# IMPLEMENTASI ENKRIPSI GAMBAR DENGAN ALGORITMA *ONE TIME PAD* MENGGUNAKAN METODE *LOGISTIC MAP* PADA FPGA

**SKRIPSI**

**LABORATORIUM SISTEM KOMPUTER DAN ROBOTIKA**

Diajukan untuk memenuhi persyaratan
mencapai gelar Sarjana Komputer


**Disusun Oleh :**
**DARMAWAN LAHRU RIATMA**
**0910683028**

**KEMENTRIAN PENDIDIKAN DAN KEBUDAYAAN**
**UNIVERSITAS BRAWIJAYA**
**PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER**
**MALANG**
**2013**

**LEMBAR PERSETUJUAN**

# IMPLEMENTASI ENKRIPSI GAMBAR DENGAN ALGORITMA *ONE TIME PAD* MENGGUNAKAN METODE *LOGISTIC MAP* PADA FPGA

**SKRIPSI**

**KONSENTRASI SISTEM KOMPUTER DAN ROBOTIKA**

Untuk memenuhi persyaratan memperoleh gelar Sarjana Komputer


Disusun oleh :

**DARMAWAN LAHRU RIATMA**

**0910683028**


Telah diperiksa dan disetujui oleh pembimbing:

| Dosen Pembimbing I | Dosen Pembimbing II |
|---|---|
| **Barlian Henryranu, P. S.T., M.T** | **Rekyan Regasari, M,P. S.T.,M.T** |
| **NIK. 82102406110254** | **NIK. 77041406120253** |

**LEMBAR PENGESAHAN**

**IMPLEMENTASI ENKRIPSI GAMBAR DENGAN ALGORITMA *ONE TIME PAD* MENGGUNAKAN METODE *LOGISTIC MAP* PADA FPGA**

**SKRIPSI**

Diajukan untuk memenuhi persyaratan mencapai gelar Sarjana Komputer

Disusun Oleh :

**Darmawan Lahru Riatma**

**NIM. 0910683028**

Skripsi ini telah diuji dan dinyatakan lulus pada

tanggal 6 Desember 2013

| Penguji I | Penguji II |
|---|---|
| **Gembong Edhi S., S.T.,M.T** | **Eko Setiawan, S.T., M.Eng.** |
| **NIP. 76120116110373** | **NIP. 87061006110256** |

Penguji III

**Hurriyatul Fitriyah, S.T., M.Sc**

**-**

Mengetahui

Ketua Program Studi Teknik Informatika

**Drs. Marji, MT**

**NIP. 19670801 199203 1 001**

**PERNYATAAN**
**ORISINALITAS SKRIPSI**

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah SKRIPSI ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis dikutip dalam naskah ini dan disebutkan dalam sumber kutipan dan daftar pustaka.

Apabila ternyata didalam naskah SKRIPSI ini dapat dibuktikan terdapat unsur-unsur PLAGIASI, saya bersedia SKRIPSI ini digugurkan dan gelar akademik yang telah saya peroleh (SARJANA) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku. (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 9 Desember 2013
Mahasiswa,

**Darmawan Lahru Riatma**
**NIM.0910683028**

# KATA PENGANTAR

*Alhamdulillahi rabbil 'alamin* Puji syukur penulis panjatkan atas kahadirat ALLAH SWT, karena Rahmat dan Hidayah-nya lah penulis dapat menyelesaikan skripsi yang berjudul **"IMPLEMENTASI ENKRIPSI GAMBAR DENGAN ALGORITMA *ONE TIME PAD* MENGGUNAKAN METODE *LOGISTIC MAP* PADA FPGA".**

Dalam pelaksanaan dan penulisan tugas akhir ini penulis mendapatkan banyak bantuan dari berbagai pihak baik secara moril maupun materiil. Dalam kesempatan ini penulis ingin mengucapkan terimakasih yang sebesar – besarnya kepada :

1. Allah SWT atas rahmat dan hidayah yang telah diberikan.
2. Rasulullah Muhammad SAW, semoga shalawat serta salam selalu tercurah kepada beliau.
3. Bapak, Ibu dan adik-adik yang saya cintai, yang telah memberikan dukungan, semangat, perhatian, nasehat, serta Doa sehingga penulis dapat menyelesaikan skripsi ini.
4. Bapak Barlian Henryranu P, S.T, M.T selaku dosen pembimbing Iyang telah memberikan Doa, masukan, dorongan, nasehat, ide, saran, serta semangat kepada penulis sehingga penulis dapat menyelesaikan skripsi ini.
5. IbuRekyan Regasari, MP. S.T, M.T selaku dosen pembimbing II yang telah memberikan Doa, masukan, dorongan, nasehat, ide, saran, serta semangat kepada penulis sehingga penulis dapat menyelesaikan skripsi ini.
6. Teman-teman komunitas robotika Universitas Brawijaya Andri Bagus Prastiyo, Ghayuh Fernanda Putra serta laboran robotikamas Aninditya Nugroho yang telah berbagi ilmu dan memberi suntikan moril serta Doa sehingga penulis dapat menyelesaikan skripsi ini.
7. Teman-teman seperjuangan dilab robotika Arif Hidayatullah, Thierry Rahman Aziz, Yusuf Oktofani, Yoga Pradana Pamungkas, Moh. Halimi, Delis Sukmawati, Alan Nur Abdan Natsir, Antarangga Bhawika Manggala, Dhimas Sawung Pamungkas, dan Nabila Mahastika Priadana.

8. Seluruh teman-teman angkatan 2009 dan pihak yang telah membantu yang tidak dapat saya sebutkan satu per satu.

Penulis menyadari bahwa dalam penyusunan tugas akhir ini masih banyak kekurangan baik format penulisan maupun isinya. Oleh karena itu, saran dan kritik membangun dari para pembaca senantiasa penulis harapkan guna perbaikan bagi tugas akhir selanjutnya. Semoga tugas akhir ini dapat memberikan manfaat bagi semua pihak, Aamin.

Malang, Oktober 2013

Penulis

## ABSTRAK

**Darmawan Lahru Riatma.2013. Implementasi Enkripsi Gambar Dengan Algoritma One Time Pad Menggunakan Metode Logistic Map Pada FPGA.** Program Teknologi Informasi dan Ilmu Komputer, Universitas Brawijaya, Malang. Dosen PembimbingBarlian Henryranu P, S.T, M.Tdan Rekyan Regasari, MP. S.T, M.T

Seiring perkembangan teknologi dewasa ini pengiriman dan penyimpanan gambar sangat rawan akan pencurian dan pengaksesan dari pihak-pihak yang tidak mempunyai hak. Sebagian besar masyarakat belum menyadari tentang pentingnya hal tersebut, oleh karena itu gambar yang bersifat private atau gambar yang mengandung informasi rahasia perlu dilindungi.Enkripsi merupakan salah satu teknik yang umum digunakan untuk melindungi gambar dari pengaksesan illegal. Sudah banyak dari kalangan ilmuwan maupun akademisi yang melakukan riset enkripsi berbasis perangkat lunak dengan berbagai metode yang terus disempurnakan. Kedepannya diharapkan sudah ada enkripsi berbasis *embedded system*.Dua dekade terakhir ini pengembangan perangkat *embedded system* begitu pesat. Salah satu foktor pendorongnya adalah perkembangan *Integrated Circuit* (IC). Saat ini sudah diciptakan teknologi *Field Programable Gate Array* (FPGA). Pada penelitian iniFPGA Spartan 3e digunakan sebagai simulator pemrograman IC untuk enkripsi gambar dengan algoritma *One Time Pad*dan menggunakan metode *Logistic Map* sebagai kunci. Diharapkan pada penelitian ini menjadi langkah awal diciptakanya enkripsi gambar berbasis *embedded system.* Pengujian dilakukan dengan dua cara yaitu pengujian validitas gambar dan pengujian sensitivitas nilai awal kunci l*ogistic map*. Dari hasil pengujian validitas pada saat gambar didekripsi dengan kunci yang sama gambar kembali seperti gambar aslinya, ketika gambar didekripsi dengan kunci yang berbeda gambar tidak kembali seperti gambar asli dan tetap acak. Pada pengujian sensitivitas ketika gambar didekripsi dengan selisih nilai awal kunci yang kecil menghasilkan gambar tetap acak tidak bermakna dan tidak kembali seperti gambar asli.


***Kata Kunci :*** Enkripsi Gambar, *FPGA*, kriptografi, *One Time Pad, Logistic Map*

## ABSTRACTS

**Darmawan Lahru Riatma.2013 . Image Encryption Implementation Analysis of Algorithms One Time Pad Using Logistic Map Method On FPGA**. Information Technology and Computer Science Programme, University of Brawijaya, Malang. Supervisor Barlian Henryranu P, ST, MT and Rekyan Regasari, MP. S.T, M.T

As the development of today's technology, images delivery and storages is very vulnerable to theft and accessing of parties who do not have the right. Most people simply do not aware about the importance of this, so private images or images that contain confidential information needs to be protected. Encryption is one of the technique commonly used to protect the image from illegal access. There has already been many of the scientists and academics who conduct research with software -based encryption methods that continue to be refined. It is expected in the future the existence of encryption -based embedded systems or portable. The last two decades the development of embedded systems is growth rapidly. One of the supporting factor is the development of the Integrated Circuit ( IC ). Now there is a technology called the technology Field Programable Gate Array ( FPGA ). In this study, Spartan 3e FPGA is used as a programming simulator with IC for image encryption algorithms One Time Pad ( OTP ) and using the Logistic Map as a key. it is expected in this research as beginning on the creation of image encryption embedded system based. Experiments is done in two ways: testing the validity of the test image and the initial value sensitivity of key logistic map. Validity of the test results at the time the image is decrypted with the same key as the image back to its original image, when the image is decrypted with a different key does not return an image to the original image and still random. On sensitivity testing when the image is decrypted with the small difference value of keys generating random image and remains insignificant and do not come back as the original image.

**Keywords** : Encryption Image, FPGA, cryptography, One Time Pad, Logistic Map

## DAFTAR ISI

## DAFTAR TABEL

## DAFTAR GAMBAR

# BAB I
# PENDAHULUAN

## 11.3 Latar Belakang

Gambar merupakan salah satu citra yang sangat penting dalam multimedia.Gambar menyajikan informasi secara visual yang informasinya lebih baik dari pada citra teks [MUN-04]. Seiring perkembangan teknologi dewasa inipengiriman dan penyimpanan gambar sangat rawan akan pencurian dan pengaksesan dari pihak-pihak yang tidak mempunyai hak.Sebagian besar masyarakat belum menyadari tentang pentingnya hal tersebut, oleh karena itugambar yang bersifat private atau gambar yang mengandung informasi rahasia perlu dilindungi.

Enkripsi merupakan salah satu teknik yang umum digunakan untuk melindungi gambar dari pengaksesan illegal.Saat ini sudah banyak dari kalangan ilmuwan maupun akademisi yang melakukan riset enkripsi berbasis perangkat lunak dengan berbagai metode yang terus disempurnakan.Kedepannya diharapkan sudah ada enkripsi berbasis *embedded system* atau bersifat portable. Pada dua dekade terakhir ini pengembangan perangkat *embedded system* begitu pesat. Peningkatan ini didorong beberapa faktor, salah satunya adalahperkembangan *Integrated Circuit* (IC) yang memicu perkembangan *embedded system*[ABD-13]. Menurut Gordon More, seorang insinyur dari Fairchild Semiconductor, menyatakan bahwa “kapasitas silikon secara konstan Meningkat dua kali lipat setiap 18 sampai 20 bulan”.

Perkembangan teknologi di bidang perangkat keras dewasa ini difokuskan dalam pembuatan perangkat keras yang mempunyai desain dan bentuk yang minimalis.Saat ini sudah diciptakan teknologi *Field Programable Gate Array* (FPGA).FPGA adalah perangkat elektroniksemikonduktor dalam sebuah IC yang dapat dikonfigurasi oleh programmer untuk keperluan simulasi system [ABD-13].FPGA sering juga disebut pemodelan atau prototyping pemrograma ICdimana pengguna dapat menggunakannya sesuai dengan kebutuhan.

Penelitian tentang enkripsi gambar telah banyak dilakukandan dikembangkan hingga saat ini.Algoritma *One Time Pad*merupakan algoritma yang mencapai kerahasian sempurna yaitu menghasilkan teks sandi yang tidak memiliki hubungan statistik terhadap teks asli sehingga analisa statistik sulit untuk dilakukan [SAD-12].Rinaldi Munir [MUN-11] melakukan penelitian tantang algoritma enkripsi citra dengan *pseudo One Time Pad* yang menggunakan sistem *chaos*.Hasil yang diperoleh dari penelitian tersebut adalah algoritma *One Time Pad*dapat mengenkripsi citra dengan baik dan mendeskripsikanya tepat seperti citra semula.Eksperimen perubahan nilai awal chaos memperlihatkan algoritma ini aman dari serangan *exhaustive attack*.

*Logistic map*merupakan salah satu metode sederhana di dalam sistem *chaos* yang bekerja dengan membangkitkan barisan kunci semi acak. Penelitian tentang *Logistic map* pernah dilakukan oleh Ravindra K. Purwar dan Prayinka [PUR-13] yang berjudul An Improved Image Encryption Schme Using Chaotic Logistic maps. Hasil dari peneliatian menunjukan bahwa algoritma Logistic map aman terhadap serangan *statistic* dan *brute force*, juga efisien dan cukup cepat untuk digunakan.

Penelitian ini penulis mengambil judul skripsi Implementasi Enkripsi Gambar dengan Algoritma *One Time Pad* Menggunakan Metode *Logistic map* Pada FPGA.Algoritma *One Time Pad* digunakan untuk mengenkripsi gambar, dan memanfaatkan Metode *Logistic map*sebagai pembangkit bilangan kunci. Pengimplementasian enkripsi gambar ini dikerjakan pada FPGA. Diharapkan pada penelitian ini menjadi langkah awal diciptakanya enkripsi gambar berbasis *embedded system*.Akan tetapi pada penelitian ini belum sampai dilakukan pengujian mengenai keamanan.

## 4.2 Rumusan Masalah

Berdasarkan uraian latar belakang di atas, maka dapat dirumuskan permasalah pada skripsi ini yaitu sebagai berikut:

1. Bagaimana implementasienkripsi gambar dengan algoritma *One Time Pad* menggunakan metode *Logistic map* pada FPGA?
2. Bagaimana pengujian validitas data gambar setelah gambar kembali didekripsi sesuai dengan aturan *One Time Pad*yang diimplementasikan pada FPGA?
3. Bagaimana pengujian sensitivitas nilai awal kunci pada *Logistic map*yangdiimplementasikan pada FPGA?

## 11.5 Batasan Masalah

Agar permasalahan yang dirumuskan dapat lebih terfokus, maka padapenelitianini dibatasi oleh hal-hal berikut :

1. Enkripsi gambar di implementasikan padaFPGA Spartan 3e.
2. Algoritma enkripsi yang digunakan adalah algoritma *One Time Pad*, sedangkan pembangkit kunci yang digunakan adalah *Logistic Map*.
3. Gambar yang digunakan pada penelitian kali ini adalah gambar dengan warna 3 bit.
4. Gambar tersebut adalah gambar buatan sendiri.
5. Pixel gambar yang digunakan tidak berjumlah lebih dari 400 pixel.

## 11.6 Tujuan

Mengimplementasikan enkripsi gambar dengan algoritma *One Time Pad* menggunakan metode *logistic map* pada papan FPGA Spartan 3e, dan menguji validitas gambar dengan aturan *One Time Pad*serta menguji sensitivitas nilai awal kunci pada *Logistic map.*Diharapkan pada penelitian ini menjadi langkah awal diciptakanya enkripsi gambar berbasis *embedded system* atau bersifat portable sebagai prosesor tujuan tertentu yang bisa diletakan antara cloud dengan enduser.

## 11.7 Manfaat

Penulisan skripsi ini diharapkan mempunyai manfaat yang baik dan berguna bagi pembaca dan penulis. Adapun manfaat yang diharapkan adalah sebagai berikut:

A. Bagi penulis

1. Sebagai media untuk pengimplementasian ilmu pengetahuan teknologi yang telah didapat selama mengikuti perkuliahan di Teknik Informatika Universitas Brawijaya.
2. Mendapatkan pengetahuan dan wawasan tentang FPGA.
3. Mendapatkan pengetahuan tentang dunia kriptografi.

B. Bagi pembaca

1. Mendapatkan wawasan akan pengimplementasian programdi FPGA.
2. Sebagai bahan dan acuan dalam mengembangkan dan melanjutkan riset yang memilik keterkaitan dengan penelitian ini.

## 11.8 Sistematika Penulisan

Sistematika penulisan dalam skripsi ini sebagai berikut:

**BAB I Pendahuluan**

Bab ini berisi tentang latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, dan sistematika penulisan.

**BAB II Dasar Teori**

Membahas teori-teori yang mendukung dalam perancangan enkripsi Gambar dengan Algoritma *One Time Pad* menggunakan *Logistic Map* pada FPGA**.**

**BAB III Metode Penelitian dan Perancangan**

Membahas tentang metode yang digunakan dalam penulisan yang terdiri dari Studi dan Pengkajian Literatur, Analisis Kebutuhan Sistem, perancangan sistem, implementasi, pengujian dan analisis.

**BAB IV Implementasi**

Menguraikan proses implementasi dari enkripsi gambar dengan Algoritma *One Time Pad* menggunakan *Logistic Map* pada FPGA.

**BAB V Pengujian dan Analisis**

Memuat hasil pengujian dan analisis terhadap sistem yang telah direalisasikan.

**BAB VI Penutup**

Memuat kesimpulan yang diperoleh dari pembuatan dan pengujian program, serta saran-saran untuk pengembangan lebih lanjut.

# BAB II
# TINJAUAN PUSTAKA DAN DASAR TEORI

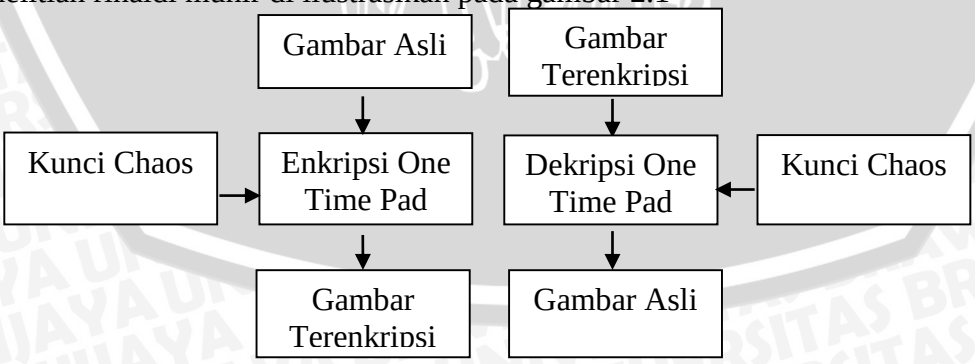
Bab ini berisi tinjauan pustaka yang meliputi kajian pustaka dan dasar teori yang diperlukan untuk penelitian.Kajian pustaka membahas penelitian yang telah ada dan yang diusulkan.Dasar teori membahas teori yang diperlukan untuk menyusun penelitian yang diusulkan.

## 13.1 Kajian Pustaka

Kajian pustaka membahas penelitian yang telah ada dan yang diusulkan.Pada penelitian ini kajian pustaka diambil dari beberapa penelitian yang relevan yang pernah dilakukan.

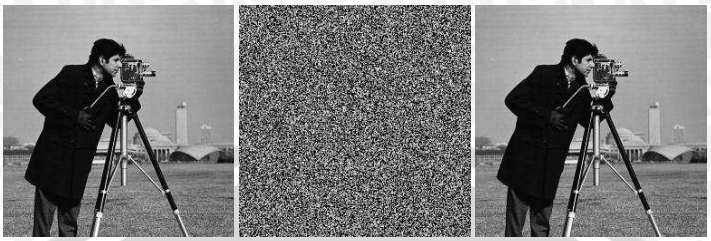
### 13.1.1 Penelitian Algoritma Enkripsi Citra dengan Pseudo One-Time Pad yang Menggunakan Sistem Chaos. Oleh Rinaldi Munir

Rinaldi Munir pernah melakukan risetmengenai enkripsi citra gambar. Pada riset tersebut berjudul '*Algoritma Enkripsi Citra dengan Pseudo One-Time Pad yang Menggunakan Sistem Chaos*'. Penelitian ini membahas mengenaialgoritma sederhana untuk mengenkripsi citra digital dengan memanfaatkan sistem *chaos*.Enkripsi dilakukan dengan algoritma *One-Time Pad*, yang dalam hal ini barisan kunci dibangkitkan dengan sistem chaos*logistic map*.Proses kerja dari penelitian rinaldi munir di ilustrasikan pada gambar 2.1




**Gambar2.1**Proses Enkripsi dan Dekripsi

Hasil riset yang dilakukan Rinaldi Munir di tunjukan pada gambar 2.2.



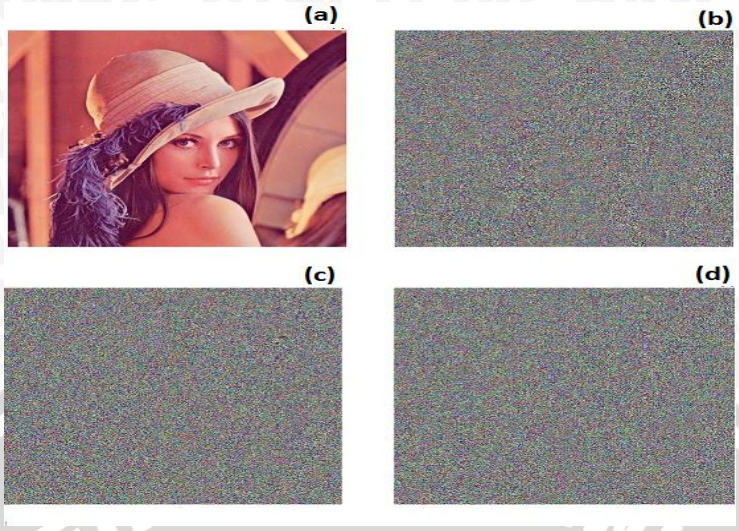
**Gambar2.2**Hasil Proses gambar asli, Enkripsi dan Dekripsi
**Sumber :**[RIN-11].

Dari hasil riset Algoritma *One Time Pad* dapat mengenkripsi citra gambar dengan baik dan dapat juga mendekripsi gambar enkripsi kembali sesuai dengan gambar asli.Eksperimen perubahan parameter nilai *chaos* memperlihatkan bahwa algoritma ini aman dari serangan *exhaustive attack*.

### 13.1.2 Penelitian An Image Encryption Schme Using Chaotic Logistic map. Oleh Ravindra K.Purwar dan Priyanka

Teori chaos telah memberikan kontribusi yang signifikan terhadap perkembangan enkripsi yang aman dan cepat. Dalam kriptografi chaos digunakan karena kesensitivanya terhadap perubahan kecil pada nilai awal, sensitif pada perubahan kecil ini berarti dengan perubahan yang kecil pada nilai awalnya akan menghasilakan keluaran yang sangat berbeda hal ini akan menyebabkan analisis statistik sulit untuk dilakukan oleh kriptanalis. Pada penelitian iniRavindra K.Purwar dan Priyanka menggunakan 80 bit kunci yang dirubah setelah enkripsi blok pada pixel ke-16 dan menggunakan dua Logistic map untuk setiap gambar.

Pengujian dan analisa dilakukan dengan menguji sensitivitas pada nilai awal, analisa waktu dan analisis histogram warna gambar.gambar 2.3 menunjukan hasil analisis sensitivitas. Table 2.1 menunjukan analisis waktu.




**Gambar2.3**pengujian sensitivitas
**Sumber :**[PUR-13]

Berdasarkan gambar 2.3 gambar bagian (a) adalahgambar asli, (b) gambar enkripsi dengan kunci BQFRAM3L12 (c) dekripsi dengan kunci BQFRAM3L13(d) gambar dekripsi dengan kunci BQFSAM3L12.

| Ukuran Gambar | Rata-rata waktu enkripsi dan dekripsi |
|---|---|
| 1024x1024 | 0,75 |
| 512x512 | 0.18 |
| 256x256 | 0.03 |

**Tabel2.1**Analisis waktu
**Sumber :**[PUR-13]

Dari hasil pengujian dan analisis Ravindra K.Purwar dan Priyanka menyimpulkan bahwa algoritma Logistic map aman terhadap serangan statistic dan brute force, juga efisien dan cukup cepat untuk digunakan.

### 13.1.3 Penelitian Parancangan dan Implementasi Algoritma Enkripsi Arcfour pada Perangkat Kriptografi Berbasis FPGA Oleh Mohamad Syahral

Pada penelitian yang dilakukan oleh Mohamad Syahral, peneliti melakukan perancangan dan implementasi enkripsi pada data teks berupa file menggunakan algoritma arcfour yang di implementasikan pada papan FPGA Spartan 2e.hasi pengujian menujukan implementasi algoritma arcfour berhasil dilakukan pada FPGA Spartan 2e, kriptografi yang diimpementasikan berhasil melakukan enkripsi

dan dekripsi pada data berupa teks. Namun pada penelitian ini masih ada kegagalan pada proses kirim dan terima file. Hasil pengujian pada penelitian yang dilakukan oleh mohamad Syahral ditunjukan pada gambar


**Gambar2.4**Hasil**P**engujian Rangkaian Kunci Arcfour
**Sumber :**[SYA-11]

Perbedaan yang dibuat penulis pada penelitian ini adalah pengimplementasian enkripsi gambar yang dilakukan di papan FPGA Spartan 3e, dengan menggunakan usulan Algoritma enkripsi yang telah dilakaukan pada riset sebelumnya.Data yang di enkripsi berupa data gambar, namun pada penelitian ini gambarnya adalah gambar buatan sendiri.Perbedaan ini akan menyebabkan masukan yang dibutuhkan dan proses akhir akan berbeda dengan penelitian sebelumnya.

## 13.2 Dasar Teori

Dasar teori membahas teori yang diperlukan untuk menyusun penelitian yang diusulkan. Dasar teori akan dibahas di subbab 2.2.1 sampai 2.2.4.

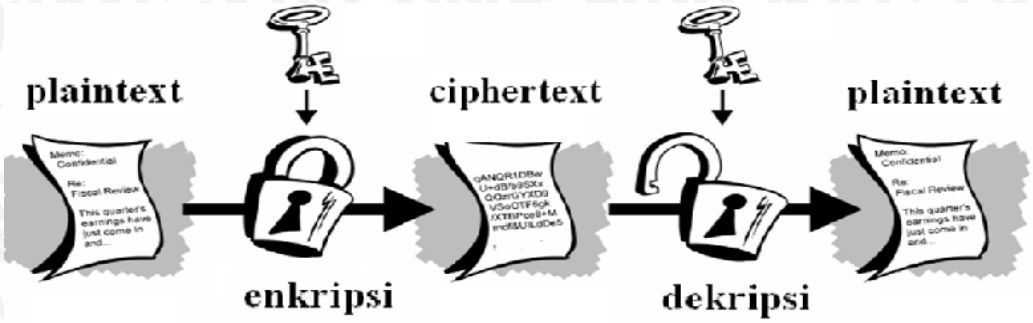
### 13.2.1 Kriptografi

Kriptografi berasal dari bahasa Yunani, “kryptós” yang berarti tersembunyi dan “gráphein” yang berarti tulisan.Sehingga kata kriptografi dapat diartikan menjadi “tulisan tersembunyi”.Kriptografi adalah teknik penyandian dengan kunci simetrik dan menyembunyikan pesan yang memiliki arti ke sebuah pesan yang nampaknya tidak memiliki arti dengan metode subtitusi ( pergantian huruf) dan transposisi (pertukaran tempat)[ABD-13].Pengertian Kriptografi dalam kamus bahasa Inggris Oxford adalah sebagai berikut “ Sebuah teknik rahasia dalam penulisan, dengan karakter khusus, dengan mengguanakan huruf dan karakter di luar bentuk aslinya, atau dengan metode-metode lain yang hanya dapat dipahami oleh pihak-pihak yang memproses kunci, juga semua hal yang ditulis dengan cara seperti ini.”Tujuan kriptografi adalah kerahasiaan, integritas data, otentikasi, dan nirpenyangkalan [MUN-06].

Di dalam kriptografi kita akan sering menemukan berbagai istilah atau terminology.Beberapa istilah yang penting untuk diketahui diberikan di bawah ini.

A. Plaintext : pesan atau data dalam bentuk aslinya yang dapat terbaca. Plaintext adalah masukan bagi algoritma enkripsi. Untuk selanjutnya digunakan istilah text asli sebagai padanan kata plaintext.
B. Secret key : secret key juga merupakan masukan bagi algoritma yang merupakan nilai yang bebas terhadap text asli dan menentukan hasil keluaran algoritma enkripsi.
C. Ciphertext : ciphertext adalah keluarga algoritma enkripsi. Ciphertext dapat dianggap sebagai pesan dalam bentuk tersembunyi. Algoritma enkripsi yang baik menghasilkan ciphertext yang terlihat acak.
D. Algoritma enkripsi: Algoritma enkripsi memiliki dua masukan teks asli dan kunci rahasia. Algoritma enkripsi melakukan transformasi terhadap teks asli sehingga menghasilkan teks sandi.
E. Algoritma dekripsi: Algoritma dekripsi memiliki dua masukan yaitu teks sandi dan kunci rahasia. Algoritma dekripsi memulihakan kembali teks sandi menjadi teks asli bila kunci rahasia yang di pakai algoritma

dekripsi sama dengan algoritma dekripsi.Gambar 2.3 merupakan skema proses kriptografi.




**Gambar2.5**Skema Proses Kriptografi
**Sumber :**[PER-08].

### 13.2.1.1 *Jenis Jenis Kriptografi*

Berdasarkan jenis kuncinya, algoritma kriptografi terbagi menjadi dua yaitu : algoritma simetris atau algoritma kunci privat dan algoritma asimetris yang sering juga disebut algoritma kunci publik.

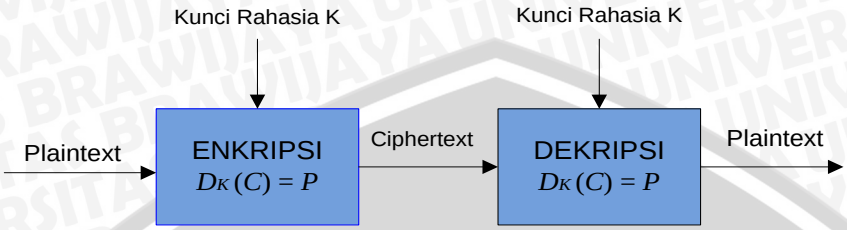
#### A. Algoritma Simetri

Algoritma Simetris merupakan salah satu jenis kunci dalam kriptografi yang mana menggunakan kunci sama pada saat proses enkripsi dan dekripsinya. Cara kerja kriptografi simetri diasumsikan sebagai pengirim dan penerima pesan yang sudah berbagi kunci yang sama sebelum bertukar pesan. Keamanan sistem kriptografi simetri terletak pada kerahasiaan kuncinya.

Secara umum, cipher yang termasuk ke dalam kriptografi simetri beroperasi dalam mode blok (block cipher), yaitu setiap kali enkripsi/dekripsi dilakukan terhadap satu blok data (yang berukuran tertentu), atau beroperasi dalam mode aliran (stream cipher), yaitu setiap kali enkripsi/dekripsi dilakukan terhadap satu bit atau satu byte data.

Salah satu kelebihan pada algoritma simetris yaitu proses enkripsi dan deskripsinya jauh lebih cepat dibandingkan dengan algoritma asimetris. Sedangkan

kelemahannya yaitu pada permasalahan distribusi kunci (key distribution) [ZEL-12].Skema kriptografi simetri ditunjukan pada gambar 2.4.

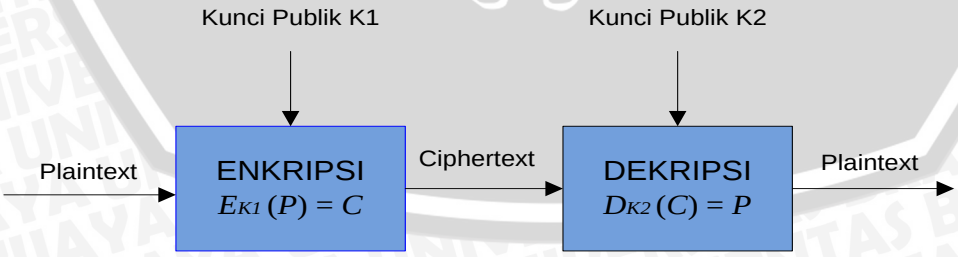



**Gambar2.6**Skema Kriptografi Simetri
**Sumber :**[MUN-06]

**B. Algoritma Asimetris**

Algoritma asimetris atau yang sering juga disebut dengan algoritma kunci public, Dimana kunci untuk enkripsi tidak rahasia (diumumkan ke publik), sementara kunci dekripsinya bersifat rahasia (hanya diketahui oleh penerima pesan).

Sistem ini memiliki dua keuntungan.Yang pertama yaitu, tidak ada kebutuhan untuk mendistribusikan kunci privat sebagaimana pada sistem kriptografi simetri. Kunci publik dapat dikirim ke penerima pesan melalui saluran yang sama dengan saluran yang digunakan untuk mengirim pesan. Saluran untuk mengirim pesan umumnya tidak aman.Skema kriptografi asimetri ditunjukan pada gambar 2.5.




**Gambar2.7**Skema Proses Kriptografi Asimetri
**Sumber :**[MUN-06]

### 13.2.2 Algoritma One Time Pad

Algoritma *one-time pad*(OTP)ditemukan oleh Mayor Joseph Mauborgne dan Gilbert Vernam pada tahun 1917.Algoritma *One Time Pad*(OTP) merupakan algoritma yang mencapai kerahasian sempurna yaitu menghasilkan teks sandi yang tidak memiliki hubungan statistik terhadap teks asli sehingga analisa statistik sulit untuk dilakukan [SAD-12]. OTP termasuk cipher aliran stream cipher yaitu cipher yang di hasilkan dari XOR antara setiap bit plaintext dengan setiap bit kuncinya.Algoritma *One Time Pad* bekerja dengan menghasilkan barisan kunci yang acak, panjang kunci harus sama dengan panjang pesan dan tidak ada perulangan kunci.Persamaan dari OTP pada saat kondisi enkripsi ditunjukan dalam (1) sedangkan persamaan dekripsi ditunjukan pada (2).

$$C_1 = P_1 \; XOR \; K_1 \qquad (1)$$

Rumus tersebut adalah rumus OTP dalam keadaan enkripsi, dimana $C_1$ adalah ciphertext yang dalam penelitian iniadalah gambar terenkripsi, $P_1$ adalah Plaintext yang dalam penelitian ini adalah gambar asli dan $K_1$ adalah kunci.

$$P_1 = C_1 \; XOR \; K_1 \qquad (2)$$

Rumus tersebut adalah rumus OTP dalam keadaan dekripsi, dimana $P_1$ adalah plaintext atau gambar asli.Untuk mengenbalikan atau mendekripsi gambar terenkrip ke gambar asli digunakan rumus seperti diatas.Gambar 2.6menunjukan skema proses enkripsi OTP untuk setiap pixel. Ilustrasi akan menunjukan cara kerja OTP pada saat proses enkripsi yang menggunaka persamaan (1).

| | | | | |
|---|---|---|---|---|
| Plaintext | 0 | O | 1 | |
| Key | 0 | 1 | 0 | XOR |
| Ciphertext | 0 | 1 | 1 | |

Pada ilustrasi tersebut di ilustrasikan proses enkripsi gambar pada satu pixel gambar. Plainteks merupakan nilai warna pada pixel 1 gambar, key merupakan kunci yang panjangnya harus sama dengan plaintexs menurut aturan OTP, sedangkan ciphertexs merupakan hasil terenkripsi pada pixel 1 gambar. Ilustrasi proses dekripsi menggunakan persamaan (2).

| | | | | |
|---|---|---|---|---|
| Ciphertext | 0 | 1 | 1 | |
| Key | 0 | 1 | 0 | XOR |
| Plaintext | 0 | 0 | 1 | |

Ilustrasi proses enkripsi berbeda dengan dekripsi akan tetapi tetap menggunakan key atau kunci yang sama. Pada proses dekripsi ciphertext atau pixel terenkripsi kembali didekripsi dengan kunci yang sama pada saat proses enkripsi. Dengan proses tersebut akan menghasilkan plaintext atau gambar asli pada pixel satu.

Meskipun OTP merupakan cipher yang sempurna aman, namun algoritma ini tidak digunakan secara universal dalam aplikasi kriptografi. Alasanya adalah ketidakpraktisan dari kuncinya, kerena panjang kunci harus sama dengan panjang pesan maka OTP hanya cocok untuk pesan berukuran kecil.Mengirimkan barisan kunci melalui saluran komunikasi sebagai media pengiriman pesan juga tidak praktis dengan pertimbangan lalu lintas/ traffic yang padat.Oleh karena itu OTP hanya dapat digunakan jika tersedia saluran komunikasi yang cukup aman untuk mengirim kunci.Kunci OTP dapat di distribusikan dengan menggunakan kurir dengan asumsi saluran komunikasi tidak aman.

### 13.2.3 Metode Chaos

Menurut Zeynep Ipek metode*chaos* merupakan sub disiplin matematika yang mempelajari sistem komplek,sistem komplek disini adalah sistem yang berisi

gerakan begitu banyak yang memerlukan computer untuk menghitung semua berbagai kemungkinan. Teori chaos berasal dari teori sistem yang memperlihatkan kemunculan yang tidak teratur, meskipun sebenarnya teori ini digunakan untuk menjelaskan kemunculan data acak [SUT-06].Karakteristik dari teori chaos adalah peka terhadap perubahan nilai awal. Kepekaan ini artinya perbedaan kecil pada nilai awal fungsi akan menghasilkan perbedaan pada barisan kunci yang dihasilkanya.

Salah satu manfaat dari metode chaos yang terlihat menarik adalah di dalam dunia kriptografi. Oleh kerena itu akhir-akhir ini telah banyak di lakukan riset dari metode chaos untuk kriptografi. Dalam kriptografi metode chaos digunakan sebagai pembangkit bilangan acak yang dalam dunia kriptografi digunakan sebagai pembangkit kunci.

Logistic map adalah salah satu metode dalam chaos yang mempunyai bentuk persamaan paling sederhana.Persamaan ini di populerkan oleh seorang ahli biologi yang bernama Robert May pada tahun 1976. Berikut ini adalah persamaan *Logistic map* [MUN-11].

$$X_{i+1} = R.X_i\ (1 - X_i) \qquad (3)$$

Dimana :

R = Laju Pertumbuhan, bilangan positif antara 0 sampai 4 ( $0 \leq r \leq 4$ )

X = Disebut dengan nilai chaos. Bilangan diantara 0 sampai 1, ($0 \leq x \leq 1$)

Persamaan ini dapat di terapkan dalam kriptografi sebagai pembangkit kunci dengan membuat fungsi sesuai diatas. Berikut adalah contoh proses iterasi dari logistic map :

Misal $r$ = 4.0 dan nilai awal $x_0$ = 0.456

$x_1 = 4.0x_0(1 - x_0) = 0.992256$

$x_2 = 4.0x_1(1 - x_1) = 0.030736$

……

$x_{99} = 4.0x_{98}(1 - x_{98}) = 0.914379$

$x_{100} = 4.0x_{99}(1 - x_{99}) = 0.313162$

Dengan nilai awal yang sama maka proses dekripsi akan menghasilkan barisan nilai fugsi yang sama juga.*Logistic Map* merupakan *pseudo-random number generator* (*PRNG*) atau pembangkit bilangan semi acak maka yang digunakan sebagai kunci pada *logistic map* adalah nilai awalnya $X_0$dengan range nilai antara 0 sampai 1.

### 13.2.4 Embedded System

Secara umum produk-produk elektronik yang banyak beredar saat ini memiliki dua komponen dasar yaitu perangkat keras dan perangkat lunak. Perangkat keras secara fisik dapat dikenali karena bentuknya yang kasat mata. Sebut saja telepon seluler, sebagai contoh perangkat elektronik, yang sehari-hari kita temui. Telepon seluler adalah perangkat keras yang dapat dikenali secara fisik yang didalamnya terdapat serangkaian proses. Salah satu prosesnya adalah pengaturan tampilan ke layar sehingga pengguna merasa nyaman. Kumpulan instruksi yang secara abstrak tertanam dalam sistem telepon seluler adalah contoh perangkat lunak. Perangkat lunak sifatnya tidak kasat mata. Secara umum, berdasarkan fungsinya, perangkat lunak dibagi dua bagian yaitu sistem operasi dan aplikasi. Sistem operasi membungkus perangkat keras dan menyediakan fungsi-fungsi yang dapat digunakan oleh aplikasi.

Secara umum dalam pembuatan produk elektronik diawali dengan pembuatan perangkat keras, perangkat lunak sistem (sistem operasi) dan perangkat lunak aplikasi. Masing-masing, perangkat keras dan perangkat lunak, memiliki kelebihan dan kelemahan dalam hal kinerja dan biaya. Perangkat keras memiliki kecepatan yang lebih tinggi, tapi biaya lebih mahal, sementara perangkat lunak biayanya rendah namun kecepatan prosesnya tidak sebaik perangkat keras. Penggabungan keduanya dengan batas-batas yang fleksibel menjadi menarik untuk menjadi area penelitian sehingga diperoleh produk akhir yang optimal. Tidak ada aturan baku berkenaan dengan prosentase implementasi perangkat keras dan perangkat lunak pada suatu sistem. Penggunaan perangkat keras dan perangkat lunak dapat dioptimasi sesuai dengan kebutuhan dan target produk yang dibuat. Sesuai dengan namanya, Embedded system adalah produk elektronik yang disisipkan kepada

produk lain dan tidak berdiri sendiri. Sebuah sistem yang dirancang untuk melakukan tugas tertentu yang diugnakan dalam produk lain. JPEG *codec* contoh produk *enbedded system*. Sebuah prosesor sederhana yang berfungsi untuk mengkompresi file gambar. Istilah *embedded system* merupakan serapan dari bahasa asing yang diterjemahkan ke dalam bahasa indonesia sebagai sistem benam, sistem tertanam dan sistem embed.

Produk akhir perangkat keras diistilahkan dengan perangkat keras *embedded system*. Istilah *embedded system* lebih mudah dipahami karena pada kenyataannya kebanyakan produk akhir yang dihasilkan tidak dapat berdiri sendiri, lebih merupakan bagian tambahan dari sistem yang telah ada. Penggunaan perangkat *embedded system* ini telah merambah ke berbagai produk yang sehari-hari digunakan seperti telepon selular, pengatur suhu ruangan, mesin Anjungan Tunai Mandiri (ATM), pengendali jarak jauh, mesin cuci dan perangkat elektronik lainnya.

Pada dua dekade terakhir pengembangan perangkat *embedded system* begitu pesat. Peningkatan ini didorong oleh berbagai faktor, yang salah satunya adalah permintaan pasar yang terus meningkat. Dalam rangka memenuhi kebutuhan pasar, maka vendor dan para perancang berusaha untuk melakukan terobosan agar proses pengembang an embedded system lebih baik. Di samping tekanan pasar, dorongan perkembangan teknologi IC juga ikut memicu perkembangan *embedded system*. Kapasitas IC terus meningkat dari waktu-ke waktu, dalam tempo kurang dari dua tahun, kapasitas IC meningkat dua kali lipat.

Kelebihan embedded system adalah memiliki fitur umum yang memungkinkan untuk ditanamkan pada berbagai sistem dan meningkatkan fasilitas sistem tersebut. Ciri-ciri umum embedded system yang membedakannya dengan sistem yang lain adalah sebagai berikut[ABD-13].

1. Dirancang untuk aplikasi tertentu

   Embedded system adalah prosesor yang dirancang untuk tujuan tertentu (single purpose processor). Fungsi-fungsi yang memerlukan kecepatan proses

diimplementasikan sebagai perangkat keras. Sistem dirancang presisi terhadap kebutuhan fungsional. Komponen utamanya unit kendali dan datapath. Prosesor yang dirancang tidak seperti prosesor yang digunakan umum untuk komputer personal. Fungsi-fungsi yang diimplementasikan sesuai dengan fungsi yang dibutuhkan. Sebuah JPEG encoder hanya melakukan fungsi kompresi file gambar ke dalam format JPEG. Prosesor pengatur suhu pada sebuah AC hanya melakukan pengaturan kendali hidup dan matinya AC berdasarkan perubahan suhu ruangan. Autofocus pada kamera hanya mengatur fokus kamera digital pada sebuah objek yang dituju.

Berbeda dengan unit fungsi yang terdapat pada prosesor umum yang mengakomodasi hampir semua operasi seperti operasi logika (and, or, not), operasi dasar aritmetika (penjumlah, pengurang, pengali, pembanding) operasi aritmetik yang lebih komplek. Pada embedded system hanya operasi-operasi tertentu yang diimplementasikan sesuai dengan fungsi yang didukungnya. Misal sebuah sistem embedded system hanya membutuhkan penjumlah, pengurang dan pembanding, maka hanya tiga unit tungsi itu saja yang diimplementasikan. Salah satu perbedaannya terletak pada unit fungsi sistem.

2. Respon Cepat

   Waktu respon yang cepat terhadap pemicu yang berasal dari lingkungan. Proses *autofocus* pada kamera digital menjadi berarti jika prosesnya memenuhi tenggat waktu tertentu. Tenggat waktu respon berbeda-beda untuk setiap sistem rancangan yang berbeda. Orde waktu respon pun berbeda disesuaikan dengan kebutuhan yang disyaratkan. Perangkat keselamatan pada mobil, *air bag sistem*, perlu bereaksi kurang dari satu detik sementara perubahan pengatur AC ruangan tidak perlu secepat itu. Respon cepat ini penting terutama untuk sistem yang dirancang real-time. Sistem pengukur jarak kendaraan untuk keperluan parkir perlu waktu respon yang cepat. Jika responnya telat maka informasi jarak yang diterima oleh pengendara tidak akurat dan menjadi tidak berarti lagi.

3. Dominan perangkat keras

   Pada embedded system, proses dapat diimplementasikan sebagai perangkat keras maupun perangkat lunak. Perangkat keras melakukan proses lebih cepat dari perangkat lunak karena tidak perlu mengakses memori instruksi dan mendefinisikan jenis instruksinya. Pada implementasi embedded system, perangkat keras menjadi dominan untuk memastikan proses-proses penting dilakukan dalam tempo waktu tertentu. Isu utama pada aplikasi embedded system adalah real time. Waktu respon menjadi hal yang penting dalam implernentasinya.

4. Murah

   Embedded system adalah fungsi tambahan yang ditanamkan pada poduk lain, karenanya harus memenuhi standar biaya tertentu. Salah satu kelemahan penggunaan perangkat keras yang dominan adalah biaya tinggi. Implementasi dengan perangkat keras lebih mahal daripada implementasi dengan perangkat lunak. Dalam mengatasi tarik ulur antara kecepatan dan biaya salah satu solusinya adalah jumlah produksinya yang besar.

5. Suplay listrik rendah

   Sistem yang ditanamkan pada sistem lain mendapatkan suplay listrik dari induknya. Sebagai sistem yang menginduk maka konsumsi listriknya harus kecil supaya tidak mengganggu sistem induknya' pengaturan perancangan perangkat kerasa/perangkat lunak menjadi penting. Operasi-operasi yang menyebabkan penggunaan listrik yang tinggi harus dihindari.

Produk-produk *embedded system* sangat banyak dan diimpelementasikan untuk berbagai keperluan sehari-hari. Berikut beberapa contoh produk *embedded system* : telepon seluler, pengendali pesawat, kamera digital, mesin cuci, pengukur jarak kendaraan, kartu pintaq robot, terminal penjualan, mesin pemanas, permainan anak, pengendali papan ketik, kartu jaringan, pengendali printer laser, telpon dengan memori, sistem biomedis, router, switch, bridge, hub, ATM bank, transaksi kartu kredit, kartu telepon bergerak, pengirim/penerima FAX, pemroses suara, pemroses gambar, kompresi dan dekompresi JPEG, videogame, sistem musik,

telepon cerdas, komputer bergerak, jaringan lokal nirkabel, video interaktif, IPv6 pita lebar, video waktu nyata, kartu kemanan, kartu enkripsi dan dekripsi dan pengukur ke dalam air. Masih banyak lagi produk lain yang dibuat berbasiskan embedded system.

Sebagian besar produk-produk *embedded system* yang dicontohkan telah digunakan dalarn keseharian. Seiring dengan jalannya waktu, kebutuhan tersebut terus meningkat sesuai dengan meningkatnya tuntutan dari konsumen. Kebutuhan atau mungkin keinginan dari konsumen, menyebabkan peningkatan permintaan perangkat *embedded system*. Ini menuntut pada perancang untuk terus berinovasi dalam mendukung layanan-layanan yang diperlukan oleh pengguna.

Peningkatan kebutuhan ini menyebabkan apa yang disebut sebagai tekanan waktu pasar, yaitu semakin pendeknya waktu produk untuk memasuki pasar. Semakin besarnya tuntuan pasar menyebabkan proses pembuatan perangkat *embedded system* harus semakin cepat sehingga tidak ketinggalan oleh perkembangan pasar.

### 13.2.5 Field Programable Gate Array (FPGA)

FPGA adalah perangkat elektronik semikonduktor dalam sebuah IC yang dapat dikonfigurasi oleh programmer untuk keperluan simulasi sistem. Papan FPGA sering juga disebut pemodelan atau prototyping pemrograman perangkat keras dimana pengguna dapat menggunakan sesuai dengan kebutuhan.

FPGA terdiri dari sejumlah gerbang yang dapat di program untuk tujuan tertentu. Komponen gerbang yang terdapat dalam sebuah FPGA meliputi gerbang-gerbang logika dasar seperti OR, AND, NOT, XOR dan sejumlah rangkaian kombinasional seperti decoder, penjumlahan, pengali dll. Terdapat juga bagian untuk menyimpan data seperti register, flip-flop atau sel memori lainya. Antar komponen dalam FPGA dihubungkan oleh sistem interkoneksi yang dirancang sedemikian rupa sehingga memudahkan programmer mengkonfigurasi ulang untuk keperluan rancangan tertentu, itulah kenapa disebut *programmable*.

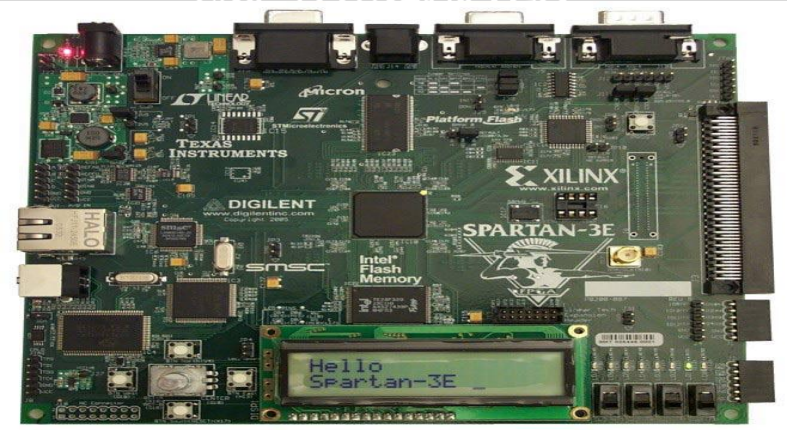
Secara umum FPGA akan lebih lambat jika dibandingkan dengan chip yang lain seperti chip *Application-specific Integrated Circuit* (ASCI). Hal ini kerena

FPGA menggunakan daya yang besar dan bentuk desain yang komplek.Beberapa kelebihan FPGA adalah harganya yang murah, bisa di program sesuai dengan kebutuhan dan bisa di program ulang untuk mengkoreksi adanya kesalahan pada program.Beberapa karakteristik FPGA adalah sebagai berikut [ABD-13].

A. Mudah di konfigurasi ulang artinya FPGA dapat di program sesuai dengan kebutuhan. Program yang tersimpan dalam FPGA bersifat sementara dan dapat diubah setiap saat.
B. Kecepatan tinggi, FPGA terdiri dari sekumpulan perangkat keras dengan kecepatan clock yang tinggi, karena itu aplikasi yang tertanam di dalamnya memiliki kecepatan yang tinggi.
C. Perlindungan hak cipta dan penggunaan ulang rancangan. Ketika sebuah rancangan diterapkan pada FPGA , aplikasinya tidak dapat dikembalikan lagi ke dalam bentuk program, jadi terlindung dari orang-orang yang tidak bertanggung jawab. Program yang sudah dirancang dapat digunakan ulang apan waktu diperlukan.

### 13.2.6 FPGA Spartan 3e

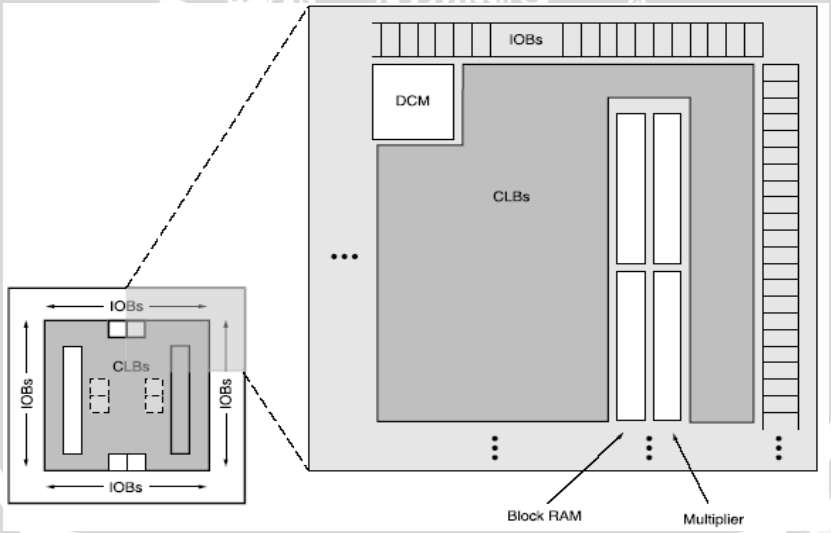
FPGA Spartan 3e adalah salah satu dari keluarga FPGA yang di produksi oleh Xilinx yang dapat dilihat di gambar 2.8.Bahasa pemrograman yang digunakan dalam FPGA ini adalah VHDL (*Very High Hardware Description Language*) dengan bantuan *software development kit* dari Xilinx.




**Gambar2.8**Fpga Spartan 3e
**Sumber :**Data Sheet Xilinx Spartan 3e

Secara umum arsitektur dalam IC FPGA terdiri dari tiga bagian yaitu *Input Output Block* (IOB), *Configurable Logic Block* (CLB), dan Interkoneksi (kanal-kanal routing). Arsitektur dalam IC FPGA Spartan 3e dapat dilihat di gambar 2.9.

1. *Input Output Block* (IOB) mengontrol aliran data antara I/O pin dan logika internal. Setiap IOB juga mendukung bagian standar sinyal.
2. *Configurable Logic Block* (CLB) memiliki Look-Up Tables (LUTs) yang fleksibel dimana dapat mengimplementasikan logika dan ada tempat penyimpanan yang digunakan sebagai flip-flop atau latches. CLB melakukan berbagai fungsi logika dan juga penyimpanan data.
3. Blok RAM menyediakan penyimpanan data dalam bentuk dual port blok 19 Kbit.
4. Block Multiplier menerima dua angka biner 18-bit sebagai input dan menghitungnya.
5. Digital block Manager menyediakan kalibrasi, delay, multipliying, dividing dan fase pergeseran sinyal clock.

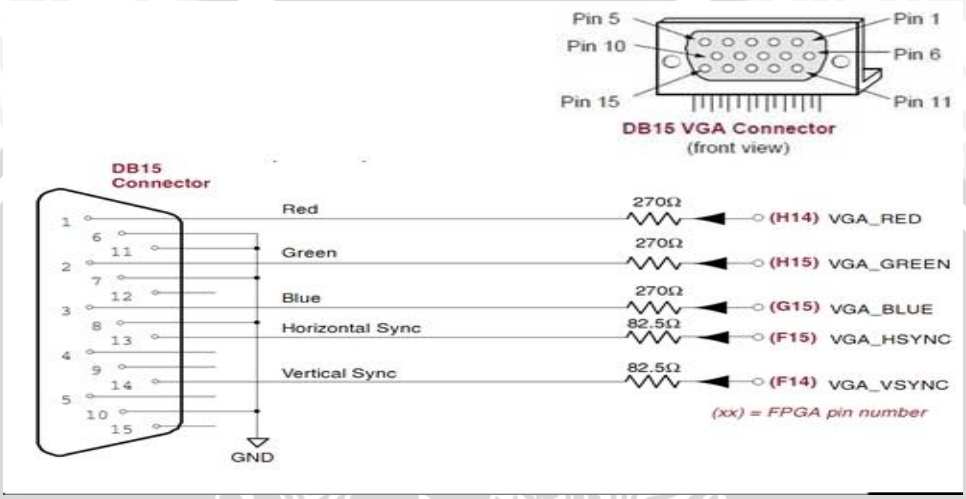



**Gambar2.9**ArsitekturFpga Spartan 3e
**Sumber :**Data Sheet Xilinx Spartan 3e

Semua bagian dari arsitektur ini teroganisir dengan baik seperti yang di tunjukan pada gambar 2.9. Cicin IOBs mengelilingi array CLBs. Setiap RAM

terdiri dari beberapa blok 18Kbit RAM. Setiap blok RAM dikaitkan dengan multiplier. DCMsdiposisikan di tengah dengan dua diata dan dua di bawah.

Di dalam Spartan 3e terdapat port VGA monitor yang bisa dihubungkan melalui konektor DB15. Port ini langsung dihubungkan ke monitor atau LCD menggunakan kabel monitor biasa.Seperti yang ditunjukan dalam gambar 2.10.




**Gambar2.10**Koneksi Port VGAFpga Spartan 3e
**Sumber :**Data Sheet Xilinx Spartan 3e

Pada gambar 2.10 ditunjukan bahwa konektor DB15 mempunyai lima port dan setiap baris warna memiliki port resistor seri. Dengan satu bit untuk masing-masing VGA_RED, VGA_GREEN, VGA_BLUE. Sedangkan VGA_HSYNC dan VGA_VSYNC digunakan sebagai horisontal dan vertikal pada monitor. Yang menggunakan I/O standar.VGA_RED, VGA_GREEN, VGA_BLUE menghasilkan delapan kombinasi warna yang ditunjukan dalam table 2.1.

| VGA_RED | VGA_GREEN | VGA_BLUE | Hasil Warna |
|---|---|---|---|
| 0 | 0 | 0 | BLACK |
| 0 | 0 | 1 | BIRU |
| 0 | 1 | 0 | HIJAU |
| 0 | 1 | 1 | CYAN |
| 1 | 0 | 0 | MERAH |

| | | | |
|---|---|---|---|
| 1 | 0 | 1 | MAGENTA |
| 1 | 1 | 0 | YELLOW |
| 1 | 1 | 1 | WHITE |

**Tabel2.1**Kombinasi Warna LCD Monitor FPGA Spartan 3e
**Sumber :**Data Sheet Xilinx Spartan 3e

### 13.2.7 Bahasa Pemrograman Hardware

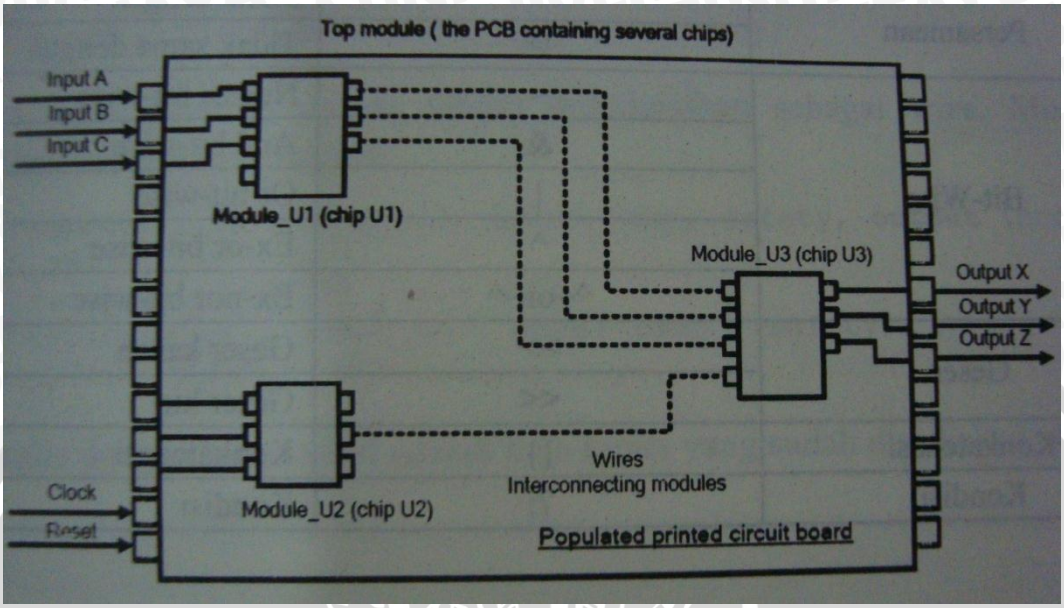
Dalam bahasa pemrograman perangkat keras (*hardware*) ada beberapa bahasa yang umum digunakan untuk pemrograman IC yaitu Verilog dan VHDL.Diantara kedua kode tersebut memiliki perbedaan yang cukup signifikan.Namun secara filosofi konsep, perbedaan dasar dari VHDL dengan Verilog adalah mengenai konteks dari kedua bahasa itu sendiri. Verilog berasal dari tradisi "bottom-up" yang telah sering digunakan dalam industri IC dalam hal rancangan dasar IC. Sedangkan kode VHDL dikembangkan lebih kepada persepektif "top-down".Tentu saja, banyak perbedaan umum dan luas dalam konteks saat ini.Namun, secara jelas dan nyata, perbedaannya dapat terlihat pada syntax dasar dan metode dari kedua kode tersebut. Satu hal penting tentang keunggulan VHDL adalah kemampuannya untuk menggunakan gabungan level dari model yang memiliki arsitektur yang berbeda. Hal tersebut memang bukanlah keunikan atau ciri khas VHDL. Namun, pada kenyataannya kode Verilog juga memiliki konsep sama walaupun hanya terdapat dalam sebuah "module". Meskipun demikian, keungulan itu secara eksplisit didefinisikan dalam VHDL dan secara praktis digunakan bersama oleh rancangan multi level dalam VHDL.

Pada subab 2.2.6.1 dan 2.2.6.2 akan dibahas mengenai bahasa pemrograman perangkat keras verilog dan VHDL. Pada subab 2.2.6.1 verilog akan dijelasakan gambaran umumnya saja karena pada penelitian ini yang digunakan adalah bahasa VHDL.

#### 13.2.7.1 Verilog

Bahasa pemodelan verilog sangat dekat kaitan sematiknya dengan VHDL.Lebih banyak samanya dari pada bedanya [ABD-13].Dalam memahami

konsep pemrograman dalam verilog dapat diambil analogi seperti halnya PCB (Printed Circuit Board) seperti yang ditujukan pada gambar 2.11.




**Gambar2.11**Analogi Pemodelan Verilog
**Sumber :**[ABD-13]

Dalam pemodelan verilog terdapat istilah- istilah berikut : module, top module, port dan write. Untuk memudahkan pemahaman istilah-istilah ini dijelaskan sesuai dengan pengertian dalam PCB.Module adalah sebuah chip yang terdapat di sebuah PCB, pada gambar 2.11 terdapat tiga module yaitu module_U1 (Chip U1), Module_U2 (Chip U2), module_U3 (Chip U3). Port menunjukan pin input/output dalam PCB, contoh pada gambar 2.11 adalah input A, Input B, Input C, Clock, reset dan output x,y,z. Write dapat diumpamakan sebagai jalur yang menghubungkan antara chip dalam PCB. Top module adalah PCB itu sendiri yang didalamnya terdapat banyak chip (module).

### 13.2.7.2 *VHDL*

VHSIC *Hardware Description Language* (VHDL) merupakan sebuah bahasa pemrograman dari VHSIC (*Very High Speed Integrated Circuit*) yang dikembangkan oleh IEEE ( *Institute Electrical and Elektronic Engineering* ) pada tahun 1987, yang digunakan untuk merancang dan memodelkan sebuah rangkaian

digital. VHDL telah mengalami modifikasi dan revisi dengan versi terbaru berlabel IEEE Std 1076-1993.

Desain digital yang akan digunakan dalam VHDL digambarkan dengan menggunakan external view dengan satu atau beberapa view. External view adalah interface dari rancangan sedangkan internal view menyatakan fungsi atau struktur dari rancangan yang dibuat, biasanya satu rancangan memiliki satu atau lebih view [KUR-12].

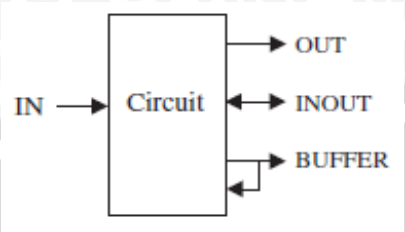
VHDL memungkinkan untuk menggambarkan sistem digital di structural, pada umumnya structural tersebut dibagi menjadi dua yaitu flow dan algoritmik. Data flow merupakan representasi yang menggambarkan bagaimana data bergerak melalui sistem, biasanya ini dilakukan dalam aliran data antara register. Aliran data penggunaan bersamaan membuat pernyataan yang dijalankan secara paralel data segera setelah tiba di masukan. Sedangkan pernyataan sekuensial dijalankan dalam urutan yang mereka tetapkan.

Secara umum pendeskripsian VHDL terdapat dua bagian, yaitu arsitektur dan entity. Bagian lainnya adalah library, sinyal, variable, konstan, array, fungsi dan prosedur. Entitymemberikan arti tentang bagaimana sebuah bagian rancangan dideskripsikan di VHDL dalam hubungannya dengan model VHDL lain dan juga memberikan nama untuk model tersebut. Didalam entityjugadiperbolehkan untuk mendefinisikan beberapa parameter yang mengambil model menggunakan hierarki. Contoh sintaks entity seperti yang ditunjukan pada gambar 2.11

```
ENTITY entity_name IS
   PORT (
      port_name : signal_mode signal_type;
      port_name : signal_mode signal_type;
      ...);
END entity_name;
```

**Gambar2.12**Penulisan Sintax Entity
**Sumber :**[APR-12]

Sinyal *mode* bisa IN, OUT, INOUT, atau BUFFER. Seperti digambarkan dalam gambar 2.11. IN dan OUT adalah pin searah, sedangkan INOUT adalah pin dua arah. BUFFERdigunakan ketika sinyal output harus digunakan internal.




**Gambar2.13**Signal Mode
**Sumber :**[APR-12]

Sinyal *type* bisa BIT, STD_LOGIC, INTEGER, dll. *Nama* pada entity pada dasarnya bisa menggunakan nama apapun.Architecture digunakan untuk mendeskripsikan rangkaian yang akan dibuat. Penulisan architecture ditunjukan pada gambar 2.13.

```
ARCHITECTURE architecture_name OF entity_name IS
    [declarations]
BEGIN
    (code)
END architecture_name;
```

**Gambar2.14**Penulisan Architecture
**Sumber :**[APR-12]

Seperti yang ditunjukkan gambar 2.13, arsitektur memiliki dua bagian: bagian *deklaratif* (opsional), di mana sinyal dan konstanta (antara lain) dinyatakan, dan bagian *kode* (dari BEGIN ke bawah). Seperti dalam kasus entity, nama arsitektur pada dasarnya dapat nama apapun (kecuali kata-kata VHDL yang telah dipesan), termasuk nama yang sama dengan entity.

Pendeskripsian VHDL dapat dilakukan dengan menggunakan metode *behavioral dan structural.*

A. Behavioral

Arsitektur dapat didesain sesuai dengan prinsip kerja alat. Kunci dalam behavior adalah proses. Proses adalah kumpulan dari sequential statement yang dipasang parallel dengan statemen yang sama dan proses yang lain.

B. Structural

Arsitektur structural menghubungkan modul-modul yang ada pada sistem FPGA.Arsitektur structural mendefinisikan struktur interkoneksi yang tepat dari sinyal dan entity.

Bagian lain yang juga penting dalam arsitektur adalah proses dan konfigurasi. Proses adalah sub dari sebuah arsitektur. Dalam suatu arsitektur bisa terdapat satu atau lebih proses. Sedangkan konfigurasi digunakan untuk mendeklarasikan kumpulan desain entity tertentu sebagai komponen.

A. Sinyal = dapat dianalogikan sebagai kabel yang menjadi penghubung antar bagian dalam sistem yang dirancang. Sinyal dideklarasikan dalam arsitektur.

B. Variable = memiliki nilai yang langsung berubah, tanpa harus menunggu selesainya suatu proses. Variable dideklarasikan didalam proses.

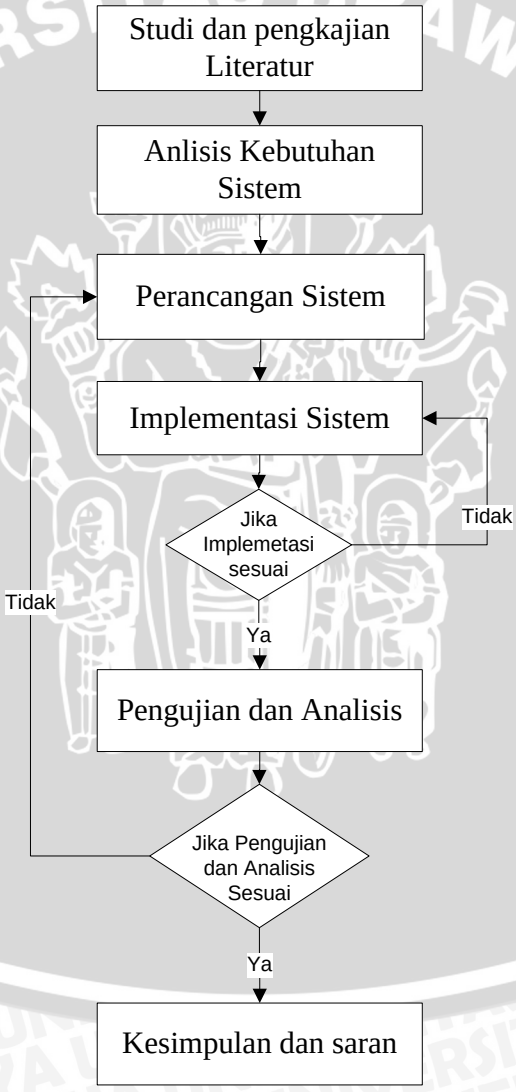
C. Fungsi = dapat menerima suatu data masukan dengan tipe data tertentu dan menghasilkan keluaran yang telah didefinisikan dalam spesifikasinya.

D. Prosedur = hampir sama dengan fungsi, namun prosedur tidak dapat mengembalikan nilai balikan. Prosedur dalam VHDL harus menyertakan argument dengan spesifikasi dengan tipe out atau inout.

# BAB III
# METODOLOGI PENELITIAN DAN PERANCANGAN

## 14.1 Metodologi Penelitian

Bab ini menjelaskan langkah-langkah yang akan ditempuh dalam penyusunan skripsi, meliputi studi dan pengkajian literature, analisis kebutuhan sistem, perancangan sistem, implementasi sistem, pengujian dan analisis, kesimpulan dan saran.




**Gambar3.1***Flowchart*Metode Penelitian

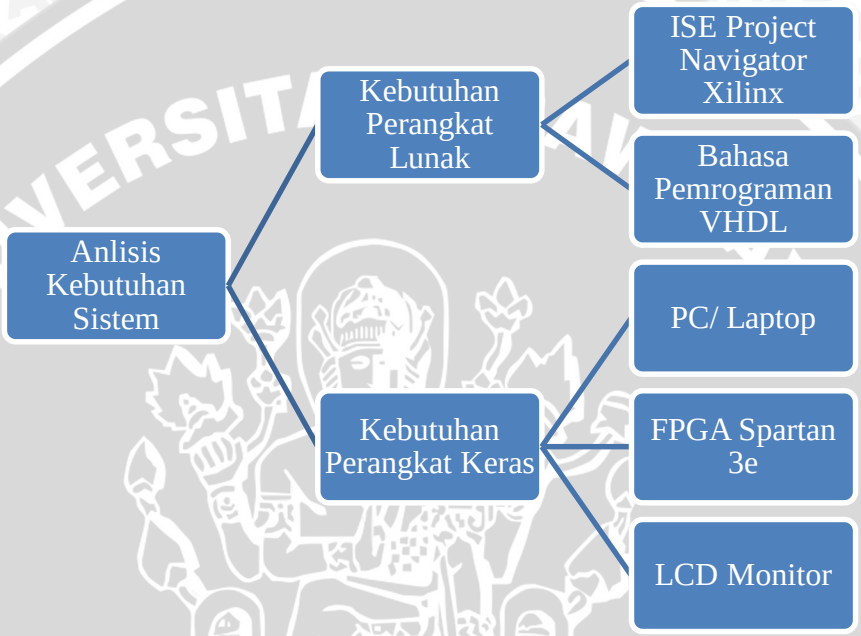
### 14.1.1 Studi dan Pengkajian Literatur

Dalam perancangan dan implementasi penelitian ini, perlu diadakan studi literatur.Literatur digunakan sebagai teori penguat dan landasan dasar dalam penelitian.Teori pendukung tersebut didapat dari buku, jurnal, paper dan internet. Literatur yang digunakan meliputi :

1. Kriptografi
    a. Jenis- jenis Algoritma Kriptografi
    b. Stream cipher
    c. Block cipher
2. Algoritma *One Time Pad*
    *a. One Time Pad*
    b. Cara kerja *One Time Pad*
    c. Persamaan *One Time Pad*
3. *Sistem Chaos*
    *1. Logistic map*
    *2.* Persamaan *Logistic map*
    3. Cara kerja *Logistic map*
4. Embedded System
    *a. FPGA*
    *b. FPGA Spartan 3e*
    *c.* Pin Out *FPGA Spartan 3e*
    *d.* VGA display monitor

### 14.1.2 Analisis Kebutuhan Sistem

Analisis kebutuhan bertujuan untuk menganalisa semua kebutuhan yang diperlukan oleh sistem yang akan dibangun. Analisis kebutuhan dilakukan dengan mengidentifikasi kebutuhan sistem.Analisis kebutuhan tersebut meliputi, kebutuhan perangkat lunak dan kebutuhan perangkat keras.



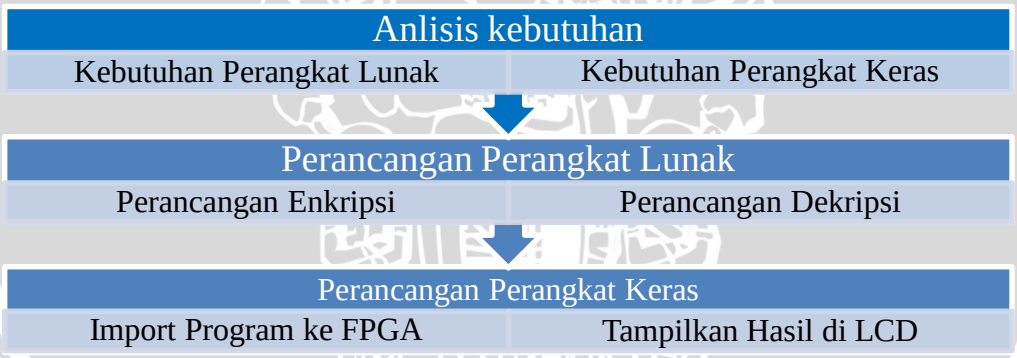

**Gambar3.2**Pohon Analisis Kebutuhan Sistem
**Sumber :**Perancangan

Kubutuhan perangkat lunak digunakan untuk meracang program enkripsi gambar dengan algoritma *One Time Pad*(OTP) dan menggunakan *Logistic map* sebagai pembangkit kunci.Pada kebutuhan perangkat lunak digunakan ISE Project Navigator sebagai *compiler* yang merupakan *software* bawaan dari Xilinx. Selain digunakan sebagai desain program ISE Project Navigator digunakan juga sebagai perantara untuk mengimportkan program kedalam FPGA. Bahasa pemrograman yang digunakan adalah VHDL, karakteristik VHDL adalah mempunyai kemampuan yang lebih untuk merancang sistem digital yang komplek. Satu hal penting tentang keunggulan

VHDLdibandingkan verilog adalah kemampuannya untuk menggunakan gabungan level dari model yang memiliki arsitektur yang berbeda. Pada kebutuhan perangkat keras digunakan FPGA Spartan 3e, FPGA digunakan sebagai simulator pemrograman IC.Selain harganya relative terjangkau untuk kalangan akdemik IC keluaran Xilinx ini dapat diprogram dan dihapus dengan waktu yang tidak terbatas sesuai dengan kebutuhan pengguna. Desain program akan dirancang di *software* kit ISE Project Navigator yang merupakan *software* bawaan Xilinx.

### 14.1.3 Perancangan Sistem

Perancangan sistem dilakukan setelah semua kebutuhan sistem didapatkan melalui tahap analisis kebutuhan yang meliputi kebutuhan perangkat lunak dan kebutuhan perangkat keras.Setelah kebutuhan sistem terpenuhi selanjutnya adalah perancangan perangkat lunak dan perancangan perangkat keras.Seperti yang di tunjukan dalam gambar 3.3.



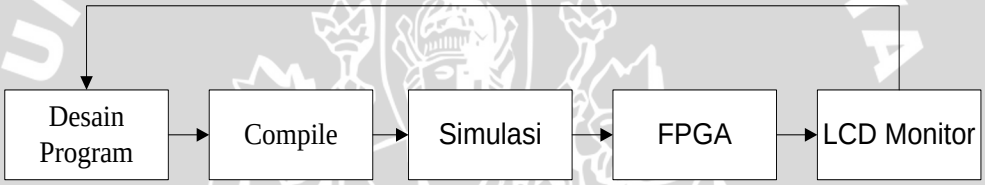

**Gambar3.3**Alur Perancangan Sistem Secara Umum
**Sumber :**Perancangan

Pada perancangan enkripsi dan dekripsi dilakukan dengan algoritma *One Time Pad*(OTP). OTP merupakan algoritma sederhana yang hanya menggunakan operasi modular XOR untuk proses enkripsi dan dekripsinya, berdasarkan penelitian sebelumnya algoritma ini mampu mengenkripsi dengan baik dan dapat mendekripsi gambar dengan tepat seperti gambar aslinya. Sedangakan kunci yang digunakan adalah *Pseudo Number Random Generator* (PNRG) *Logistic map*(LM).LM adalah pembangkit bilangan semi acak,

LMmerupakan salah satu metode sederhana dalam sistem chaos. LM mulai digunakan dalam kriptografi karena memiliki keacakan dari nilai yang dihasilkanya. Pada perancangan perangkat keras terdapat proses import program ke FPGA dan tampilkan hasil dalam LCD Monitor pada proses tersebut digunakan ISE Project Navigator sebagai perantara atau penghubung dalam setiap proses tersebut.

### 14.1.4 Implementasi

Implementasi merupakan perancangan program sampai hasil akhir yang di tampilkan dalam LCD Monitor.Seperti yang digambarkan dalam gambar 3.4.



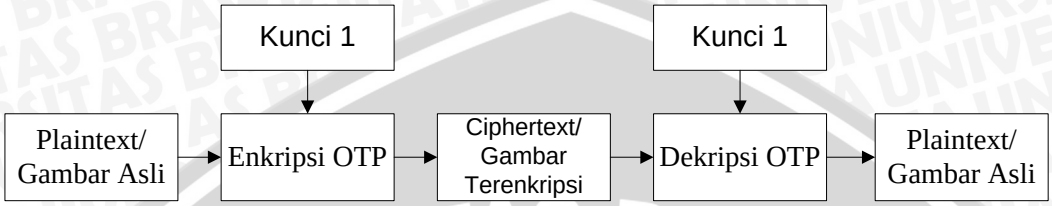

**Gambar3.4**Gambaran Implementasi Secara Umum
**Sumber :**Perancangan

Desain program, *compile* sampai simulasi dijalankan pada lunak pendukung bawaan dari Xilinx yang dalam hal ini adalah ISE Project Navigator. Jika program sukses, *import*program dalam papan FPGA yang kemudian hasilnya akan ditampilkan dalam LCD Monitor. Proses ini mungkin tidak berjalan satu kali, Jika dirasa program belum sesuai kebutuhan, program akan didesain kembali dalam perangkat lunak.

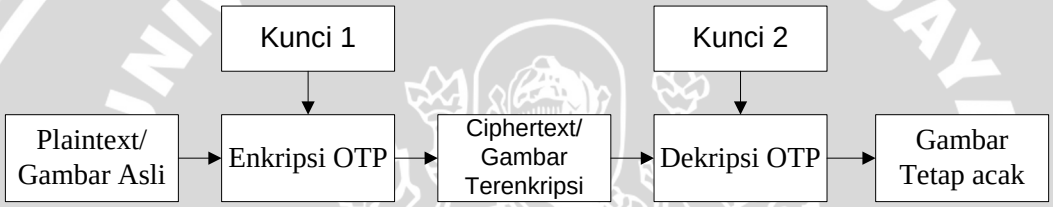
### 14.1.5 Pengujian dan Analisis

Pengujian dilakukan dengan pengujian validitas sesuai aturan *One Time Pad.*Pengujian validitas dilakukan dengan menggunakan dua kasus uji. Kasus uji pertamayaitu menguji dengan caradekripsimenggunakan kunci yang sama jika gambar hasil dekripsi kembali sesuai gambar asli maka dikatakan valid. Kasus uji

kedua adalah dengan caradekripsimenggunakan kunci yang berbeda, jikagambarhasil dekripsi tidak kembali ke gambar asli dan tetap acak maka dikatakan valid.Skenario pengujian validitas dengan kunci sama maupun dengan kunci berbeda yang ditunjukan pada gambar 3.5 dan gambar3.6.




**Gambar3.5**Skenario Pengujian Kasus Uji Pertama
**Sumber :**Perancangan




**Gambar3.3**Skenario Pengujian Kasus Uji Kedua
**Sumber :**Perancangan

Hasil dari pengujian validitas tersebut akan di masukan kedalam table validitas. Berikut merupakan contoh tabel yang akan digunakan sebagai pengelompokan hasil dari pengujian validitas.

| NO | Kasus Uji | Hasil yang diharapkan | Hasil yang didapatkan | Status Validitas |
|---|---|---|---|---|
| 1 | Enkripsi dan dekripsi dengan kunci sama | | | |
| 2 | Enkripsi dan dekripsi dengan kunci berbeda | | | |

**Tabel 3.1**Hasil Uji Validitas
**Sumber :**Perancangan

Pengujian kedua adalah pengujian sensitivitas pada nilai awal kunci *logistic map*.Pengujian ini dilakukan dengan caramelakukan perubahan kecil padanilai awal kunci dekripsi pada *logistic map*.

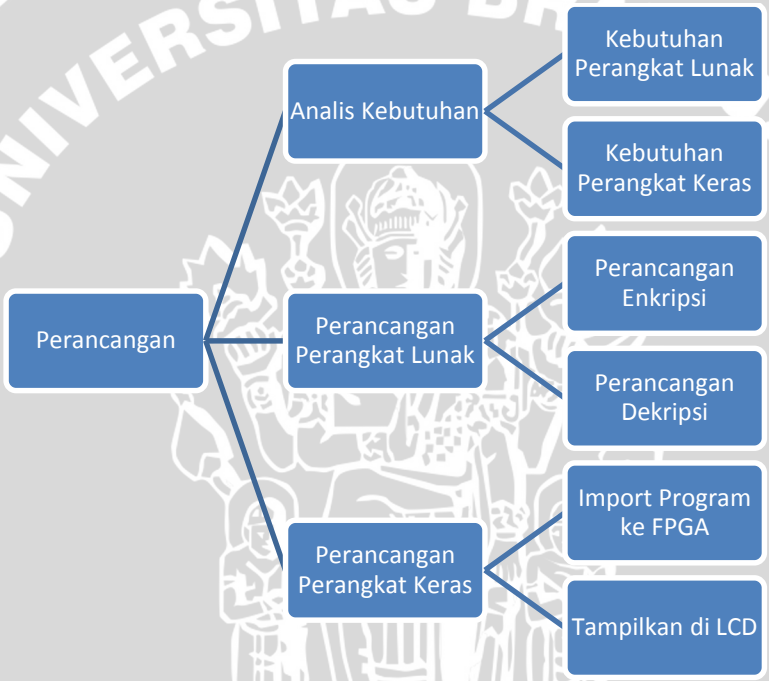
### 14.1.6 Kesimpulan

Pengambilan kesimpulan dilakukan setelah semua tahapan perancangan, implementasi, pengujian dan analisis sistem telah selesai dilakukan.Kesimpulan diambil dari hasil pengujian dan analisis terhadapsistem yang dibuat. Kesimpulan diambil untuk mengtahui validitas dari hasil enkripsi dan dekripsi dengan kunci yang sama maupun berbeda mengacu pada algoritma *One Time Pad*.Selain pengujian validitas juga dilakukan pengujian sensitivitas pada nilai awal kunci *logistic map*. Pengujian ini dilakukan dengan cara melakukan perubahan kecil pada nilai awal kunci dekripsi pada *logistic map*.

Tahap akhir yang dilakukan adalah saran untuk memperbaiki kesalahan-kesalahan yang terjadi dalam pembuatan enkripsi gambar dengan algoritma One Time Pad yang dan menggunakan metode logistic map pada FPGA dan menyempurnakan penulisan serta untuk memberikan pertimbangan atas pengembangan aplikasi selanjutnya.

## 14.2 Perancangan

Perancangan enkripsi gambar dengan algoritma *One Time Pad* dan menggunakan metode *Logistic map* pada FPGA. Perancangan sistem dilakukanberdasarkan subbab 3.1.3.Tahap awal yaitu tahap analisis kebutuhan yang meliputi kebutuhan perangkat lunak dan kebutuhan perangkat keras.Setelah analisis kebutuhan sistem terpenuhi selanjutnya adalah perancangan perangkat lunak dan perancangan perangkat keras. Gambaran umum perancangan akan dijelaskan dalam pohon perancangan sistem pada gambar 3.7.




**Gambar3.7**Pohon Perancangan Sistem
**Sumber :**Perancangan

### 14.2.1 Analisis kebutuhan

Berdasarkan pohon perancangan sistem pada gambar 3.7, analisis kebutuhan merupakan langkahawal sebelum dilakukan perancangan perangkat lunak dan perancangan perangkat keras. Analisis kebutuhan terbagi menjadi dua yaitu analisis kebutuhan perangkat lunak dan analisis kebutuhan perangkat keras.Berikut adalah masing-masing kebutuhan.

#### 7.2.1.1. Kebutuhan Perangkat keras

a. PC atau laptop yang digunakan untuk perancangan program.

b. FPGA Spartan 3e digunakan sebagai simulator pemrograman *hardware*.

c. LCD monitor yang nantinya digunakan sebagai output atau keluaran dari hasil akhir program yang kita buat

d. Kabel USB digunakan untuk menghubungkan antara laptop/ PC dengan FPGA Spartan 3e

e. Kabel VGA Monitor digunakan untuk menghubungkan antara FPGA Spartan 3e dengan monitor.

#### 7.2.1.2. Kebutuhan Perangkat Lunak

a. Xilinx ISE Design Suite 13.4 digunakan sebagai compiler perancangan program enkripsi

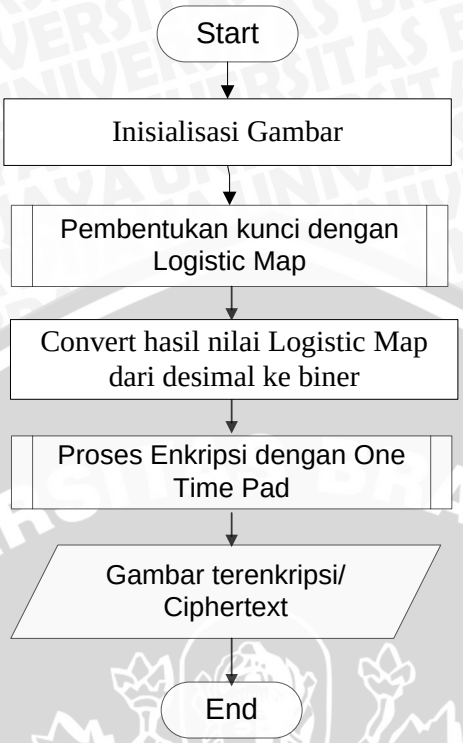
b. VHDL sebagai bahasa pemrograman *hardwere*.

### 14.2.2 Perancangan Perangkat Lunak

Mengacu pada pohon perancangan pada gambar 3.7.Perancangan perangkat lunak dilakukan setelah analisis kebutuhan. Pada penelitian ini perancangan perangkat lunak terdapat dua bagian yaitu perancangan perangkat lunak enkripsi dan perancangan perangkat lunak dekripsi. Berikut adalah masing-masing proses enkripsi dan dekripsi.

#### 4.2.2.1 Perancangan Enkripsi

Perancangan enkripsi merupakan bagian perancangan perangkat lunak.Enkripsi gambar dilakukan dengan Algoritma *One Time Pad*dan menggunakan metode*Logistic map* sebagai pembangkit kuncinya. Diagram alir dari proses enkripsi dijelaskan pada gambar 3.8.




**Gambar3.8**Proses Enkripsi dengan OTP dan Lm
**Sumber :**Perancangan

**A. Proses Inisialisasi Gambar Asli**

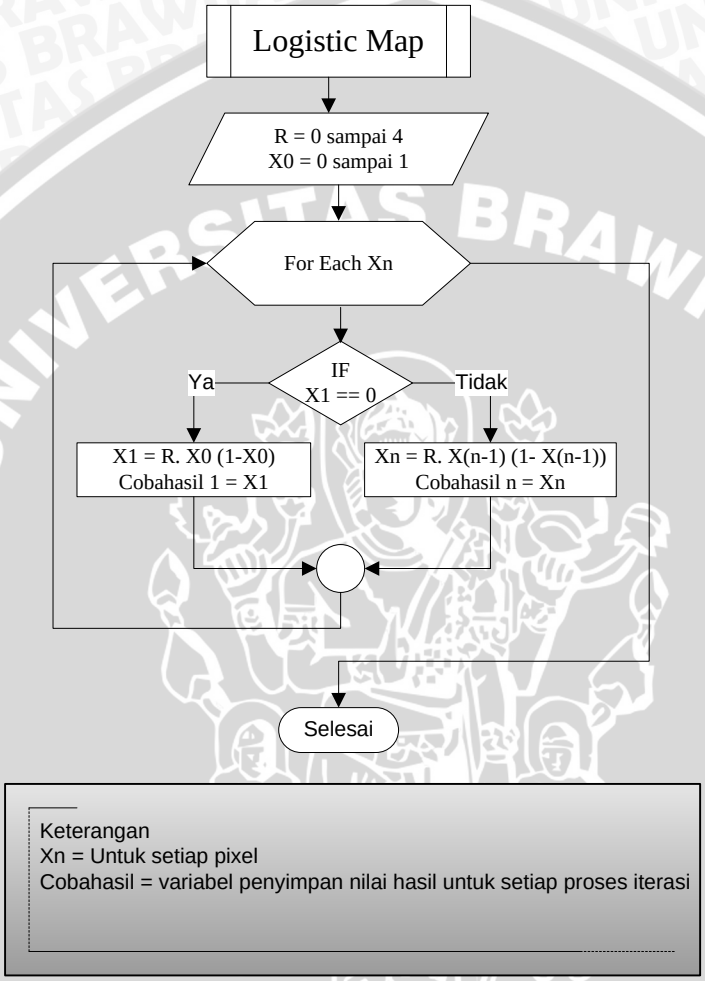
Proses ini dilakukan untuk menampilakan gambar asli di dalam LCD monitor yang nantinya akan di enkripsi. Proses tersebut akan di jelaskan di tabel 3.2.

| Nama Algoritma : Inisialisasi Gambar/ menampilkan gambar |
|---|
| Deskripsi :<br>➔ Masukan : biner nilai warna gambar<br>Proses :<br>• Untuk setiap pixel gambar:<br>o Tandai koordinat tiap pixel pada lcd monitor<br>o Beri nilai warna dengan format biner 3 bit.<br>Output : tampilan gambar dengan 3bit warna |

**Tabel3.2**Algoritma Menampilkan Gambar dalam LCD Monitor
**Sumber :**Perancangan

## B. Proses Pembentukan Kunci dengan Logistic map

Pembentukan kunci dilakukan menggunakan metode *Logistic map*. Flowchart proses pembentukan kunci*logistic map* untuk setiap iterasi pada proses enkripsi ditunjukan pada gambar 3.9.




**Gambar3.9**flowchart pembentukan kunci Logistic Map
**Sumber :**Perancangan

Gambar 3.9 adalah penjelasan proses pembentukan kunci dengan Logistic Map, sedangkan pada tabel 3.3, dijelaskan juga proses pembentukan kunci dengan tabel algoritma.

| Nama Algoritma : pembentukan kunci/Logistic map |
|---|
| Deskripsi : |

| |
|---|
| ➔ Masukan : R. nilai R antara 0 sampai 4<br>➔ Masukan : X. nilai X antara 0 sampai 1<br>Proses :<br>• Untuk setiap iterasi:<br>o Proses masukan nilai dengan persamaan (3)<br>o Simpan hasil iterasi di dalam variable cobahasil1<br>o Convert hasil iterasi ke dalam biner 3 bit.<br>o Simpan hasil convert di dalam variable key1<br><br>Output : menghasilkan barisan kunci yang disimpan dalam variable |

**Tabel3.3**Algoritma Proses Iterasi Kunci Logistic Map
**Sumber :**Perancangan

### C. Algoritma Convert nilai decimal ke biner

Convert atau perubahan nilai dari decimal ke biner 3 bit dilakukan untuk memenuhi persyaratan dari *One Time Pad*. Yaitu panjang kunci harus sama dengan panjang plaintext, karena panjang plaintext 3 bit maka kunci yang akan digunakan untuk mengenkripsi gambar juga harus 3 bit. Proses tersebut dijelaskan pada table 3.4

| Nama Algoritma :convert nilai decimal ke biner |
|---|
| Deskripsi :<br>➔ Masukan : nilai desimal hasil iterasi Logistic map<br>Proses :<br>• Untuk setiap iterasi :<br>o Bagi nilai antara 0 sampai 1 dengan 8<br>o Jika hasil proses iterasi antara 0- 0.125 dirubah jadi biner 000<br>o Jika hasil proses iterasi antara 0.126 - 0.251 diconvert jadi biner 001<br>o Jika hasil proses iterasi antara 0.252 - 0.377 diconvert jadi biner 010<br>o Jika hasil proses iterasi antara 0.378 - 0.503 diconvert jadi biner 100<br>o Jika hasil proses iterasi antara 0.504 - 0.629 diconvert jadi biner 011 |

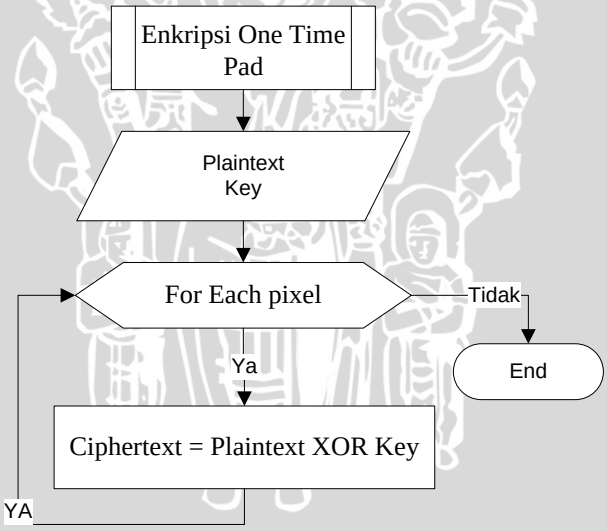
- Jika hasil proses iterasi antara 0.630 - 0.755 diconvert jadi biner 101
- Jika hasil proses iterasi antara 0.756 - 0.881 diconvert jadi biner 110
- Jika hasil proses iterasi antara 0.882 – 1.0 diconvert jadi biner 111

Output : menghasilkan barisan kunci yang disimpan dalam variable key1

**Tabel3.4**Algoritma convert nilai decimal ke biner
**Sumber :**Perancangan

## D. Algoritma enkripsi One Time Pad

Proses enkripsi dilakukan dengan algoritma OTP yang menggunakan persamaan (1). Proses enkripsi dilakukan di setiap pixel gambar. Flowchart proses enkripsi dijelaskan pada gambar 3.10




**Gambar3.10**Flowchart Proses Enkripsi One Time Pad
**Sumber :**Perancangan

Pada gambar 3.10 dijelaskan proses enkripsi dengan algoritma OTP, untuk lebih jelasnya dijelaskan lagi menggunakan algoritma, seperti yang ditujukan pada table 3.4

| Nama Algoritma :Algoritma enkripsi One Time Pad |
| --- |
| Deskripsi :<br>➔ Masukan : Biner nilai warna pixel<br>➔ Masukan : nilai hasil covert dari proses Logistic map<br>Proses :<br>• Untuk setiap iterasi/ pixel 1:<br>o Proses kedua masukan dengan persamaan (1)<br>o Nilai warna pada pixel 1 gambar di XOR dengan nilai LM<br>Output : pixel 1 terenkripsi |

**Tabel3.5**AlgoritmaProses Enkripsi One Time Pad
**Sumber :**Perancangan

#### 14.2.2.2 Perancangan Dekripsi

Mengacu pada pohon perancangan pada gambar 3.7.Perancangan dekripsi merupakan bagian perancangan perangkat lunak.Dekripsi gambar dilakukan dengan Algoritma *One Time Pad*dan menggunakan Logistic map sebagai pembangkit kuncinya. Berikut adalah diagram alir dari proses tersebut.


**Gambar3.11**Proses Dekripsi dengan OTP dan LM
**Sumber :**Perancangan

## A. Proses Inisialisasi Gambar Dekripsi

Proses ini dilakukan untuk menapilakan gambar terenkripsi di dalam LCD monitor yang nantinya akan didekripsi. Proses tersebut akan di jelaskan di tabel 3.5.

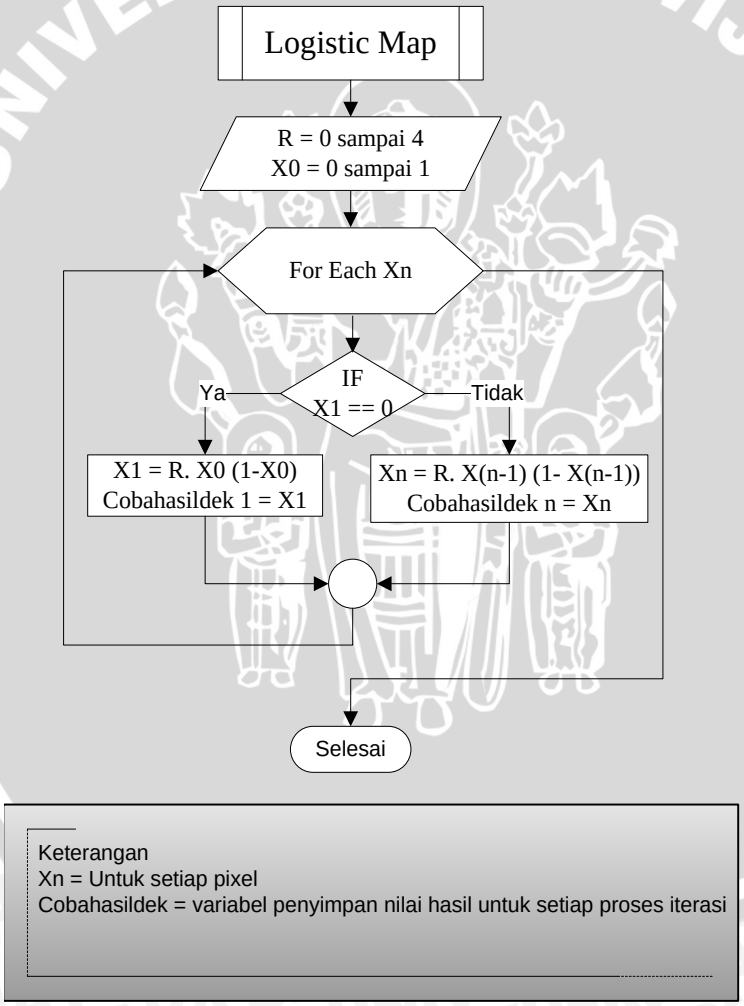
| Nama Algoritma : Inisialisasi Gambar/ menampilkan gambar |
|---|
| Deskripsi :<br>➔ Masukan : biner nilai warna gambar terenkripsi<br>Proses :<br>• Untuk setiap pixel gambar :<br>○ Tandai koordinat tiap pixel pada lcd monitor<br>○ Dengan nilai warna yang sudah terenkripsi. |

| |
|---|
| Output : tampilan gambar terenkripsi |

**Tabel3.6**Algoritma Menampilkan Gambar dalam LCD Monitor
**Sumber :**Perancangan

## B. Proses Pembentukan Kunci dengan Logistic map

Pembentukan kunci pada proses dekripsi pada dasarnya prosesnya sama dengan pembentukan kunci pada saat proses enkripsi. Pembentukan kunci dilakukan dengan metode Logistic map. Gambar 3.11 menunjukan flowchart proses pembentukan kunci logistic map pada proses dekripsi.




**Gambar3.12**Proses Pembentukan Kunci Dekripsi
**Sumber :**Perancangan

Gambar 3.12 dijelaskan proses pembentukan kunci dekripsi dengan logistic map. Untuk lebih jelasnya proses pembentukan kunci logistic map ditunjukan pada Tabel 3.6

| Nama Algoritma : pembentukan kunci/Logistic map |
|---|
| Deskripsi :<br>➔ Masukan : R. nilai R antara 0 sampai 4<br>➔ Masukan : X. nilai X antara 0 sampai 1<br>Proses :<br>• Untuk setiap iterasi :<br>o Proses masukan nilai dengan persamaan (3)<br>o Simpan hasil iterasi di dalam variable hasil1<br>o Convert hasil iterasi ke dalam biner 3 bit.<br>o Simpan hasil convert di dalam variable key1<br>Output : menghasilkan barisan kunci yang disimpan dalam variable |

**Tabel3.7**AlgoritmaProses Iterasi Logistic Map
**Sumber :**Perancangan

### C. Algoritma Convert nilai desimal ke biner

Convert atau perubahan nilai dari decimal ke biner 3 bit dilakukan untuk memenuhi persyaratan dari *One Time Pad*. Yaitu panjang kunci harus sama dengan panjang plaintext, karena panjang plaintext 3 bit maka kunci yang akan digunakan untuk mengenkripsi gambar juga harus 3 bit. Proses tersebut akan dijelaskan pada table 3.7

| Nama Algoritma : convert nilai decimal ke biner |
|---|
| Deskripsi :<br>➔ Masukan : nilai desimal hasil iterasi Logistic map<br>Proses :<br>• Untuk setiap iterasi :<br>o Bagi nilai antara 0 sampai 1 dengan 8<br>o Jika hasil proses iterasi antara 0- 0.125 dirubah jadi biner 000 |

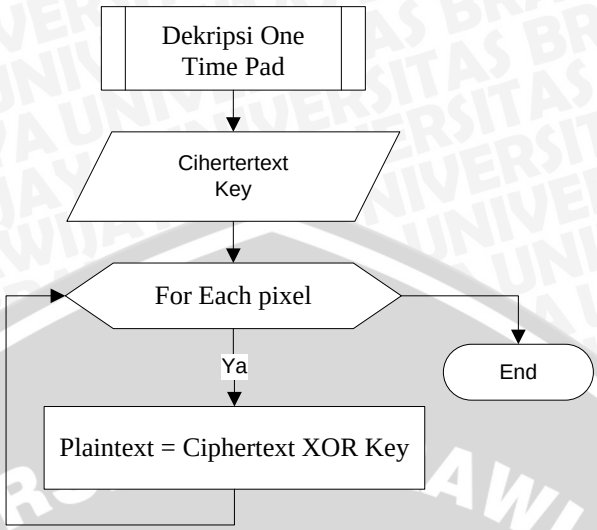
- Jika hasil proses iterasi antara 0.126 - 0.251 diconvert jadi biner 001
- Jika hasil proses iterasi antara 0.252 - 0.377 diconvert jadi biner 010
- Jika hasil proses iterasi antara 0.378 - 0.503 diconvert jadi biner 100
- Jika hasil proses iterasi antara 0.504 - 0.629 diconvert jadi biner 011
- Jika hasil proses iterasi antara 0.630 - 0.755 diconvert jadi biner 101
- Jika hasil proses iterasi antara 0.756 - 0.881 diconvert jadi biner 110
- Jika hasil proses iterasi antara 0.882 – 1.0 diconvert jadi biner 111

Output : menghasilkan barisan kunci yang disimpan dalam variable key1

**Tabel3.8**AlgoritmaProses Convert nilai Decimal ke Biner
**Sumber :**Perancangan

## E. Algoritma Dekripsi One Time Pad

Proses dekripsi dilakukan dengan algoritma OTP yang menggunakan persamaan (2). Proses dekripsi dilakukan di setiap pixel gambar. proses tersebut dijelaskan pada table 3.8 Flowchart proses enkripsi dijelaskan pada gambar 3.13



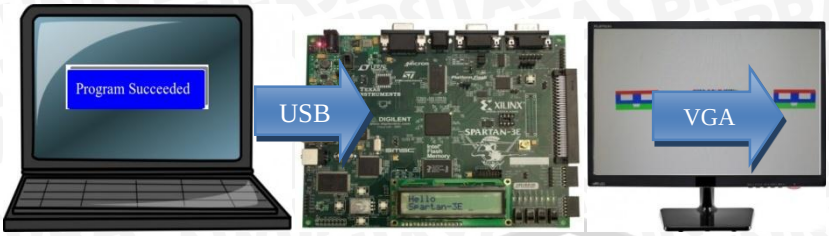

**Gambar3.13**Flowchart Proses Enkripsi One Time Pad
**Sumber :**Perancangan

| Nama Algoritma :Algoritma enkripsi One Time Pad |
|---|
| Deskripsi :<br>➔ Masukan : Biner pixel 1 nilai warna gambar terenkripsi<br>➔ Masukan : nilai hasil covert dari proses Logistic map<br>Proses :<br>• Untuk setiap iterasi/ pixel 1 :<br>o Proses kedua masukan dengan persamaan (2)<br>o Nilai warna pixel 1 gambar terenkripsi di XOR dengan nilai Logistic map<br><br>Output : pixel 1 kembali ke pixel semula/ plainteks |

**Tabel3.9**AlgoritmaProses Enkripsi One Time Pad
**Sumber :**Perancangan

### 14.2.3 Perancangan Perangkat keras

Berdasarkan pada pohon perancangan pada gambar 3.7.Perancangan perangkat keras di lakukan setelah analisis kebutuhan dan perancangan perangkat lunakselesai dilakukan.Perancangan perangkat keras ditunjukan pada gambar 3.10.




**Gambar 3.10**Perancangan Perangkat Keras

**Sumber** : Perancangan

Perancangan perangkat keras dilakukan setelah perancangan perangkat lunak selesai dilakukan. Seperti yang sudah disebutkan di dalan analisis kebutuhan perangkat keras, Pada perancangan perangkat keras terdapat laptop/PC,FPGA Spartan 3e,LCD monitor.

Setelah desain program selesai dan program sukses langkah selanjutnya adalah mengimportkan program kedalam FPGA Spartan 3e, pada proses ini dibutuhkan kabel USB untuk menghubungkan antara laptop dengan FPGA Spartan 3e. Selanjutnya setelah program sukses diimportkan didalam papan FPGA Spartan 3e, langkah terakhir adalah menampilkan hasil program yang sudah diimportkan di papan FPGA Spartan 3e kedalam LCD monitor. Pada proses ini di butuhkan kabel VGA sebagai penghubung antara FPGA Spartan 3e dengan LCD monitor.

# BAB IV
# IMPLEMENTASI

## 15.1 Lingkungan Implementasi

Lingkungan implementasi dari enkripsi gambar dengan algoritma *One Tima Pad*danmenggunakan metode*Logistic map* pada fpga meliputi lingkungan perangkat lunak (Software) dan lingkungan perangkat keras (Hardware).

### 15.1.1 Lingkungan Perangkat Lunak

Lingkungan perangkat lunak yang digunakan dalam mengimplementasikan enkripsi gambar dengan algoritma one time pad dan menggunakan metode logistic mapadalah sebagai berikut:

1. Sistem operasi windows 7 ultimate 32-bit sebagai lingkungan proses implementasi.
2. ISE Project Navigator 13.4 merupakan *software development*bawaan dari Xilinx spartan3e yang digunakan untuk pemrograman.

### 15.1.2 Lingkungan Perangkat Keras

Lingkungan perangkat keras yang digunakan dalam mengimplementasikan enkripsi gambar dengan algoritma one time pad dan menggunakan metode logistic map pada fpga adalah sebagai berikut:

1. FPGA Xilinx Spartan 3e
2. LCD Monitor
3. Kabel USB
4. VGA Monitor
5. Laptop prosesor intel(R) core (TM) i3 CPU M330
6. RAM 2GB

## 15.2 Batasan Implementasi

Beberapa batasan implementasi dalam implementasi enkripsi gambar dengan algoritma One Time Pad dan menggunakan metode Logistic Map pada FPGA sebagai berikut :

1. Enkripsi gambar diimplementasikan di FPGA Spartan 3e menggunakan bahasa pemrograman VHDL.
2. Gambar yang digunakan adalah gambar buatan sendiri.
3. Gambar yang digunakan dengan kedalaman warna 3 bit.
4. Pixel gambar yang digunakan tidak berjumlah lebih dari 400 pixel.

## 15.3 Implementasi Perangkat Lunak

Implementasi enkripsi gambar dengan algoritma one time pad dan menggunakan metode logistic map pada fpga berdasarkan perancangan sistem pada BAB III.Pada implementasi perangkat lunak terdapat dua bagian yaitu implementasi enkripsi dan implementasi dekripsi.Bahasa yang digunakan adalah bahasa VHDL.

### 15.3.1 Implementasi Enkripsi

Padaproses implementasi enkripsi terdapat beberapa proses yaitu inisialisasi gambar, pembentukan kunci dengan *logistic map*, convert hasil nilai logistic map dari decimal ke biner, proses enkripsi dengan *One Time Pad.*

#### 15.3.1.1 Inisialisasi Gambar

Tahap awal dalam proses enkripsi adalah inisialisasi gambar, inisilisasi gambar adalah menampilkan gambar pada LCD monitor yang gunanya untuk mengetahui hasil proses enkripsi dan dekripsi. Source code inisialisasi gambar akan di jelaskan pada source code 4.1.

```
Port (R,G,B, Vsync, Hsync, clk);
architecture Behavioral of Bismillah is
signal photonX : integer range 0 to 640 := 0;
signal photonY : integer range 0 to 480 := 0;
signal halfClock : STD_LOGIC;
```

```
signal horizontalPosition : integer range 0 to 800 := 0;
signal verticalPosition : integer range 0 to 521 := 0;
signal hsyncEnable : STD_LOGIC;
signal vsyncEnable : STD_LOGIC;
signal color : STD_LOGIC_VECTOR (2 downto 0) := "000";
begin
    if photonX >= 225 and photonX <= 225 and photonY >= 82
    and photonY <= 82 then color <= "100";
    elsif photonX >= 226 and photonX <= 226 and photonY >=
    82 and photonY <= 82 then  color <= "100";
    Else color <= "111";
        end if;
    end process colorSetter1;
    draw : process(photonX, photonY, halfClock)
    begin
        if halfClock'event and halfClock = '1' then
            Hsync <= hsyncEnable;
            Vsync <= vsyncEnable;
            if (photonX < 640 and photonY < 480) then
                R <= color(2);
                G <= color(1);
                B <= color(0);
              else
                R <= '0';
                G <= '0';
                B <= '0';
            end if;
    end process draw;
end Behavioral;
```

**Source Code4.1**ProsesInisialisasi Gambar Tiap Pixel

#### 15.3.1.2 Pembentukan Kunci EnkripsiLogistic map

Pembentukan kunci dilakukan dengan metode logistic map, logistic map merupakan psoudo number random generator (PNRG).Logistic map merupakan pembangkit bilangan semi acak, oleh kerena itu yang digunakan sebagai kunci adalah nilai awal X. Dengan nilai awal x akan menghasilkan barisan nilai sejumlah n atau sejumlah pixel. Source code pembentukan kunci dengan logistic mapakan di jelaskan pada source code 4.2.

```
variable r : real ;
variable x_awal: real ;
variable cobahasil: real ;
r := 3.999;
x_awal := 0.596;
cobahasil := r * x_awal *(1.0 - x_awal);
```

**Source Code4.2**Pembentukan Kunci Logistic map

Proses selanjutnya merubah hasil nilai iterasi dari persamaan logistic map dari decimal ke biner yang dapat dilihat di source code 4.3.convert nilai dari decimal kebiner dilakukan untuk memenuhi persyaratan dari one time pad.pada source code 4.3 di jelaskan proses iterasi untuk satu kali proses convert nilai decimal ke biner.

```
Variable Cobahasil:real;
signal key1 :  STD_LOGIC_VECTOR (2 downto 0);
if  0.0 < cobahasil and cobahasil < 0.125 then
     key1 <= "000" ;
     elsif  0.126 < cobahasil and cobahasil < 0.251 then
     key1 <= "001" ;
     elsif  0.252 < cobahasil and cobahasil < 0.377 then
     key1 <= "010" ;
     elsif  0.378 < cobahasil and cobahasil < 0.503 then
```

```
        key1 <= "011" ;
        elsif  0.504 < cobahasil and cobahasil < 0.629 then
        key1 <= "100" ;
        elsif  0.630 < cobahasil and cobahasil < 0.755 then
        key1 <= "101" ;
        elsif  0.756 < cobahasil and cobahasil < 0.881 then
        key1 <= "110" ;
        elsif  0.882 < cobahasil and cobahasil < 1.0 then
        key1 <= "111" ;
end if;
```

**Source Code4.3**Convert Nilai Logistic mapDari Desimal ke Biner

#### 15.3.1.3 Proses Enkripsi Gambar dengan One Time Pad

Tahap ketiga adalah proses enkripsi gambar dengan Algoritma One Time Pad, proses ini dilakukan setelah tahap kedua yaitu pembentukan kunci selesai dilakukan. Proses enkripsi akan dijelaskan pada source code 4.4.

| | |
|---|---|
| 1 | `elsif photonX >= 225 and photonX <= 225 and photonY >= 82 and photonY` |
| 2 | `<= 83 then  color <= "100" xor key1;` |
| 3 | `elsif photonX >= 226 and photonX <= 226 and photonY >= 82 and photonY` |
| 4 | `<= 83 then  color <= "100" xor key2;` |

**Source Code4.4**Proses Enkripsi

Pada source code 4.4. dijelaskan proses enkripsi untuk dua pixel. Pada baris 1 dan 2adalah proses enkripsi pada pixel pertama. Pada proses tersebut nilai warna 001 di XOR kan dengan nilai hasil iterasi logistic map yang sudah diconvert dalam bentuk biner yang disimpan dalam variable key1. Pada baris 3 dan 4 adalah proses enkripsi pada pixel kedua.

### 15.3.2 Implementasi Dekripsi

Padaproses implementasi dekripsi terdapat beberapa proses yang hampir sama dengan proses enkripsi yaitu inisialisasi gambar, pembentukan kunci dengan *logistic map*, convert hasil nilai logistic map dari desimal ke biner, proses dekripsi dengan *One Time Pad.*

#### 15.3.2.1 Inisialisasi Gambar

Tahap awal dalam proses dekripsi adalah inisialisasi gambar, inisilisasi gambar adalah menampilkan gambar pada LCD monitor yang gunanya untuk mengetahui hasil proses enkripsi dan dekripsi. Source code inisialisasi gambar akan di jelaskan pada source code 4.5.

```
Port (R,G,B, Vsync, Hsync, clk);
architecture Behavioral of Bismillah is
signal photonX : integer range 0 to 640 := 0;
signal photonY : integer range 0 to 480 := 0;
signal halfClock : STD_LOGIC;
signal horizontalPosition : integer range 0 to 800 := 0;
signal verticalPosition : integer range 0 to 521 := 0;
signal hsyncEnable : STD_LOGIC;
signal vsyncEnable : STD_LOGIC;
signal color : STD_LOGIC_VECTOR (2 downto 0) := "000";
begin
elsif photonX >= 367 and photonX <= 368 and photonY >= 82
and photonY <= 83 then color <= "100";
elsif photonX >= 369 and photonX <= 370 and photonY >= 82
and photonY <= 83 then color <= "100" ;
     Else color <= "111";
          end if;
     end process colorSetter1;
     draw : process(photonX, photonY, halfClock)
```

```
    begin
        if halfClock'event and halfClock = '1' then
            Hsync <= hsyncEnable;
            Vsync <= vsyncEnable;
            if (photonX < 640 and photonY < 480) then
                R <= color(2);
                G <= color(1);
                B <= color(0);
            else
                R <= '0';
                G <= '0';
                B <= '0';
            end if;
    end process draw;
end Behavioral;
```

**Source Code4.5**ProsesInisialisasi Gambar Tiap Pixel

Source code 4.5 adalah proses inisialisasi gambar dekrip prosesnya hampir sama dengan inisialisasi gambar asli dan terenkripsi perbedaanya terletak pada penempatan nilai koordinat pixelnya saja.

### 15.3.2.2 Pembentukan Kunci DekripsiLogistic map

Tahap kedua proses dekripsi adalah pembentukan kunci dengan menggunakan metode logistic map. Pada proses dekripsi juga di perlukan kunci yang sama untuk menghasilkan gambar asli yang mana pada penelitian ini digunakan metode logistic map. Pembentukan kunci akan dijelaskan pada source code 4.6.

```
variable Rdek : real ;
variable x_awaldek: real ;
variable cobahasildek: real ;
r := 3.999;
```

```
x_awal := 0.596;

cobahasil := Rdek * x_awaldek *(1.0 - x_awaldek);
```

**Source Code4.6**Pembentukan kunci dekripsi Logistic map

Setelah nilai hasil iterasi dari persamaan logistic map didapatkan langkah selanjutnya adalah merubah nilai hasil iterasi dari decimal ke biner.Convert nilai dari decimal kebiner dilakukan untuk memenuhi persyaratan dari one time pad. pada source code 4.7 di jelaskan proses iterasi untuk satu kali proses convert nilai decimal ke biner.

```
if  0.0 < cobahasildek and cobahasildek < 0.125 then

     keydek1 <= "000" ;

     elsif  0.126 < cobahasildek and cobahasildek < 0.251
then

     keydek1 <= "001" ;

     elsif  0.252 < cobahasildek and cobahasildek < 0.377
then

     keydek1 <= "010" ;

     elsif  0.378 < cobahasildek and cobahasildek < 0.503
then

     keydek1 <= "011" ;

     elsif  0.504 < cobahasildek and cobahasildek < 0.629
then

     keydek1 <= "100" ;

     elsif  0.630 < cobahasildek and cobahasildek < 0.755
then

     keydek1 <= "101" ;

     elsif  0.756 < cobahasildek and cobahasildek < 0.881
then

     keydek1 <= "110" ;
```

```
    elsif  0.882 < cobahasildek and cobahasildek < 1.0
then

    keydek1 <= "111" ;

end if;
```

**Source Code4.7**Convert Nilai Logistic mapDari Desimal ke Biner

### 15.3.2.3 Proses Dekripsi Gambar dengan One Time Pad

Tahap ketiga adalah proses dekripsi gambar dengan Algoritma One Time Pad, proses ini dilakukan setelah tahap kedua yaitu pembentukan kunci selesai dilakukan. Proses enkripsi akan dijelaskan pada source code 4.8.

| | |
|---|---|
| 1 | `elsif photonX >= 367 and photonX <= 367 and photonY >= 82 and photonY` |
| 2 | `<= 83 then color <= ("100" xor key1) xor keydek1;` |
| 3 | `elsif photonX >= 368 and photonX <= 368 and photonY >= 82 and photonY` |
| 4 | `<= 83 then color <= ("100" xor key2) xor keydek2;` |

**Source Code4.8**Proses Dekripsi

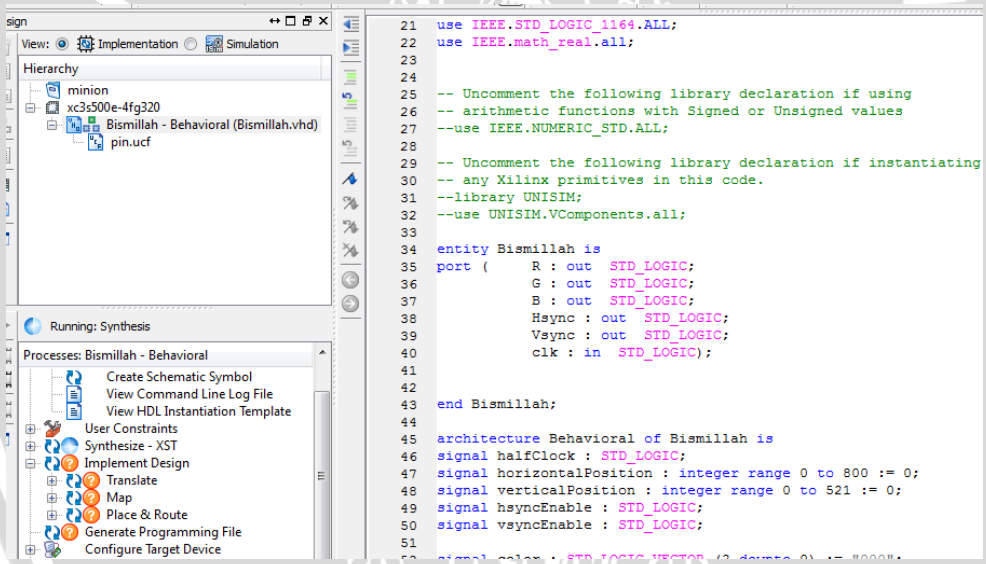
Pada source code 4.8, dijelaskan proses enkripsi untuk dua pixel. Pada baris 1 dan 2 adalah proses dekripsi pada pixel pertama. Pada proses tersebut nilai warna 100 di XOR kan dengan nilai hasil iterasi logistic map yang sudah diconvert dalam bentuk biner yang disimpan dalam key1 yang digunakan pada proses enkripsi kemudian di XOR kan lagi variable keydek1 yang berisi nilai hasil iterasi logistic map yang sudah diconver dalam bentuk biner. Pada baris 3 dan 4 adalah proses dekripsi untuk pixel kedua.

## 15.3 Implementasi Perangkat Keras

Implementasi enkripsi gambar dengan algoritma one time pad dan menggunakan metode logistic map pada fpga berdasarkan perancangan sistem pada BAB III. Implementasi perangkat keras terdapat dua bagian yaitu implementasi import program ke FPGA dan implementasi menampilkan hasilnya dalam LCD monitor. FPGA yang digunakan adalah FPGA Spartan 3e keluaran Xilinx.

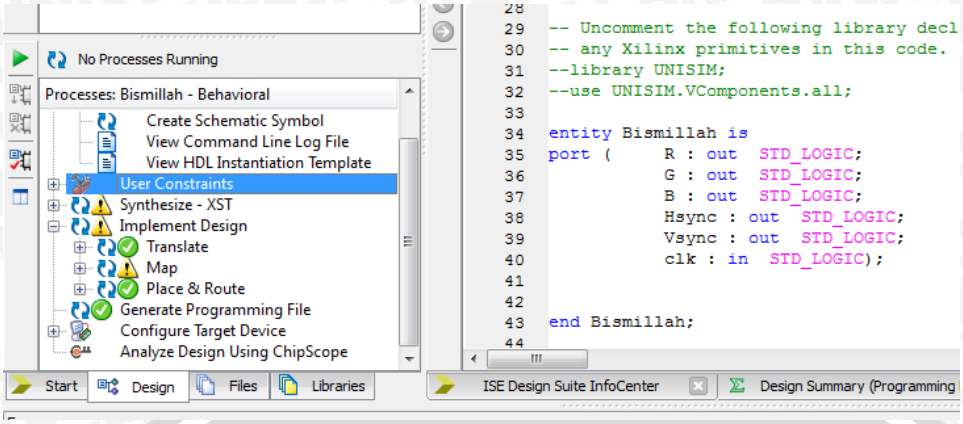
### 15.3.1 Implementasi Import Program Kedalam FPGA

Tahap awal yang dilakukan sebelum import program kedalam FPGA adalah memastikan bahwa program yang telah di desain didalam ISE Project Navigator selesai, berhasil dijalankan dan tidak ada error dalam program. Proses compile program seperti yang ditunjukan pada gambar 4.1.



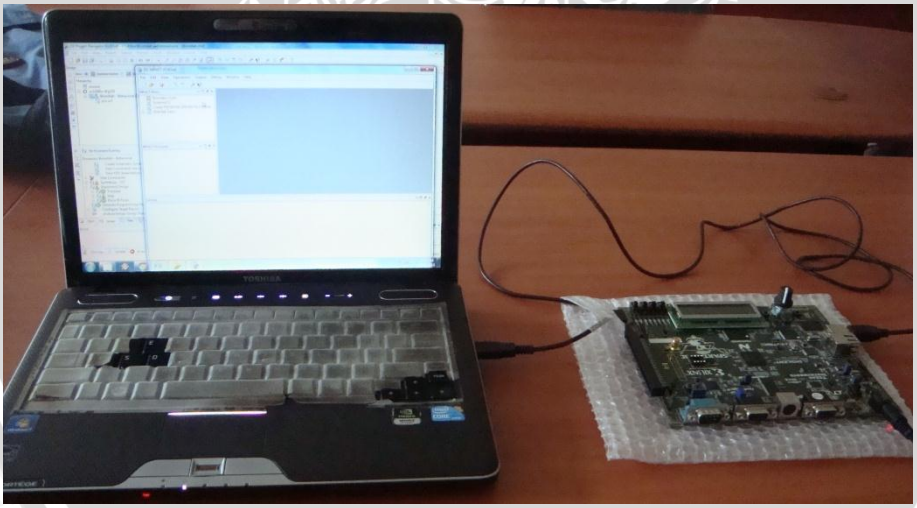

**Gambar4.1**Proses Compile Program
**Sumber**: Implementasi

Program pada synthesize, translate, map, place and route dan generate program harus berhasil dan berwarna hijau atau berwarna kuning. Warna kuning menunjukan dalam program masih beberapa warning namun warning dalam program tidak menggagalkan program dalam artian program bisa diimportkan dalam FPGA. Tanda selesainya program dan siap di importkan dalam FPGA ditunjukan dalam gambar 4.2.




**Gambar4.2**Program sukses
**Sumber**: Implementasi

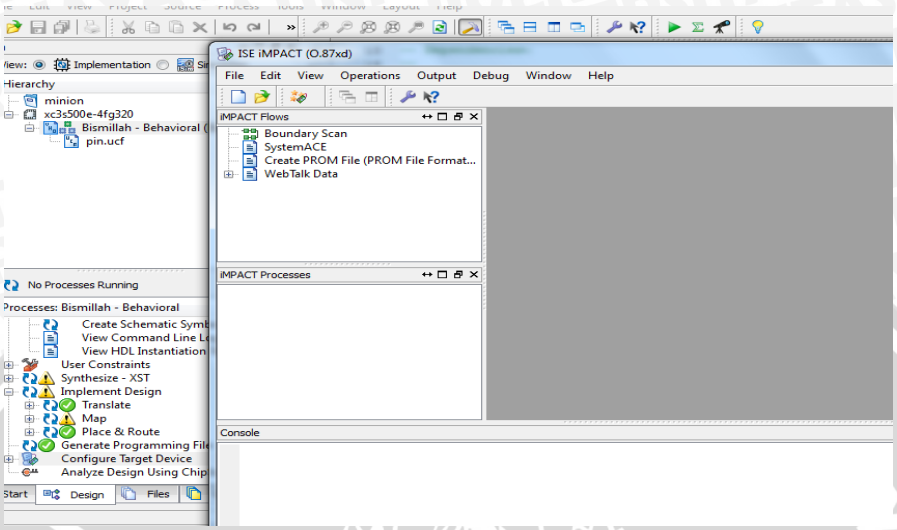
dalah proses import program didalam FPGA. Dalam hal ini dibutuhkan kabel USB untuk menghubungkn antara laptop dengan FPGA Spartan 3e.Desainnya ditunjukan dalam gambar 4.3.



**Gambar4.3**Menghubungkan Laptop dengan FPGA
**Sumber**: Implementasi

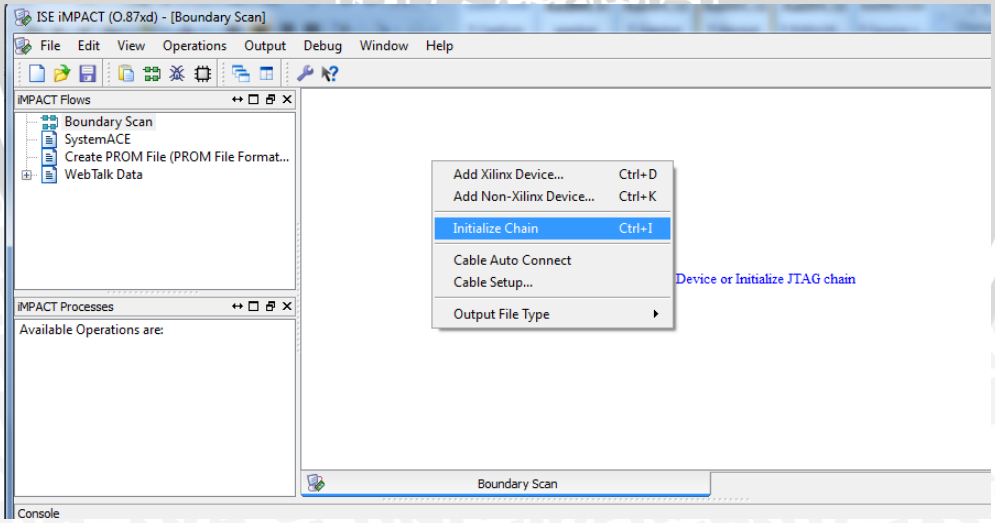
Setelah laptop dan FPGA dihubungkan dengan kabel USB langkah selanjutnya adalah *configure target service* hal ini dilakukan untuk mengidentifikasi target import program yang dalam hal ini digunakan FPGA Spartan 3e. Pada saat klik *configure target service*akan keluar ISE impact pada modul ini akan digunakan

untuk mengidentifikasi FPGA Spartan 3e. gambar 4.4 menunjukan proses *configure target service.*



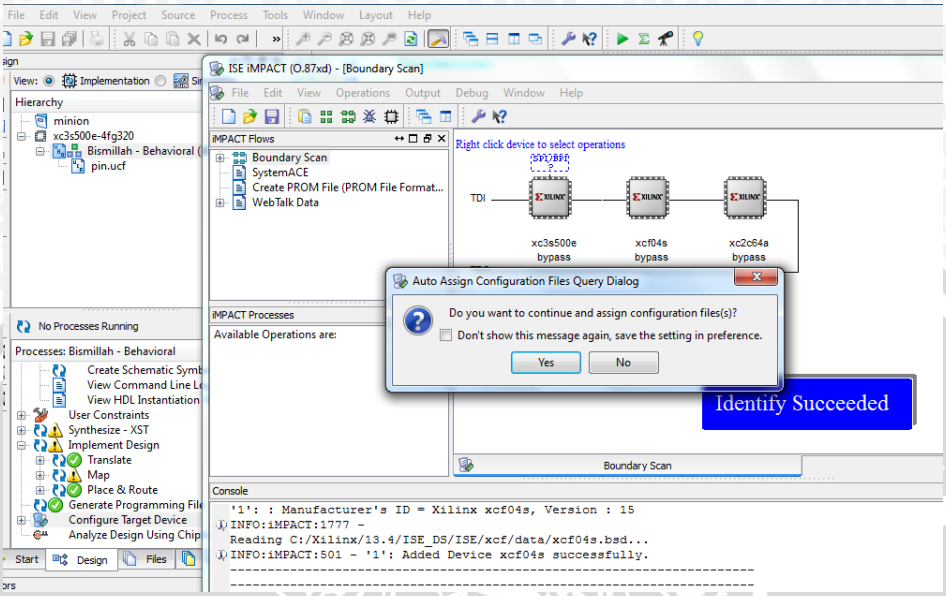

**Gambar4.4**Configure Target Service
**Sumber**: Implementasi

Langkah selanjutnya pilih boundary scan pada ISE Impact setelah itu klik kanan pada ISE Impact dan pilih initialize chain hal ini untuk identifikasi IC yang akan dipilih seperti yang ditunjukan pada gambar 4.5.



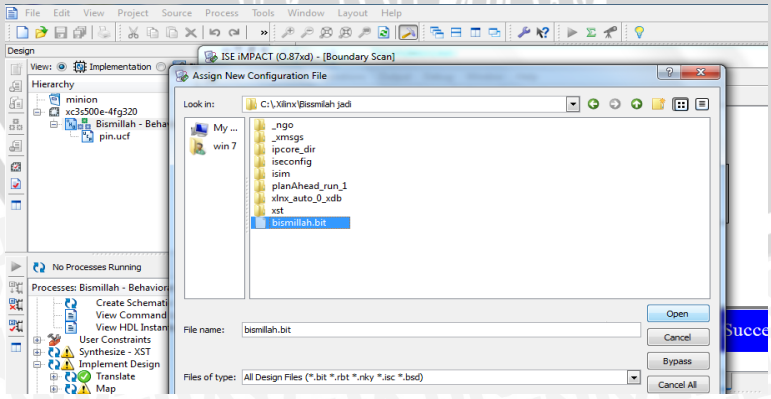

**Gambar4.5**Initialize Chain
**Sumber**: Implementasi

Setelah pilih initialize chain akan keluar pilihan IC yang akan digunakan dalam FPGA Spartan 3e dan pilihan yes untuk langkah selanjutnya seperti yang ditunjukan pada gambar 4.6



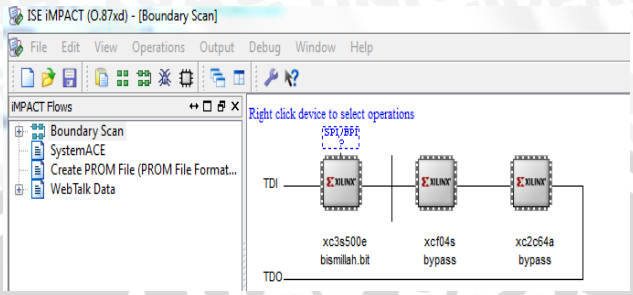

**Gambar4.6**Identifikasi IC
**Sumber**: Implementasi

Setelah pilih yes akan dengan sendirinya mengarah pada IC xc3s500s yang akan kita gunakan untuk mengimport program. Selanjutnya pilih dimana file program disimpan yang dalam penelitian ini program bernama bissmilah.bit dan klik open. Proses pemilihan program ditunjukan dalam gambar 4.7



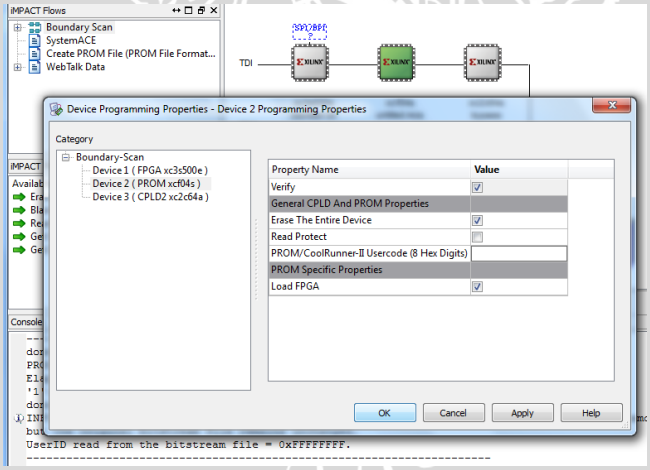

**Gambar4.7**Import Program
**Sumber**: Implementasi

Setelah program berhasil diimportkan pada IC xc3s500s akan berganti nama dengan nama program yang kita importkan yang dalam penelitian ini bernama bissmilah.bit seperti yang ditunjukan dalam gambar 4.8.




**Gambar4.8**Import Program Berhasil
**Sumber**: Implementasi

Langkah selanjutnya menyimpn program dalam memory FPGA yang bernama PROM, pilih load FPGA dan klik ok seperti yang ditunjukan dalam gambar 4.9.

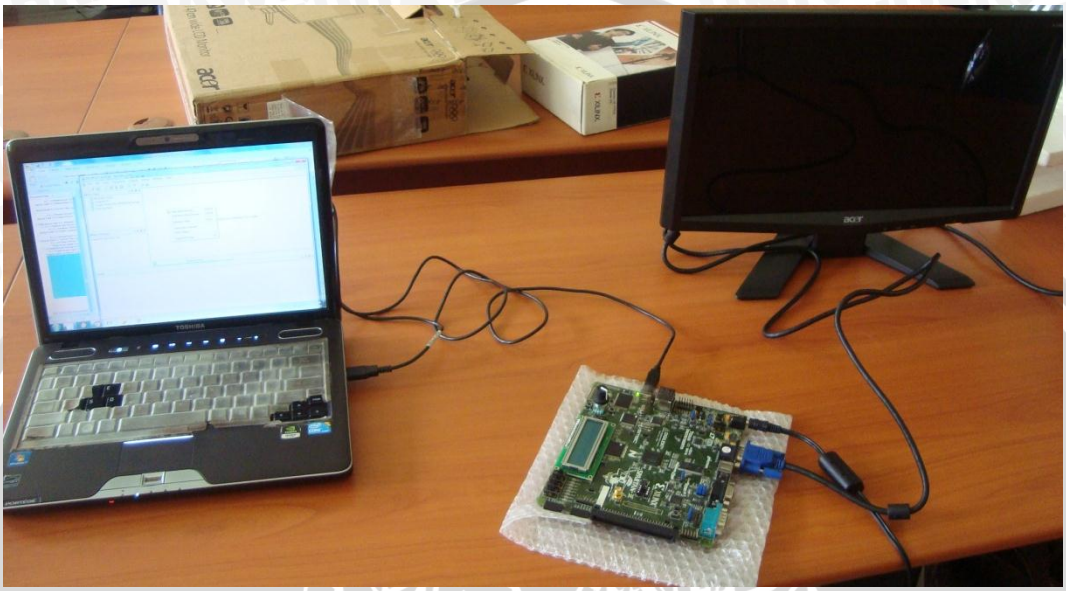



**Gambar4.1**Simpan Gmabar dalam Memory FPGA
**Sumber**: Implementasi

Pada langkah ini merupakan langkah terakhir dari proses import program dalam FPGA dan dalam proses ini program sudah berhasil diimportkan di dalam FPGA.

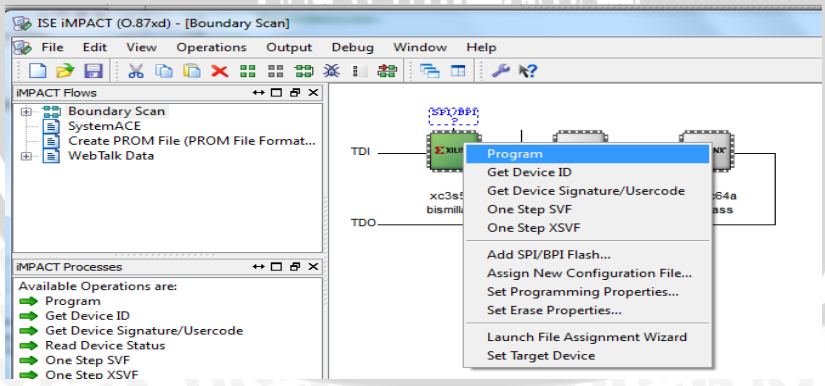
### 15.3.2 Menampilkan Program dalam LCD Monitor

Langkah pertama adalah menghubungkan antara laptop dengan kabel USB dan menghubungkan LCD monitor pada FPGA Spartan 3e dengan kabel HDMI seperti yang ditunjukan dalam gambar 4.10



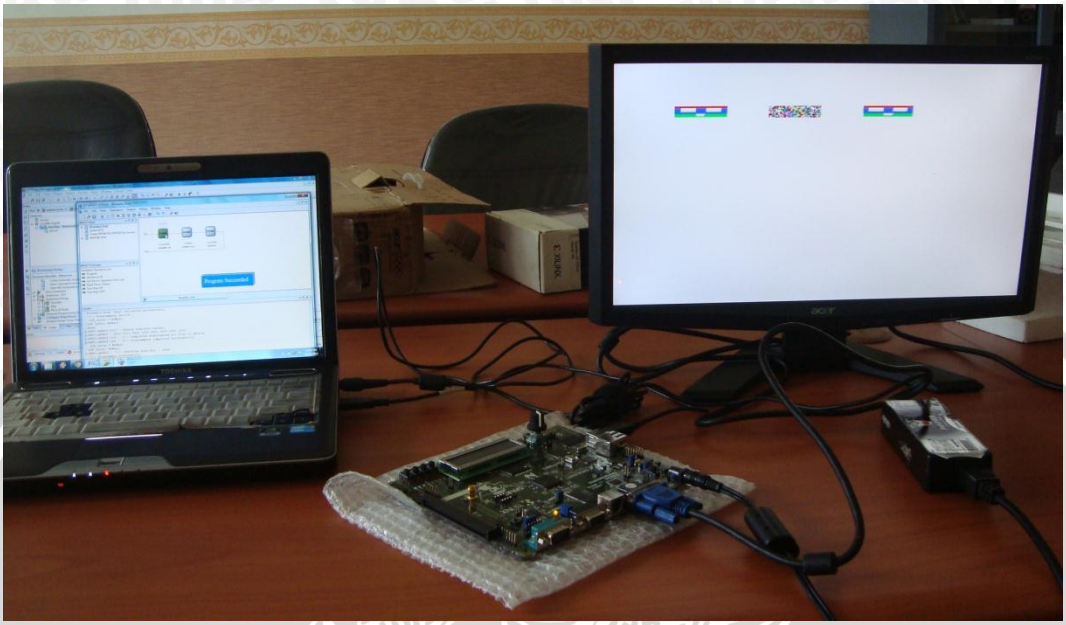
**Gambar4.10**Menghubungkan Laptop FPGA dan LCD Monitor

Setelah laptop FPGA dan LCD Monitor berhasil dihubungkan langkah selanjutnya adalah, klik kanan pada IC xc3s500s yang digunakan dan pilih program seperti yang ditunjukan dalam gambar 4.11




**Gambar4.11**Tampilkan Program
**Sumber**: Implementasi

Setelah dilakukan proses pada gambar 4.11 dengan sendirinya hasil program yang kita rancang akan ditampilkan dalam FPGA Spartan 3e seprti yang ditunjukan pada gambar 4.12



**Gambar4.12**Program Sukses Ditampilkan
**Sumber**: Implementasi

## 15.4 Kendala dan Solusi

Dalam penelitian ini ada beberapa kendala yang dihadapi. Salah satu kendala yang cukup sulit adalah proses menampilkan gambar berformat contoh : JPG, PNG, BMP,dll, kedalam LCD Monitor dan menyimpanya kedalam memory IP CORE. Proses tersebut cukup sulit dilakukan karena gambar dengan format yang sudah dicontohkan harus dikonversikan terlebih dahulu dalam format COE, karena pada IP FPGA hanya bisa membaca gambar dengan format COE. Kendala selanjutnya adalah menyimpan gambar dalam 3 memory IP CORE, hal tersebut cukup suit dilakukan.

Untuk mensiasati kendala yang dihadapi dalam menampilkan gambar dalam LCD Monitor.Dalam penelitian ini gambar yang digunakan adalah gambar buatan sendiri.Gambar buatan sendiri tersebut adalah gambar yang dibuat dengan memberi

warna pada setiap pixel LCD Monitor.Gambar buatan sendiri tersebut cukup memberikan solusi dalam penelitian ini.

# BAB V
# PENGUJIAN DAN ANALISIS

## 16.1 Strategi Pengujian

Strategi pengujian terhadap enkripsi gambar dengan algoritma *one time pad* dan menggunakan metode *logistic map* pada FPGA. Berdasarkan gambar 3.5 dan gambar 3.6 pada BAB III,Pengujian validitas dilakukan dengan dua kasus uji. Kasus uji pertama yaitu gambar didekripsi kunci yang sama, sedangkan kasus uji kedua adalah gambar didekripsi dengan kunci yang berbeda. Pada kasus uji pertama bertujuan untuk mengetahui hasil keakuratan gambar setelah didekripsi menggunakan kuci yang sama. Kasus uji kedua bertujuan untuk mengetahui hasil gambar dekripsi apabila didekripsi menggunakan kunci yang salah.Hasil pengujian validitas ini akan dimasukan pada tabel3.1. Hasil yang diharapkan dari pengujian validitas ini adalah gambar kembali terdekripsi dengan kunci yang sama dan gambar tidak kembali seperti gambar asli apabila didekripsi dengan kunci yang berbeda.

Pengujian kedua adalah pengujian sensitivitas pada kuncilogistic map.Pengujian sensitivitas dilakukan dengan melakukanperubahan kecil pada kunci dekripsi *logistic map*.Pengujian ini dilakukan untuk mengetahui hasil gambar dekripsiketika dilakukan perubahan kecil pada kunci dekripsi. Daftar perubahan kunci yang digunakan untuk pengujian dapat dilihat pada table 5.1.

| NO | Perubahan Kecil Pada Kunci Dekrip |
|---|---|
| 1 | X = 0,997 |
| 2 | X = 0,995 |
| 3 | X = 0,986 |

**Tabel5.1**Daftar Perubahan Kunci Pengujian

Mengacu pada table 5.1 kunci pada nomer 1 sampai 3 adalah kunci yang digunakan sebagai pengujian sensitivitas. Dengan kunci untuk gambar terenkripsi adalah X= 0,996. Pengujian sensitivitas dilakukan dengan menggunakan kunci dekripsi nomer 1 sampai 3.

## 16.2 Hasil Pengujian dan Analisa

Berdasarkan rancangan pengujian dan analisis pada bab 3, ada dua pengujian yang dilakukan yaitu pengujian validitas sesuai dengan atura *one time pad* dan pengujian sensitivitas nilai kunci *Logistic map.*

*Logistic map* merupakan *pseudo-random number generator* (*PRNG*) atau pembangkit bilangan semi acak maka kunci yang digunakan pada logistic map adalah nilai awalnya $X_0$dengan nilai antara 0 sampai 1. Pada logistic map terdapat dua inputan nilai yaitu nilai R dan $X_0$. Berdasarkan percobaan yang telah dilakukan dengan mencoba beberapa kemungkinan nilai R pada Logistic map, nilai R akan menghasilkan gambar yang benar-benar acak apabila nilai R bernilai antara 3,700 sampai 4. Table 5.2 menunjukan hasil percobaan dengan mencoba beberapa nilai pada R antara 0 sampai 4.

| NO | Hasil Gambar | Keterangan |
|---|---|---|

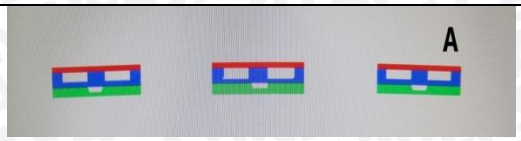
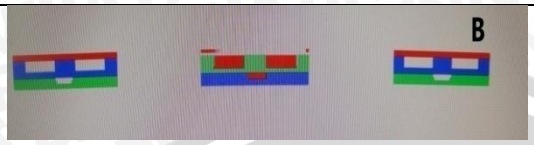
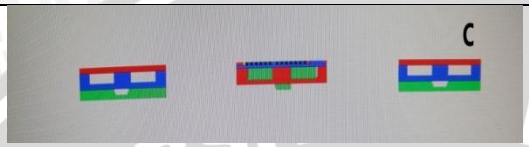
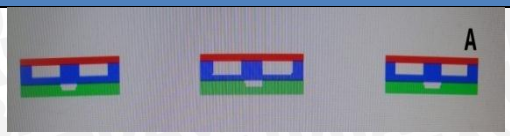
| | | |
|---|---|---|
| 1 |   | Hasil percobaan gambar dengan nilai R = 0 |
| 2 |   | Hasil percobaan gambar dengan nilai R = 2 |
| 3 |   | Hasil percobaan gambar dengan nilai R = 3 |
| 4 |   | Hasil percobaan gambar dengan nilai R = 3,560 |
| 5 |   | Hasil percobaan gambar dengan nilai R = 3,600 |
| 6 |   | Hasil percobaan gambar dengan nilai R = 3,700 |
| 7 |   | Hasil percobaan gambar dengan nilai R = 4,0 |

**Table5.2**Hasil Percobaan Perubaahan Nilai R

Gambar 5.2 menunjukan hasil percobaan yang dilakukan dengan mencoba beberapa nilai R pada metode *logistic map*.Nilai R pada metode *logistic map*menunjukan laju pertumbuhan dengan range nilai antara 0 sampai 4 dan dengan nilai awal kunci $X_0$ =0,996.

Percobaan gambar 5.1 pada gambar bagian A dilakukan iterasi dengan nilai R 0 menunjukan hasil gambar enkripsi tetap dan tidak berubah hal ini karena hasil seluruh iterasi menghasilkan nilai O semua, sehingga jika diconvert kedalam biner

akan menghasilkan biner 000 dan ketika diXORkan dengan warna aslinya akan menghasilkan warna gambar yang tetap. Percobaan pada gambar bagian B dilakukan dengan nilai R = 2 pada gambar bagian B ada sedikit perubahan pada gambar enkripsi hal ini karena dengan nilai R = 2 pada setiap iterasi menghasilkan selisih nilai yang sedikit hampir sama dan ketika di convert menadi biner akan menghasilkan nilai yang sama. Percobaan pada gambar bagian C dilakukan dengan nilai R = 3, dengan R sebesar 3 hasil gambar enkripsi yang dihasilkan ada sedikit perubahan seperti yang ditunjukan pada gambar bagian C. kasus hal ini hampir sama dengan apa yang terjadi pada saat R bernilai 2.Percobaan pada gambar D dilakukan dengan nilai R = 3,560, dari hasil yang ditunjukan pada gambar bagian D menunjukan hasil gambar enkripsi mulai terlihat acak keseluruhan namun masih terlihat sedikit bermakna seperti gambar aslinya.Percobaan pada gambar E dilakukan dengan nilai R = 3,600 dari hasil yang ditunjukan pada gambar bagian E menunjukan hasil gambar enkripsi mulai terlihat acak keseluruhan namun masih terlihat sedikit bermakna seperti gambar aslinya, kasus ini hampir sama dengan gambar bagian D.Percobaan pada gambar F dilakukan dengan nilai R = 3,700 dari hasil yang ditunjukan pada gambar bagian F menunjukan gambar enkripsi sudah teracak di tiap pixel dan sudah tidak bermakna lagi seperti yang ditunjukan pada gambar bagian E dan D yang masih sedikit bermakna.Percobaan pada gambar G dilakukan dengan nilai R = 4,0 dari hasil yang ditunjukan pada gambar bagian F menunjukan gambar enkripsi sudah teracak di tiap pixel dan sudah tidak bermakna lagi.Pada nilai awal kunci $X_0$juga dilakukan beberapa kali percobaan dengan mencoba beberapa kemungkinan kunci. Dari hasil percobaannilai awal kunci akan menghasilkan barisan nilai acak apabila nilai awal kunci bernilai antara 0.001 sampai 1 dengan range antara 0 sampai 1. Hasil percobaan ditunjukan pada table 5.3.

| NO | Hasil Gambar | Keterangan |
|---|---|---|
| 1 |  | Hasil percobaan gambar dengan nilai $X_0 = 0$ |

| | | |
|---|---|---|
| 2 |   | Hasil percobaan gambar dengan nilai $X_0$ = 0,001 |
| 3 |   | Hasil percobaan gambar dengan nilai $X_0$ = 0,99 |
| 4 |   | Hasil percobaan gambar dengan nilai $X_0$ = 0,503 |
| 5 |   | Hasil percobaan gambar dengan nilai $X_0$ = 0,881 |

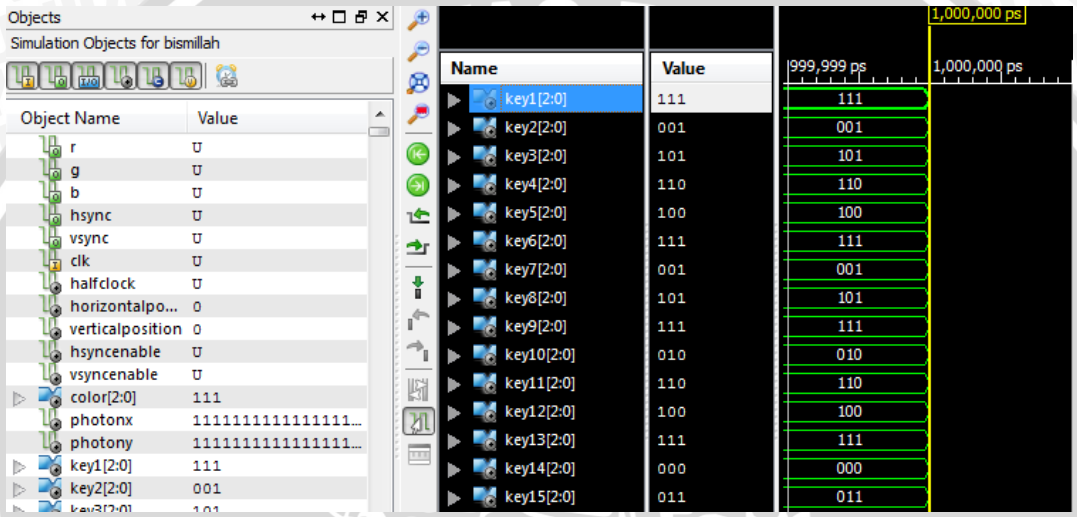
**Table5.3**Hasil Percobaan Perubahan Nilai Awal Kunci $X_0$

Gambar 5.2 menunjukan hasil percobaan yang dilakukan dengan mencoba beberapa nilai X pada metode *logistic map*.Nilai X pada metode *logistic map*menunjukan nilai acak yang dalam kriptografi digunakan sebagai kunci, dengan range nilai antara 0 sampai 1. Pada gambar 5.2 Ryang digunakan adalah 3,999.

Pada gambar bagian A nilai awal kunci $X_0$ yang digunakan adalah 0.0, menunjukan hasil gambar enkripsi tetap dan tidak berubah hal ini karena hasil seluruh iterasi menghasilkan nilai O semua, sehingga jika diconvert kedalam biner akan menghasilkan biner 000 dan ketika diXORkan dengan warna aslinya akan menghasilkan warna gambar yang tetap.Pada gambar bagian B nilai awal kunci yang digunakan adalah 0,001, menunjukan gambar enkripsi acak dan tidak bermakna. Pada gambar bagian C nilai awal kunci yang digunakan adalah 0,99 menunjukan gambar enkripsi acak dan tidak bermakna. Pada gambar bagian D nilai awal kunci yang digunakan adalah 0,503 menunjukan gambar enkripsi acak dan tidak bermakna.Pada gambar bagian E nilai awal kunci yang digunakan adalah 0,881 menunjukan gambar enkripsi acak dan tidak bermakna. Dari hasil percobaan

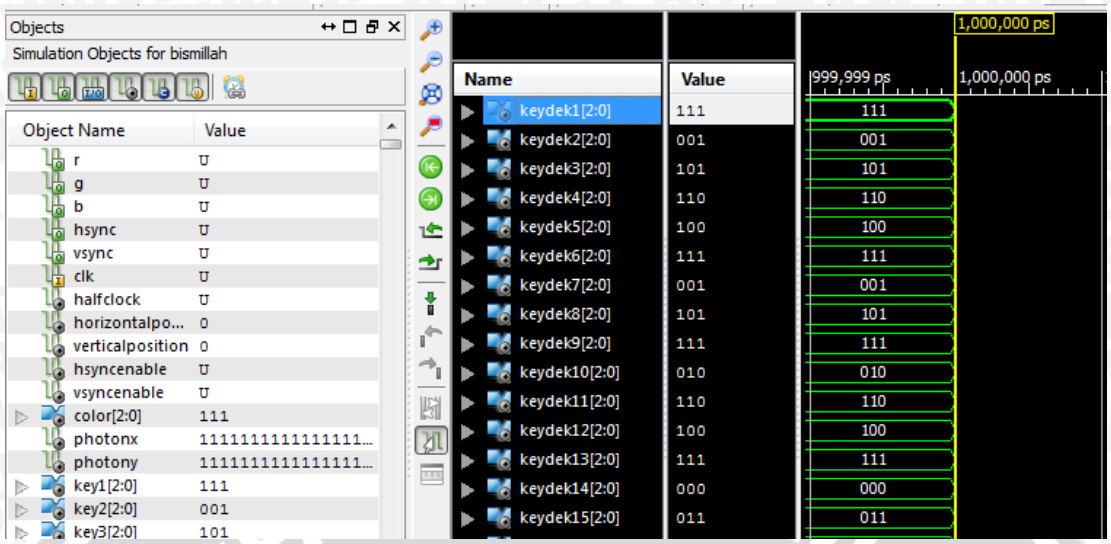
dengan mencoba beberapa kemungkinan pada nilai R dan nilai awal kunci $X_0$, disarankan menggunakan nilai R antara 3,700 sampai 4 dan nilai awal kunci antara 0,001 sampai 1.

Analisi waktu pada proses enkripsi dan dekripsi juga dilakukan percobaan, percobaan ini dilakukan sebagai tambahan informasi tentang berapa lama waktu pada proses generate nilai kunci logistic map sampai enkripsi dan dekripsi. Analisa waktu pada proses penbentukan kunci dan enkripsi ditunjukan pada gambar 5.1.




**Gambar5.1 Analisa Waktu Enkripsi**

Pada gambar 5.1 ditunjukan proses generate kunci logistic map dan proses enkripsi memerlukan waktu 1,000,000 picosecond (ps). Sedangkan pada proses dekripsi juga dilakukan analisa waktu, gambar 5.2 menunjukan analisa waktu pada proses pembentukan kunci logistic map dan proses dekripsi.




**Gambar5.2**Analisa Waktu Dekripsi

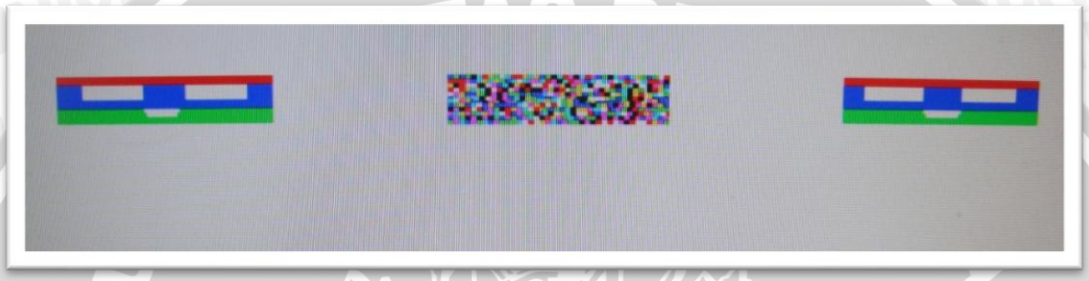
Pada gambar 5.1 ditunjukan proses generate kunci logistic map dan proses dekripsi memerlukan waktu 1,000,000 picosecond (ps). Hasil analisa waktu pembentukan kunci logistic map sampai proses enkripsi dan dekripsi sama-sama memerlukan waktu eksekusi 1,000,000 ps, hal ini karena diimplementasikan pada hardware yang prosesnya akan dijalankan atau dieksekusi secara pararel atau bersamaan.

### 16.2.1 Hasil Pengujian Validitas

Pengujian validitas dilakukan dengan menggunakan dua kasus uji.Kasus uji yang pertama adalah dekripsi dengan kunci yang sama sedangkan kasus uji yang kedua adalah dekripsi dengan kunci berbeda.Pengujian validitas dilakukan sesuai dengan cara kerja algoritma *One Time Pad*. Pada kasus uji pertama bertujuan untuk mengetahui keakuratan gambar setelah di dekripsi menggunakan kunci yang sama. Prosedur uji dilakukan dengan cara menggunakan kunci logistic map yang sama pada saat proses enkripsi dan dekripsi. Hasil yang diharapkan dari pengujian pada kasus uji pertama ini adalah gambar dekripsi dapat berubah kembali seperti gambar asli. Pada kasus uji kedua bertujuan untuk mengetahui hasil gambar dekripsi apabila di dekripsi menggunakan kunci yang berbeda atau kunci yang salah.Prosedur uji dilakukan dengan cara menggunakan kunci logistic map yang berbeda pada saat proses enkripsi dan dekripsi. Hasil yang diharapkan dari

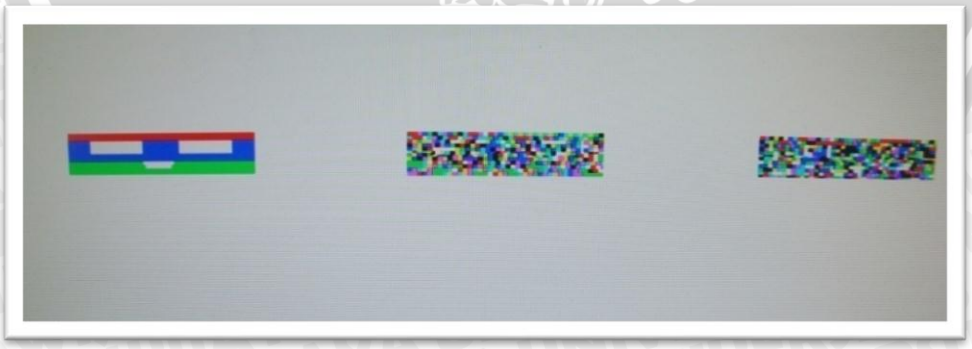
pengujian pada kasus uji kedua ini adalah gambar dekripsi tetap acak dan tidak kembali seperti gambar asli.

Kasus uji pertama menggunakan kunci yang sama pada saat proses enkripsi dan dekripsi. Pada kasus uji pertama nilai awal kunci $X_0$yang digunakan pada proses enkripsi dan dekripsi adalah sama yaitu0.755 dan nilai R = 3.999, dengan menggunakan kunci yang sama gambar dekripsi tepat akurat kembali seperti gambar aslinya.Hasil pengujian ditunjukan pada gambar 5.1



**Gambar5.1**Hasil Pengujian Kasus Uji Pertama

Pada kasus uji keduamenggunakan kunci berbeda pada saat proses enkripsi dan dekripsi. Kasus uji kedua pada saat proses enkripsi menggunakan nilai awal kunci $X_0$ = 0,996 dan nilai R = 4.0, sedangkan pada saat proses dekripsi menggunakan nilai awal kunci $X_0$ = 0,986. Dekripsi menggunakan kunci yang berbeda menghasilkan gambar dekripsi tetap acak dan tidak kembali seperti gambar asli.Hasil pengujian ditunjukan pada gambar 5.2.



**Gambar5.2**Hasil Pengujian Kasus Uji Kedua

Dari pengujian yang dilakukan pada kasus uji pertama dan kasus uji kedua pada pengujan validitas hasil pengujian validitas dimasukan pada tabel validitas seperti yang ditunjukan pada tabel 5.4

| NO | Kasus Uji | Hasil yang diharapkan | Hasil yang didapatkan | Status Validitas |
|---|---|---|---|---|
| 1 | Dekripsi dengan kunci sama | Gambar dekripsi dapat berubah kembali seperti gambar asli | Gambar dekripsi dapat berubah kembali seperti gambar asli | Valid |
| 2 | Dekripsi dengan kunci berbeda | Gambar dekripsi tetap acak dan tidak kembali seperti gambar asli | Gambar dekripsi tetap acak dan tidak kembali seperti gambar asli | Valid |

**Tabel5.4**Hasil Pengujian Validitas

Berdasarkan hasil pengujian validitas seperti yang ditunjukan pada tabel 5.2, pada kasus uji pertama ketika gambar didekripsi menggunakan kunci yang sama hasil yang diharapkan sesuai dengan hasil yang didapatkan maka statusnya dikatakan valid. Pada kasus uji kedua ketika gambar didekripsi menggunakan kunci yang berbeda hasil yang diharapkan sesuai dengan hasil yang didapatkan maka statusnya dikatakan valid.

### 16.2.2 Hasil Pengujian Sensitivitas Kunci

Pengujian sensitivitas pada nilai awal kunci dilakukan berdasarkan karakteristik dari sistem chaos yaitu sensitive terhadap perubahan kecil pada nilai awal kuncinya.Menurut Ravindra K.Purwar dan Priyanka algoritma enkripsi citra yang baik adalah bergantung dengan kunci yang digunakan, semakin sensitive kunci yang digunakan dan bahkan jika seseorang atau kiptanalis mencoba mendekripsi gambar enkripsi menggunakan kunci yang hampir sama tetap tidak mendeskripsi gambar seperti gambar asli. Pada penelitian ini pengujian ini bertujuan untuk menguji sensitivitas kunci logistic map yang diterapkan pada

gambar 3bit, dan mengetahui hasil gambar dekripsi apabila dilakukan perubahan sedikit pada nilai awal kuncinya.

Prosedur pengujian dilakukan dengan melakukan perubahan sedikit pada kunci dekripsi$X_0$.Pengujian sensitivitas ini dilakukan tiga kali pengujian dengan perubahan kunci dilakukan berdasarkan table 5.1. Kunci dekripsi pertama yang digunakan adalah $X_0$ = 0,997, kunci kedua adalah $X_0$ = 0,995 dan kunci ketiga adalah $X_0$ =0,986 dengan kunci enkripsi yang sebenarnya adalah $X_0$= 0,996 dan dengan nilai R = 4.0.Hasil pengujian sensitivitas ditunjukan pada table 5.5

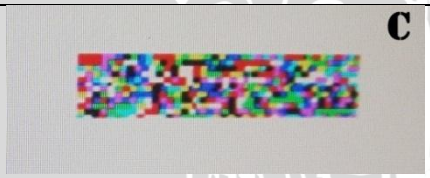
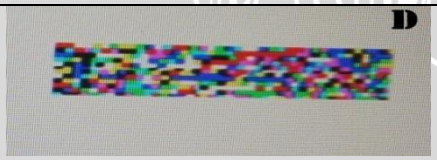
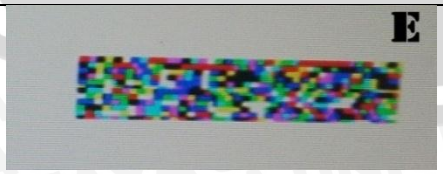
| NO | Gambar | Keterangan |
|---|---|---|
| 1 |   | Gambar Asli |
| 2 |   | Gambar enkripsi dengan kunci $X_0$= 0,996 |
| 3 |   | Gambar Dekripsi dengan kunci $X_0$= 0,997 |
| 4 |   | Gambar enkripsi dengan kunci $X_0$= 0,995 |
| 5 |   | Gambar enkripsi dengan kunci $X_0$= 0,986 |

**Table5.5**Hasil Pengujian Sensitivitas

Berdasarkan gambar 5.3 gambar A adalah gambar asli atau plaintext, gambar B adalah gambar enkripsi dengan nilai awal kunci $X_0$= 0,996.

Gambar bagian C adalah gambar dekripsi yang menggunakan nilai awal kunci $X_0$ = 0,997, dengan perubahan kecil pada nilai awal kunci gambar yang dihasilkan tetap acak dan gambar tidak kembali seperti gambar aslinya. Sensitivitas kunci pada gambar C didapatkan presentase 7,75 %gambar yang sama, penghitungan presentase nilai pixel yang sama tersebut dujelaskan pada lampiran. Gambar bagian D adalah gambar dekripsidengan menggunakan nilai awal kunci $X_0$ = 0,995, dengan perubahan kecil pada nilai awal kuncinya, gambar dekripsi yang dihasilkan tetap acak dan tidak kembali seperti gambar aslinya seperti yang ditunjukan pada gambar bagian D. Sensitivitas kunci pada gambar D didapatkan presentase 7,75 % gambar yang sama, penghitungan presentase nilai pixel yang sama tersebut dijelaskan pada lampiran.Gambar bagian E adalah gambar dekripsi dengan menggunakan nilai awal kunci $X_0$ =0,986.padapengujian yang sudah dilakukan gambar dekripsi yang dihasilkan tetap acak dan tidak kembali seperti gambar aslinya seperti yang ditunjukan pada gambar bagian E.Sensitivitas kunci pada gambar E didapatkan presentase 5,50 % gambar yang sama, penghitungan presentase nilai pixel yang sama tersebut dijelaskan pada lampiran.

Dari hasil pengujian sensitivitas dengan mencoba beberapa nilai awal kunci dengan selisih kecil pada kunci dekripsi, menghasilkan gambar tetap acak dan tidak kembali seperti gambar aslinya.Sensitivitas kunci ini berarti nilai awal kunci sensitive terhadap perubahan kecil pada saat proses dekripsi, seperti yang sudah dilakukan pada pengujian. Berdasarkan hasil pengujian yang telah dilakukan menunjukan bahwa, sensitivitas nilai awal kunci*logistic map* terpenuhi pada penelitian ini yang menggunakan gambar dengan kedalaman warna 3 bit, namun dari hasil pengujian sensitivitas tersebut belum bisa diukur kuantitatifnya karena keterbatasan dari FPGA Spartan 3e. Sensitivitas nilai awal kunci pada *Logistic map*terpenuhi dengan syarat nilai inputan pada R dan $X_0$sesuai ketentuan acak berdasarkan percobaan yang telah dilakukan.

# BAB VI
# PENUTUP

## 17.1 Kesimpulan

Berdasarkan hasil perancangan, implementasi dan pengujian yang dilakukan, maka diambil kesimpulan sebagai berikut :

1. Implementasi enkripsi gambar dengan algoritma *One Time Pad* menggunakan metode *logistic map*pada FPGA dimulai dengan desain program dan*compile*. Cek hasil proses iterasi dari kunci *logistic map* pada simulasi semua proses ini dijalankan pada lunak pendukung bawaan dari Xilinx yang dalam hal ini adalah ISE Project Navigator, jika proses iterasi pada kunci masih belum berhasil ulangi pada desain program. Apabila kunci sudah berhasil dibuat dalam setiap iterasinya, *import* program dalam FPGA yang kemudian hasilnya akan ditampilkan dalam LCD Monitor. Proses ini mungkin tidak berjalan satu kali, Jika dirasa program belum sesuai kebutuhan, program akan didesain kembali dalam ISE Project Navigator.
2. Berdasarkan hasil pengujian yang dilakukan pada kasus uji pertama dan kasus uji kedua pada pengujian validitas, menghasilkan hasil yang sama-sama sesuai dengan harapan. Maka status validitasnya dikatakan valid 100%.
3. Berdasarkan hasil pengujian sensitivitas nilai awal pada kunci pengujian dilakukan dengan nilai $X_0$ = 0,997, $X_0$ = 0,995, $X_0$ =0,986 menghasilkan gambar tetap acak dan tidak kembali seperti gambar aslinya.
4. Sensitivitas kunci pada gambar C dengan nilai awal kunci $X_0$ = 0,997 didapatkan presentase 7,75 % pixel gambar sama yang muncul. Sensitivitas kunci pada gambar D dengan nilai awal kunci $X_0$ = 0,995 didapatkan presentase 7,75 % pixel gambar sama yang muncul.Sensitivitas kunci pada gambar E dengan nilai awal kunci $X_0$ = 0,986 didapatkan presentase 5,50 % pixel gambar sama yang muncul.

5. Hasil pengujian juga menunjukan bahwa sensitivitas nilai awal kunci *Logistic Map* juga terpenuhi pada penelitian ini yang menggunakan gambar dengan kedalaman warna 3 bit.
6. Sensitivitas nilai awal kunci pada *Logistic map* terpenuhi dengan syarat nilai inputan pada R lebih besar atau sama dengan 3,700 dan $X_0$lebih besar atau sama dengan 0.001.Sesuai dengan ketentuan acak berdasarkan percobaan yang telah dilakukan.

## 17.2 Saran

Saran yang dapat diberikan untuk pengembangan penelitian ini antara lain :

1. Untuk pengembangan lebih lanjut, diharapkan diciptakan juga converter gambar dari format JPG, BMP, PNG menjadi format coe, karena pada IC FPGA hanya bisa menerima gambar dengan format coe.
2. Untuk pengembangan lebih lanjut, gambar yang digunakan harus gambar inputan yang menggunakan memory *IP CORE* pada FPGA, sehingga gambar yang berupa file dapat disimpan dalam memory dan dapat juga ditampilkan dalam LCD Monitor.
3. Untuk pengembangan lebih lanjut, diharapkan sudah bisa diimplementasikan didalam IC yang sebenarnya agar enkripsi berbasis *embedded system* segera tercipta.
4. Untuk pengembangan lebih lanjut, bisa dilakukan pengujian dengan serangan seperti *exhaustive attack* atau *brute force*.