
    a:=g;
    g:=z;
  end;
  result:=a;
end;

Function Prima(X: longint): Boolean;
var Y, yn : longint;
    dibagi : boolean;
begin
  dibagi := false;
  if ((X mod 2 = 0) And (X <> 2)) or (X = 1) or (X <= 0) then
     dibagi := true
  else
  begin
    Y := 3;
    yn := trunc(sqrt(X)) + 1;
    while (Y < yn) do
    begin
      if (X mod Y = 0) then
      begin
        dibagi := true;
        break;
      end;
        inc(Y,2);
    end;
  end;
  Prima := Not dibagi;
end;


function exp_encrypt(M,e,n:longint):longint;
```

```
var C1 : longint;
    i,k:integer;
    e_biner : array[1..32] of integer;
begin
  k:=1;
  while (e <> 0) do
  begin
    if (e and 1 = 1) then
    begin
      e_biner[k] := 1;
    end
    else
    begin
      e_biner[k] := 0;
    end;
  e := e shr 1;
  inc(k);
  end;
  dec(k);
  C1:=1;
  for i:=k downto 1 do
  begin
    C1 := (C1 * C1) mod n;
    if (e_biner[i] = 1) then
        C1 := (C1 * M) mod n;
  end;
  result := C1;
end;

procedure TForm2.Edit1Exit(Sender: TObject);
begin
  try
    Prima1:=Prima(strtoint(edit1.Text));
    if not Prima1  then
    begin
      messagedlg('Angka      '+edit1.Text+'      bukan      bilangan
                  prima'+chr(13),
      mtwarning,[mbOk],0);
      edit1.SetFocus;
    end;
  except
    Edit1.SetFocus;
  end;
end;

procedure TForm2.Edit2Exit(Sender: TObject);
begin
  try
    Prima2:=Prima(strtoint(edit2.text));
    N:=strtoint(edit1.Text) * strtoint(edit2.Text);
    if (StrToInt(Edit1.Text) = StrToInt(Edit2.Text)) then
    begin
      messagedlg('Nilai p dan q harus berbeda.'+chr(13),
      mtwarning,[mbOk],0);
      edit1.SetFocus;
    end
```

```
    else if not Prima2 then
    begin
      messagedlg('Angka      '+edit2.Text+'      bukan      bilangan
                 prima'+chr(13),
      mtwarning,[mbOk],0);
      edit2.SetFocus;
    end
    else if (N > 256) then
    begin
      messagedlg('Hasil kali bilangan prima1 dan prima2 harus <
                 256!',mtwarning,[mbOK],0);
      edit1.SetFocus;
    end;
  except
    Edit2.SetFocus;
  end;
end;

procedure TForm2.Edit3Exit(Sender: TObject);
begin
  try
    Kunci_e:=strtoint(edit3.text);
    Q:=(strtoint(edit1.Text)-1) * (strtoint(edit2.Text)-1);
    if (kunci_e <= 1) or (kunci_e >= Q) then
    begin
      messagedlg ('Kunci e harus diantara 1 dan '+ IntToStr(Q)
                  +'.',
      mtinformation,[mbOk],0);
      edit3.SetFocus;
    end
    else if gcd(Kunci_e,Q) <> 1 then
    begin
      messagedlg (IntToStr(Kunci_e)+' tidak relatif prima terhadap
                  '+IntToStr(Q)+'.',
      mtinformation,[mbOk],0);
      edit3.SetFocus;
    end
    else if gcd(Kunci_e,Q) = 1 then
    begin
      proses_enkrip.Enabled:=true;
      proses_enkrip.SetFocus;
    end;
  except
    if (Edit3.Text = '') then
    begin
      ShowMessage('Nilai Kunci e belum terisi');
      Edit3.SetFocus;
    end;
  end;
end;

procedure TForm2.FormCreate(Sender: TObject);
begin
  statusbar1.Panels[0].Text:='Form Enkripsi Citra';
  statusbar1.Panels[1].Text:='insert';
  statusbar1.Panels[2].Text:='By : Rosalia S. Karo (995314063)';
```

```
  edit2.Text:='';
  edit3.Text:='';
  proses_enkrip.Enabled:=false;
  save.Enabled:=false;
  saveas.Enabled:=false;
end;

procedure TForm2.FormClose(Sender: TObject; var Action:
TCloseAction);
begin
  gambar_enkrip.Picture:=nil;
  gambar_asli.Picture:=nil;
  prima_1.Enabled:=false;
  prima_2.Enabled:=false;
  kunciumum_e.Enabled:=false;
  edit1.Text:='';
  edit2.Text:='';
  edit3.Text:='';
  proses_enkrip.Enabled:=false;
end;

procedure TForm2.SaveClick(Sender: TObject);
var myFile: File of myData;
    i: LongWord;
begin
  if (CurrentFile <> EmptyStr) then
  begin
    gambar_enkrip.Picture.SaveToFile(CurrentFile);
    if (jlh > 1) then
    begin
      AssignFile(myFile, CurrentFile + '.tmp');
      ReWrite(myFile);
      for i := 1 to jlh-1 do
      begin
        Gauge1.Progress := Round(i/(jlh-1) * 100);
        Write(myFile, nilai[i]);
      end;
      CloseFile(myFile);
    end;
    gauge1.Progress:=0;
  end
  else
    SaveAsClick(sender);
end;

procedure TForm2.SaveAsClick(Sender: TObject);
begin
  if savepicturedialog1.Execute then
  begin
    CurrentFile:=savepicturedialog1.FileName;
    SaveClick(sender);
  end;
  statusbar1.Panels[0].Text:=savepicturedialog1.FileName+'('+intto
  str(gambar_enkrip.Width)+'x'+inttostr(gambar_enkrip.Height)+')';
end;
```

```
procedure TForm2.proses_enkripClick(Sender: TObject);
var
   i,j,panjang,lebar:word;
begin
  proses_enkrip.Enabled:=false;
  scrollbox1.Hide;
  gambar_enkrip.Picture:=gambar_asli.Picture;
  panjang:=gambar_enkrip.Picture.height;
  lebar:=gambar_enkrip.Picture.width;
  luas := panjang * lebar;
  m1:=1;
  x:=1;
  jlh:=1;
  for i := 0 to lebar - 1 do
  begin
    for j := 0 to panjang - 1 do
    begin
      gauge1.Progress := round(m1/luas * 50);
      datawarna[x]:=getRvalue(gambar_enkrip.Canvas.Pixels[i,j]);
      datawarna[x+1]:=getGvalue(gambar_enkrip.Canvas.Pixels[i,j]);
      datawarna[x+2]:=getBvalue(gambar_enkrip.Canvas.Pixels[i,j]);
      if (datawarna[x] >= N) or (datawarna[x+1] >= N)
          or (datawarna[x+2] >= N) then
      begin
        nilai[jlh].xr := datawarna[x] div N;
        nilai[jlh].xg := datawarna[x+1] div N;
        nilai[jlh].xb := datawarna[x+2] div N;
        nilai[jlh].px := i;
        nilai[jlh].py := j;
        Inc(jlh);
      end;
      x:=x+3;
      inc(m1);
    end;
  end;
  sel := panjang * lebar * 3;
  x := 1;
  while x <= sel+1 do
  begin
    C := exp_encrypt(datawarna[x],Kunci_e,N);
    datawarna[x]:=C;
    x:=x+1;
  end;
  x:=3;
  for i := 0 to lebar - 1 do
  begin
    for j := 0 to panjang - 1 do
    begin
      gauge1.Progress := round(m1/luas * 50);
      gambar_enkrip.Canvas.Pixels[i,j]:=rgb(datawarna[x-
      2],datawarna[x-1],datawarna[x]);
      x:=x+3;
      inc(m1);
    end;
  end;
  gauge1.Progress:=0;
```

```
  scrollbox1.Show;
  edit1.Text:=' ';
  edit2.Text:=' ';
  edit3.Text:=' ';
  input_kunci.Visible:=false;
  save.Enabled:=true;
  saveas.Enabled:=true;
end;
end.
```

Program Dekripsi Citra

```
unit UnitDekripsi;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics,
  Controls, Forms, Dialogs, ComCtrls, StdCtrls, Buttons, ExtCtrls,
  ExtDlgs, Menus, Gauges;

const max=10000000;

type
  TForm3 = class(TForm)
    OpenPictureDialog1: TOpenPictureDialog;
    SavePictureDialog1: TSavePictureDialog;
    StatusBar1: TStatusBar;
    GroupBox1: TGroupBox;
    GroupBox2: TGroupBox;
    open1: TBitBtn;
    Label1: TLabel;
    Bevel1: TBevel;
    gambar_enkrip: TImage;
    Label2: TLabel;
    ScrollBox1: TScrollBox;
    gambar_dekrip: TImage;
    input_kuncidekrip: TGroupBox;
    prima_1: TLabel;
    prima_2: TLabel;
    kunciumum_e: TLabel;
    Edit1: TEdit;
    Edit2: TEdit;
    Edit3: TEdit;
    Panel1: TPanel;
    proses_dekripsi: TBitBtn;
    Gauge1: TGauge;
    Panel2: TPanel;
    save: TBitBtn;
    saveas: TBitBtn;
    procedure open1Click(Sender: TObject);
    procedure gambar_enkripClick(Sender: TObject);
    procedure gambar_dekripClick(Sender: TObject);
    procedure Edit1Exit(Sender: TObject);
    procedure Edit2Exit(Sender: TObject);
```

```
    procedure Edit3Exit(Sender: TObject);
    procedure FormShow(Sender: TObject);
    procedure saveClick(Sender: TObject);
    procedure saveasClick(Sender: TObject);
    procedure    FormClose(Sender:    TObject;    var    Action:
    TCloseAction);
    procedure proses_dekripsiClick(Sender: TObject);

  private
    { Private declarations }
  public
    { Public declarations }
  CurrentFile:string;
  end;
  mydata = record
    px, py: Word;
    xr, xg, xb: Byte;
  end;

var datawarna:array[1..max] of Longint;
    nilai: array [1..max] of myData;
    x,sel,luas,m1:Longint;
    Prima1, Prima2: boolean;
    Kunci_e,Kunci_d,N,Q:longint;
    M:Longint;
    Ftext: Boolean;
    jlh: LongWord;

  Form3: TForm3;

implementation

uses UnitMenuUtama,JPEG;

{$R *.dfm}


Function gcd(a,b:integer):integer;
var g,z:integer;
begin
  g:=b;
  If b<>0 then
  while g<>0 do
  begin
    z:=a mod g;
    a:=g;
    g:=z;
  end;
  result:=a;
end;

function invers_d(a,b: longint):longint;
var i,hasil_kali,hasil_bagi:longint;
begin
  i:=1;
  while (i < b+1) do
```

```
  begin
    hasil_kali:=i * a;
    hasil_bagi:=hasil_kali mod b;
    if hasil_bagi = 1 then
       break;
    inc(i,2);
  end;
  result:=i;
end;

Function Prima(X: longint): Boolean;
var Y, yn : longint;
    dibagi : boolean;
begin
  dibagi := false;
  if ((X mod 2 = 0) And (X <> 2)) or (X = 1) or (X <= 0) then
     dibagi := true
  else
  begin
    Y := 3;
    yn := trunc(sqrt(X)) + 1;
    while (Y < yn) do
    begin
      if (X mod Y = 0) then
      begin
        dibagi := true;
        break;
      end;
      inc(Y,2);
    end;
  end;
  Prima := Not dibagi;
end;

function exp_decrypt(C,e,n:longint):longint;
var M1 : longint;
    i,k:integer;
    e_biner : array[1..32] of integer;
begin
  k:=1;
  while (e <> 0) do
  begin
    if (e and 1 = 1) then
    begin
      e_biner[k] := 1;
    end
    else
    begin
      e_biner[k] := 0;
    end;
  e := e shr 1;
  inc(k);
  end;
  dec(k);
  M1:=1;
  for i:=k downto 1 do
```

```
  begin
    M1 := (M1 * M1) mod n;
    if (e_biner[i] = 1) then
        M1 := (M1 * C) mod n;
  end;
  result := M1;
end;

procedure TForm3.open1Click(Sender: TObject);
var myFile: File of myData;
begin
  if openpicturedialog1.Execute then
  begin
    try
      gambar_enkrip.Picture.LoadFromFile(openpicturedialog1.FileNa
      me);
      AssignFile(myFile, openpicturedialog1.FileName + '.tmp');
      Ftext := True;
      jlh  := 1;
      Reset(myFile);
      while not eof(myFile) do
      begin
        Read(myFile,nilai[jlh]);
        Seek(myFile, jlh);
        Inc(jlh);
      end;
      CloseFile(myFile);
    except
      Ftext := False;
    end;
    Statusbar1.Panels[0].Text:=openpicturedialog1.FileName+'('+int
    tostr(gambar_enkrip.Width)+'x'+inttostr(gambar_enkrip.Height)+
    ')';
    input_kuncidekrip.Visible:=true;
    prima_1.Enabled:=true;
    prima_2.Enabled:=true;
    kunciumum_e.Enabled:=true;
    Edit1.Text:='';
    Edit2.Text:='';
    Edit3.Text:='';
    save.Enabled:=false;
    saveas.Enabled:=false;
    Edit1.SetFocus;
  end;
end;

procedure TForm3.gambar_enkripClick(Sender: TObject);
begin
  bevel1.BringToFront;
  gambar_dekrip.BringToFront;
end;

procedure TForm3.gambar_dekripClick(Sender: TObject);
begin
  gambar_dekrip.BringToFront;
end;
```

```
procedure TForm3.Edit1Exit(Sender: TObject);
begin
  try
    Prima1:=Prima(strtoint(edit1.Text));
    if not Prima1  then
    begin
      messagedlg('Angka      '+edit1.Text+'      bukan      bilangan
                  prima'+chr(13),
      mtwarning,[mbOk],0);
      edit1.SetFocus;
    end;
  except
    edit1.SetFocus;
  end;
end;

procedure TForm3.Edit2Exit(Sender: TObject);
begin
  try
    Prima2:=Prima(strtoint(edit2.text));
    N:=strtoint(edit1.Text) * strtoint(edit2.Text);
    if (StrToInt(Edit1.Text) = StrToInt(Edit2.Text)) then
    begin
      messagedlg('Nilai p dan q harus berbeda.'+chr(13),
      mtwarning,[mbOk],0);
      edit1.SetFocus;
    end
    else if not Prima2 then
    begin
      messagedlg('Angka      '+edit2.Text+'      bukan      bilangan
                  prima'+chr(13),
      mtwarning,[mbOk],0);
      edit2.SetFocus;
    end
    else if (N > 256) then
    begin
      messagedlg('Hasil kali bilangan prima1 dan prima2 harus <
                  256!',mtwarning,[mbOK],0);
      edit1.SetFocus;
    end;
  except
    edit2.SetFocus;
  end;
end;

procedure TForm3.Edit3Exit(Sender: TObject);
begin
  try
    Kunci_e:=strtoint(edit3.text);
    Q:=(strtoint(edit1.Text)-1) * (strtoint(edit2.Text)-1);
    if (kunci_e <= 1) or (kunci_e >= Q) then
    begin
      messagedlg ('Kunci e harus diantara 1 dan '+ IntToStr(Q)
                   +'.',mtinformation,[mbOk],0);
      edit3.SetFocus;
    end
```

```
    else if gcd(Kunci_e,Q) <> 1 then
    begin
      messagedlg (IntToStr(Kunci_e)+' tidak relatif prima terhadap
                  '+IntToStr(Q)+'.',
      mtinformation,[mbOk],0);
      edit3.SetFocus;
    end
    else if gcd(Kunci_e,Q) = 1 then
    begin
      proses_dekripsi.Enabled:=true;
      proses_dekripsi.SetFocus;
    end;
  except
    if (Edit3.Text = '') then
    begin
      ShowMessage('Nilai Kunci e belum terisi');
      Edit3.SetFocus;
    end;
  end;
end;

procedure TForm3.FormShow(Sender: TObject);
begin
  gambar_dekrip.Picture:=nil;
  gambar_enkrip.Picture:=nil;
  input_kuncidekrip.Visible:=false;
  prima_1.Enabled:=false;
  prima_2.Enabled:=false;
  kunciumum_e.Enabled:=false;
  edit1.Text:='';
  edit2.Text:='';
  edit3.Text:='';
  proses_dekripsi.Enabled:=false;
  save.Enabled:=false;
  saveas.Enabled:=false;
end;

procedure TForm3.saveClick(Sender: TObject);
begin
  if CurrentFile <> EmptyStr then
     gambar_dekrip.Picture.SaveToFile(savepicturedialog1.FileName)
  else
     saveasClick(sender);
end;

procedure TForm3.saveasClick(Sender: TObject);
begin
  if savepicturedialog1.Execute then
  begin
    CurrentFile:=savepicturedialog1.FileName;
    saveclick(sender);
  end;
  statusbar1.Panels[0].Text:=savepicturedialog1.FileName+'('+intto
  str(gambar_dekrip.Width)+'x'+inttostr(gambar_dekrip.Height)+')';
end;
```

```
      begin
        datawarna[x-2] := datawarna[x-2]+nilai[k].xr*N;
        datawarna[x-1] := datawarna[x-1]+nilai[k].xg*N;
        datawarna[x] := datawarna[x]+nilai[k].xb*N;
        Inc(k);
        if (k = jlh) then
           Ftext := False;
      end;
      gambar_dekrip.Canvas.Pixels[i,j]:=rgb(datawarna[x-
      2],datawarna[x-1],datawarna[x]);
      gauge1.Progress:=round(m1/luas * 50);
      x:=x+3;
      inc(m1);
    end;
  end;
  gauge1.Progress:=0;
  scrollbox1.Show;
  edit1.Text:=' ';
  edit2.Text:=' ';
  edit3.Text:=' ';
  input_kuncidekrip.Visible:=false;
  save.Enabled:=true;
  saveas.Enabled:=true;
end;
end.
```

Program Cek File Citra

```
unit UnitCekFileCitra;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics,
  Controls, Forms, Dialogs, ExtDlgs, StdCtrls, Buttons, ExtCtrls,
  ComCtrls, Gauges, JPEG;

type
  TForm6 = class(TForm)
    OpenPictureDialog1: TOpenPictureDialog;
    GroupBox1: TGroupBox;
    GroupBox2: TGroupBox;
    open_gambar1: TBitBtn;
    open_gambar2: TBitBtn;
    Label1: TLabel;
    Label2: TLabel;
    ScrollBox1: TScrollBox;
    ScrollBox2: TScrollBox;
    gambar_1: TImage;
    gambar_2: TImage;
    file_1: TLabel;
    file_2: TLabel;
    StatusBar1: TStatusBar;
    ket_benar: TLabel;
    ket_salah: TLabel;
```

```
    Panel1: TPanel;
    cekfilecitra: TBitBtn;
    procedure open_gambar1Click(Sender: TObject);
    procedure open_gambar2Click(Sender: TObject);
    procedure cekfilecitraClick(Sender: TObject);
    procedure FormShow(Sender: TObject);
    procedure     FormClose(Sender:     TObject;     var     Action:
    TCloseAction);
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form6: TForm6;
  datawarna_gambar1: Array[1..(300 * 300)] of integer;
  datawarna_gambar2: Array[1..(300 * 300)] of integer;
  m1, luas : integer;
implementation

{$R *.dfm}

function JPEG2Bitmap(JPEGFile : String) : TBitmap;
var
   myJPEG:TJPEGImage;
begin
  myJPEG := TJPEGImage.Create ;
  try
    myJPEG.LoadFromFile(JPEGFile);
    Result := TBitmap.Create;

    myJPEG.PixelFormat:=pf24bit;
  finally
    myJPEG.Free;
  end;
end;

procedure TForm6.open_gambar1Click(Sender: TObject);
var
   myBmp: TBitmap;
   temp: String;
begin
  if   OpenPictureDialog1.Execute then
  begin
//     Load JPEG Image for Display
    temp := openpicturedialog1.FileName;
    try
      myBmp := JPEG2Bitmap(temp);
      gambar_1.Picture.Graphic:=myBmp;
    except
//     Load Bitmap Image for Display
      gambar_1.Picture.LoadFromFile(temp);
    end;
      file_1.Visible:=true;
```

```
      file_1.Caption:=OpenPictureDialog1.FileName;
  end;
end;

procedure TForm6.open_gambar2Click(Sender: TObject);
var
   myBmp: TBitmap;
   temp: String;
begin
  if   OpenPictureDialog1.Execute then
  begin
//     Load JPEG Image for Display
    temp := openpicturedialog1.FileName;
    try
      myBmp := JPEG2Bitmap(temp);
      gambar_2.Picture.Graphic:=myBmp;
    except
//     Load Bitmap Image for Display
      gambar_2.Picture.LoadFromFile(temp);
    end;
      file_2.Visible:=true;
      file_2.Caption:=OpenPictureDialog1.FileName;
      cekfilecitra.Enabled:=true;
  end;
end;

procedure TForm6.cekfilecitraClick(Sender: TObject);
var i, j, jlh_p, lebar1, tinggi1 : integer;
begin
  cekfilecitra.Enabled:=false;

  lebar1:=gambar_1.Width;
  tinggi1:=gambar_1.Height;

  luas:=lebar1 * tinggi1;
  jlh_p:=1;
  for i := 0 to lebar1 - 1 do
  begin
    for j := 0 to tinggi1 - 1 do
    begin
      datawarna_gambar1[jlh_p] := gambar_1.Canvas.Pixels[i,j];
      datawarna_gambar2[jlh_p] := gambar_2.Canvas.Pixels[i,j];
      if((getRvalue(datawarna_gambar1[jlh_p]))<>
         (getRvalue(datawarna_gambar2[jlh_p]))) or
        ((getGvalue(datawarna_gambar1[jlh_p]))<>
         (getGvalue(datawarna_gambar2[jlh_p]))) or
        ((getBvalue(datawarna_gambar1[jlh_p]))<>
         (getBvalue(datawarna_gambar2[jlh_p]))) then
        begin
          MessageDlg('Gambar   I   tidak   sama   dengan   Gambar
          II',mtInformation,[mbOk], 0);
          exit;
        end
      else
        begin
```

```
          MessageDlg('Gambar I sama dengan Gambar
          II',mtInformation,[mbOk], 0);
          exit;
        end;
      inc(jlh_p);
    end;
  end;
  file_1.Caption:='';
  file_2.Caption:='';
end;

procedure TForm6.FormShow(Sender: TObject);
begin
  gambar_1.Picture:=nil;
  gambar_2.Picture:=nil;
  cekfilecitra.Enabled:=false;
end;

procedure TForm6.FormClose(Sender: TObject; var Action:
TCloseAction);
begin
  gambar_1.Picture:=nil;
  gambar_2.Picture:=nil;
  cekfilecitra.Enabled:=false;
  file_1.Caption:='';
  file_2.Caption:='';
end;

procedure TForm6.FormCreate(Sender: TObject);
begin
  statusbar1.Panels[0].Text:='Form Pengecekan File Citra Asli
                              Dengan File Citra Hasil Penyandian';
  statusbar1.Panels[1].Text:='insert';
  statusbar1.Panels[2].Text:='By : Rosalia S. Karo (995314063)';
end;
end.
```

Program Contents

```
unit Unit5;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics,
  Controls, Forms, Dialogs, StdCtrls, ExtCtrls, Buttons;

type
  TForm5 = class(TForm)
    Panel1: TPanel;
    Label1: TLabel;
    Label2: TLabel;
    BitBtn1: TBitBtn;
    Memo1: TMemo;
    procedure BitBtn1Click(Sender: TObject);
```

```
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form5: TForm5;

implementation

uses UnitMenuUtama;

{$R *.dfm}

procedure TForm5.BitBtn1Click(Sender: TObject);
begin
  form5.Hide;
  form1.show;
end;
end.
```

Program About_Us

```
unit UnitAbout;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics,
  Controls, Forms, Dialogs, ExtCtrls, jpeg, StdCtrls, Buttons;

type
  TForm4 = class(TForm)
    Panel1: TPanel;
    BitBtn1: TBitBtn;
    Label1: TLabel;
    Label2: TLabel;
    Image1: TImage;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    Label6: TLabel;
    Label7: TLabel;
    procedure BitBtn1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form4: TForm4;

implementation
uses UnitMenuUtama;
```

```
{$R *.dfm}

procedure TForm4.BitBtn1Click(Sender: TObject);
begin
  form4.Hide;
  form1.show;
end;
end.
```