BAB 4
IMPLEMENTASI DAN PENGUJIAN SISTEM

Setelah melakukan tahapan analisis dan perancangan, maka yang harus dilakukan selanjutnya adalah mengimplementasikan apa saja yang telah di analisis dan dirancang sebelumnya. Tahapan-tahapan implementasi tersebut berupa implementasi perangkat keras, implementasi perangkat lunak, implementasi antar muka, implementasi program dan pengujian program.

4.1. Implementasi

Pada tahap implementasi, analisis dan perancangan yang telah dilakukan akan diterapkan pada aplikasi enkripsi dan dekripsi. Implementasi meliputi yaitu implementasi perangkat keras, implementasi perangkat lunak, implementasi aplikasi dan implementasi antarmuka.

4.2.1. Implementasi Perangkat Keras

Implementasi perangkat keras merupakan realisasi dari analisis dan perancangan kebutuhan perangkat keras. Implementasi perangkat keras merupakan salah satu syarat dalam implementasi perangkat lunak yang kana dilakukan pada tahap berikutnya. Implementasi perangkat keras yang dilakukan meliputi perangkat keras yang diperuntukan bagi komputer yang akan digunakan untuk aplikasi ini. Berikut ini merupakan spesifikasi perangkat keras untuk komputer yang digunakan

Tabel 4.1 Spesifikasi Perangkat Keras

No	Perangkat Keras	Spesifikasi
1.	Processor	Intel(R) Core i3-2330M
2.	RAM	2GB DDR 3
3.	Harddisk	500GB HDD
4.	Layar Monitor	Resolusi 1366 x 768 pixel
5.	Server	12 Core V3, 16GB *RAM*

4.2.2. Implementasi Perangkat Lunak

Implementasi perangkat lunak merupakan proses instalasi perangkat lunak, sehingga dapat beroperasi dengan benar. Proses implementasi perangkat lunak meliputi

Tabel 4.2 Spesifikasi Perangkat Lunak

No	Perangkat Lunak	Spesifikasi
1.	Sistem Operasi	Windows 7 Ultimate 64 bit
2.	Bahasa Pemograman	Java
3.	Perangkat Pendukung	Java Development Kit
4.	Aplikasi Tools	Java NetBeans IDE 8.1
5.	Picture Viewer	Windows Photo Viewer

4.2.3. Implementasi Jaringan

Implementasi jaringan dibedakan menjadi 2 yaitu implementasi jaringan server dan implementasi jaringan client.

4.2.3.1. Spesifikasi Komputer Server

Spesifikasi komputer server adalah sebagai berikut :

Tabel 4.3 Spesifikasi Komputer *Server*

No	Perangkat Keras	Spesifikasi
1.	Processor	Intel Server Board S5000
2.	RAM	2GB DDR2
3.	VGA	Onboard
4.	Ethernet	DFE-528 TX Ethernet
5.	HDD	SATA 500 GB
6.	Monitor	LG 710S
7.	Optional Disc	CD Room

Spesfikasi perangkat lunak pada komputer server adalah Windows NT Server, Windows Server 2003, Linux (Red Hat, Fedore) , dan Unix (Solaris). Sedangkan *software* pendukung meliputi Crystal Report, Autodesk Invertor, PHP, Visual Basic, Antivirus dan Delphi.

4.2.3.2. Spesifikasi Komputer Client

Spesifikasi komputer client adalah sebagai berikut:

Tabel 4.4 Spesifikasi Komputer *Client*

No	Perangkat Keras	Spesifikasi
1.	Processor	Inter Core i3
2.	RAM	1GB DDR2
3.	VGA	Onboard
4.	Ethernet	DFE-528 TX Ethernet
5.	HDD	SATA 500 GB
6.	Monitor	LG 710S
7.	Optional Disc	CD Room

Spesifikasi Perangkat lunak pada komputer client adalah Windows XP, Windows Vista, dan Windows 7 dan beberapa software pendukung lainya seperti Microsoft Office, Adobe Reader, dan lain–lain

4.2.4. Implementasi Data Storage

Implementasi data yang dilakukan adalah dengan menggunakan Mysql 5. Berikut paparan mengenai implementasi data yang dimaksud.

1. Pembuatan Database

Query

```
CREATE DATABASE img_key;
```

2. Pembuatan Tabel Key

Query

```
CREATE TABLE key (
  name varchar(64)NOT NULL UNIQUE,
  w int(11),
  h int(11),
  p tinyint(4),
  q tinyint(4),
  m int(4),
  x0 double,
  r double
);
```

4.2.5. Implementasi Antarmuka

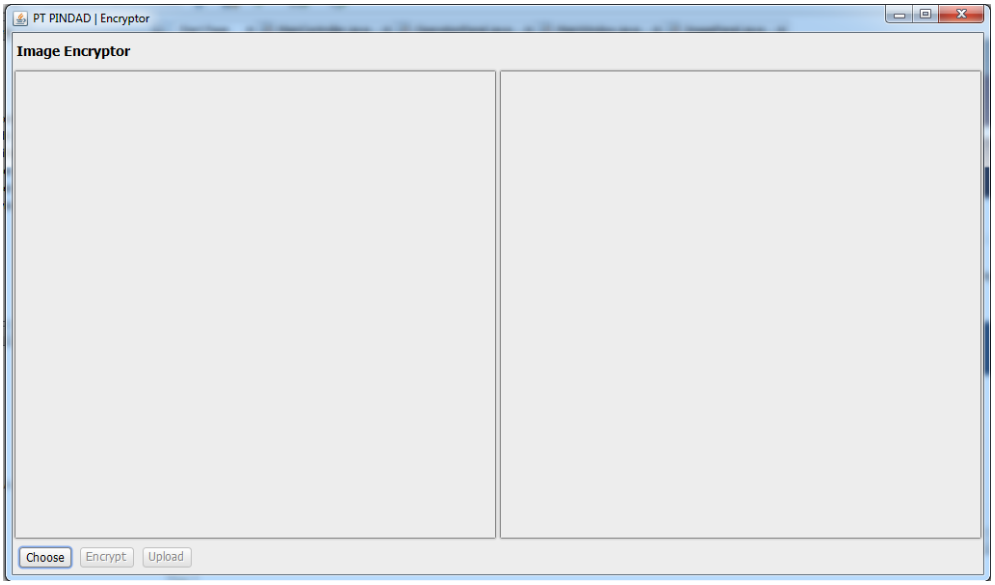
Antarmuka merupakan bagian penting pada aplikasi sistem keamanan enkripsi citra digital. Implementasi antarmuka merupakan penerapan antarmuka dari perancangan antarmuka yang telah dilakukan sebelumnya.

Tabel 4.5 Implementasi Antarmuka

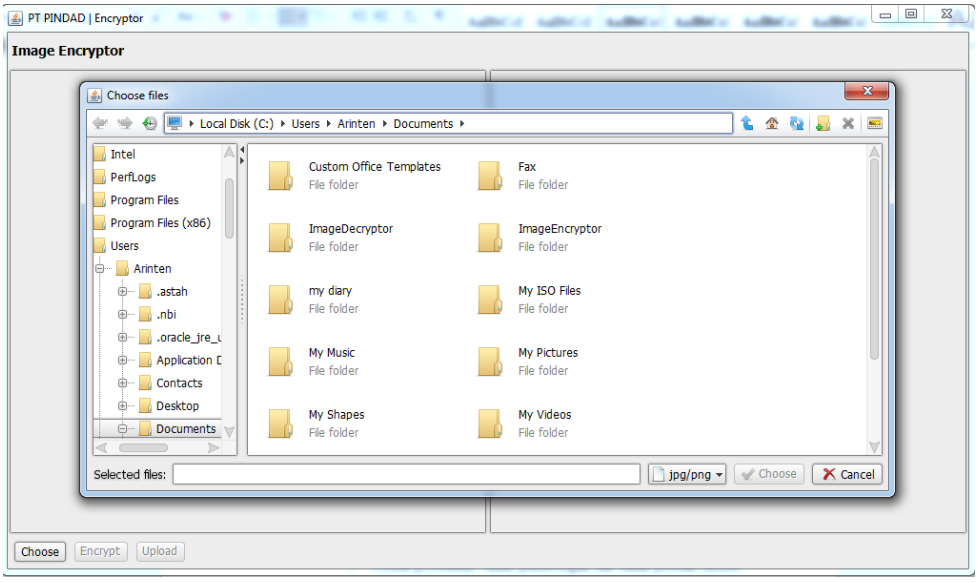
No	Nama Antarmuka	Keterangan
1.	Encrypt	Menampilkan form untuk mengenkripsi gambar yang akan dikirim.
2.	Decrypt	Menampilkan form untuk mendekripsi gambar menjadi *plain image*.

1. Encrypt

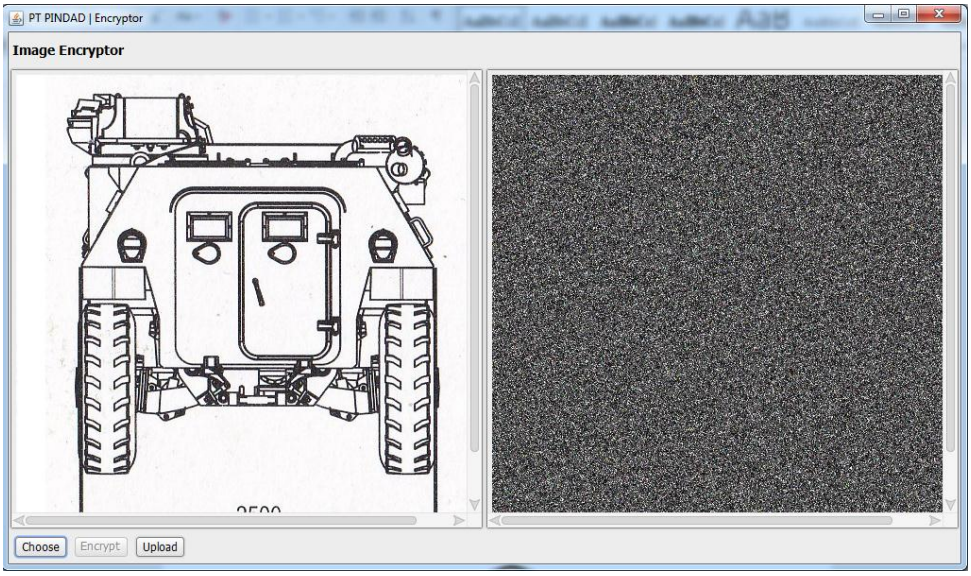
Di dalam form *encrypt* terdapat 3 proses yaitu memilih gambar, mengenkripsi gambar kemudian upload gambar yang telah dienkripsi.



Gambar 4.1 Antarmuka Encrypt



Gambar 4.2 Antarmuka Choose Image



Gambar 4.3 Antarmuka Proses Encrypt

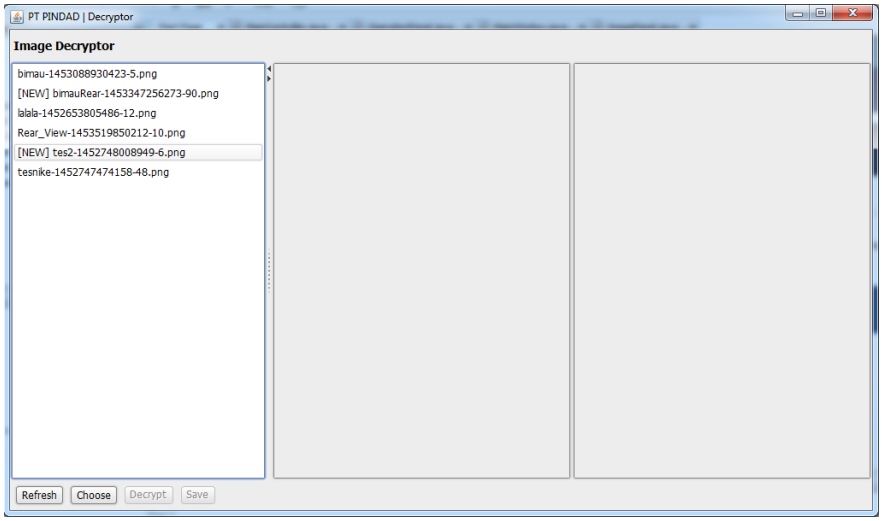
a. *Choose* : memilih gambar yang akan dienkripsi dengan format gambar berupa *.jpg atau *.png

b. *Encrypt* : melakukan proses enkripsi pada gambar yang telah dipilih sebelumnya.

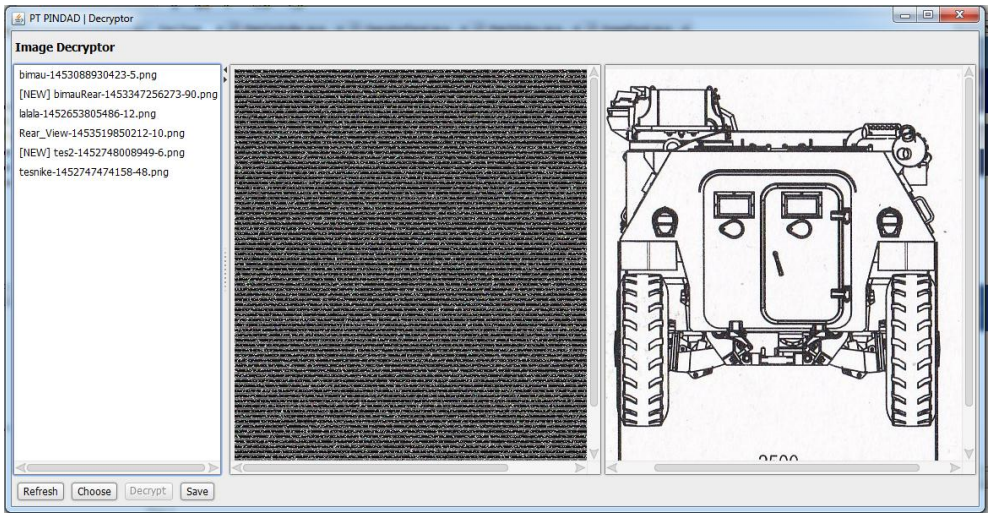
c. *Upload* : mengirim gambar ke divisi produksi.

2. Decrypt

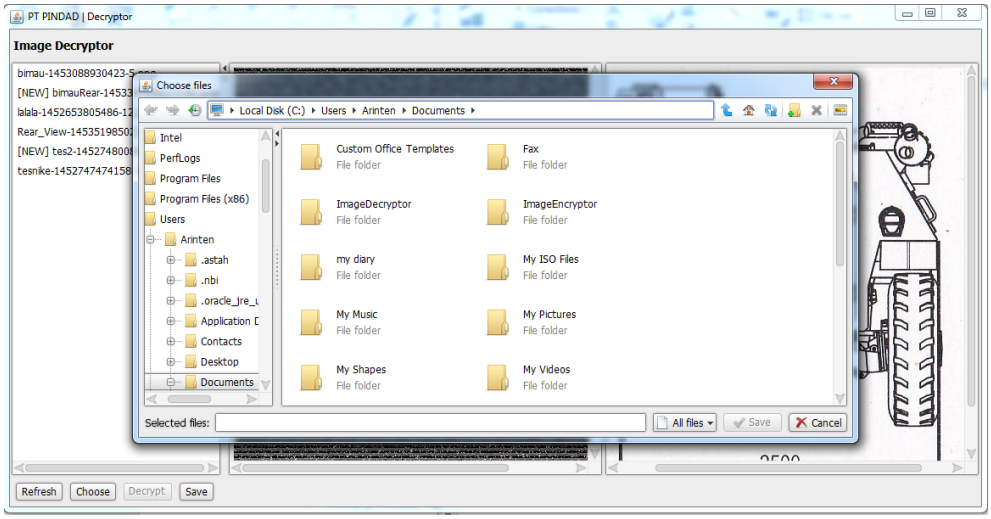
Di dalam form *decrypt* ini terdapat list notifikasi bahwa ada gambar baru yang telah dikirim dari divisi engineering, mendekripsikan gambar, dan menyimpan gambar pada direktori yang diinginkan.



Gambar 4.4 Antarmuka Decrypt



Gambar 4.5 Antarmuka Decrypt Choose Image



Gambar 4.6 Antarmuka Decrypt Save Image

a. *Refresh* : untuk me-refresh apakah ada gambar baru muncul di list gambar.

b. *Choose* : untuk menampilkan gambar *cipher image* pada panel gambar.

c. *Decrypt* : untuk mendekripsikan gambar menjadi *plain image* yang dimunculkan pada sebelah kiri gambar *cipher*.

d. *Save* : untuk menyimpan gambar pada direktori yang diinginkan.

4.2. Pengujian Sistem

Setelah melakukan tahap implementasi, maka tahapan selanjutnya yaitu pengujian program yang telah dibuat. Pengujian ini dilakukan untuk mengetahui apakah rancangan dan implementasi yang sudah dilakukan berjalan sesuai dengan prosedur yang diinginkan atau tidak.

4.2.1. Rencana Pengujian

Jenis pengujian yang dilakukan pada pembangunan aplikasi sistem keamnan enkripsi digital ini adalah pengujian *blackbox* dan pengujian *whitebox*. Rencana pengujian yang akan dilakukan pada aplikasi ini dapat dilihat pada Tabel 4.6.

Tabel 4.6 Rencanan Pengujian Aplikasi

No	Proses	Jenis Pengujian
1.	Pengujian *upload*	*Blackbox*
2.	Pengujian *Arnold Cat Map*	*Whitebix*
4.	Pengujian *Generate XOR*	*Whitebox*

4.2.2. Pengujian Blackbox

Pengujian *blackbox* bertujuan mengukur kinerja dari perangkat lunak apakah fungsinya dapat berjalan dengan baik atau tidak. Berikut pgnujian black box yang dilakukan.

4.2.2.1. Kasus dan Hasil Pengujian Aplikasi

Berdasarkan rencana pengujian yang disusun, maka dilakukan pengujian seperti yang dicantumkan dibawah ini.

Tabel 4.7 Pengujian Upload

Aktivitas yang dilakukan	Yang diharapkan	Pengamatan	Kesimpulan
Memilih gambar yang akan di enkripsi	Gambar berhasil ditampilakn ada panel image	Berhasil Menampilkan gambar	Diterima

Mengenkripsi gambar yang telah dipilih	Menampilkan hasil enkripsi pada panel image enkripsi	Berhasil menampilkan gambar *cipher image*	Diterima
Mendekripsi gambar yang dipilih	Menampilkan gambar *cipher* pada panel image	Berhasil menampilkan *plain image*	Diterima

4.2.3. Pengujian Whitebox

Pengujian *white box* digunakan untuk mengetahui logika yang dibuat pada sebuah perangkat lunak apakah berjalan dengan baik atau tidak. Pengujian *white box* akan digunakan pada algoritma Chaos Map dan *selective technique,* untuk mengukur kinerja logika berdasarkan *pseudocode* yang telah dibuat pada tahap analisis.

1. Langkah pertama ubah *pseudocode* menjadi *flowchart*
2. Ubah *flowchart* menjadi *flowgraph* ke dalam bentuk yang lebih sederhana.
3. Tahap pengujian, dimana tahap pengujian ini dilakukan dengan 5 cara yaitu, menghitung region, menghitung Cyclomatic Complexity, menghitung independent path, menggunakan graph matriks, menghitung predicate node.

4.2.2.1. Pengujian Pembangkit Kunci ACM

Pengujian pembangkit kunci ACM berikut dimulai dengan merubah pseudocode menjadi *flowchart*, kemudian *flowchart* dirubah menjadi *flowgraph*. Tabel Tabel 4.8 merupakan *source code* pengujian pembangkir kunci ACM.

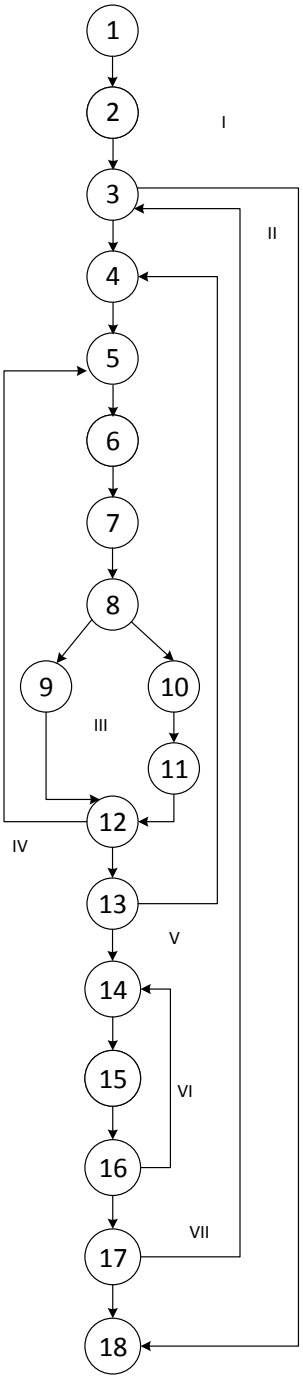
1. *Source code pembangkit kunci ACM*

Tabel 4.8 Algoritma Arnold Cat Map

Line	Source
1	`Integer[] hasil = new Integer[2];`
2	`Integer[][] result = new Integer[N][N];`
3	`For (int i = 0; i < this.acmIteration; i++) {`
4	`  for (int a = 0; a < N; a++) {`
5	`    for (int b = 0; b < N; b++) {`
6	`      hasil[0] = (M[0][0]*a + M[0][1]*b) % N;`
7	`      hasil[1] = (M[1][0]*a + M[1][1]*b) % N;`
8	`      if (status.equals("encrypt")) {`
9	`       result[a][b] = matrix[hasil[0]][hasil[1]];`
10	`       } else {`
11	`       result[hasil[0]][hasil[1]]= matrix[a][b];}`
12	`    }`
13	`  }`
14	`  for (int a = 0; a < N; a++) {`
15	`    System.arraycopy(result[a], 0, matrix[a], 0, N);`
16	`  }`
17	`}`
18	`return result;`

2. *Flowgraph* Arnold Cat Map

Berikut tahap *source code Arnold cat map imenjadi iflowgraph* dapat dilihat pada Gambar 4.7.



Gambar 4.7 Flow Graph Arnold Cat Map

3. Tahap Pengujian
   a. Hitung Region = 7
   b. $V_{(G)}$ = Edge – Node + 2
      = 23 – 18 + 2
      = 7

c. Independent Path

Path 1 = 1-2-3-18

Path 2 = 1-2-3-4-5-6-7-8-10-11-12-13-14-15-16-17-18

Path 3 = 1-2-3-4-5-6-7-8-9-12-13-14-15-16-17-18

Path 4 = 1-2-3-4-5-6-7-8-9-12-5-6-7-8-9-12-13-14-15-16-17-18

Path 5 = 1-2-3-4-5-6-7-8-9-12-13-5-6-7-8-9-12-13-14-15-16-17-18

Path 6 = 1-2-3-4-5-6-7-8-9-12-13-14-15-16-14-15-16-17-18

Path 7 = 1-2-3-4-5-6-7-8-9-12-13-14-15-16-17-3-4-5-6-7-8-9-12-13-14-15-16-17-18

d. *Graph Matriks* ACM

Graph matriks ACM dapat dilihat pada Tabel 4.9.

Tabel 4.9 *Graph Matriks* ACM

Node	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	Sum
1		1																	0
2			1																0
3				1														1	1
4					1														0
5						1													0
6							1												0
7								1											0
8									1	1									1
9												1							0
10											1								0
11												1							0
12					1								1						1
13				1										1					1
14															1				0
15																1			0
16															1		1		1
17			1															1	1
18																			0
Total																			6

$V_{(G)}$ = Jumlah *Graph Matriks* + 1

= 6 + 1

= 7

e. *Predicate Code*

$V_{(G)}$ = Jumlah node yang memiliki jalur lebih dari 1 jalur

= 6 + 1

= 7

4. Kesimpulan

Berdasarkan pengujian pada setiap metode, dihasilkan nilai *Cyclomatic complexity* yang sama yaitu 7. Maka dapat disimpulkan bahwa pengujian *white box* pada proses *Algoritma ACM* berjalan dengan baik, karena setiap pengujian menghasilkan nilai yang sama.

4.2.2.2. Pengujian Generate XOR

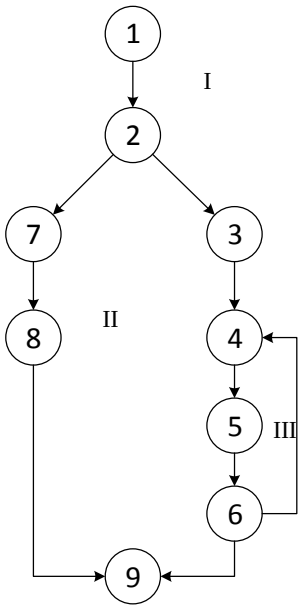
Pengujian *Generate* XOR dilakukan pada algoritma teknik selective yang meng-XOR-kan nilai 4-bit MSB dengan *keystream* yang dibangkitkan dari Logistic Map. Pada Tabel 4.10 merupakan *source code* pengujian *generate* XOR.

1. *Source code* Generate XOR

Tabel 4.10 *Source Code Generate* XOR

Line	Source
1	`Integer[] result;`
2	`  if (a.length == b.length) {`
3	`      result = new Integer[a.length];`
4	`      for (int i = 0; i < a.length; i++) {`
5	`           result[i] = a[i] ^ b[i]; }`
6	`      return result;`
7	`      } else {`
8	`      return null;`
9	`}`

2. *Fowgraph Generate* XOR



Gambar 4.8 *Flowgraph Generate* XOR

3. Tahap Pengujian
   a. Hitung Region = 3
   b. Hitung *cyclomatic complexity* yaitu sebagai berikut:

      $V_{(G)}$ = Edge – Node + 2

      = 10 – 9 + 2

      = 3

   c. Berdasarkan pada hasil *cyclomatic complexity* maka didapat dua *independent path* yaitu :

      Path 1 = 1-2-3-4-5-6-9

      Path 2 = 1-2-7-8-9

      Path 3 = 1-2-3-4-5-6-4-5-6-9

   d. *Graph Matriks Generate* XOR

      Graph matriks generate XOR dapat dilihat pada Tabel 4.11.

Tabel 4.11 *Graph Matriks Generate* XOR

Node	1	2	3	4	5	6	7	8	9	Sum
1		1								0
2			1				1			1
3				1						0
4					1					0
5						1				0
6				1					1	1
7								1		0
8									1	0
9										0
Total										2

$V_{(G)}$ = Jumlah *Graph Matriks* + 1

= 2 + 1

= 3

e. *Predicate Node*

$V_{(G)}$ = Jumlah node yang memiliki jalur lebih dari 1 jalur

= 2 + 1

= 3

4. Kesimpulan

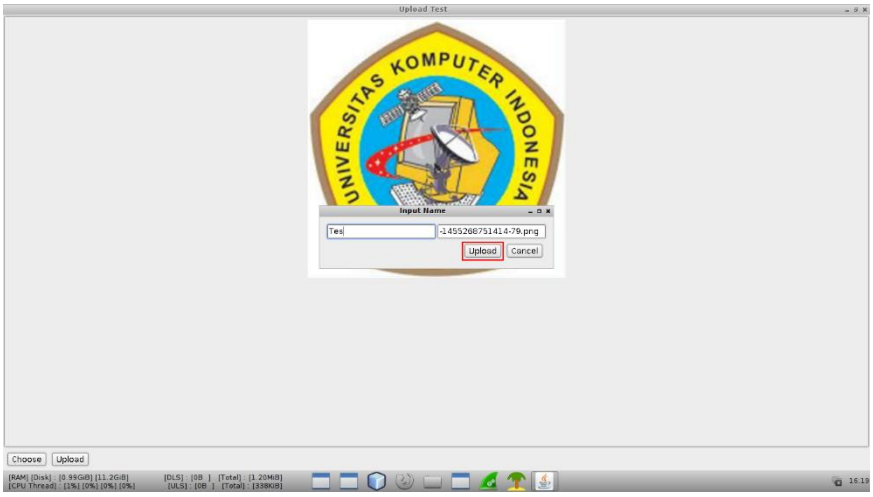
Berdasarkan pengujian pada setiap metode, dihasilkan nilai *Cyclomatic complexity* yang sama yaitu 3. Maka dapat disimpulkan bahwa pengujian *white box* pada proses *generate* XOR berjalan dengan baik, karena setiap pengujian menghasilkan nilai yang sama.

4.2.4. Pengujian Keamanan Gambar

Pengujian keamanan gambar dilakukan untuk mengetahui aman atau tidak nya gambar yagn dikirimkan. Pengujian ini dilakukan dengan menggunakan aplikasi *wireshark 2.0.1*. *Wireshark* dapat menangkap proses – proses yang sedang berjalan di dalam jaringan.

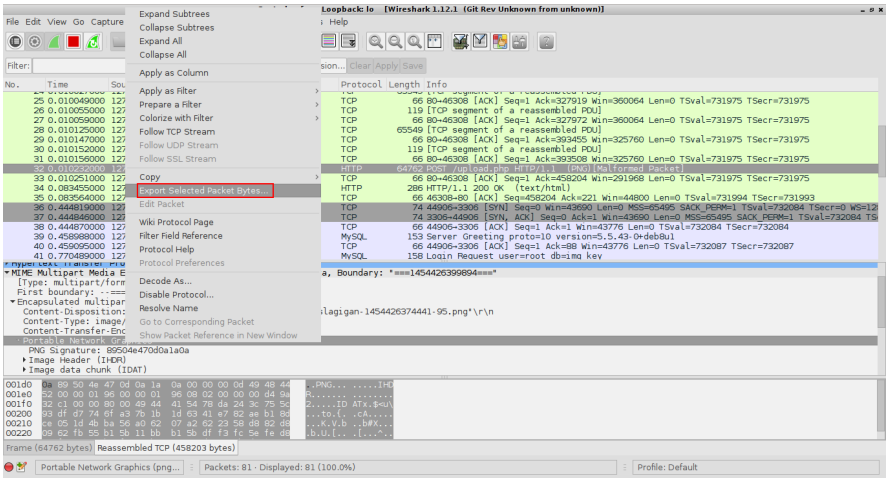
Pada pengujian pertama gambar yang dikirim merupakan gambar yang tidak di enkripsi. Berikut ini merupakan langkah-langkah dan hasil analisa keamanan pengiriman gambar yang ditangkap oleh *wireshark.*

1. Pertama upload gambar yang melalui aplikasi engiriman gambar. Berikut merupakan gambar yang akan dikirim dapat dilihat pada Gambar 4.9.



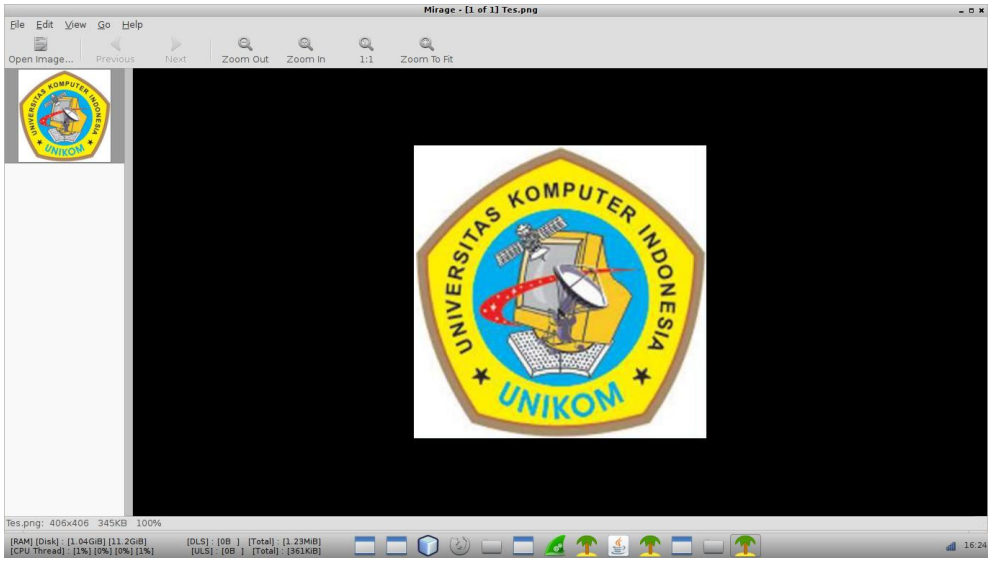
Gambar 4.9 Pengiriman Gambar yang Tidak di Enkripsi

2. Setelah gambar diupload kemudian buka aplikasi *wireshark* dan temukan nama gambar yang telah diupload. Pilih selected packet byte dan pilih PNG. Kemudian pilih save.



Gambar 4.10 Extrak Gambar yang Telah dipilih

3. Berikut hasil *sniffing* gambar yang telah diupload pada server dengan menggunakan aplikasi *wireshark*. Gambar terlihat seperti pada saat upload. Hal ini menyebabkan perlunya pengiriman gambar secara aman yaitu dengan cara mengenkripsi terlebih dahulu.

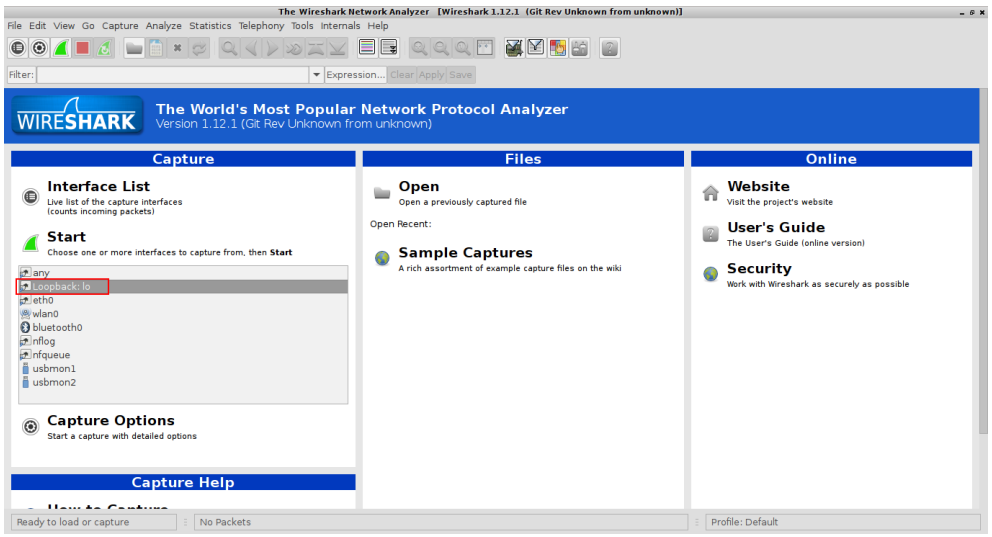


Gambar 4.11 Tampilan Gambar yang Telah di Deteksi Oleh Wireshark

Berikut ini merupakan langkah-langkah dan hasil analisa keamanan pengiriman gambar yang ditangkap oleh *wireshark* untuk gambar yang telah dienkripsi sebelumnya.

1. Tampilan *wireshark*

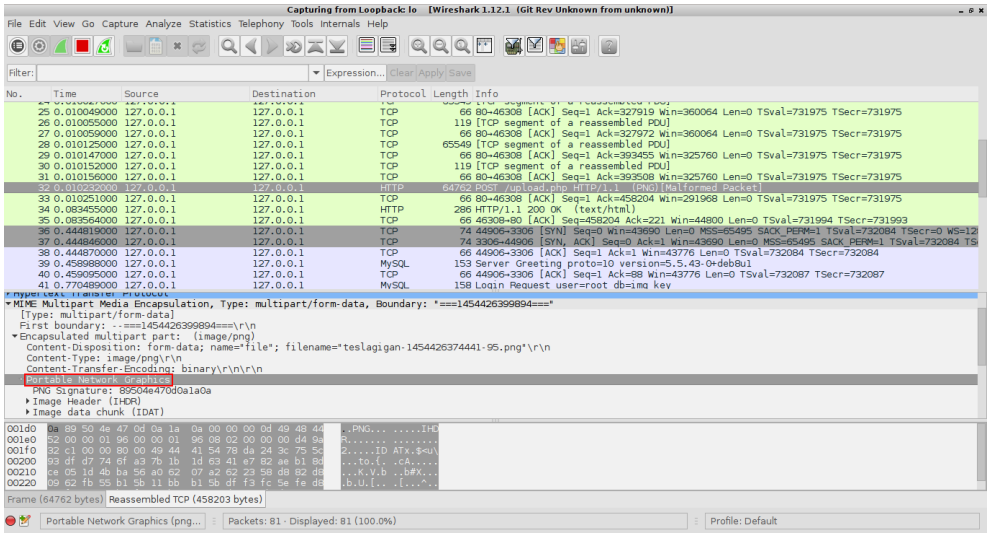
   Pilih menu start untuk menjalankan *sniffing* sehingga *wireshark* dapat mendeteksi proses yang ada di dalam jaringan. Berikut tampilan home pada *wireshark*.



Gambar 4.12 Tampilan Home *Wireshark*

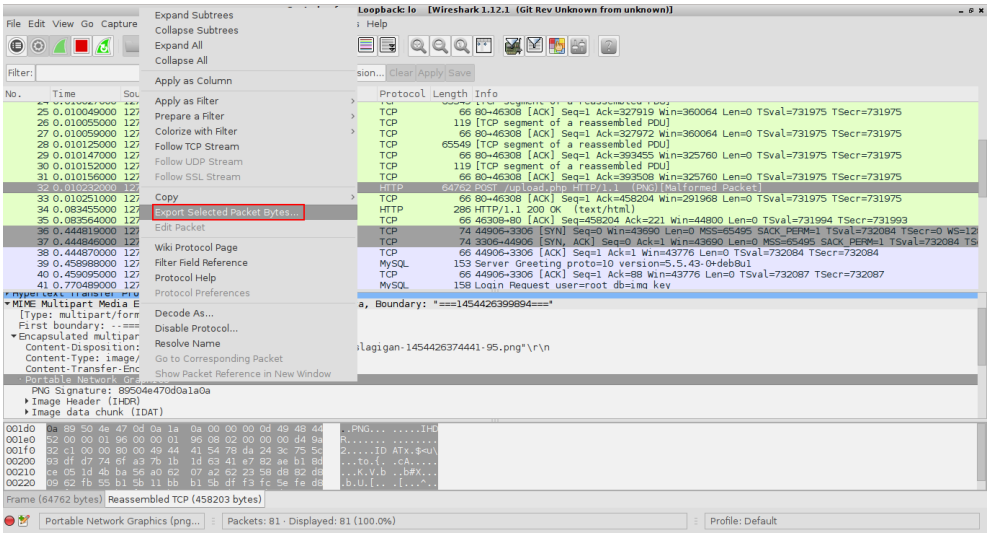
2. Setelah di jalankan aplikasi *wireshark* maka semua aktifitas di dalam dapat dilihat. Jika divisi engineering melakukan upload gambar ke server

maka *wireshark* dapat mendeteksi hal tersebut. Berikut tampilan *wireshark* dapat dilihat pada Gambar 4.13.



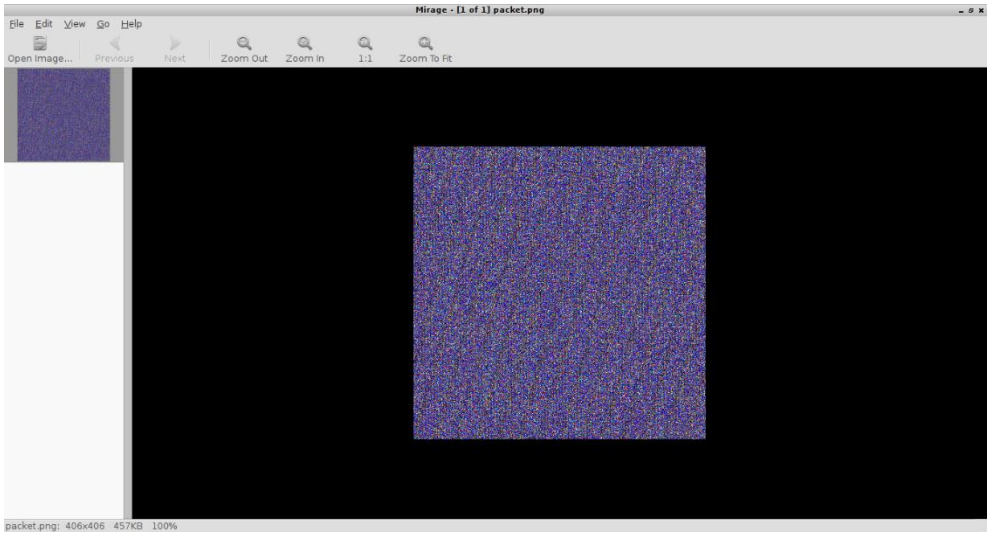
Gambar 4.13 Tampilan *Wireshark* Saat Mendeteksi Proses di Dalam Jaringan

3. Kemudian pilih export selected packet byte untuk mengetahui jenis packet yang dikirim.



Gambar 4.14 Tampilan untuk Mengetahui Jenis Packet yang Dikirim

4. Setelah memilih export selected packet byte maka file disimpan terlebih dahulu. Tampilan file yang disimpan berupa file gambar dan jika dibuka maka dapat dilihat pada Gambar 4.15



Gambar 4.15 Tampilan Gambar yang Telah Disimpan

5. Kesimpulan pengujian pengiriman gambar yang dilakukan oleh aplikasi *wireshark* yaitu, aplikasi *wireshark* dapat mendeteksi pengiriman gambar, namun gambar yang telah dideteksi merupakan *chiper image* sehingga gambar aman dari penyadapan.