

**IMPLEMENTASI WATERMAKING PADA CITRA DIGITAL
MENGUNAKAN METODE LINEAR CONGRUENTIAL
GENERATOR (LCG) DI BADAN METEOROLOGI
KLIMATOLOGI DAN GEOFISIKA
PROVINSI BENGKULU**

SKRIPSI



OLEH :

ALEK WIJAYA
NPM : 21010159P

**PROGRAM STUDI INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS DEHA SEN
BENGKULU
2023**

**IMPLEMENTASI WATERMAKING PADA CITRA DIGITAL
MENGUNAKAN METODE LINEAR CONGRUENTIAL
GENERATOR (LCG) DI BADAN METEOROLOGI
KLIMATOLOGI DAN GEOFISIKA
PROVINSI BENGKULU**

SKRIPSI

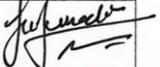
OLEH :

ALEK WIJAYA
NPM : 21010159P

Telah dipertahankan di depan Tim Penguji Universitas Dehasen Bengkulu Pada :

Hari : Sabtu
Tanggal : 17 Juni 2023
Tempat : Ruang Sidang Universitas Dehasen Bengkulu

Skripsi Telah Diperiksa dan Disahkan Oleh :

Penguji	Nama	NIDN	Tanda Tangan
Ketua	Juju Jumadi, S.Kom, M.Kom	02.111282.01	
Anggota	Abdussalam Al Akbar, S.Kom, M.Kom	02.051092.01	
Anggota	Sapri, S.Kom, M.Kom	02.150171.02	
Anggota	Eko Prasetyo Rohmawan, S.Kom, M.Kom	02.130488.01	

Mengetahui,

**Dekan
Fakultas Ilmu Komputer**


Siswanto, SE, S.Kom, M.Kom
NIDN : 02.240363.01



ABSTRAK

IMPLEMENTASI WATERMAKING PADA CITRA DIGITAL MENGUNAKAN METODE LINEAR CONGRUENTIAL GENERATOR (LCG) DI BADAN METEOROLOGI KLIMATOLOGI DAN GEOFISIKA PROVINSI BENGKULU

Oleh :

Alek Wijaya¹

Juju Jumadi²

Abdussalam Al Akbar²

Watermarking merupakan teknik untuk menanamkan suatu pesan rahasia atau tanda pada citra digital, sehingga dapat digunakan untuk melacak asal usul citra digital tersebut. *Watermarking* pada citra digital dapat diterapkan pada BMKG untuk melindungi hak cipta dan keaslian data citra yang dihasilkan. Salah satu algoritma yang dapat digunakan untuk melakukan watermarking pada citra digital adalah algoritma *Linear Congruential Generator* (LCG). Algoritma ini memiliki kelebihan dalam menghasilkan deret bilangan acak dengan cepat dan mudah diimplementasikan. Selain itu, algoritma LCG juga dapat diatur untuk menghasilkan kunci acak

Dari hasil pengujian yang dilakukan dapat disimpulkan bahwa watermarking pada citra digital menggunakan algoritma LCG terbukti dapat memberikan watermarking yang baik, hal ini dikarenakan hasil citra watermarking dengan citra asli tidak memiliki perbedaan yang signifikan atau kasat mata.

Kata kunci : Watermaking, *Linear Congruential Generator*, Citra Digital

1. Mahasiswa
2. Pembimbing

ABSTRACT

THE IMPLEMENTATION OF WATERMARKING IN DIGITAL IMAGES USING THE LINEAR CONGRUENTIAL METHOD GENERATOR (LCG) AT METEOROLOGICAL, CLIMATOLOGICAL, AND GEOPHYSICAL AGENCY IN BENGKULU PROVINCE

By:

Alek Wijaya¹

Juju Jumadi²

Abdussalam Al Akbar²

Watermarking is a technique for embedding a secret message or mark on a digital image, therefore it can be used to trace the origin of the digital image. Watermarking on digital images can be applied to BMKG to protect copyrights and the authenticity of the resulting image data. One of the algorithms that can be used to perform watermarking on digital images is the Linear Congruential Generator (LCG) algorithm. This algorithm has the advantage of generating a series of random numbers quickly and is easy to implement. In addition, the LCG algorithm can also be set to generate random keys. From the results of the tests carried out, it can be concluded that watermarking on digital images using the LCG algorithm is proven to provide good watermarking, this is because the results of the watermarking image with the original image do not have a significant or visible difference.

Keywords: Watermaking, Linear Congruential Generator, Digital Image

1. Student

2. Supervisors

**SURAT PERNYATAAN ORISINILITAS DAN PERSETUJUAN
AKADEMIK SKRIPSI**

Yang bertanda tangan dibawah ini :

Nama : Alek Wijaya

NPM : 21010159P

Fakultas /Program Studi : Ilmu Komputer / Informatika

Dengan ini menyatakan dengan sesungguhnya bahwa Skripsi dengan Judul :

**IMPLEMENTASI WATERMAKING PADA CITRA DIGITAL
MENGUNAKAN METODE LINEAR CONGRUENTIAL
GENERATOR (LCG) DI BADAN METEOROLOGI
KLIMATOLOGI DAN GEOFISIKA
PROVINSI BENGKULU**

1. Adalah benar dibuat oleh saya sendiri untuk memenuhi persyaratan kelulusan akademik.
2. Pada bagian-bagian tertentu dalam penulisan skripsi ini yang saya kutip dari hasil karya orang lain telah ditulis sumbernya secara jelas sesuai dengan norma, kaidah dan cara penulisan ilmiah.
3. Jika dikemudian hari diketahui berdasarkan bukti-bukti yang kuat ternyata skripsi tersebut dibuat oleh orang lain atau diketahui bahwa skripsi tersebut merupakan *plagiat/mencontek/menjiblak* hasil karya tulis ilmiah orang lain, maka dengan inisaya bersedia menerima sanksi pencabutan gelar akademik yang saya peroleh dan sanksi-sanksi lainnya sesuai dengan peraturan yang berlaku.
4. Dan atas orisinilitas tersebut diatas, maka saya menyetujui untuk memberi kepada Universitas Dehasen Bengkulu hak atas bebas royalti non-eksklusif untuk menyimpan, mengalih mediakan, mendistribusikan dan mempublikasikan skripsi saya tanpa perlu meminta izin selama mencantumkan nama saya sebagai penulis/pencipta.
5. Saya bersedia menanggung secara pribadi tanpa melibatkan Universitas Dehasen Bengkulu segala bentuk tuntutan hukum yang timbul atas pelanggaran Hak Cipta dalam Karya Ilmiah saya ini

Demikian surat pernyataan ini dibuat dengan sebenarnya dan untuk dipergunakan sebagaimana mestinya.

Bengkulu, Juni 2023
Hormat Saya

Alek Wijaya

KATA PENGANTAR

Puji Syukur saya panjatkan kehadirat Allah SWT yang telah memberikan rahmat dan karunia-NYA, sehingga skripsi yang berjudul “**Implementasi Watermaking Pada Citra Digital Menggunakan Metode Linear Congruential Generator (LCG) Di Badan Meteorologi Klimatologi dan Geofisika Provinsi Bengkulu**” dapat diselesaikan dalam waktu yang telah ditetapkan.

Pada kesempatan ini penulis ingin menyampaikan ucapan terima kasih kepada pihak-pihak yang telah banyak membantu penulis selama penyusunan skripsi ini kepada :

1. Bapak Prof. Dr. Husaini, SE., M.Si, Ak, CA, CRP selaku Rektor Universitas Dehasen Bengkulu
2. Bapak Siswanto, SE, S.Kom, M.Kom selaku Dekan Fakultas Ilmu Komputer Universitas Dehasen Bengkulu.
3. Ibu Liza Yulianti, M.Kom selaku Ketua Prodi Informatika Universitas Dehasen Bengkulu.
4. Bapak Juju Jumadi, S.Kom, M.Kom selaku pembimbing utama yang telah membimbing dengan sabar dan memberikan masukan serta saran kepada penulis
5. Bapak Abdussalam Al Akbar, S.Kom, M.Kom Selaku pembimbing pendamping yang telah memberikan masukan dan saran kepada penulis.
6. Pimpinan dan pegawai BMKG Provinsi Bengkulu yang telah membantu dalam penelitian ini.

7. Buat teman-teman yang tidak bisa disebutkan satu persatu baik formal dan non formal, terima kasih atas bantuannya selama penyelesaian penulisan skripsi ini

Penulis menyadari masih banyak kekurangan dalam penulisan skripsi ini, namun penulis mengharapkan saran dan kritik yang sifatnya membangun guna menunjang perkembangan ilmu pengetahuan khususnya ilmu komputer.

Bengkulu, Juni 2023

Penulis

DAFTAR ISI

	Halaman
HALAMAN JUDUL	ii
HALAMAN PENGESAHAN	iii
HALAMAN PERSETUJUAN.....	iv
RIWAYAT HIDUP	vii
MOTO DAN PERSEMBAHAN	vi
ABSTRAK	vii
ABSTRACT	vii
PERNYATAAN ORIGINAL	ix
KATA PENGANTAR.....	x
DAFTAR ISI.....	xii
DAFTAR GAMBAR.....	xvi
DAFTAR TABEL.....	xviii
 BAB I PENDAHULUAN	
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	3
1.3 Batasan Masalah	3
1.4 Tujuan Penelitian	3
1.4.1 Tujuan Umum	3
1.4.2 Tujuan Khusus	4
1.5 Manfaat Penelitian	4
 BAB II LANDASAN TEORI	
2.1 Pengertian Watermarking.....	5
2.2 Linear Congruential Generator (LCG)	11
2.3 Citra Digital	12
2.3.1 Pengolahan Citra Digital	16

2.3.2	Tujuan Pengolahan Citra Digital	17
2.4	Tinjauan Umum Visual Studio 2010	17
2.4.1	Integrated Development Environment (IDE)	18
2.4.2	Toolbox Windows Form	19
2.4.3	Jendela Explorer	19
2.4.4	Jendela Properties	20
2.5	UML (Unified Modeling Language)	20
2.5.1	Use Case Diagram	21
2.5.2	Activity Diagram	24
2.5.3	Class Diagram	25
2.6	Flowchart	27

BAB III METODOLOGI PENELITIAN

3.1	Subjek Penelitian	29
3.1.1	Gambaran Umum BMKG	29
3.1.2	Struktur Organisasi	32
3.1.3	Tempat Dan Waktu Penelitian	32
3.2	Metode Penelitian	32
3.3	Software dan Hardware yang digunakan	33
3.4	Metode Pengumpulan Data	33
3.5	Analisa Perancangan Sistem	34
3.5.1	Analisa Sistem Aktual	34
3.5.2	Analisa Watermarking LCG	34
3.5.3	Perancangan Sistem Baru	37
3.6	Perancangan Pengujian	45

BAB IV HASIL DAN PEMBAHASAN

4.1	Hasil Aplikasi	46
4.2	Pembahasan	46
4.3	Pengujian Sistem	54

BAB IV KESIMPULAN DAN SARAN

5.1 Hasil Aplikasi	55
5.2 Saran	55

DAFTAR PUSTAKA

LAMPIRAN

DAFTAR GAMBAR

Gambar	Halaman
2.1 Ilustrasi Watermarking pada Citra Digital.....	11
2.2 Ilustrasi Citra Digital	14
2.3 Citra Biner	15
2.4 Citra Grayscale 4 Bit	15
2.5 Citra Warna.....	16
2.6 Komponen Visual Basic 2010	18
2.7 Toolbox.....	18
2.8 Jendela Explorer	19
2.9 Jendela Properties	19
3.1 Potongan Citra Sample	34
3.2 Use Case Diagram	37
3.3 Activity Diagram Penyisipan.....	38
3.4 Activity Diagram Ekstraksi	39
3.5 Flowchart Sistem	40
3.6 Rancangan Form Utama	41
3.7 Rancangan Form Penyisipan	42
3.8 Rancangan Form Ekstraksi	44
4.1 Form Menu Utama Aplikasi	47
4.2 Form Penyisipan	47
4.3 Pemilihan Citra Digital Untuk disisipi pesan	48
4.4 Input Citra Kedalam Sistem	49
4.5 Proses Embed Pesan Pada Citra	50
4.6 Hasil Penyisipan Watermarking	50
4.7 Penyimpanan Hasil Watermarking.....	51
4.8 Ekstraksi Data Dari Citra Watermarking.....	52
4.9 Proses Ekstraksi	53

DAFTAR TABEL

Tabel	Halaman
2.1 Hasil Pembangkitan Bilangan Acak dengan Metode LCG	13
2.2 Simbol Use Case Diagram.....	21
2.3 Notasi <i>Activity</i> Diagram.....	24
2.4 Simbol Class Diagram	25
2.5 Simbol dan Fungsi <i>Flowchart</i>	27
4.1 Pengujian Black Box	54

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi dan informasi telah membawa perubahan besar bagi media digital. Media digital yang dapat berupa teks, citra, audio dan video dapat dengan mudah didistribusikan melalui internet. Kemudahan distribusi media digital melalui internet disisi lain dapat menimbulkan permasalahan diantaranya, pencurian hak cipta, pemalsuan, hingga rekayasa pada citra digital menjadi kasus yang semakin sering terjadi. Untuk melindungi citra digital, dapat dilakukan dengan memberikan suatu otentifikasi atau identitas bahwa citra digital tersebut adalah milik seseorang atau lembaga.

Salah satu solusi untuk perlindungan hak cipta pada citra digital adalah dengan menggunakan teknologi *watermarking*. *Watermarking* adalah teknik untuk menanamkan suatu pesan rahasia atau tanda pada citra digital, sehingga dapat digunakan untuk melacak asal usul citra digital tersebut. *Watermarking* dapat dilakukan secara *visible* atau *invisible* pada citra digital, tergantung pada kebutuhan.

Dalam melakukan tugasnya, Badan Meteorologi Klimatologi dan Geofisika atau biasa disebut BMKG merupakan instansi dibawah Non Kementrian yang bergerak di bidang cuaca menghasilkan berbagai jenis citra seperti peta cuaca, peta gempa bumi, dan citra satelit. Citra-citra tersebut merupakan hasil dari pengolahan data yang telah dilakukan oleh

BMKG Provinsi Bengkulu. Oleh karena itu, untuk menjaga keaslian dan keamanan data citra tersebut, perlu dilakukan tindakan preventif seperti *watermarking*.

Watermarking pada citra digital dapat diterapkan pada BMKG untuk melindungi hak cipta dan keaslian data citra yang dihasilkan. BMKG memiliki tugas utama dalam melakukan pengamatan, pengolahan, dan penyajian data cuaca dan iklim serta gempa bumi dan tsunami yang terjadi di Indonesia. Data-data yang dihasilkan oleh BMKG memiliki nilai penting dalam berbagai bidang seperti pertanian, transportasi, perikanan, dan keamanan nasional.

Salah satu algoritma yang dapat digunakan untuk melakukan *watermarking* pada citra digital adalah algoritma *Linear Congruential Generator* (LCG). Algoritma ini memiliki kelebihan dalam menghasilkan deret bilangan acak dengan cepat dan mudah diimplementasikan. Selain itu, algoritma LCG juga dapat diatur untuk menghasilkan kunci acak yang berbeda-beda dengan mengubah nilai *seed*, konstanta a , dan modulus m .

Dengan menerapkan *watermarking* pada citra-citra yang dihasilkan oleh BMKG menggunakan algoritma LCG, maka citra tersebut akan terlindungi dari penggunaan yang tidak sah serta dapat digunakan sebagai bukti autentikasi keaslian data citra tersebut. Hal ini tentunya akan meningkatkan kepercayaan dan kredibilitas BMKG sebagai lembaga resmi yang menghasilkan data cuaca, iklim, dan gempa bumi di Indonesia.

Berdasarkan latar belakang diatas maka penulis tertarik untuk melakukan penelitian yang dituangkan dalam bentuk proposal skripsi yang

berjudul “**Implementasi *Watermaking* Pada Citra Digital Menggunakan Metode *Linear Congruential Generator* (LCG) Di Badan Meteorologi Klimatologi dan Geofisika Provinsi Bengkulu**”.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang dikemukakan diatas, maka yang menjadi rumusan masalah pada penelitian ini adalah bagaimana implementasi *watermaking* pada citra digital menggunakan metode *Linear Congruential Generator* (LCG) di Badan Meteorologi Klimatologi dan Geofisika Provinsi Bengkulu.

1.3 Batasan Masalah

Agar permasalahan tidak meluas dan menyimpang dari pembahasan maka penelitian ini dibatasi pada :

1. Citra yang digunakan dalam penelitian ini bertipe bitmap (BMP)
2. Sistem akan dibangun dengan menggunakan bahasa pemrograman Visual Studio 2010.

1.4 Tujuan Penelitian

1.4.1 Tujuan Umum

Adapun tujuan umum dari penelitian yang dilakukan oleh penulis yaitu untuk memenuhi salah satu syarat kurikulum dalam menyelesaikan Strata Satu (S1) Pada Program Studi Informatika Universitas Dehasen Bengkulu.

1.4.2 Tujuan Khusus

Adapun tujuan khusus dari penelitian ini adalah untuk mengimplementasi *watermaking* pada citra digital menggunakan metode *Linear Congruential Generator* (LCG) di Badan Meteorologi Klimatologi dan Geofisika Provinsi Bengkulu.

1.5 Manfaat Penelitian

1.5.1 Bagi BMKG Provinsi Bengkulu

- a. Memberikan keamanan tambahan dalam autentikasi kepemilikan data sebelum data dapat disebarluaskan.
- b. Memberikan sebuah penanda bagi citra untuk menjaga hak milik BMKG Provinsi Bengkulu.

1.5.2 Bagi Pembaca

- a. Dapat dijadikan referensi yang digunakan dengan tema penelitian yang sama.
- b. Dapat digunakan sebagai tambahan ilmu dan pengetahuan.

BAB II

LANDASAN TEORI

2.1 Pengertian Watermaking

Watermarking sudah ada sejak 700 tahun yang lalu. Pada akhir abad 13, pabrik kertas di Fabriano, Italia, membuat kertas yang diberi watermark atau tanda air dengan cara menekan bentuk cetakan gambar atau tulisan pada kertas yang baru setengah jadi. Ketika kertas dikeringkan terbentuklah suatu kertas yang berwatermark. *Watermarking* merupakan suatu bentuk dari *steganography*, yaitu bidang ilmu yang mempelajari bagaimana menyembunyikan suatu data baik itu citra maupun teks pada data yang lain (Situmorang, 2018)

Watermarking adalah metode untuk menyisipkan suatu informasi pada suatu data digital dengan tujuan untuk perlindungan kepemilikan data tersebut. Informasi tersebut dapat berupa citra digital, audio maupun video (Ikromina & Ujianto, 2019). *Watermarking* atau tanda air bisa diartikan sebagai suatu teknik penyembunyian data atau informasi rahasia ke dalam suatu data lainnya dengan cara menumpanginya (kadang disebut *host data*), tanpa orang lain menyadarinya adanya data tambahan pada data *host*-nya (Aulia, 2019)

Teknik *watermarking* memanfaatkan kekurangan pada indera manusia (mata dan telinga). Pada dasarnya, teknik *watermarking* adalah proses dari menambahkan kode identifikasi secara permanen ke digital data. Kode identifikasi dapat berupa teks, suara, gambar atau video (Rahim, dkk, 2018).

Informasi yang disisipkan tersebut disebut dengan *watermarking* yang dapat dideteksi oleh komputer. Ada dua *watermarking* yaitu *watermarking* yang terlihat dan tidak terlihat untuk mengindikasikan gambar pada *pixel* atau dalam kode *byte* (Supiyandi, dkk, 2018).

Berdasarkan pendapat terdahulu diatas dapat diambil kesimpulan bahwa penerapan *watermarking* pada data dapat memberikan autentikasi, pengamanan, dan juga klaim terhadap data yang dimiliki oleh seseorang tanpa menimbulkan kerusakan terhadap data atau merubah tampilan informasi yang dimiliki oleh data.

Ada beberapa masalah yang melatar belakangi munculnya *watermarking*, antara lain :

a. Masalah kepemilikan

Pemalsuan atas kepemilikan produk *digital* sering terjadi disekitar Kita. Sebagai contoh adalah foto *digital*, tidak memiliki suatu label atau informasi yang melekat pada foto tersebut. Apabila ada klaim dari pihak lain yang juga mengaku sebagai pemilik sah atas foto *digital* tersebut, pemilik foto asli tidak dapat membantah karena ia tidak mempunyai bukti otentik yang menandakan kepemilikan terhadap foto tersebut.

b. Masalah pelanggaran hak cipta

Penggunaan yang tidak memiliki izin atas produk *digital* dapat merugikan pemiliknya, sebab pemilik produk *digital* tidak mendapat royalti apapun terhadap proses penggunaan tersebut. Banyak cara yang sudah ditempuh untuk memberikan atau melindungi data *digital*, seperti : *encryption*, *copy protection*, *visible marking*, *header marking*, dan

sebagainya, tetapi semua cara tersebut memiliki kelemahan masing-masing.

c. Masalah keaslian

Produk *digital* sangat mudah untuk dilakukan perubahan. Perubahan tersebut dapat berupa rekayasa terhadap produk asli, baik perubahan yang dapat dipersepsi maupun tidak. Perubahan yang timbul dapat menyebabkan informasi penting yang terdapat produk *digital* tersebut hilang. Teknik kriptografi biasa saja tidak dapat menyelesaikan masalah-masalah diatas. Meskipun produk-produk *digital* dienkrpsi, menggunakan algoritma RSA sekalipun, cukup sekali saja diperlukan dekripsi produk-produk *digital* tersebut. Setelah enkripsi dihilangkan, produk-produk *digital* tadi dapat langsung diperbanyak dan disebarakan tanpa perlu melakukan dekripsi lagi. Selain itu, tidak terdapat jejak yang dapat menunjukkan bahwa seseorang bertanggung jawab atas penyebarluasan produk *digital* ataupun otentikasi mengenai hak seseorang atas produk *digital* tersebut.

Data *watermark* yang disisipkan ke dalam data *digital* biasanya berupa teks, gambar, *video* dan suara. Semua jenis data *watermark*, baik teks, gambar, *video*, suara maupun berkas *digital* yang lain dapat digunakan sebagai *watermark*, dengan catatan bahwa ukuran dari data *watermark* cukup (tidak lebih besar dari media penampung) untuk disisipkan pada media penampung. Berdasarkan tipe dokumen yang diberi *watermark*, *watermarking* dapat diklasifikasikan menjadi) :

1. *Image Watermarking*
2. *Video Watermarking*

3. *Audio Watermarking*

4. *Text Watermarking*

Watermark berupa data suara, *video*, maupun gambar lebih disukai karena kesalahan pada beberapa *bit watermark* tidak menghasilkan perubahan yang cukup berarti pada saat dilakukan ekstraksi. Hasil ekstraksi tersebut masih dapat dipersepsi dengan memanfaatkan kelemahan pada sistem kelemahan penglihatan manusia (*Human Visual System/HVS*) maupun sistem pendengaran manusia (*Human Auditory System/HAS*). Dengan memahami karakteristik dari HVS dan HAS maka akan menghasilkan metode yang baik untuk *watermarking*.

Sebuah teknik *watermarking* yang bagus harus memenuhi persyaratan berikut :

- a. *Imperceptibility* : keberadaan *watermark* tidak dapat dipersepsi oleh indera *visual*. Hal ini bertujuan untuk menghindari gangguan pengamatan *visual*.
- b. *Key uniqueness* : kunci yang berbeda seharusnya menghasilkan *watermark* yang berbeda. Ini berarti penggunaan kunci yang salah dapat menyebabkan hasil ekstraksi/deteksi *watermark* yang salah pula
- c. *Non invertibility*: secara komputasi sangat sukar menemukan *watermark* bila diketahui hanya citra ber*watermark* saja
- d. *Image dependency*: suatu kunci menghasilkan sebuah *watermark* tunggal, tetapi *watermark* bergantung pada isi citra. Salah satu pendekatan yang digunakan adalah membangkitkan *watermark* dari nilai *hash (message digest)* citra asli, sebab nilai *hash* bergantung pada isi citra.

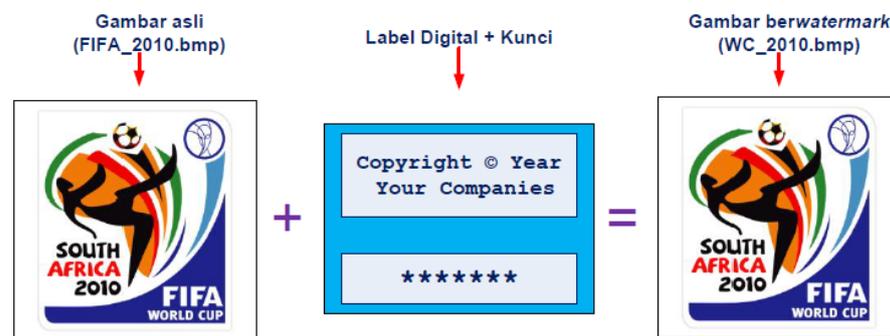
- e. *Robustness*: *watermark* seharusnya tetap kokoh terhadap berbagai serangan yang dilakukan pada citra ber*watermark*. Ini berarti manipulasi yang dilakukan terhadap citra ber*watermark* tidak merusak *watermark* (*watermark* masih dapat dideteksi). Manipulasi citra meliputi operasi seperti penambahan derau aditif (*Gaussian* atau *non-Gaussian*), kompresi (seperti JPEG), transformasi geometri (seperti rotasi, perbesaran, perkecilan), penapisan (baik penapisan lanjar maupun nirlanjar), konversi *digital* ke *analog* (D/A) atau A/D, seperti pemindaian citra.
- f. *Fidelity*: perubahan yang disebabkan oleh *watermark* semestinya tidak mempengaruhi nilai isi, idealnya *watermark* harusnya tidak dapat dilihat, sehingga tidak dapat dibedakan antara data yang ber*watermark* dan data yang asli. Salah satu *trade-off* antara karakteristik *watermarking* yang sangat terlihat adalah antara *robustness* dengan *fidelity*. Dalam beberapa literatur *fidelity* kadang disebut dengan *invisibility* untuk jenis data citra dan *video*. Yang dimaksud *fidelity* disini adalah derajat degradasi *host* data sesudah diberi *watermark* dibandingkan dengan sebelum diberi *watermark*. Biasanya bila *robustness* dari *watermark* tinggi maka memiliki *fidelity* yang rendah, sebaliknya *robustness* yang rendah dapat membuat *fidelity* yang tinggi. Jadi sebaiknya dipilih *trade-off* yang sesuai, sehingga keduanya dapat tercapai sesuai dengan tujuan aplikasi. Untuk *host* data yang berkualitas tinggi maka *fidelity* dituntut setinggi mungkin sehingga tidak merusak data aslinya, sedangkan *host* data yang memiliki *noise* (kualitas kurang) maka *fidelity*-nya bisa rendah.

g. *Security*: *watermarking* harus tahan terhadap usaha sengaja memindahkan atau menggandakan *watermark* dari satu multimedia data ke multimedia data lainnya.

Watermarking pada citra *digital* (*image watermarking*) dapat dikategorikan sebagai berikut :

- a. Berdasarkan persepsi manusia, *image watermarking* dibedakan menjadi *visible watermarking* dan *invisible watermarking*. Yang dimaksud dengan *visible watermarking* adalah *watermark* terlihat jelas oleh sistem penglihatan manusia (HVS). Sedangkan *invisible watermarking* adalah *watermark* tidak terlihat jelas atau tidak tampak oleh sistem penglihatan manusia (HVS).
- b. Berdasarkan proses verifikasi *watermark*, *image watermarking* dibedakan menjadi *blind watermarking* dan *non blind watermarking*. Yang dimaksud dengan *blind watermarking* adalah pada saat terjadi proses verifikasi citra *watermark* tidak membutuhkan citra asal. Sedangkan *non blind watermarking* adalah *watermarking* yang pada saat terjadi proses verifikasi citra *watermark* memerlukan citra asli/asal.
- c. Berdasarkan tingkat kekokohan *watermark*, *image watermarking* dibedakan menjadi *secure watermarking*, *robust watermarking*, dan *fragile watermarking*. *Secure watermarking* berarti *watermark* harus tetap bertahan terhadap serangan (*malicious* dan *non malicious*). Serangan *malicious* diasumsikan langsung menyerang pada algoritma *watermarking*, sedangkan serangan *non malicious* seperti operasi kompresi, operasi penapisan, penambahan derau, penskalaan,

penyuntingan, operasi geometri, dan *cropping*. *Robust watermarking* berarti *watermark* harus tetap bertahan terhadap serangan *non malicious*. Sedangkan *fragile watermarking* berarti *watermark* akan mudah rusak jika terjadi serangan, namun kehadirannya tidak terdeteksi oleh indera manusia. Berikut ini diperlihatkan ilustrasi penyisipan label *digital* pada gambar bitmap :



Gambar 2.1 Ilustrasi *Watermarking* pada Citra *Digital*

2.2 Linear Congruential Generator (LCG)

Linear Congruential Generator (LCG) atau jika diubah kedalam bahasa Indonesia menjadi Pembangkit Bilangan Acak Kongruen-Lanjar merupakan pembangkit bilangan acak yang sederhana, mudah dimengerti teorinya, dan juga mudah untuk diimplementasikan (Putra & Harianto, 2018).

Linear Congruential Generator (LCG) merupakan algoritma yang menghasilkan deret pseudo random yang dihitung berdasarkan kongruensi linear. Kongruensi linear yang dimaksud adalah sebagai berikut (Nanda & Gelar, 2022) :

$$Z_i = (aZ_{i-1} + c) \bmod m$$

Dimana :

Z_i = bilangan acak ke-i dari deretnya

Z_{i-1} = bilangan acak sebelumnya

a = faktor pengali

c = increment

m = modulus

kunci pembangkit adalah Z_0 yang disebut umpan (*seed*)

Penentuan nilai awal Z_0 atau Z_{-1} dan konstanta (a , b , dan m) akan menentukan kualitas bilangan acak yang dihasilkan. Bilangan acak yang baik (pada umumnya) apabila terjadinya perulangan atau munculnya bilangan acak yang sama, dapat terjadi setelah sekian banyak pembangkitan bilangan acak (semakin banyak akan semakin baik) serta tidak bisa diprediksi kapan terjadi perulangannya. Periode dari LCG umumnya adalah sebesar nilai m . Masalah pada LCG adalah lower order bit yang digenerasi mempunyai periode yang lebih pendek dari deretan secara keseluruhan jika m di-set menjadi pangkat 2. Tanpa desain yang benar, dengan m yang sangat besar, bisa jadi periode bilangan acak yang dihasilkan tidak akan maksimal, bahkan mungkin jauh lebih pendek daripada periode maksimalnya.

Kunci pembangkit adalah Z_0 yang disebut umpan (*seed*). LCG mempunyai periode tidak lebih besar dari m . Jika a , b , dan m dipilih secara tepat (misalnya b seharusnya relatif prima terhadap m dan $b < m$), maka LCG akan mempunyai periode maksimal, yaitu $m - 1$. Sebagai contoh : Untuk membangkitkan bilangan acak sebanyak 10 kali dengan $a=13$, $b=7$,

$m=11$, dan $0 = 2$. Dengan menggunakan rumus pada metode LCG di atas, akan diperoleh hasil sesuai tabel 2.1

Tabel 2.1. Hasil Pembangkitan Bilangan Acak dengan Metode LCG

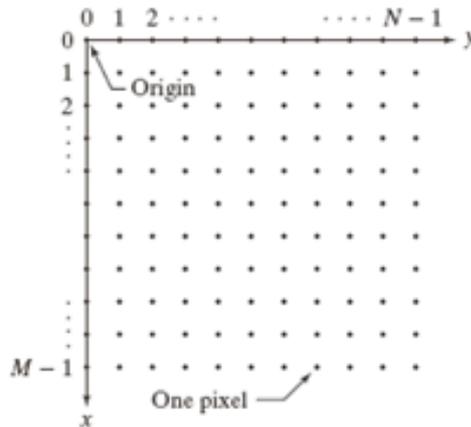
n	X_{n-1}	a	b	$a.X_{n-1} + b$	m	$X_n = (a.X_{n-1} + b) \bmod m$
1	2	13	7	33	11	0
2	0	13	7	7	11	7
3	7	13	7	98	11	10
4	10	13	7	137	11	5
5	5	13	7	72	11	6
6	6	13	7	85	11	8
7	8	13	7	111	11	1
8	1	13	7	20	11	9
9	9	13	7	124	11	3
10	3	13	7	46	11	2

2.3 Citra Digital

Citra (*image*) adalah gambar pada bidang dwimatra (dua dimensi). Ditinjau dari sudut pandang matematis, citra merupakan fungsi menerus (*continue*) dari intensitas cahaya pada bidang dwimatra. Sumber cahaya menerangi objek, objek memantulkan kembali sebagian dari berkas cahaya tersebut. Pantulan cahaya ini ditangkap oleh alat-alat optik, misalnya mata pada manusia, kamera, pemindai (*scanner*), dan sebagainya, sehingga bayangan objek yang disebut citra tersebut terekam. (Arsy, et.al, 2017)

Citra yang ditangkap oleh kamera dan telah dikuantisasi dalam bentuk nilai diskrit disebut sebagai citra digital (*digital image*). Foto hasil cetak dari printer tidak dapat disebut sebagai citra digital, namun foto yang tersimpan dalam *file* gambar (bmp, jpg, png atau format lainnya) pada komputer dapat disebut sebagai citra digital. (Sinaga, 2017).

Citra digital dibentuk oleh kumpulan titik yang dinamakan piksel nonton siccin (*pixel* atau “*picture element*”). Setiap piksel digambarkan sebagai satu kotak kecil. Setiap piksel mempunyai koordinat posisi. Sistem koordinat yang dipakai untuk menyatakan citra digital ditunjukkan pada Gambar 2.1 berikut



Gambar 2.2 Ilustrasi Citra Digital

Ada banyak cara untuk menyimpan citra digital di dalam memori. Cara penyimpanan menentukan jenis citra digital yang terbentuk. Format citra digital yang banyak dipakai adalah sebagai berikut :

A. Citra Biner

Citra biner adalah citra digital yang hanya memiliki dua kemungkinan nilai pixel yaitu hitam dan putih. Citra biner juga disebut sebagai citra *B&W (black and white)* atau citra monokrom. hanya dibutuhkan 1 bit untuk mewakili nilai setiap pixel dari citra biner. Citra biner (*monochrome*) atau disebut juga *binary image*, merupakan citra digital yang setiap *pixel*-nya hanya memiliki 2 kemungkinan derajat keabuan, yaitu 0 dan 1 (Sinaga, 2017). Nilai 0 mewakili warna hitam,

dan nilai 1 mewakili warna putih, di mana setiap *pixel*-nya membutuhkan media penyimpanan sebesar 1 bit



Gambar 2.3 Citra Biner

B. Citra Grayscale

Citra *grayscale* menangani gradasi warna hitam dan putih, yang tentu saja menghasilkan warna abu-abu. Dalam hal ini intensitas berkisar antara 0 sampai dengan 255. Nilai 0 menyatakan hitam dan nilai 255 menyatakan putih. Banyaknya warna tergantung pada jumlah bit yang disediakan dimemori untuk menampung kebutuhan warna ini. Semakin besar jumlah bit warna yang disediakan di memori, semakin halus gradasi warna yang terbentuk. Gambar 24 menunjukkan perbandingan gradasi warna untuk jumlah bit tertentu



Gambar. 2.4 Citra Grayscale 4 Bit

C. Citra Warna

Setiap piksel pada citra warna memiliki warna yang merupakan kombinasi dari tiga warna dasar RGB (*Red, Green, Blue*). Setiap warna dasar menggunakan penyimpanan 8 bit = 1 byte, yang berarti setiap

warna mempunyai gradasi sebanyak 255 warna. Berarti setiap piksel mempunyai kombinasi warna sebanyak $28 \cdot 28 \cdot 28 = 224 = 16$ juta warna lebih. Itulah yang menjadikan alasan format ini disebut dengan *true color* karena mempunyai jumlah warna yang cukup besar sehingga bias dikatakan hampir mencakup semua warna di alam. Penyimpanan citra *true color* di dalam memori berbeda dengan citra *grayscale*. Setiap piksel dari citra *grayscale* 256 gradasi warna diwakili oleh 1 byte. Sedangkan 1 piksel citra *true color* diwakili oleh 3 byte, dimana masing-masing byte merepresentasikan warna merah, hijau dan biru



Gambar. 2.5 Citra Warna

2.3.1 Pengolahan Citra Digital

Pengolahan citra adalah disiplin ilmu yang mempelajari hal-hal yang berkaitan dengan perbaikan kualitas gambar (peningkatan kontras, transformasi warna, restorasi), transformasi gambar (rotasi, translasi, skala, transformasi geometrik), melakukan pemilihanciri citra (*feature extraction*) yang optimal untuk bertujuan analisis, melakukan proses penarikan informasi atau deskripsi objek atau pengenalan objek yang terkandung pada citra, melakukan kompresi atau reduksi data untuk tujuan penyimpanan, transmisi dan waktu proses data. (Sinaga, 2017).

Pengolahan citra merupakan cabang ilmu dalam Artificial Intelligence yang menggunakan objek citra dalam bentuk digital untuk penyelesaian kasusnya. Metode dalam citra dapat digunakan baik

perhitungan matematis pada objek secara piksel ataupun geometris. Masing-masing objek citra memiliki nilai perbedaan yang dapat diperhitungkan secara matematis, sehingga menunjukkan ciri yang berbeda antara objek satu dengan yang lain. Penciri dari perbedaan setiap objek dapat ditentukan dari warna, tekstur, ataupun bentuk (Widyaningsih, 2017). Dengan memanfaatkan informasi digital ini pengelompokkan atau clustering dapat di implementasikan terhadap objek.

2.3.2 Tujuan Pengolahan Citra Digital

Pengolahan citra digital banyak dimanfaatkan oleh berbagai bidang mulai dari kewanitaan, kesehatan, pendidikan dan bidang – bidang yang lain. Berikut beberapa tujuan dari kegiatan pengolahan citra digital

1. Memperbaiki kualitas gambar dilihat dari aspek *radiometric* (peningkatan kontras, transformasi warna, restorasi citra) dan dari aspek *geometric* (rotasi, translasi, skala, transformasi geometrik).
2. Melakukan proses penarikan informasi atau deskripsi objek atau pengenalan objek yang terkandung pada citra.
3. Melakukan kompresi atau reduksi data untuk tujuan penyimpanan data, transmisi data, dan waktu proses data.

2.4 Tinjauan Umum Visual Basic.Net 2010

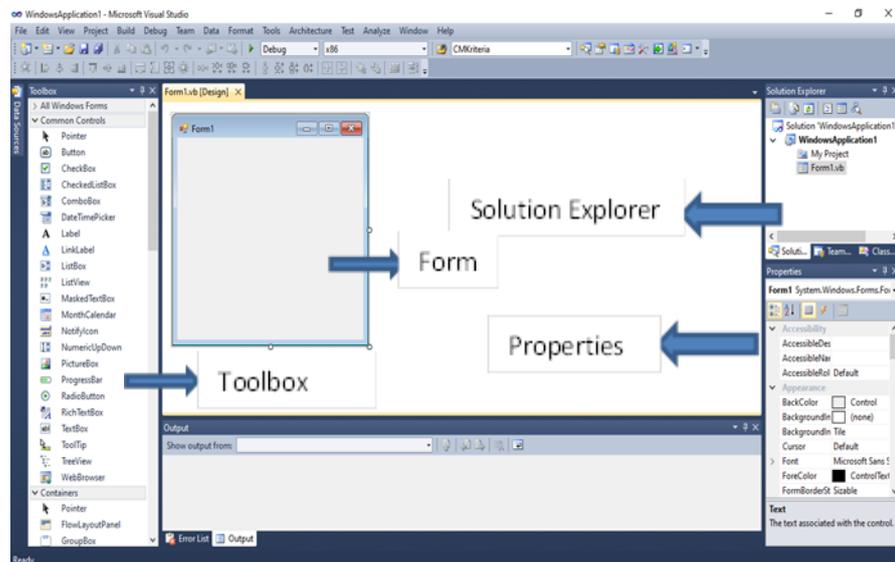
Microsoft Visual Basic.Net adalah sebuah alat untuk mengembangkan dan membangun aplikasi yang bergerak di atas sistem .NET Framework, dengan menggunakan bahasa BASIC. Dengan menggunakan alat ini, para programmer dapat membangun aplikasi *Windows Forms*, Aplikasi web

berbasis ASP.NET, dan juga aplikasi *command-line*. Alat ini dapat diperoleh secara terpisah dari beberapa produk lainnya (seperti *Microsoft Visual C++*, *Visual C#*, atau *Visual J#*), atau juga dapat diperoleh secara terpadu dalam Microsoft Visual Studio (R.H Sianipar, 2017).

Visual Basic.Net merupakan salah satu *Development Tool* yaitu alat bantu untuk membuat berbagai macam program komputer, khususnya yang menggunakan sistem operasi *Windows*. Visual Basic merupakan salah satu bahasa pemrograman komputer yang mendukung object (*Object Oriented Programming = OOP*)

2.4.1 Menu Utama *Integrated Development Environment*

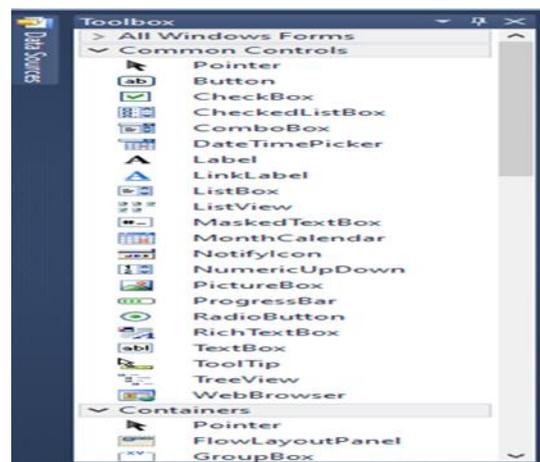
Di dalam menu utama *Integrated Development Environment* (IDE) tersedia perintah-perintah dan disertai pula dengan submenu-submenunya. Pada umumnya menu juga dapat ditampilkan dalam bentuk toolbar, tetapi tidak semua opsi tersedia pada saat itu juga. Ada kalanya opsi-opsi tersebut tidak dapat diterapkan pada tempat IDE. Ini berarti opsi tersebut dalam keadaan *invisible* atau *disabled*



Gambar 2.6 Komponen Visual Basic 2010

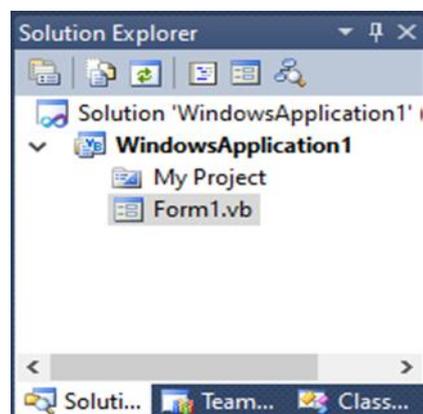
2.4.2 *Toolbox Windows Form*

Toolbox berisi berbagai control yang dapat anda gunakan untuk mendesain antarmuka grafis. *Toolbox* mempunyai pengaturan automatic hiding sehingga akan tertutup jika tidak diperlukan

**Gambar 2.7** *Toolbox*

2.4.3 *Jendela Explorer*

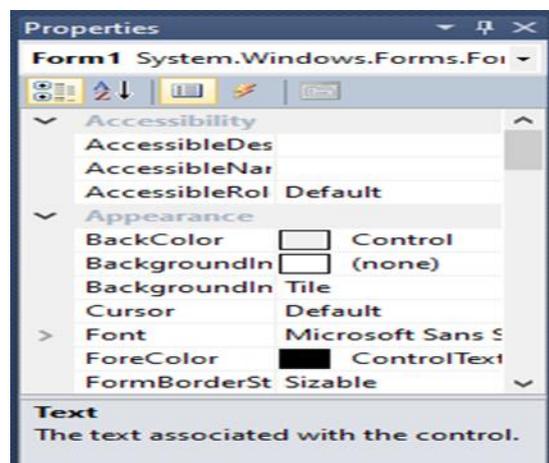
Jendela *explorer* merupakan tempat ditampilkannya daftar-daftar komponen secara hirarki. Dalam Jendela *explorer* dimungkinkan adanya beberapa proyek, dan dalam proyek ini masih ada beberapa item lagi seperti *form*, *module*, dan lain-lain



Gambar 2.8 *Jendela Explorer*

2.4.4 Jendela *Properties*

Jendela propertis ini berfungsi untuk menampilkan semua *property* dari komponen yang dipilih beserta settingannya. Dengan jendela ini kita dapat mengatur *property* dari masing-masing kontrol yang telah dibuat



Gambar 2.9 *Jendela Properties*

2.5 UML (*Unified Modeling Language*)

UML adalah salah satu *tool* atau model untuk merancang pengembangan *software* yang berbasis *object-oriented*. UML sendiri juga memberikan standar penulisan sebuah sistem *blueprint*, yang meliputi konsep proses bisnis, penulisan kelas-kelas dalam bahasa program yang spesifik, skema database, dan komponen yang diperlukan dalam sistem *software* (Sonata & Sari, 2019)

UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung”. Beberapa pemodelan yang termasuk kedalam pemodelan UML

seperti *use case diagram*, *class diagram*, *activity diagram*, dan *sequence diagram* (Syarif & Nugraha, 2020)

Adapun tujuan dari UML adalah:

1. Merancang perangkat lunak.
2. Sarana komunikasi antara perangkat lunak dengan proses bisnis.
3. Menjabarkan sistem secara rinci untuk analisa dan mencari apa yang diperlukan sistem.
4. Mendokumentasi sistem yang ada, proses-proses dan organisasinya

2.5.1 Use Case Diagram

Use case atau *diagram use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat (Syarif & Nugraha, 2020). *Use case* merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, meng-*create* sebuah daftar belanja, dan sebagainya.

Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu. *Use case diagram* dapat sangat membantu bila kita sedang menyusun *requirement* sebuah sistem, mengkomunikasikan rancangan dengan klien, dan merancang *test case* untuk semua *feature* yang ada pada sistem. Sebuah *use case* dapat meng-*include* fungsionalitas *use case* lain sebagai bagian dari proses dalam dirinya.

Secara umum diasumsikan bahwa *use case* yang di-*include* akan dipanggil setiap kali *use case* yang meng-*include* dieksekusi

secara normal. Sebuah *use case* dapat di-*include* oleh lebih dari satu *use case* lain, sehingga duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar fungsionalitas yang *common*. Sebuah *use case* juga dapat meng-*extend* *use case* lain dengan *behaviour*-nya sendiri. Sementara hubungan generalisasi antar *use case* menunjukkan bahwa *use case* yang satu merupakan spesialisasi dari yang lain.

Tabel 2.2 Simbol Use Case Diagram

No	Gambar	Nama	Keterangan
1		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri(<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>).
3		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).

No	Gambar	Nama	Keterangan
4		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara <i>eksplisit</i> .
5		<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
6		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
9		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).

No	Gambar	Nama	Keterangan
10		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

2.5.2 Activity Diagram

Diagram aktivitas atau *activity diagram* menggambarkan *work flow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak (Syarif & Nugraha, 2020). Pada dasarnya, *activity diagram* merupakan variasi dari statechart diagram. *Activity diagram* mempunyai peran seperti halnya *flowchart*, akan tetapi perbedaannya dengan *flowchart* adalah *activity diagram* bisa mendukung perilaku paralel sedangkan *flowchart* tidak bisa. Berikut adalah notasi *activity diagram*

Tabel 2.3 Simbol-Simbol Activity Diagram

No	Gambar	Nama	Keterangan
1		<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		<i>Action</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi
3		<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.

4		<i>Activity Final Node</i>	Bagaimana objek dibentuk dan dihancurkan
5		<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran

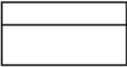
2.5.3 Class Diagram

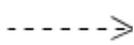
Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi definisi kelas-kelas yang akan dibuat untuk membangun sistem (Syarif & Nugraha, 2020). *Class diagram* membantu dalam memvisualisasikan struktur kelas-kelas dari suatu sistem dan merupakan tipe diagram yang paling banyak dipakai. Selama tahap desain, *class diagram* berperan dalam menangkap struktur dari semua kelas yang membentuk arsitektur sistem yang dibuat. Kelas-kelas yang ada pada struktur sistem harus dapat melakukan fungsi-fungsi sesuai dengan kebutuhan sistem sehingga pembuat perangkat lunak atau *programmer* dapat membuat kelas-kelas di dalam program perangkat lunak sesuai dengan perancangan diagram kelas. Susunan struktur kelas yang baik pada diagram kelas sebaiknya memiliki jenis-jenis kelas berikut :

1. Kelas Main, kelas yang memiliki fungsi awal dieksekusi ketika sistem dijalankan
2. Kelas yang menangani tampilan sistem (*view*), kelas yang mendefinisikan dan mengatur tampilan ke pemakai

3. Kelas yang diambil dari pendefinisian *use case (controller)*, kelas yang menangani fungsi-fungsi yang baru ada didalam dari pendefinisian *use case*, kelas bisanya disebut dengan proses yang menangani proses bisnis pada perangkat lunak
4. Kelas yang diambil dari pendefinisian data (model), kelas yang digunakan untuk memegang atau membungkus data menjadi sebuah kesatuan yang diambil maupun akan disimpan ke basis data

Tabel 2.4 Simbol-Simbol *Class Diagram*

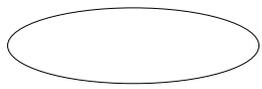
No	Gambar	Nama	Keterangan
1		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk
2		<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4		<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor
5		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.

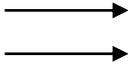
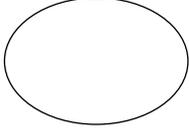
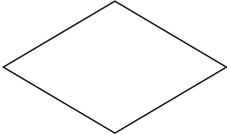
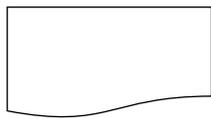
No	Gambar	Nama	Keterangan
6		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri
7		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya

2.6 Flowchart

Flowchart adalah representasi secara simbolik dari suatu algoritma atau prosedur untuk menyelesaikan suatu masalah, dengan menggunakan *flowchart* akan memudahkan pengguna melakukan pengecekan bagian-bagian yang terlupakan dalam analisis masalah, disamping itu *flowchart* juga berguna sebagai fasilitas untuk berkomunikasi antara pemrogram yang bekerja dalam tim suatu proyek. *Flowchart* membantu memahami urutan-urutan logika yang rumit dan panjang. *Flowchart* membantu mengkomunikasikan jalannya program ke orang lain (bukan pemrogram) akan lebih mudah (Santoso & Nurmalina, 2017).

Tabel 2.5 Simbol dan Fungsi *Flowchart*

Gambar	Nama Simbol	Keterangan
	<i>Start / Mulai</i> <i>End / Selesai</i>	Simbol yang digunakan untuk memulai / selesai

Gambar	Nama Simbol	Keterangan
	<i>Flow</i>	Simbol arus/ <i>flow</i> yang menyatakan jalannya proses
	<i>Connector</i>	Simbol <i>connector</i> , (menyatakan sambungan dari proses ke proses lainnya dalam hal yang sama)
	<i>Process</i>	Simbol proses yaitu menyatakan suatu tindakan
	<i>Manual Operation</i>	Simbol manual, menyatakan suatu tindakan
	<i>Decision</i>	Simbol <i>decision</i> , menunjukkan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan
	<i>Keying Operation</i>	Simbol <i>keying operation</i> menyatakan segala jenis operasi yang diproses dengan menggunakan suatu mesin yang mempunyai keyboard.
	<i>Input/Output</i>	Simbol <i>input</i> atau <i>output</i> menyatakan proses <i>input</i> atau <i>output</i>
	<i>Document</i>	Simbol dokumen mencetak keluaran dalam bentuk dokumen

BAB III

METODOLOGI PENELITIAN

3.1 Subjek Penelitian

3.1.1 Gambaran Umum Badan Meterologi, Klimatologi dan Geofisika (BMKG)

Sejarah pengamatan meteorologi dan geofisika di Indonesia dimulai pada tahun 1841 diawali dengan pengamatan yang dilakukan secara perorangan oleh Dr. Onnen, Kepala Rumah Sakit di Bogor. Tahun demi tahun kegiatannya berkembang sesuai dengan semakin diperlukannya data hasil pengamatan cuaca dan geofisika.

Pada tahun 1866, kegiatan pengamatan perorangan tersebut oleh Pemerintah Hindia Belanda diresmikan menjadi instansi pemerintah dengan nama *Magnetisch en Meteorologisch Observatorium* atau *Observatorium Magnetik dan Meteorologi* dipimpin oleh Dr. Bergsma.

Pada tahun 1879 dibangun jaringan penakar hujan sebanyak 74 stasiun pengamatan di Jawa. Pada tahun 1902 pengamatan medan magnet bumi dipindahkan dari Jakarta ke Bogor. Pengamatan gempa bumi dimulai pada tahun 1908 dengan pemasangan komponen horisontal seismograf Wiechert di Jakarta, sedangkan pemasangan komponen vertikal dilaksanakan pada tahun 1928.

Pada tahun 1912 dilakukan reorganisasi pengamatan meteorologi dengan menambah jaringan sekunder. Sedangkan jasa meteorologi mulai digunakan untuk penerangan pada tahun 1930. Pada masa pendudukan Jepang antara tahun 1942 sampai dengan 1945, nama instansi meteorologi dan geofisika diganti menjadi *Kisho Kauso Kusho*.

Setelah proklamasi kemerdekaan Indonesia pada tahun 1945, instansi tersebut dipecah menjadi dua: Di Yogyakarta dibentuk Biro Meteorologi yang berada di lingkungan Markas Tertinggi Tentara Rakyat Indonesia khusus untuk melayani kepentingan Angkatan Udara. Di Jakarta dibentuk Jawatan Meteorologi dan Geofisika, dibawah Kementerian Pekerjaan Umum dan Tenaga.

Pada tanggal 21 Juli 1947 Jawatan Meteorologi dan Geofisika diambil alih oleh Pemerintah Belanda dan namanya diganti menjadi Meteorologisch en Geofisiche Dienst. Sementara itu, ada juga Jawatan Meteorologi dan Geofisika yang dipertahankan oleh Pemerintah Republik Indonesia, kedudukan instansi tersebut di Jl. Gondangdia, Jakarta.

Pada tahun 1949, setelah penyerahan kedaulatan negara Republik Indonesia dari Belanda, Meteorologisch en Geofisiche Dienst diubah menjadi Jawatan Meteorologi dan Geofisika dibawah Departemen Perhubungan dan Pekerjaan Umum. Selanjutnya, pada tahun 1950 Indonesia secara resmi masuk sebagai anggota Organisasi Meteorologi Dunia (World Meteorological Organization atau WMO) dan Kepala Jawatan Meteorologi dan Geofisika menjadi Permanent Representative of Indonesia with WMO.

Pada tahun 1955 Jawatan Meteorologi dan Geofisika diubah namanya menjadi Lembaga Meteorologi dan Geofisika di bawah Departemen Perhubungan, dan pada tahun 1960 namanya dikembalikan menjadi Jawatan Meteorologi dan Geofisika di bawah Departemen Perhubungan Udara. Pada tahun 1965, namanya diubah menjadi Direktorat Meteorologi dan Geofisika, kedudukannya tetap di bawah Departemen Perhubungan Udara. Pada tahun 1972, Direktorat Meteorologi

dan Geofisika diganti namanya menjadi Pusat Meteorologi dan Geofisika, suatu instansi setingkat eselon II di bawah Departemen Perhubungan, dan pada tahun 1980 statusnya dinaikkan menjadi suatu instansi setingkat eselon I dengan nama Badan Meteorologi dan Geofisika, dengan kedudukan tetap berada di bawah Departemen Perhubungan. Pada tahun 2002, dengan keputusan Presiden RI Nomor 46 dan 48 tahun 2002, struktur organisasinya diubah menjadi Lembaga Pemerintah Non Departemen (LPND) dengan nama tetap Badan Meteorologi dan Geofisika.

Terakhir, melalui Peraturan Presiden Nomor 61 Tahun 2008, Badan Meteorologi dan Geofisika berganti nama menjadi Badan Meteorologi, Klimatologi, dan Geofisika (BMKG) dengan status tetap sebagai Lembaga Pemerintah Non Departemen. Pada tanggal 1 Oktober 2009 Undang-Undang Republik Indonesia Nomor 31 Tahun 2009 tentang Meteorologi, Klimatologi dan Geofisika disahkan oleh Presiden Republik Indonesia, Susilo Bambang Yudhoyono. (unduh Penjelasan UU RI Nomor 31 Tahun 2009

Badan Meteorologi, Klimatologi, dan Geofisika (disingkat BMKG), sebelumnya bernama Badan Meteorologi dan Geofisika (disingkat BMG) adalah Lembaga Pemerintah Non Kementrian di Indonesia yang mempunyai tugas melaksanakan tugas pemerintahan di bidang meteorologi, klimatologi, dan geofisika sesuai dengan ketentuan perundang-undangan yang berlaku. Adapaun tujuan dari BMKG adalah mengamati dan memahami fenomena meteorologi, klimatologi, kualitas udara dan geofisika. Menyediakan data, informasi dan jasa meteorologi, klimatologi, kualitas udara dan geofisika yang handal dan terpercaya.

3.1.2 Struktur Organisasi

Struktur Organisasi BMKG Provinsi Bengkulu dapat dilihat di lampiran I.

3.1.3 Waktu dan Tempat Penelitian

Waktu penelitian dimulai dari 01 Nopember 2022 sampai dengan 31 April 2023. Penelitian dilakukan di BMKG Provinsi Bengkulu yang beralamat di jalan Depati Payung Negara, Komplek Bandar Udara Fatmawati Soekarno *Bengkulu*.Kel. Pekan Sabtu, Kec. Selebar, Kota *Bengkulu, Bengkulu* – 38213

3.2 Metode Penelitian

Dalam melakukan penelitian ini, penulis menggunakan metode terapan (*applied research*). Penelitian terapan dilakukan berkenaan dengan kenyataan-kenyataan praktis, penerapan, dan pengembangan ilmu pengetahuan yang dihasilkan oleh penelitian dasar dalam kehidupan nyata. Penelitian terapan berfungsi untuk mencari solusi tentang masalah-masalah tertentu. Tujuan utama penelitian terapan adalah pemecahan masalah sehingga hasil penelitian dapat dimanfaatkan untuk kepentingan manusia baik secara individu atau kelompok.

3.3 Software dan Hardware yang digunakan

Adapun perangkat lunak (*Software*) dan perangkat keras (*Hardware*) yang digunakan pada penelitian ini adalah sebagai berikut

3.3.1 Perangkat Lunak (*Software*)

- a. Sistem operasi Windows 10.
- b. Bahasa Pemrograman Visual Basic.Net 2010
- c. *Microsoft Office* 2010

3.3.2 Perangkat Keras (*Hardware*)

- a. Laptop Asus Core i3 Ram 4Gb
- b. *Mouse Optic*
- c. *Printer Cannon 2770*

3.4 Metode Pengumpulan Data

Adapun metode pengumpulan data yang penulis gunakan pada penelitian ini adalah sebagai berikut :

1. Observasi

Dalam pengumpulan data melalui observasi, penulis mengamati dan menganalisa bagaimana tahapan atau langkah-langkah dari metode *watermarking* dan kriptografi pada objek citra digital.

2. Studi Pustaka

Metode dimana penulis mempelajari dan mencari data yang berasal dari buku dan referensi yang berhubungan dengan masalah yang ditulis.

3.5 Analisa Perancangan Sistem

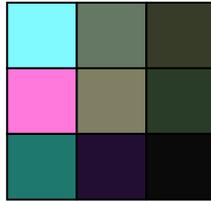
3.5.1 Analisa Sistem Aktual.

Dalam melakukan tugasnya, BMKG menghasilkan berbagai jenis citra seperti peta cuaca, peta gempa bumi, dan citra satelit. Citra-citra tersebut merupakan hasil dari pengolahan data yang telah dilakukan oleh BMKG Provinsi Bengkulu. Oleh karena itu, untuk menjaga keaslian dan keamanan data citra tersebut, perlu dilakukan tindakan preventif. *Watermarking* diterapkan untuk melindungi hak cipta dan keaslian data citra yang dihasilkan. BMKG memiliki tugas utama dalam melakukan pengamatan, pengolahan, dan penyajian data cuaca dan iklim serta gempa bumi dan tsunami yang terjadi. Data-data yang dihasilkan oleh BMKG memiliki nilai penting

dalam berbagai bidang seperti pertanian, transportasi, perikanan, dan keamanan nasional.

3.5.2 Analisa *Watermarking LCG*

Sebagai bahan analisis maka akan digunakan piksel potongan citra digital yang dapat dilihat pada gambar 3.1 berikut.



Gambar 3.1 Potongan Citra Sample

Potongan piksel citra sample seperti yang terlihat pada gambar 3.1 merupakan piksel dengan ruang warna RGB yang tiap piksel terdiri dari 24 bit. Berikut tabel warna RGB dari piksel citra sample.

P1(128,250,255)	P2(100,120,100)	P3(55,60,40)
P4(255,120,220)	P5(128,127,100)	P6(40,60,40)
P7(30,120,110)	P8(34,14,50)	P9(10,10,10)

Berikut proses dari transformasi warna HSV :

$$M = \max\{128,250,255\}$$

$$M = 255$$

$$m = \min\{128,250,255\}$$

$$m = 128$$

$$V = \frac{255}{255} = 1$$

$$S = 1 - \left(\frac{m}{M}\right) = 1 - \left(\frac{128}{255}\right) = 0.5$$

$$H = \text{Cos}^{-1} \left[\frac{R - 0.5G - 0.5B}{\sqrt{R^2 + G^2 + B^2 - RG - RB - GB}} \right] = 1.99$$

Maka diperoleh nilai HSV dari Piksel pertama adalah (1,0.5,1.99). Berikutnya adalah analisa penyisipan piksel kedalam ruang warna HSV yang dapat dilakukan seperti berikut :

Dimisalkan bit yang akan ditanam adalah $W = "1"$:

$$S' = ((WxV) + S) \text{ mod } 2\pi = ((1x1) + 0.5) = 1.5 \text{ mod } 1 = 0.5$$

Berikutnya adalah melakukan konversi balik menjadi ruang warna RGB dengan tahap sebagai berikut :

$$M = 255V = 255$$

$$m = M(1 - S) = 128$$

$$Z = \left| \left(\frac{H}{60} \right) \text{ mod } 2 - 1 \right| = 0.986$$

$$R = m = 128$$

$$G = m + z = 250$$

$$B = M = 255$$

Linear Congruential Generator merupakan pembangkit bilangan acak semu yang didasarkan pada kemunculan linear. Kelebihan utama dari *linear congruential generator* adalah kecepatan yang tinggi dan memiliki property statistic yang baik berdasarkan pemilihan parameter a, b, M . Contoh untuk membangkitkan bilangan acak menggunakan *linear congruential generator* adalah sebagai berikut.

Contoh Pembangkitan Bilangan Acak LCG :

Diketahui :

$$a = 5$$

$$b = 1$$

$$m = 16$$

$$X(0) = 1$$

Bilangan Acak Sebanyak 7 bilangan :

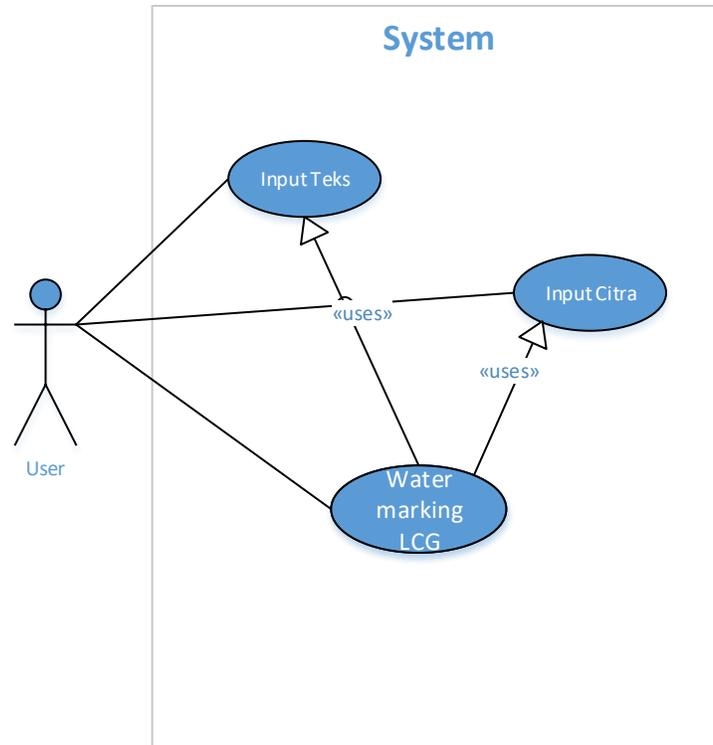
X	Formula	Random Number
1	$X_1 = (5.1 + 1) \text{ mod } 16$	6
2	$X_2 = (5.6 + 1) \text{ mod } 16$	15
3	$X_3 = (5.15 + 1) \text{ mod } 16$	12
4	$X_4 = (5.12 + 1) \text{ mod } 16$	13
5	$X_5 = (5.13 + 1) \text{ mod } 16$	2
6	$X_6 = (5.2 + 1) \text{ mod } 16$	11
7	$X_7 = (5.11 + 1) \text{ mod } 16$	8

3.5.3 Perancangan Sistem Baru

Perancangan sistem baru adalah suatu proses yang menggambarkan bagaimana suatu sistem dibangun untuk memenuhi kebutuhan pada fase analisis.

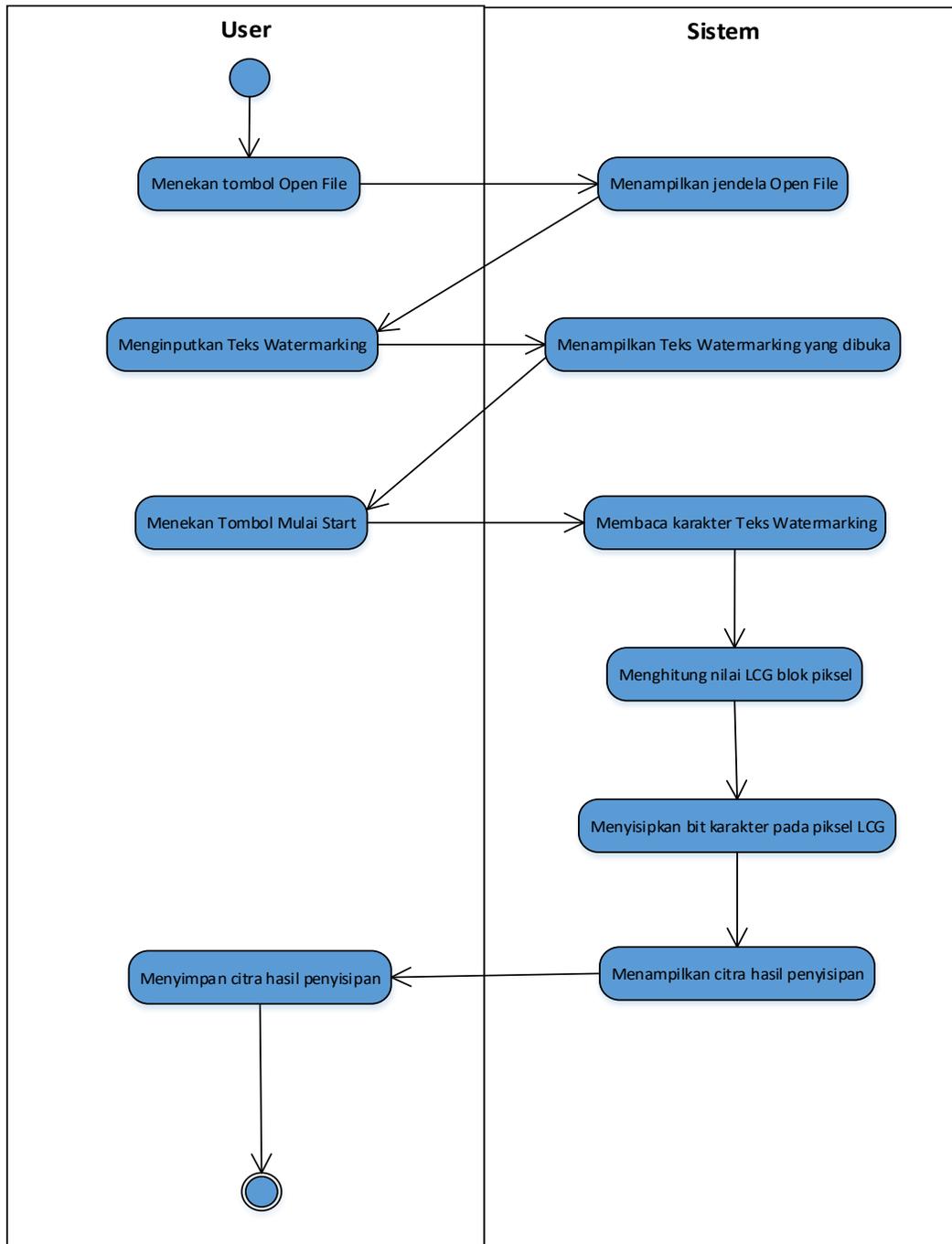
Perancangan sistem dilakukan dengan merancang *use case diagram*, *activity diagram*, diagram alir atau *flowchart* dan perancangan antar muka.

A. Use Case Diagram

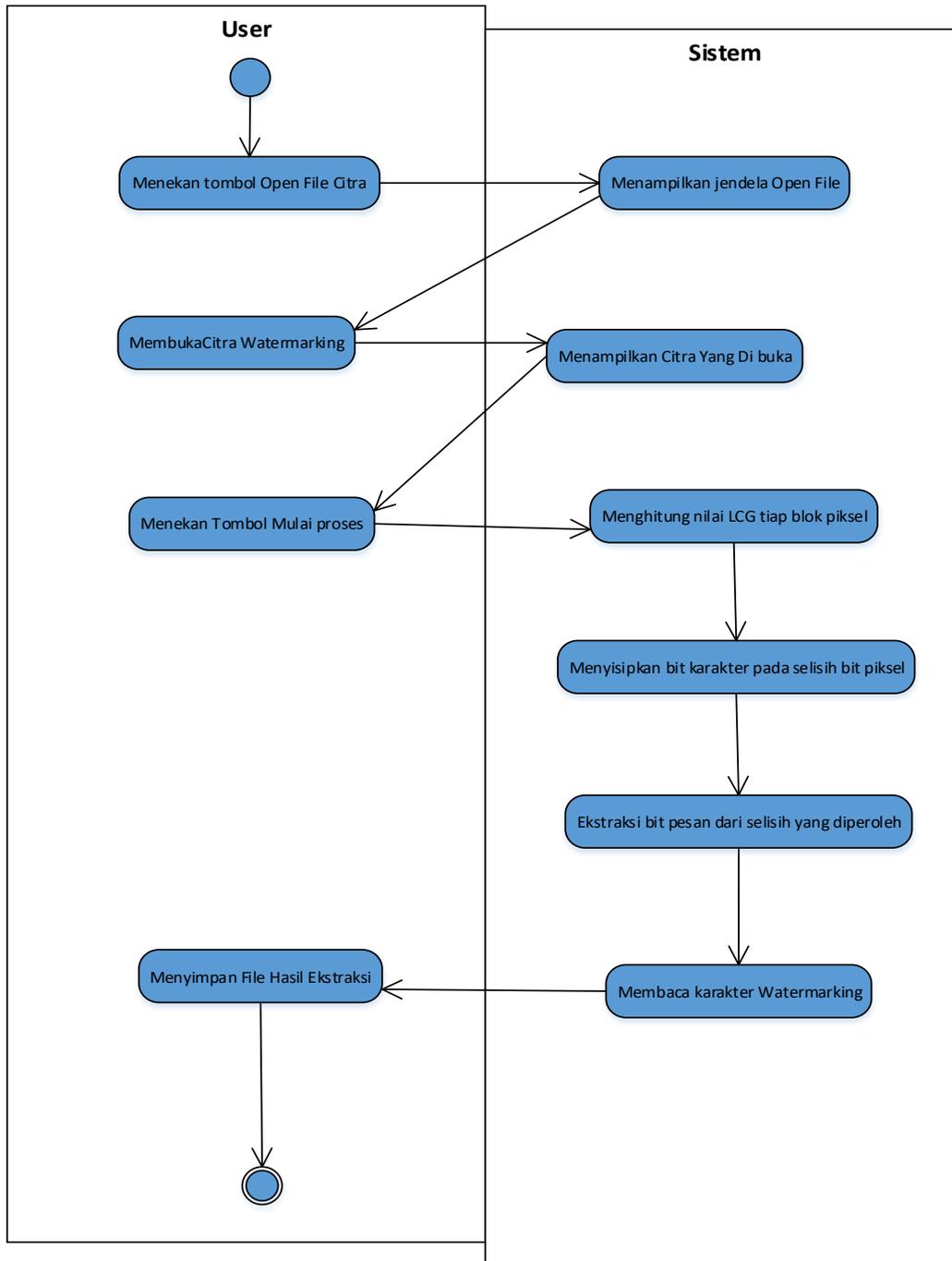


Gambar 3.2 Use Case Diagram

B. Acitivity Diagram



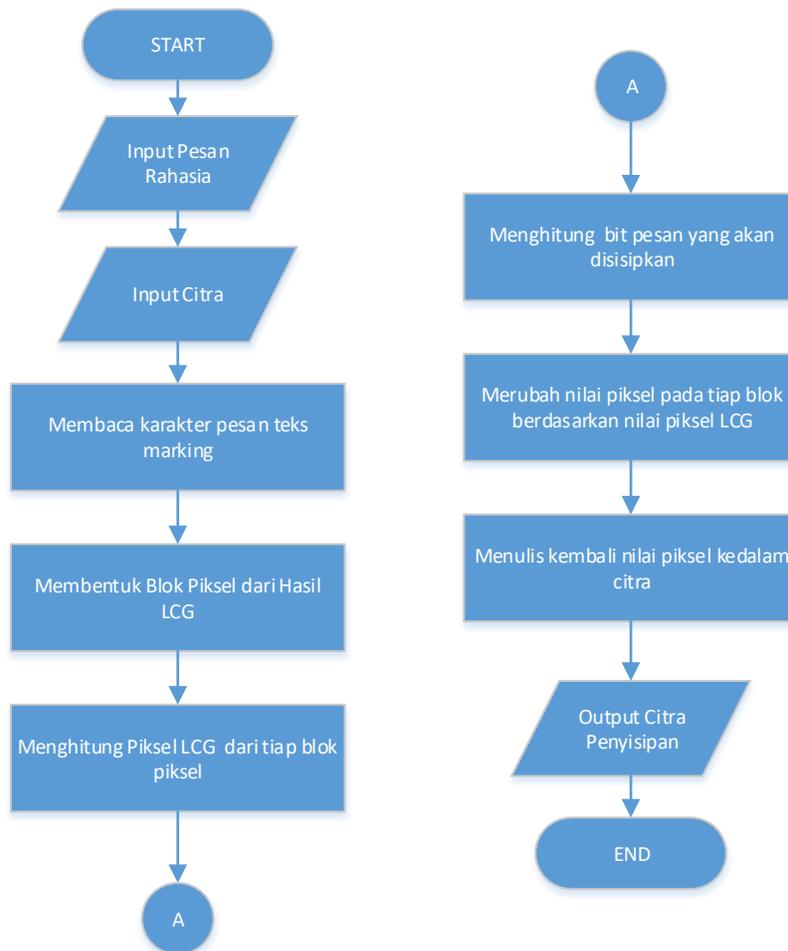
Gambar 3.3 Activity Diagram Penyisipan



Gambar 3.4 Activity Diagram Ekstraksi

C. Flowchart Sistem

Diagram alir yang dirancang pada penelitian ini membahas alur dari proses yang digunakan pada aplikasi yang dibangun. Adapun diagram *flowchart* sistem dapat dilihat pada gambar 3.5 berikut



Gambar 3.5 Flowchart Sistem

Keterangan :

Dari gambar 3.5 dapat dilihat gambaran umum aplikasi yang dibangun pada tugas akhir ini. Pengguna menginput data citra dan pesan yang akan disisipkan. Pesan pertama sekali dibaca dan konversi menjadi kode ASCII dan biner, hasil

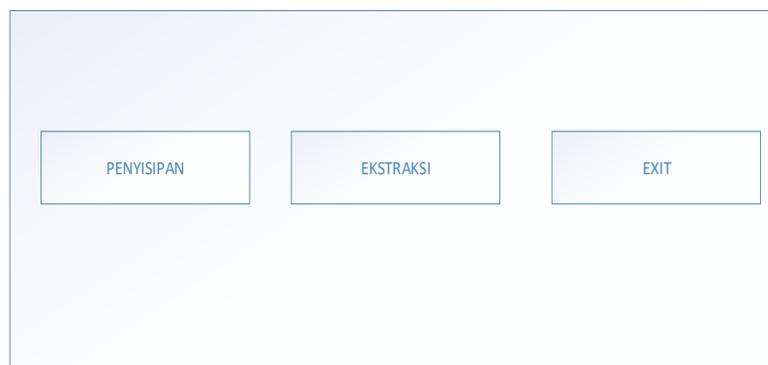
transformasi kemudian digunakan untuk menghitung nilai LCG dan penyisipan pada piksel citra penampung. Output citra yang telah disisipkan kemudian ditampilkan kepada pengguna

D. Rancangan Antarmuka Aplikasi

Sistem akan dibangun menggunakan bahasa pemrograman Visual Basic dengan menggunakan *software Microsoft Visual Studio 2010*. Rancangan antarmuka disesuaikan dengan kebutuhan dan *software* yang digunakan. Antarmuka menggunakan 3 *form*, yaitu : *form utama*, *form penyisipan*, dan *form ekstraksi*. *Form utama* berfungsi sebagai tampilan utama dari sistem dimana terdapat *menu utama* dan logo

1. Rancangan Form Menu Utama

Antarmuka yang pertama kali akan user gunakan pada saat menggunakan aplikasi ini adalah halaman utama. Pada halaman ini pengguna dapat memilih untuk melakukan proses penyisipan ataupun ekstraksi terlebih dahulu. Adapun rancangan dari antarmuka utama dapat dilihat pada gambar 3.6 berikut



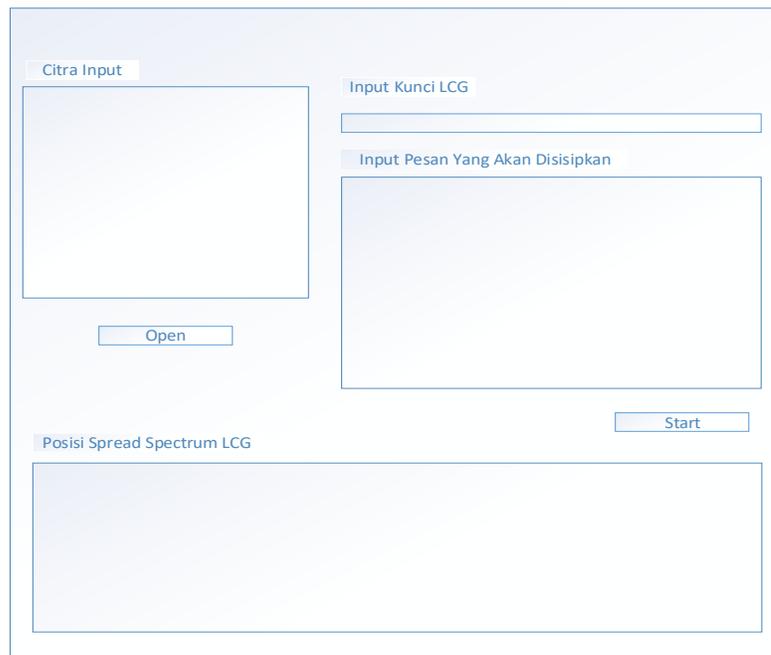
Gambar 3.6 Rancangan Form Utama

Komponen yang dipakai untuk membangun antar muka *form Utama* pada gambar 3.6 adalah sebagai berikut :

- a. *Tombol* “Penyisipan” : tombol yang digunakan untuk masuk kedalam halaman penyisipan
- b. *Tombol* “Ekstraksi” : tombol yang digunakan untuk masuk kedalam halaman ekstraksi
- c. *Tombol* “Exit” : untuk keluar dari aplikasi

2. Rancangan Antarmuka *Form Penyisipan*

Halaman penyisipan ini adalah halaman yang digunakan untuk menyisipkan marking pada citra dengan memanfaatkan algoritma LCG. Pada halaman ini disediakan semua kebutuhan pengguna untuk dapat menyisipkan watermarking pada citra. Berikut tampilan dari halaman penyisipan



The image shows a software interface for watermarking. It features a light blue background with several input fields and buttons. On the left, there is a large rectangular area labeled "Citra Input" with an "Open" button below it. On the right, there is a smaller rectangular area labeled "Input Kunci LCG" with a text input field below it, and another larger rectangular area labeled "Input Pesan Yang Akan Disisipkan" with a text input field below it. At the bottom right, there is a "Start" button. At the bottom left, there is a label "Posisi Spread Spectrum LCG" above a large empty rectangular area.

Gambar 3.7 Rancangan *Form Penyisipan*

Komponen yang dipakai untuk membangun antar muka *form* penyisipan pada gambar 3.7 adalah sebagai berikut :

- a. *Image 1* : control ini berfungsi untuk menampilkan citra yang akan di marking
- b. *Image 2* : control ini berfungsi untuk menampilkan marking yang akan di sisipan pada citra
- c. *Image 3* : control ini berfungsi untuk menampilkan piksel mana saja yang akan digunakan sebagai piksel penampung
- d. *Text1* : control ini menampung marking yang akan disisipkan pada citra penampung.
- e. Tombol “Open” : tombol yang berfungsi untuk membuka dialog untuk memilih citra penampung.
- f. Tombol “Start” : tombol yang berfungsi untuk memulai proses.
- g. Input Kunci LCG: disini akan dimasukkan kunci yang akan menjadi kunci *watermarking* LCG

3. Rancangan Antarmuka *Form* Ekstraksi

Halaman ekstraksi ini adalah halaman yang digunakan untuk mengembalikan hasil yang telah diterapkan pada penyisipan dengan memanfaatkan algoritma LCG.. Berikut tampilan dari halaman ekstraksi

The image shows a software interface with the following components:

- Citra Input**: A large rectangular area for displaying the input image.
- Open**: A button located below the 'Citra Input' area.
- Input Kunci LCG**: A text input field for entering the LCG key.
- Pesan Hasil Ekstraksi**: A large rectangular area for displaying the extraction results.
- Start**: A button located to the right of the 'Pesan Hasil Ekstraksi' area.
- Posisi Spread Spectrum LCG**: A large rectangular area at the bottom for displaying the LCG spread spectrum positions.

Gambar 3.8 Rancangan *Form* Ekstraksi

Komponen yang dipakai untuk membangun antar muka *form* ekstraksi pada gambar 3.8 adalah sebagai berikut :

1. *Image 1* : control ini berfungsi untuk menampilkan citra yang telah di marking
2. *Image 2* : control ini berfungsi untuk menampilkan marking yang di sisipan pada citra
3. *Image 3* : control ini berfungsi untuk menampilkan piksel mana saja yang akan digunakan sebagai piksel penampung
4. *Text1* : control ini menampung marking yang akan disisipkan pada citra penampung.
5. Tombol “Open” : tombol yang berfungsi untuk membuka dialog untuk memilih citra penampung.
6. Tombol “Start” : tombol yang berfungsi untuk memulai proses.

7. Input Kunci LCG : disini akan dimasukkan kunci yang akan menjadi kunci *watermarking* LCG

3.6 Perancangan Pengujian

Pengujian sistem merupakan proses eksekusi sistem dengan tujuan mencari kesalahan atau kelemahan dari program tersebut. Proses tersebut dilakukan dengan mengevaluasi kemampuan program. Suatu program yang diuji akan dievaluasi apakah keluaran atau output yang dihasilkan telah sesuai dengan yang diinginkan atau tidak. Metode pengujian yang dipakai dalam sistem ini adalah metode *black box*. Pengujian *black box* berfokus pada persyaratan fungsional perangkat lunak. Pengujian ini memungkinkan analisis sistem memperoleh kumpulan kondisi input yang akan mengerjakan seluruh keperluan fungsional program. Tujuan metode ini mencari kesalahan pada:

1. Fungsi yang salah atau hilang.
2. Kesalahan pada *interface*.
3. Kesalahan pada struktur data atau akses database.
4. Kesalahan performansi.
5. Kesalahan inisialisasi dan tujuan akhir