BAB 4
IMPLEMENTASI DAN PENGUJIAN SISTEM

Tahap implementasi dan pengujian sistem adalah tahap dimana akan dilakukannya implementasi hasil analisis serta pengujian terhadap sistem. Tahap ini dilakukan setelah tahap analisis dan perancangan sistem., berguna untuk melihat apa saja kekurangan yang terdapat pada alikasi yang dibangun untuk pengembangan aplikasi.

4.1 Implementasi

Merupakan tahap menerjemahkan perancangan yang telah dilakuakan pada tahap sebelumnya yang bertujuan untuk melihat respon pengguna terhadap aplikasi yang telah dibangun, respon dari pengguna akan dijadikan masukan bagi pengembang aplikasi. Selain itu kebutuhan perangkat lunak dan kebutuhan perangkat keras menjadi pembahasan pada tahap ini.

4.1.1 Lingkungan Implementasi

Dalam rangka menerapkan aplikasi yang telah dibuat maka dibutuhkan lingkungan yang sesuai untuk aplikasi tersebut. Lingkungan yang dibutuhkan meliputi penggunaan perangkat keras dan perangkat lunak untuk menerapkan aplikasi ini.

4.1.1.1 Implementasi Perangkat Keras yang Digunakan

Implementasi perangkat keras yang digunakan untuk menjalankan Aplikasi Kriptografi pada Citra Digital dapat dilihat seperti berikut :

Tabel 4.1 Perangkat keras yang digunakan

No	Perangkat Keras	Spesifikasi
1	Komputer	Hardisk 500 GB
		Prosesor 2,5 GB
		VGA 4 GB
		Memory RAM 2 GB

No	Perangkat Keras	Spesifikasi
		Monitor LCD dengan resolusi 1024x768
		Keyboard Standard port USB
		Mouse Standard port USB

4.1.1.2 Implementasi Perangkat Lunak yang Digunakan

Implementasi perangkat lunak yang digunakan untuk menjalankan Aplikasi Kriptografi pada Citra digital dapat dilihat pada seperti berikut :

Tabel 4.2 Perangkat lunak yang digunakan

No	Perangkat Lunak	Spesifikasi
1	Sistem Operasi	Microsoft Windows 7 dan Microsoft Windows 8.1

4.1.2 Batasan Implementasi

Pembatasan implementasi dimaksudkan agar ruanglingkup implementasi lebih jelas. Batasan implementasi untuk aplikasi kriptografi pada citra digital adalah sebagai berikut :

1. Perangkat lunak yang digunakan tidak berbasis client-server
2. Gambar yang digunakan sebagai masukan adalah gambar yang berektensi .bmp dan .jpg
3. Aplikasi ini tidak menangani pengiriman password untuk klien
4. Aplikasi ini menggunakan bahasa C# untuk pembangunannya
5. Aplikasi yang digunakan berupa *exutable* (.exe) tanpa *installer*

4.1.3 Implementasi Program

Aplikasi kriptografi pada citra digital ini merupakan aplikasi yang dapat digunakan secara langsung tanpa melalui proses instalasi. File *executable* yang ada dapat langsung digunakan.

Implementasi program ini dapat terlihat dari implementasi antarmuka yang disesuaikan dengan alur kerja dari program. Dalam implementasi antarmuka akan terlihat fungsional yang berjalan dari rancangan program sebelumnya.

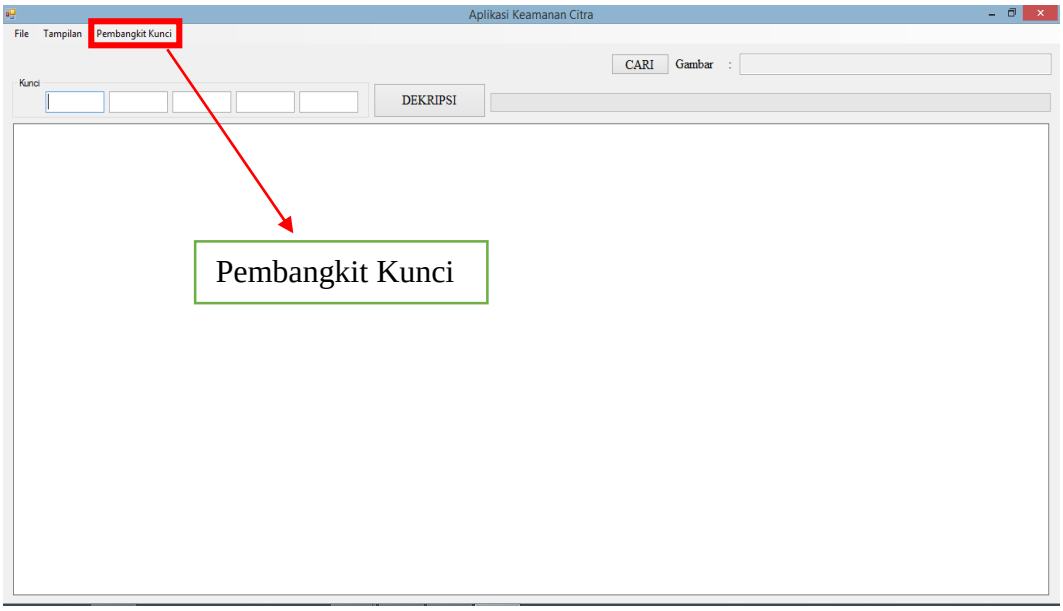
4.1.4 Implementasi Antarmuka

Dalam implementasi antarmuka ini akan terlihat alur proses kerja dari aplikasi kriptografi pada citra digital yang dibuat. Penyajian implementasi antar muka ini mengikuti tahapan proses yang terjadi pada program.

Antarmuka dari aplikasi ini terdiri dari dua bagian yaitu antramuka enkripsi dan antarmuka dekripsi. Antarmuka dekripsi memiliki satu antarmuka turunan yaitu antarmuka pembangkit kunci. Berikut rincian dari implementasi antarmuka yang dibuat:

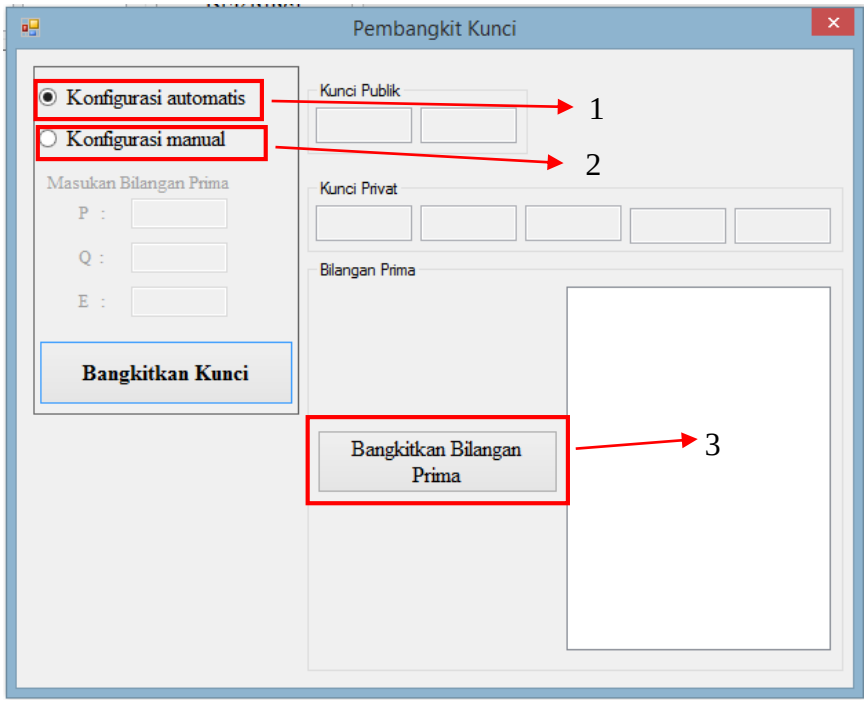
4.1.4.1 Implementasi Antarmuka Pembangkit Kunci

Antarmuka pembangkit kunci merupakan antarmuka yang berada pada antarmuka utama dekripsi. Antarmuka ini disajikan pertama kali karena untuk melakukan proses enkripsi dan dekripsi dibutuhkan kunci yang dibangkitkan dalam form antarmuka ini. Berikut merupakan tampilan antarmuka pembangkit kunci :



Bagan 4.1Menu Pembangkit Kunci pada Antarmuka Dekripsi

Setelah kita pilih menu tersebut selanjutnya akan terlihat tampilan antarmuka pembangkit kunci sebagai berikut :

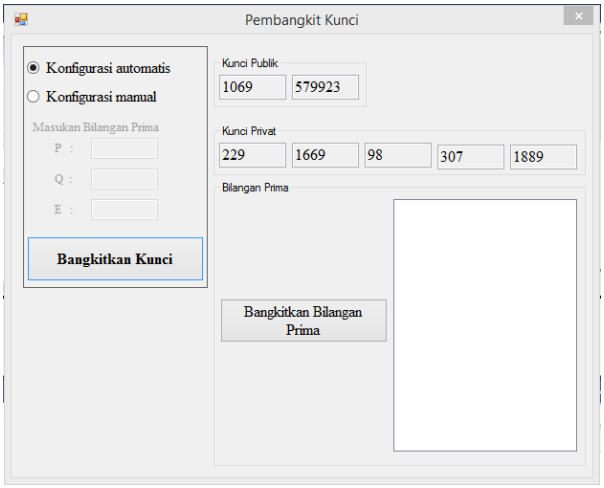


Bagan 4.2 Antarmuka Pembagkit Kunci

Pembangkit kunci ini merupakan hasil implementasi dari algoritma pembangkit kunci yang telah dibahas di bab 3 (3.1.3.1) pada tabel 3.1 (algoritma pembangkit kunci RSA – CRT). Dari tampilan diatas terlihat bahwa ada dua metode pembangkit kunci. Yaitu sebagai berikut :

1. Konfigurasi automatis

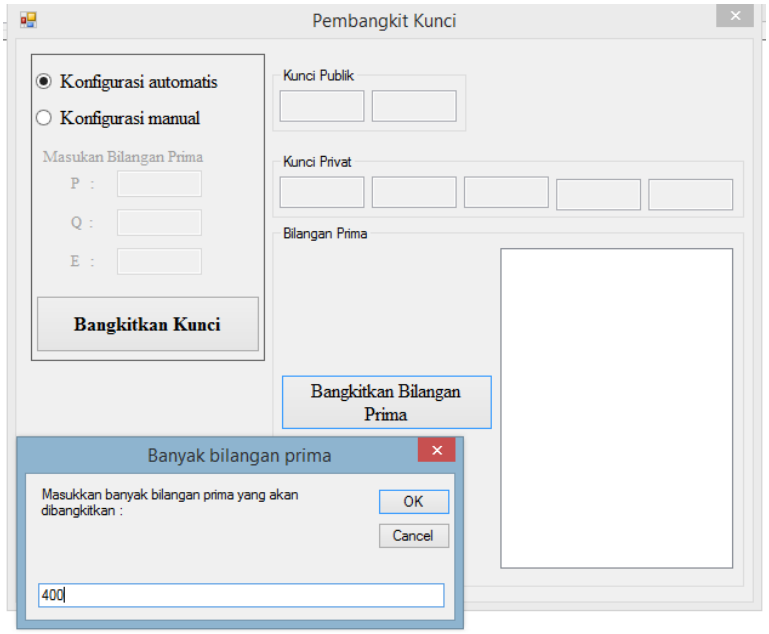
   Konfigurasi automatis ini akan membangkin nilai P, Q dan E secara acak sehingga pengguna tidak perlu memasukan nilai P, Q dan E. Dengan fasilitas ini maka pengguna dapat langsung membangkitkan kunci. Berikut merupakan hasil dari konfigurasi automatis sebagai berikut :



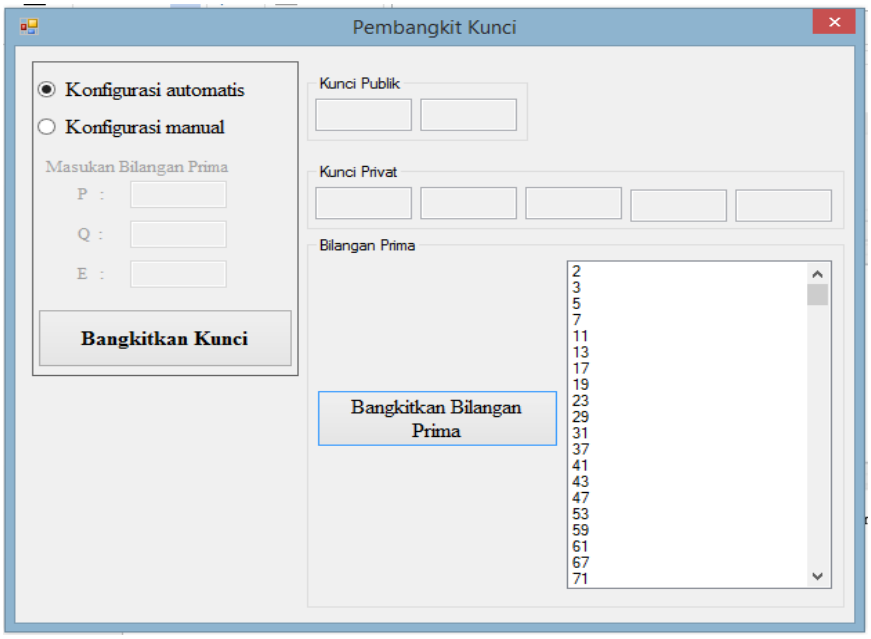
Bagan 4.3 Pembangkit kunci dengan konfigurasi automatis

2. Pembangkit bilangan prima

   Pembangkit merupakan fasilitas yang digunakan untuk membantu pengguna dalam membangkitkan bilangan prima. Mengingat tidak semua pengguna ingat tentang bilangan prima. Berikut hasil dari pembangkit bilangan prima :

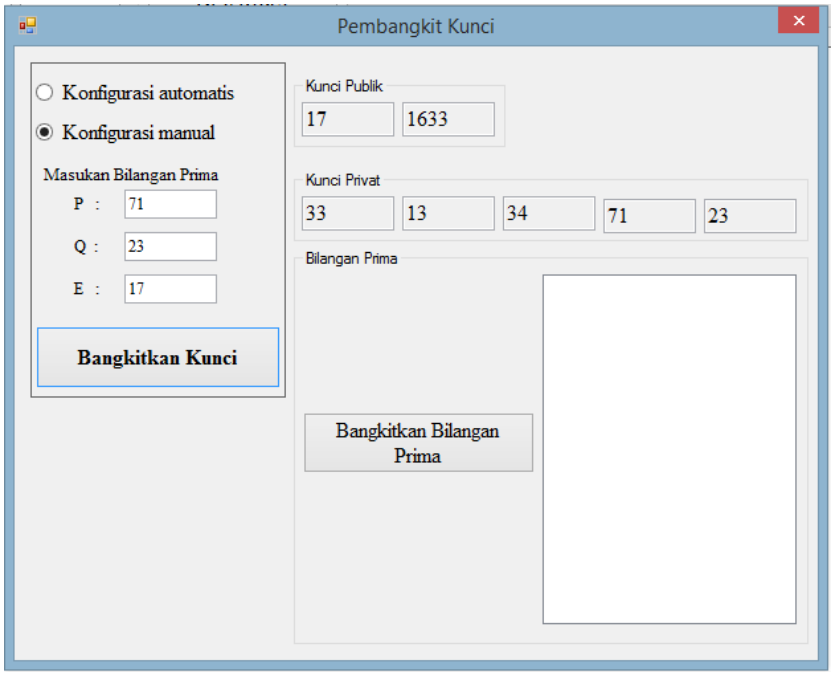


Bagan 4.4 Antarmuka masukan bilangan prima



Bagan 4.5 Antarmuka pembangkit bilangan prima

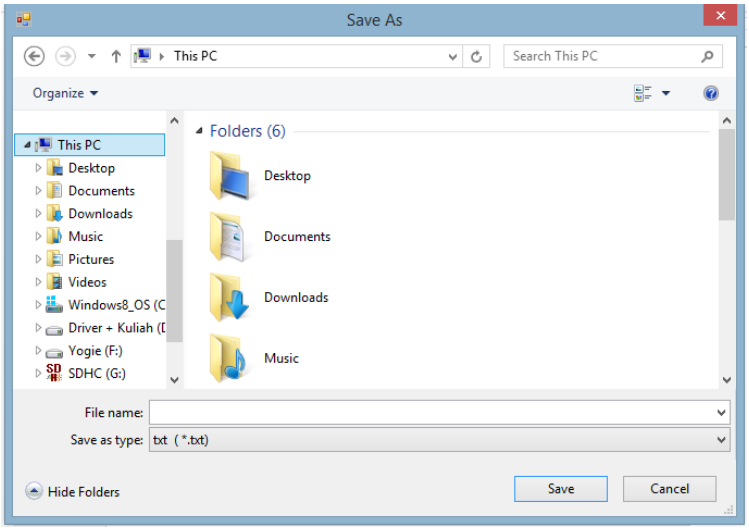
3. Konfigurasi manual ini disediakan sebagai fasilitas tambahan jika dirasa kunci yang dihasilkan melalui proses konfigurasi automatis kurang kuat. Dengan fasilitas ini bilangan prima untuk P, Q dan E yang dibangkitkan dapat beragam dan lebih besar konfigurasi manual. Berikut gambar pembangkit kunci dengan konfigurasi manual :



Bagan 4.6 Pembangkit kunci konfigurasi manual

4.1.4.1.1 Penyimpanan Kunci

Setelah kunci dibanngkitkan maka akan ada antarmuka pilihan untuk melakukan penyimpanan kunci yang telah didapatkan. Hal ini berlaku untuk konfigurasi automatis dan konfigurasi manual. Berikut tampilan antarmuka dari penyimpanan tersebut :



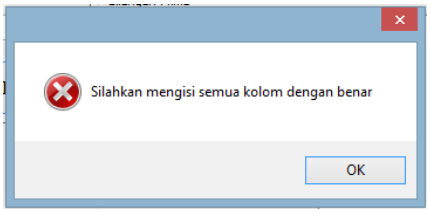
Bagan 4.7 Antarmuka penyimpanan kunci

4.1.4.1.2 Antarmuka Penanganan Masalah Pembangkit Kunci

Terdapat beberapa masalah saat melakukan pembangkitan kunci. Masalah itu antara lain sebagai berikut :

1. Kolom P, Q atau E tidak terisi

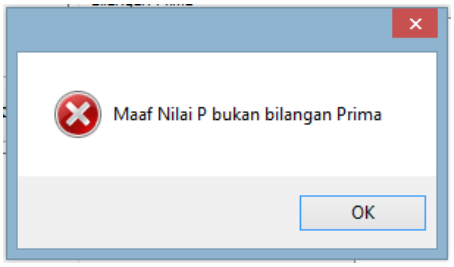
   Sewaktu membangkitkan kunc dengan konfigurasi manual ada kemungkinan bahwa pengguna langsung mengaktifkan tombol pembangkit kunci. Jika hal ini dilakukan tanpa mengisi nilai P, Q dan E maka akan program akan terganggu untuk menangani hal tersebut maka dilakukan penanganan sebagai berikut :



Bagan 4.8 Antarmuka nilai tidak terisi

2. Nilai P, Q atau E bukan bilangan prima

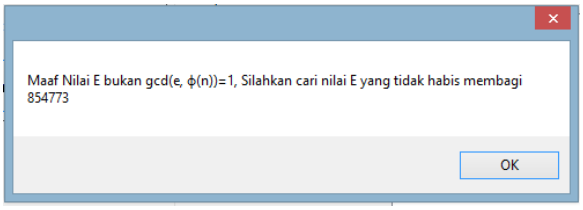
Untuk mengisi nilai P, Q atau E dibutuhkan bilangan prima maka jika nilai yang dimasukan bukan bilangan prima akan ada pesan kesalahan sebagai berikut :



Bagan 4.9 Antarmuka Bukan bilangan prima

3. Antarmuka gcd(e, ϕ(n))=1

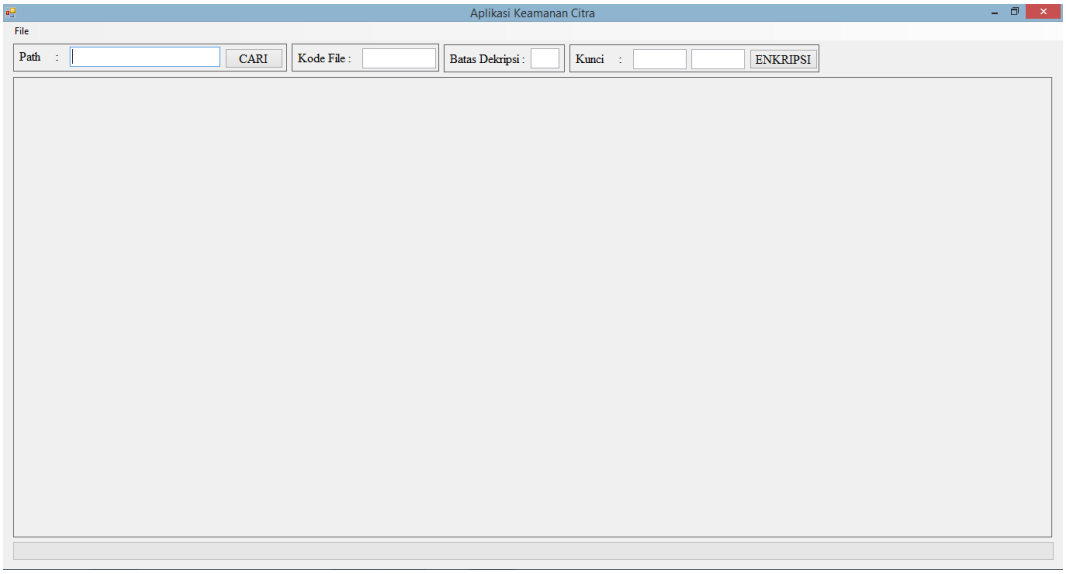
   Ketika nilai E adalah bukan prima relatif dimana merupakan salah satu faktor penyusun ϕ(n) maka akan ditampikan pesan error sebagai berikut :



Bagan 4.10 Antarmuka gcd(e, ϕ(n))=1

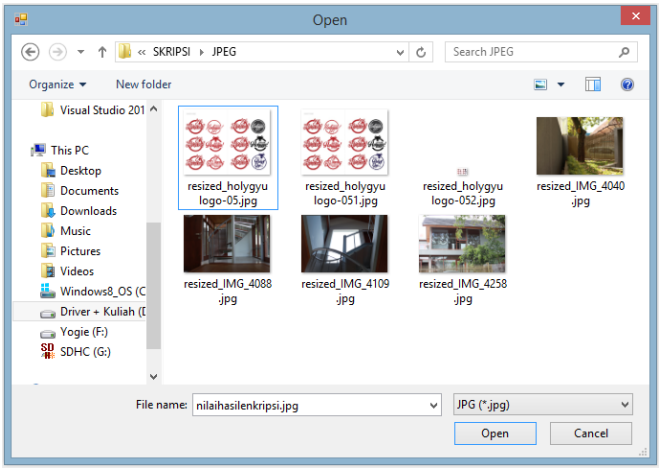
4.1.4.2 Implementasi Antarmuka Enkripsi

Setelah melakukan pembangkitan kunci maka perusahaan bisa melakukan enkripsi terhadap gambar yang diinginkan dengan menggunakan kunci publik. Berikut adalah tampilan antarmuka enkripsi :



Bagan 4.11 Antarmuka Enkripsi

langkah selanjutnya adalah pemilihan gambar yang akan dilakukan enkripsi :



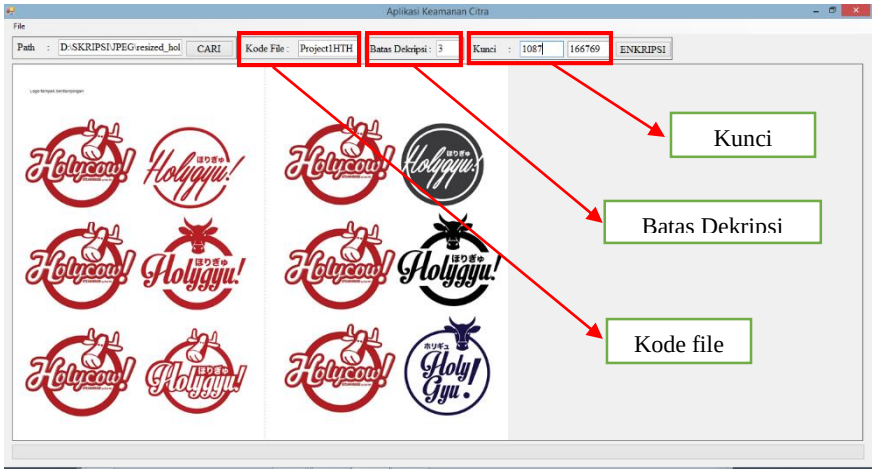
Bagan 4.13 Antarmuka Pemilihan Gambar

dan hasilnya akan dipilih gambar berikut sebagai bahan untuk dilakukan Enkripsi :



Bagan 4.12 Gambar yang akan dienkripsi

langkah selanjutnya adalah kita memasukan semua isian dari tiap kolom dekripsi sebagai berikut :



Bagan 4.14 Pengisian nilai-nilai bahan enkripsi

1. Kode File

   Kode file dapat berupa isian apapun yang merepresentasikan file yang akan dienkripsi dan harus secara unik. Unik berarti setiap gambar yang akan dienkripsi harus memiliki kode yang berbeda.

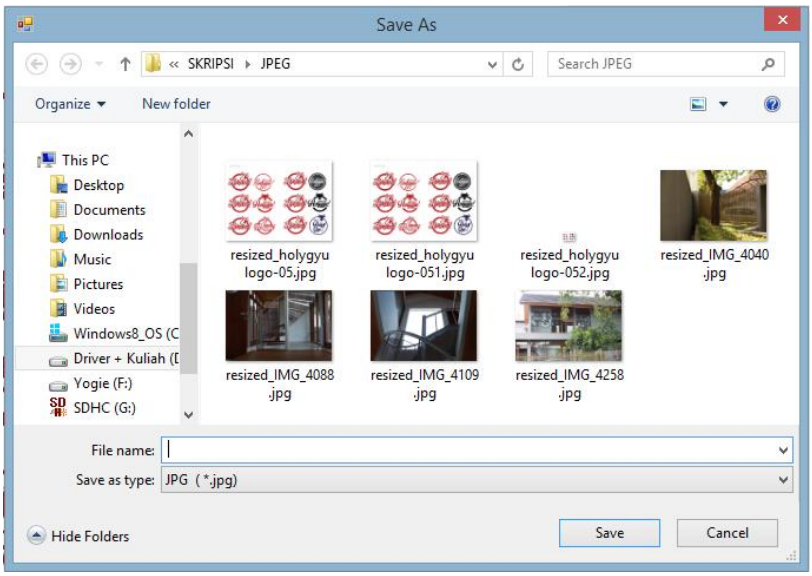
2. Batas Dekripsi

   Batas Dekripsi digunakan untuk menentukan berapa kali orang lain dapat melakukan dekripsi terhadap file citra terenkripsi tersebut.

3. Kunci

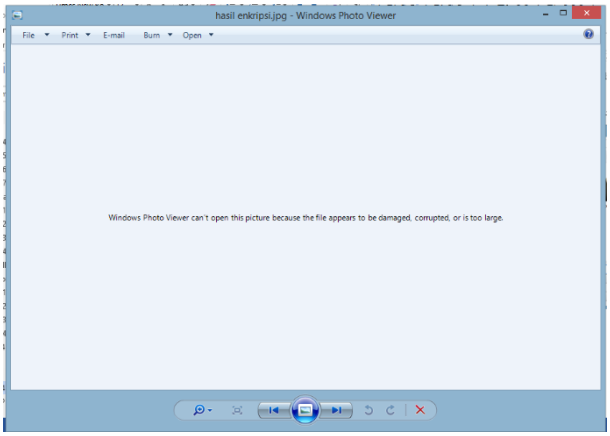
   Kunci berisi kunci publik yang telah dibangkitkan sebelumnya.

Setelah dilakukan enkripsi maka gambar akan disimpan sebagai berikut :



Bagan 4.15 Antarmuka Penyimpanan file hasil enkripsi

Dan hasil enkripsi gambar yang tidak bisa dibuka adalah sebagai berikut :



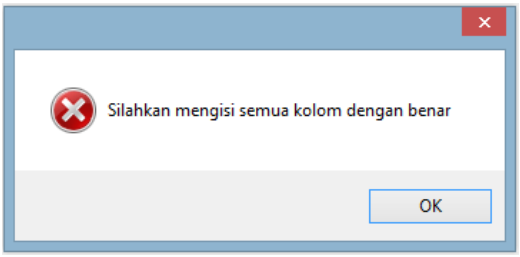
Bagan 4.16 Gambar hasil enkripsi

4.1.4.2.1 Antarmuka Penanganan Masalah Enkripsi

Terdapat beberapa masalah saat melakukan enkripsi. Masalah itu antara lain sebagai berikut :

1. Masalah nilai yang kosong

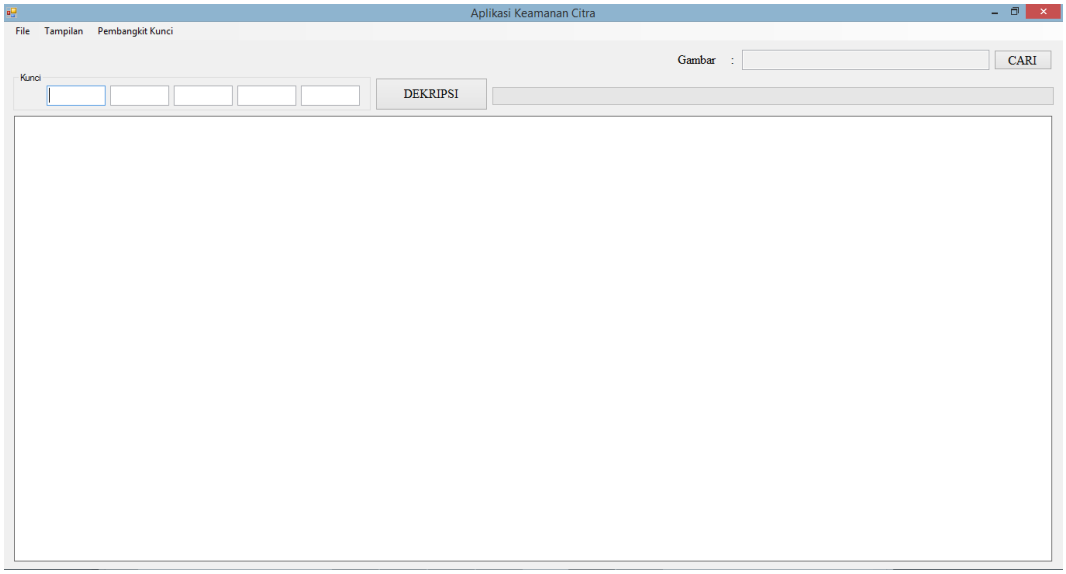
Masalah utama dalam proses enkripsi adalah salah satu kolom tidak diisi dan untuk menangani masalah tersebut maka dilakukan penanganan sebagai berikut :



Bagan 4.17 Antarmuka isian nilai kosong enrkipsi

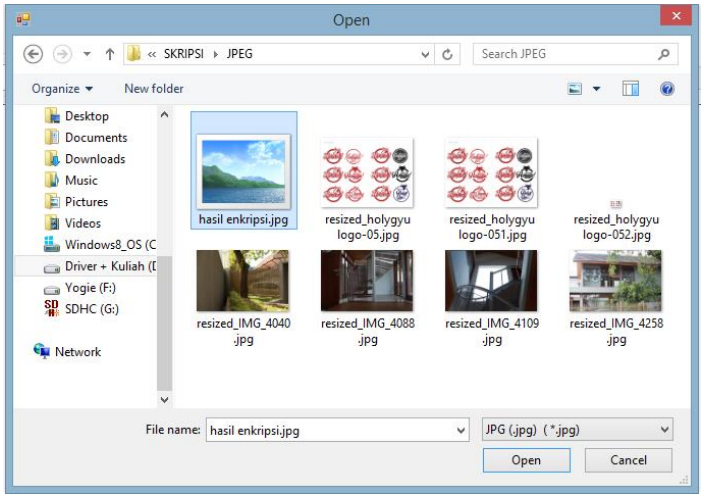
4.1.5 Implementasi Antarmuka Dekripsi

Setelah dilakukan enkripsi maka gambar akan dikirimkan kepada klien. Untuk melihat gambar klien harus memiliki aplikasi dekripsi dan kunci privat. Berikut tampilan antarmuka dekripsi :



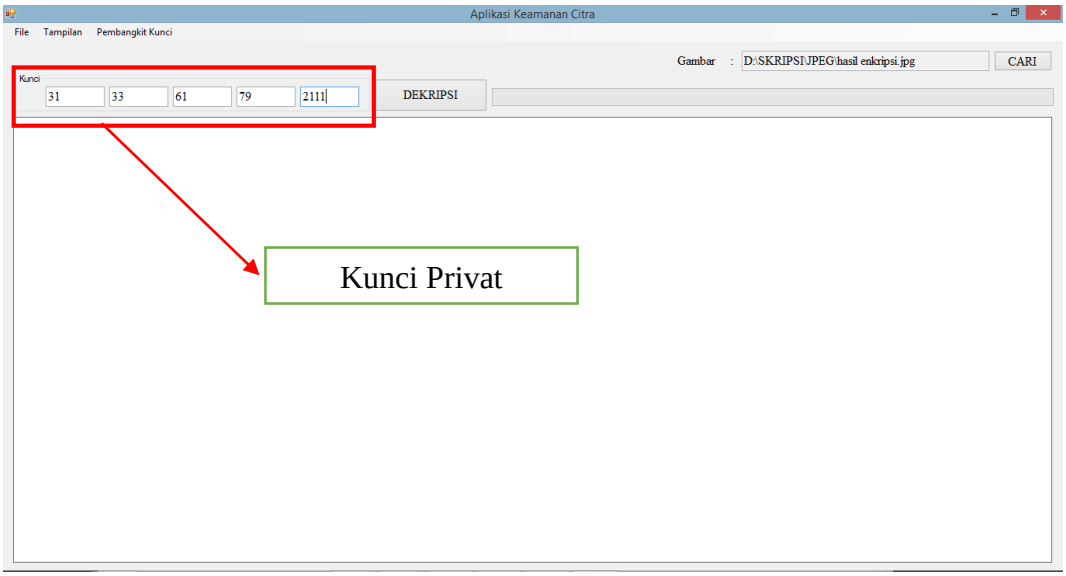
Bagan 4.18 Tampilan antarmuka dekripsi

selanjutnya adalah kita akan mencari gambar yang akan dilakukan dekripsi sebagai berikut :



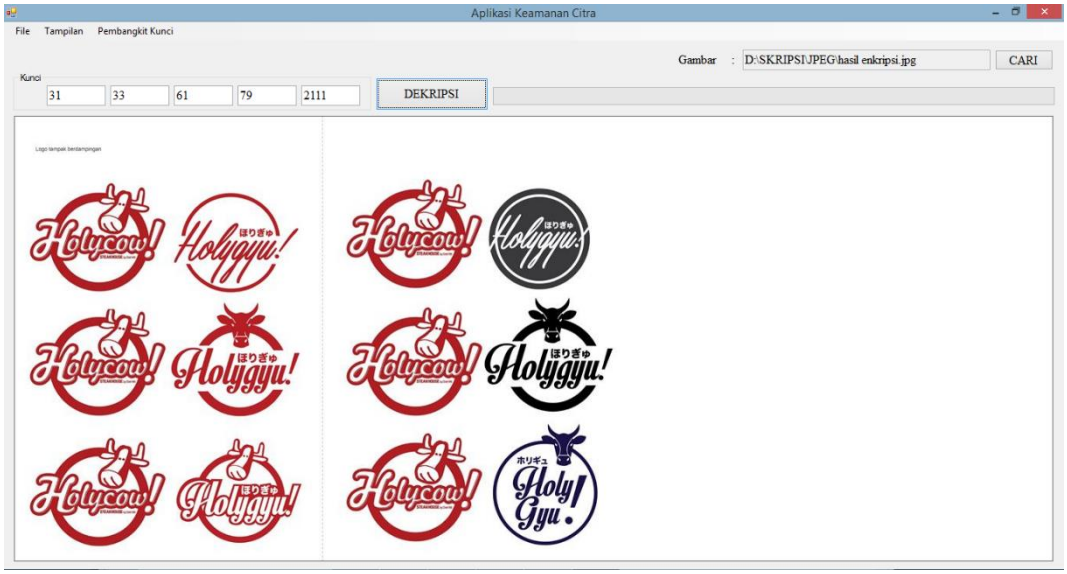
Bagan 4.20 Pemilihan gambar untuk didekripsi

selanjutnya kita akan mengisi kunci private yang ada pada form dekripsi sebagai berikut :



Bagan 4.19 Pengisian kunci dekripsi

Setelah melakukan memasukan kunci privat maka akan dilakukan proses dekripsi pada gambar yang dipilih dan hasilnya adalah sebagai berikut :



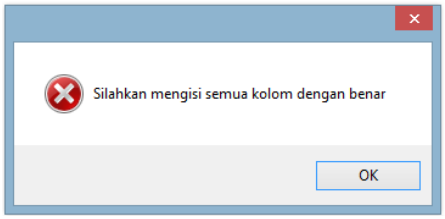
Bagan 4.21 Hasil Proses Dekripsi

4.1.5.1 Antarmuka Penanganan Masalah Dekripsi

Dalam proses dekripsi terdapat beberapa masalah yang mungkin terjadi. Antarmuka penanggulangan masalah yang terjadi seperti berikut :

1. Masalah nilai kosong

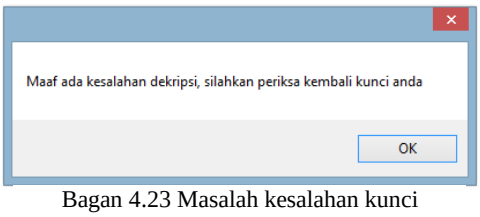
   Masalah dalam proses dekripsi adalah salah satu kolom tidak diisi dan untuk menangani masalah tersebut maka dilakukan penanganan sebagai berikut :



Bagan 4.22 Antanmuka penanganan nilai kosong dekripsi

2. Masalah kesalahan kunci

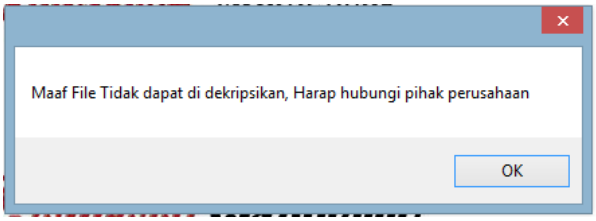
   Terdapat kemungkinan kesalahan dalam memasukan kunci privat. Berikut merupakan antarmuka penangan kesalahan kunci :



Bagan 4.23 Masalah kesalahan kunci

3. Masalah kesalahan melebihi batas dekripsi

   Terdapat masalah lain yaitu ketika proses dekripsi mencapai batas dekripsi yang ditentukan. Berikut antarmuka penanganan kesalahan ketika proses dekripsi melebihi batas :



Bagan 4.24 Antarmuka kelebihan batas dekripsi

4.2 Pengujian Sistem

Tahap Pengujian Sistem merupakan satu tahap yang penting dimana aplikasi yang dibangun diuji untuk menemukan kesalahan atau kerusakan yang ada. Pengujian yang dilakukan bertujuan untuk mengetahui apakan aplikasi yang dibangun telah sesuai atau tidak. Pengujian yang dilakukan dalam penelitian ini adalah pengujian Black Box dan Pengujian Beta terhadap Aplikasi Kriptografi Pada Citra Digital berikut hasil ujinya.

4.2.1 Rencana Pengujian BlackBox

Rencana pengujian yang dilakukan pada Aplikasi Kriptografi Pada Citra Digital sebagai berikut :

Tabel 4.3 Rencana Pengujian Aplikasi

Pengujian Aplikasi			
NO	**Komponen Yang Diuji**	**Skenario Pengujian**	**Hasil Yang Diharapkan**
1.	Halaman Pembangkit Kunci	Memasukan Banyak Nilai Bilangan Prima Yang Dibangkitkan	*Black Box*
		Memasukan Nilai Prima Untuk P, Q dan E	*Black Box*
		Memasukan Nilai P, Q dan E	*Black Box*
2.	Halaman Enkripsi	Memasukan Alamat Gambar, Kode File, Batas Dekripsi, Kunci	*Black Box*

Pengujian Aplikasi			
NO	**Komponen Yang Diuji**	**Skenario Pengujian**	**Hasil Yang Diharapkan**
3.	Halaman Dekripsi	Memilih Tombol Pembangkit Kunci	*Black Box*
		Memasukkan Alamat Gambar, Kunci	*Black Box*
		Memasukan Kunci Yang Benar	*Black Box*
		Melakukan Dekripsi Berulang Kali	*Black Box*

4.2.2 Kasus dan Hasil Pengujian BlackBox

Proses pengujian *black-box* dilakukan untuk mengetahui apakah data yang dimasukkan sudah sesuai dengan yang diharapkan. Berdasarkan rencana pengujian, maka dapat dilakukan pengujian *black-box* pada sistem seperti berikut:

1. Pengujian Pembangkit Bilangan Prima

Tabel 4.4 Pengujian Pembakit Bilangan Prima (Uji Data Benar)

Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Nilai n : 10	Menampilkan Bilangan Prima Sebanyak 10 Buah	Menampilkan Bilangan Prima Sebanyak 10 Buah	**Diterima**

Tabel 4.5 Pengujian Pambangkit Bilangan Prima (Uji Data Salah)

Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Nilai n : (kosong)	Menampilkan Masukan Nilai n Kembali	Menampilkan Masukan Nilai n Kembali	**Diterima**

2. Pengujian Uji Nilai Prima

Tabel 4.6 Pengujian Uji Bilangan Prima (Uji Data Benar)

Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
P : 11 Q : 23 E : 7	Tidak Menampilkan Pesan Error	Tidak Menampilkan Pesan Error	**Diterima**

Tabel 4.7 Pengujian Uji Bilangan Prima (Uji Data Salah)

Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
P : 11 Q : 8 E : 7	Menampilkan Pesan Q Bukan Bilangan Prima	Menampilkan Pesan Q Bukan Bilangan Prima	**Diterima**

3. Pengujian Pembangkit Kunci Uji Masukan Kosong

Tabel 4.8 Pengujian Pembangkit Kunci Uji Masukan Kosong (Uji Data Benar)

Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
P : 11 Q : 23 E : 7	Tidak Menampilkan Pesan Error	Tidak Menampilkan Pesan Error	**Diterima**

Tabel 4.9 Pengujian Pembangkit Kunci Uji Masukan Kosong (Uji Data Salah)

Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
P : 11 Q : (Kosong) E : 7	Menampilkan Pesan Data Masukan Tidak Benar	Menampilkan Pesan Data Masukan Tidak Benar	**Diterima**

4. Pengujian Enkripsi Uji Masukan Kosong

Tabel 4.10 Pengujian Enkripsi Uji Masukan Kosong (Uji Data Benar)

Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Alamat Gambar : C:\User\Desktop\Gambar.jpg Kode File : HTHProject1 Batas Dekripsi : 1 Kunci : 1087 - 166769	Tidak Menampilkan Pesan Error	Tidak Menampilkan Pesan Error	**Diterima**

Tabel 4.11 Pengujian Enkripsi Uji Masukan Kosong (Uji Data Salah)

Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Alamat Gambar : C:\User\Desktop\Gambar.jpg Kode File : HTHProject1 Batas Dekripsi : (kosong) Kunci : 1087 - 166769	Menampilkan Pesan Error Data Masukan Tidak Lengkap	Menampilkan Pesan Error Data Masukan Tidak Lengkap	**Diterima**

5. Pengujian Tombol Pembuka Halaman Kunci

Tabel 4.12 Pengujian Tombol Pembuka Halaman Kunci (Uji Data Benar)

Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Memilih Tombol Pembangkit Kunci Pada Halaman Dekripsi	Membuka Halaman Pembangkit Kunci	Membuka Halaman Pembangkit Kunci	**Diterima**

6. Pengujian Dekripsi Uji Masukan Kosong

Tabel 4.13 Pengujian Dekripsi Uji Masukan Kosong (Uji Data Benar)

Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Alamat Gambar : C:\User\Desktop\Gambar1.jpg Kunci : 31 - 33 - 61 - 79 - 2111	Tidak Menampilkan Pesan Error	Tidak Menampilkan Pesan Error	**Diterima**

Tabel 4.14 Pengujian Dekripsi Uji Masukan Kosong (Uji Data Salah)

Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Alamat Gambar : C:\User\Desktop\Gambar1.jpg Kunci : (kosong)	Menampilkan Pesan Error Data Masukan Tidak Lengkap	Menampilkan Pesan Error Data Masukan Tidak Lengkap	**Diterima**

7. Pengujian Validasi Kunci

Tabel 4.15 Pengujian Validasi Kunci (Uji Data Benar)

Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Kunci : 31 - 33 - 61 - 79 - 2111	Gambar berhasil didekripsikan	Gambar berhasil didekripsikan	**Diterima**

Tabel 4.16 Pengujian Validasi Kunci (Uji Data Salah)

Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Kunci : 31 - 33 - 61 - 79 - 2111	Menampilkan pesan error bahwa kunci salah	Menampilkan pesan error bahwa kunci salah	**Diterima**

8. Pengujian Batas Dekripsi

Tabel 4.17 Pengujian Batas Dekripsi (Uji Data Benar)

Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Memilih tombol dekripsi	Gambar terdekripsi	Gambar terdekripsi	**Diterima**

Untuk batas dekripsi yang ditetapkan adalah sebanyak 1 kali seperti pada masukan enkripsi.

Tabel 4.18 Pengujian Batas Dekripsi (Uji Data Salah)

Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Memilih tombol dekripsi (untuk yang kedua kali)	Gambar tidak terdekripsi	Gambar tidak terdekripsi	**Diterima**

4.2.3 Kesimpulan Pengujian BlackBox

Pada tahapan pengujian blackbox didapatkan bahwa semua fungsional yang dirancang berjalan sesuai dengan yang diharapkan. Selain itu juga pada pengujian blackbox ini didapatkan kesimpulan bahwa fungsional pembatasan dekripsi dan pemberian kunci dapat menjawab tujuan dari penelitian ini.

4.2.4 Pengujian Beta

Pengujian beta merupakan pengujian yang dilakukan untuk mengetahui sejauh mana kegunaan aplikasi yang dibangun, apakah telah memenuhi harapan atau

belum. Untuk itu dalam pengujian beta ini akan dilakukan penelitian terhadap hak akses dari citra tersebut dengan cara mencoba menemukan kunci privatnya menggunakan metode *bruteforce* dan perhitungan matematis.

1. Metode BruteForce

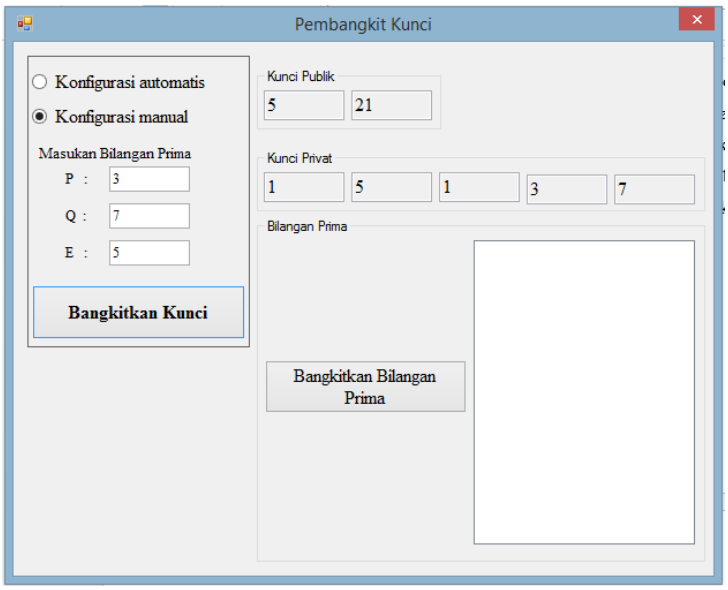
Metode BruteForce adalah metode yang melakukan percobaan terhadap semua kemungkinan yang ada atas segala masukan. Dalam kasus ini maka untuk melakukan metode *bruteforce* ini akan sangat sulit dilakukan karena komponen kunci yang banyak selain itu juga panjang kunci yang bervariasi dari setiap komponennya. Sebagai contoh sederhana telah dibangkitkan kunci dengan komponen pembangkit terkecil sebagai berikut:

P = 3

Q = 7

E = 5

Dari aplikasi pembangkit kunci yang ada maka didaptkan hasil sebagai berikut:



Bagan 4.25 Hasil Pembangkit Kunci Sederhana

Didapatkana bahwa kunci publik = 5 – 21 dan kunci privat = 1 – 5 – 1 – 3 – 7. Terlihat bahwa angka terbesar yang berhasil dibangkitkan adalah 7 yang masih dalam range 10 angka awal. Jika saja seseorang mencoba memecahkan kunci ini dengan range 10 angka awal maka didapatkan kemungkinan sebanyak :

Kemungkinan = 10 x 10 x 10 x 10 x 10

= 100.000 kali percobaan.

Dan dari percobaan itu jika dilakukan percobaan dengan mengurutkan angka maka kunci akan didapatkan pada percobaan ke 15137 atau 15137/100000 percobaan.

Angka yang besar ini maka akan sulit ditemukan terutama jika menggunakan percobaan manual. Selain itu kemungkinan ini akan terus bertambah seiring dengan komponen pembangkit kunci (P, Q, dan E) yang semakin besar nilainya. Oleh karena itu direkomendasikan untuk menggunakan nilai yang besar untuk membangkitkan kunci secara manual.

2. Metode Matematis

Perhitungan matematis ini bisa dilakukan untuk mengurangi banyak percobaan yang dilakukan. Metode ini tujuannya hanya untuk menemukan nilai P dan Q yang merupakan penyusun nilai N. Namun untuk menjalankan metode matematis ini diperlukan beberapa syarat sebagai berikut :

a. Memiliki kunci publik.
b. Mengetahui bahwa kriptografi menggunakan metode RSA – CRT.

Sebagai contoh kita menggunakan kunci yang berada pada contoh sebelumnya yaitu kunci publik = 5 – 21 dan kunci privat = 1 – 5 – 1 – 3 – 7. Dengan skenario seperti diatas maka akan dilakukan dekomposisi terhadap kunci ada sebagai berikut:

a. Tentukan mana nilai N dan E pada kunci publik (5, 21)
   1) Tentukan N = 5 dan E = 21

      Maka dari nilai tersebut akan dekomposisi nilai N karena nilai N merupakan komponen yang dibentuk oleh nilai P dan Q. Pendekomposisian ini dilakukan dengan cara memfaktorkan nilai N.

      Faktor N = Faktor 5 = (1, 5)

      Dari faktor tersebut maka didapatkan faktor N = (1, 5) dan jelas ini bukan faktor pembentuk nilai N karena syarat untuk P dan Q adalah bilangan prima dan 1 bukan merupakan bilangan prima.

2) Tentukan N = 21 dan E = 5

Maka dari nilai tersebut akan didapatkan faktor sebagai berikut :

Faktor N = Faktor 21 = (3, 7)

Dari nilai 3 dan 7 tersebut harus ditentukan mana nilai P dan Q. Dengan menggunakan algoritma pembangkit kunci yang telah dibahas pada bab 3 maka kita akan mendapatkan nilai sebagai berikut :

Tabel 4.19 Perbandingan nilai P dan Q

A	**B**
P = 3	P = 7
Q = 7	Q = 3
E = 5	E = 5
Kunci Privat : 1 – 5 – 1 – 3 – 7	Kunci Privat : 5 – 1 – 5 – 7 – 3

Dari hasil diatas untuk nilai N yang memiliki 1 faktor pembentuk kunci memiliki 2 kemungkinan untuk membangun sebuah kunci yaitu (3, 7) dan (7,3). Untuk setiap kemungkinan terdapat 5 komponen kunci yang menghasilkan 5 ! (lima faktorial) = 120 kombinasi kunci yang bisa dibentuk. Jadi kemungkinan percobaan untuk melakukan serangan terhadap kunci privat ini setelah nilai P dan Q ditemukan adalaha sebagai berikut :

Kemungkinan yang ada = 2 x 120

= 240 Kemungkinan

Kemungkinan itu akan terus bertambah bergantung pada besarnya nilai P dan Q pembentuk nilai N. Pertambahan itu yaitu sebanyak 120 x 2f. Dimana f merupakan banyaknya pasangan faktor penyusun nilai N. Sehingga sangat penting untuk memilih nilai P dan Q dengan nilai prima yang cukup besar dan sangat disarankan untuk memilih nilai P dan Q lebih dari 3 digit angka (ratusan) sehingga membuat nilai kemungkinan yang terbentuk akan semakin susah untuk diretas.