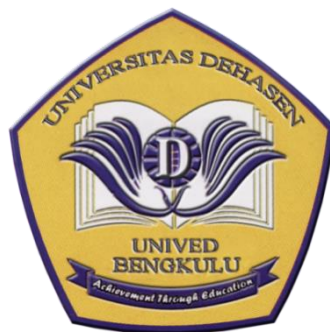


**PERANCANGAN APLIKASI STEGANOGRAFI PADA CITRA DIGITAL
MENGUNAKAN METODE PIXEL VALUE DIFFERENCING**

SKRIPSI



OLEH :

ANDRE GUSTIAWAN
NPM : 16010131

**PROGRAM STUDI INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS DEHASEN
BENGKULU
2023**

**PERANCANGAN APLIKASI STEGANOGRAFI PADA CITRA DIGITAL
MENGUNAKAN METODE PIXEL VALUE DIFFERENCING**

SKRIPSI

OLEH :

ANDRE GUSTIAWAN
NPM : 16010131

Diajukan Untuk Memperoleh Gelar Sarjana Komputer (S.Kom)
Pada Program Studi Informatika

**PROGRAM STUDI INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS DEHASEN
BENGKULU**

2023

**PERANCANGAN APLIKASI STEGANOGRAFI PADA CITRA DIGITAL
MENGUNAKAN METODE PIXEL VALUE DIFFERENCING**

SKRIPSI

OLEH :

ANDRE GUSTIAWAN
NPM : 16010131

Disetujui Oleh :

Pembimbing Utama



Ir. Jusuf Wahyudi, M.Kom
NIDN. 02.101060.01

Pembimbing Pendamping



Eko Suryana, S.Kom, M.Kom
NIDN. 02.151174.01

**Mengetahui,
Ketua Program Studi Informatika**



Liza Yulianti, S.Kom, M.Kom
NIDN : 02.160772.01

**PERANCANGAN APLIKASI STEGANOGRAFI PADA CITRA DIGITAL
MENGUNAKAN METODE PIXEL VALUE DIFFERENCING**

SKRIPSI




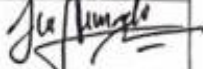
OLEH :

ANDRE GUSTIAWAN
NPM : 16010131

Telah dipertahankan di depan Tim Penguji Universitas Dehasen Bengkulu Pada :

Hari : Rabu
Tanggal : 14 juni 2023
Tempat : Ruang Sidang Universitas Dehasen Bengkulu


Skripsi Telah Diperiksa dan Disahkan Oleh :

Penguji	Nama	NIDN	Tanda Tangan
Ketua	Ir. Jusuf Wahyudi, M.Kom	02.101060.01	
Anggota	Eko Suryana, S.Kom, M.Kom	02.151174.01	
Anggota	Sapri, S.Kom, M.Kom	02.150171.02	
Anggota	Juju Jumadi, S.Kom, M.Kom	02.111282.01	

Mengetahui,

Dekan
Fakultas Ilmu Komputer




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

DAFTAR RIWAYAT HIDUP



Andre Gustiawan Lahir di desa Parada Suka, Provinsi Bengkulu Kaur pada tanggal 09 April 1998. Penulis lahir dari pasangan Burlin dan Pirma Sismia merupakan anak Pertama dari tiga bersaudara yakni Agil Dwi Runaldo, dan Alqi Cahya Linara.

Penulis menempuh Pendidikan pertama pada tahun 2006 di SDN 01 desa Kedataran, Kaur dan selesai pada tahun 2011. Pada tahun yang sama penulis melanjutkan Pendidikan di SMPN 03 Maje, Provinsi Bengkulu dan selesai pada tahun 2014. Pada tahun yang sama Penulis melanjutkan Pendidikan di SMAN 1 Kaur, Provinsi Bengkulu dan selesai pada tahun 2016. Pada Tahun 2016 Penulis Terdaftar di salah satu Perguruan Tinggi Swasta Jurusan Informatika Universitas Dehasen Bengkulu dan Alhamdulillah selesai Tahun 2023. Berkat Pertolongan dan Petunjuk Allah SWT, Usaha disertai Doa dan kedua orang tua dalam menjalani aktivitas akademik di perguruan tinggi Universitas Dehasen Bengkulu. Alhamdulillah Penulis Dapat Menyelesaikan tugas akhir Skripsi yang berjudul "PERANCANGAN APLIKASI STEGANOGRAFI PADA CITRA DIGITAL MENGGUNAKAN METODE PIXEL VALUE DIFFERENCING"

MOTTO DAN PERSEMBAHAN.

MOTTO :

Cara terbaik untuk memulai adalah diam dan mulai bekerja.

- ❖ *Jangan pernah menyesali sehari dalam hidupmu. Hari-hari baik memberimu kebahagiaan, hari yang buruk memberimu pengalaman, hari yang sia-sia memberikan pelajaran, dan hari terbaik memberikan kenangan*
- ❖ *Ketika kamu fokus pada masalah, kamu akan mendapatkan lebih banyak masalah, dan ketika kamu fokus pada kemungkinan, kamu akan mendapat banyak kesempatan*
- ❖ *Only stupidity that underestimates Education (Hanya kebodohan yang meremehkan pendidikan)*

PERSEMBAHAN :

Dengan rasa syukur yang mendalam kupersembahkan Skripsi ini untuk :

- ❖ Bangunku, Tidurku, Sujudku, Do'a ku, bahagiaku dan sedihku aku tujukan kepada Allah SWT, yang selalu melindungi dan menerangi tiap jalanku
- ❖ Ayahnda Burlin dan Ibunda Pirma Sismia yang tercinta terima kasih telah membesarkan, menyayangi, dan selalu memberikan motivasi, support serta contoh terbaik dalam membentuk karakterku, terima kasih untuk segala do'a yang telah kalian berikan padaku. Dan terima kasih juga dalam membiayai studiku, meskipun semua itu dilakukan tidak mudah, tapi kalian mampu melakukannya dengan penuh pengorbanan yang luar biasa.
- ❖ Ibu Liza Yulianti, S.Kom, M.kom dan ibu Devi Sartika, S.Kom, M.Kom yang memberikan bimbingan dan masukan kepada saya dalam menyelesaikan Skripsi
- ❖ Untuk Istriku Anisti Mamela, S.Sos Terima kasih untuk semangat serta support terbaik. terima kasih telah menemani setiap proses sidang Skripsi ini

- ❖ Adekku, Alqi Cahya Linara Dan Agil Dwi Runaldo terima kasih untuk support dan semangat serta bantuan yang tiada hentinya untukku
- ❖ Sanak dan Saudara yang telah memberikan dukungan dan Motivasi
- ❖ Sahabat Sahabat yang selalu mendukungku dalam menyelesaikan Skripsi ini. Salam sehat dan sukses untuk kita semua.
- ❖ Almamater kuning yang saya banggakan

**SURAT PERNYATAAN ORISINILITAS DAN PERSETUJUAN
AKADEMIK SKRIPSI**

Yang bertanda tangan dibawah ini :

Nama : Andre Gustiawan
NPM : 16010131
Program Studi : Informatika
Fakultas : Ilmu Komputer

Dengan ini menyatakan dengan sesungguhnya bahwa Skripsi dengan Judul :

**PERANCANGAN APLIKASI STEGANOGRAFI PADA CITRA DIGITAL
MENGUNAKAN METODE PIXEL VALUE DIFFERENCING**

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 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, Mei 2023
Hormat Saya



Andre Gustiawan

Andre Gustiawan

KATA PENGANTAR

Puji Syukur saya panjatkan kehadiran Allah SWT yang telah memberikan rahmat dan karunia-NYA, sehingga skripsi yang berjudul “**Perancangan Aplikasi Steganografi Pada Citra Digital Menggunakan Metode Pixel Value Differencing**” 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 (UNIVED) 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 Ir. Jusuf Wahyudi, M.Kom selaku pembimbing I yang telah membimbing dengan sabar dan memberikan masukan serta saran kepada penulis
5. Bapak Eko Suryanan, S.Kom, M.Kom Selaku pembimbing II yang telah memberikan masukan dan saran kepada penulis.
6. 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, Februari 2023

Penulis

ABSTRAK

PERANCANGAN APLIKASI STEGANOGRAFI PADA CITRA DIGITAL MENGUNAKAN METODE PIXEL VALUE DIFFERENCING

Oleh :

Andre Gustiawan¹

Ir. Jusuf Wahyudi²

Eko Suryana²

Saat ini keamanan data sangat penting terutama keamanan pada bidang sistem informasi, hal ini dikarenakan pengguna komputer pada kehidupan setiap hari telah menjadi kebutuhan utama dalam kegiatan mengelola data dan informasi, dimana kegiatan transaksi data ataupun penyimpanan data sangatlah penting untuk dijaga keamanannya. Salah satu teknik untuk melindungi data dan informasi adalah dengan menggunakan steganografi *Pixel Value Differncing* (PVD). Algoritma ini bekerja dengan cara membagi citra menjadi blok-blok, dimana setia blok terdiri dari dua *pixel* yang bertetangga secara horizontal. Metode penelitian yang digunakan adalah menggunakan metode terapan (*applied research*).

Hasil dari penelitian ini adalah proses Steganografi menggunakan metode *Pixel Value Differencing* dapat bekerja dengan baik dimana proses penyisipan dan ekstraksi dapat berjalan dengan baik dan optimal.

Kata kunci : Keamanan Informasi, Steganografi , *Pixel Value Differncing*

1. Mahasiswa
2. Pembimbing

DAFTAR ISI

	Halaman
COVER DEPAN	i
HALAMAN JUDUL	ii
HALAMAN PERSETUJUAN	iii
HALAMAN PENGESAHAN	iv
RIWAYAT HIDUP	v
MOTTO DAN PERSEMBAHAN	vi
PERNYATAAN	vii
KATA PENGANTAR	ix
ABSTRAK	xi
ABSTRACT	xii
DAFTAR ISI	xiii
DAFTAR GAMBAR	xv
DAFTAR TABEL	xvi
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	3
1.5 Manfaat Penelitian.....	4
BAB II LANDASAN TEORI	
2.1 Pengertian Aplikasi	5
2.2 Pengertian Steganografi.....	5
2.2.1 Kriteria Steganografi	7

2.2.2	Teknik Steganografi	8
2.3	Pengertian Pixel Value Differencing (PVD)	9
2.4	Citra Digital	12
2.4.1	Pengertian Citra Digital	12
2.4.2	Jenis Citra	14
2.5	Tinjauan Umum Visual Basic.Net 2010	16
2.5.1	Lingkungan Kerja Visual Studio 2010	18
2.6	Pengertian UML	21
2.7	Simbol dan Fungsi Flowchart.....	26

BAB III METODOLOGI PENELITIAN

3.1	Waktu dan Tempat Penelitian	28
3.2	Metode Penelitian	28
3.3	Perangkat Keras dan Perangkat Lunak.....	29
3.4	Metode Pengumpulan Data	29
3.5	Analisa Perancangan Sistem.....	30
3.5.1	Analisa Sistem Aktual	30
3.5.2	Analisa Kebutuhan Sistem.....	30
3.5.3	Analisa Steganografi Pada Citra Digital Menggunakan Metode <i>Pixel Value Differencing</i>	31
3.5.4	Analisa Perancangan Sistem Baru	37
3.6	Perancangan Pengujian.....	46

BAB IV HASIL DAN PEMBAHASAN

4.1	Pembahasan Dan Implementasi	47
4.1.1	Sistem Aplikasi.....	47
4.1.2	Implementasi Sistem	48
4.2	Pengujian Sistem.....	57

BAB V KESIMPULAN DAN SARAN

1.1	Hasil Aplikasi.....	59
1.2	Saran.....	60

DAFTAR PUSTAKA
TIME SCHEDULE
LAMPIRAN

DAFTAR GAMBAR

Gambar	Halaman
2.1 Ilustrasi Citra Digital	13
2.2 Citra Biner	15
2.3 Citra Perbandingan Gradasi Warna 1 bit, 2 bit, 5 bit, 6	15
2.4 8 Bit true color	16
2.5 Tampilan Halaman Muka Visual Basic 2010	18
2.6 Lingkungan kerja Visual Basic 2010	19
2.7 Toolbox	19
2.8 Solution Explorer	20
2.9 Properties	20
2.10 Toolbar	21
3.1 Proses Penyisipan Pesan	32
3.2 Nilai Pixel Citra Penampung	32
3.3 Nilai Pixel Citra Setelah Disisipi Pesan	35
3.4 Proses Ekstraksi Pesan	35
3.5 Nilai Pixel Citra yang Telah Disisipi Pesan	35
3.6 Use Case Diagram	38
3.7 Activity Diagram Penyisipan	40
3.8 Activity Diagram Ekstraksi	41
3.9 Flowchart Sistem	42
3.10 Rancangan Interface Menu Utama	43
3.11 Rancangan Interface Penyisipan Pesan	44
3.12 Rancangan Interface Ekstrak Pesan	45
4.1 Sistem Steganografi PVD	48
4.2 Menu Penyisipan	49
4.3 Tombol “Open”	50
4.4 Dialog Memilih Citra Penyisipan	50
4.5 Proses Penyisipan	51
4.6 Hasil Penyisipan	51

4.7	Dialog Simpan Hasil Penyisipan	52
4.8	Perbandingan Citra Asli (a) dan Citra Hasil Penyisipan (b).....	53
4.9	Menu Ekstraksi	54
4.10	Menu Open Ekstraksi	54
4.11	Dialog Memilih Citra Penyisipan	55
4.12	Tombol “Unhide” Ekstraksi	56
4.13	Proses Ekstraksi	56

DAFTAR TABEL

Tabel	Halaman
2.1 Nilai Continues Range.....	10
2.2 Notasi Activity Diagram.....	22
2.3 Simbol Use Case Diagram.....	23
2.4 Simbol Class Diagram	25
2.5 Simbol dan Fungsi Flowchart.....	26
3.1 Perangkat Keras dan Perangkat Lunak.....	29
3.2 Nilai Continues Range Dari Nilai Difference Value	32

BAB I

PENDAHULUAN

1.1 Latar Belakang

Kemajuan teknologi informasi dan ilmu komputer berperan penting pada kehidupan manusia. Dari hal yang sederhana sampai berbagai hal yang sangat kompleks dan rumit sekalipun bisa dikerjakan menggunakan teknologi komputer. Saat ini keamanan data sangat penting terutama keamanan pada bidang sistem informasi, hal ini dikarenakan pengguna komputer pada kehidupan setiap hari telah menjadi kebutuhan utama dalam kegiatan mengelola data dan informasi, dimana kegiatan transaksi data ataupun penyimpanan data sangatlah penting untuk dijaga keamanannya.

Berbagai teknik telah banyak digunakan untuk melindungi data-data penting tersebut, di antara salah satunya adalah steganografi. Steganografi merupakan seni menyembunyikan pesan ke dalam pesan lainnya sedemikian rupa sehingga orang lain tidak menyadari ada sesuatu di dalam pesan tersebut. Kata steganografi berasal dari bahasa Yunani yaitu *steganos* yang artinya tersembunyi atau terselubung dan *graphein*, yang artinya menulis, sehingga kurang lebih artinya adalah “menulis tulisan yang tersembunyi atau terselubung”. Steganografi berbeda dengan kriptografi atau metode keamanan informasi lainnya, metode ini yaitu menyembunyikan informasi atau pesan kedalam media lain seperti citra digital, teks, suara atau video sehingga tidak menimbulkan kecurigaan orang lain. Steganografi membutuhkan dua properti, yaitu informasi dan media penampung.

Media penampung yang banyak digunakan untuk menyembunyikan informasi yaitu citra digital. penyisipan informasi pada media citra digital dilakukan pada bit –bit pixel yang terdapat pada citra. Penggunaan citra digital sebagai media penampung mempunyai kelebihan karena indera penglihatan manusia memiliki keterbatasan terhadap warna, sehingga dengan keterbatasan tersebut manusia sulit membedakan citra digital yang asli dengan citra digital yang telah disisipi pesan rahasia.

Seiring beragamnya media yang digunakan pada steganografi, makin banyak pula metode steganografi yang dapat digunakan seperti EOF (*End Of File*), *MSB (Most Significant Bit)*, *LSB (Least Significant Bit)*, *Spread Spectrum*, *Discrete Cosine Transform (DCT)*, *Discrete Wavelet Transform (DWT)*, *Bit-Plane Complexity Segmentation (BPCS)* (Ratnasari & Dwiyanto, 2020). Selain itu juga ada metode *Pixel Value Differncing (PVD)*, metode ini bekerja dengan cara membagi citra menjadi blok-blok, dimana setia blok terdiri dari dua *pixel* yang bertetangga secara horizontal. Pada penelitian tentang steganografi telah banyak dilakukan sebelumnya yaitu membahas steganografi untuk menyisipkan pesan teks ke dalam gambar dengan menggunakan metode *LSB (Least Significant Bit)*. Metode *LSB* merupakan salah satu metode yang digunakan dalam teknik steganografi dengan proses kerja mengubah pesan ke dalam bentuk bilangan biner yang kemudian di urutkan nilai *pixel* gambar sesuai dengan urutan dari yang terkecil (Sitorus, 2015)

Berdasarkan uraian latar belakang diatas, maka penulis tertarik untuk melakukan penelitian yang dituangkan dalam proposal skripsi yang diberi

judul “ **Perancangan Aplikasi Steganografi Pada Citra Digital Menggunakan Metode *Pixel Value Differencing* (PVD).**”

1.2 Rumusan Masalah

Adapun rumusan masalah yang dibahas pada proposal skripsi ini adalah bagaimana merancang dan mengimplementasikan metode *Pixel Value Differencing* (PVD) pada objek citra digital?

1.3 Batasan Masalah

Dalam proposal skripsi ini, penulis menentukan batasan-batasan pembahasan sehingga nantinya pembahasan tidak menjadi terlalu luas dan menjadi tidak jelas. Berikut batasan-batasan yang penulis susun dalam proposal skripsi ini :

1. Pembahasan dibatasi pada metode Steganografi PVD pada pesan teks pada citra digital saja.
2. Aplikasi dibangun dengan menggunakan Visual Studio.Net 2010.

1.4 Tujuan Penelitian

1.4.1 Tujuan Umum

Adapun tujuan umum dari penelitian ini adalah sebagai salah satu syarat menyelesaikan kurikulum pendidikan Strata Satu (S-1) pada Program Studi Informatika Fakultas Ilmu Komputer Universitas Dehasen (UNIVED) Bengkulu

1.4.2 Tujuan Khusus

Tujuan yang ingin dicapai adalah untuk memahami bagaimana teknik steganografi dengan menggunakan metode *PVD*,

serta bagaimana teknik rekonstruksi kembali steganografi dengan metode yang sama.

1.5 Manfaat Penelitian

Manfaat dari penelitian skripsi ini adalah :

1. Memahami bagaimana cara kerja metode Steganografi PVD dalam pengamanan data.
2. Memberikan analisis hasil penggunaan untuk membantu pemilihan dan referensi penggunaan algoritma penyembunyian dan pengamanan data.
3. Menambah pengetahuan dan referensi dalam steganografi

BAB II

LANDASAN TEORI

2.1 Pengertian Aplikasi

Aplikasi adalah satu unit perangkat lunak yang dibuat untuk melayani kebutuhan akan beberapa aktivitas seperti sistem perniagaan, game, pelayanan masyarakat, periklanan, dan hampir semua proses kegiatan (Rohayah, dkk, 2015). Yang dimaksud dengan perangkat lunak adalah sub kelas perangkat lunak komputer yang memanfaatkan kemampuan komputer langsung untuk melakukan tugas yang diinginkan pengguna. Contoh utama dari perangkat lunak aplikasi adalah pengolah kata, lembar kerja dan pemutar media.

Aplikasi adalah program siap pakai yang dapat digunakan untuk menjalankan perintah-perintah dari pengguna aplikasi tersebut dengan tujuan mendapatkan hasil yang lebih akurat sesuai dengan tujuan pembuatan aplikasi tersebut, aplikasi mempunyai arti yaitu pemecahan masalah yang menggunakan salah satu teknik pemrosesan data aplikasi yang biasanya berpacu pada sebuah komputasi yang diinginkan atau diharapkan maupun pemrosesan data yang diharapkan (Lestari, dkk, 2017).

2.2 Pengertian Steganografi

Steganografi merupakan seni menyembunyikan pesan ke dalam pesan lainnya sedemikian rupa sehingga orang lain tidak menyadari ada sesuatu di dalam pesan tersebut. Kata steganografi (steganography) berasal dari bahasa

Yunani yaitu *steganos* yang artinya tersembunyi atau terselubung dan *graphein* yang artinya menulis, sehingga kurang lebih artinya adalah "menulis tulisan yang tersembunyi atau terselubung (Manulu, 2018).

Kata steganografi (*steganography*) berasal dari bahasa Yunani yaitu *steganos* yang artinya tersembunyi atau terselubung dan *graphein*, yang artinya menulis, sehingga kurang lebih artinya adalah “menulis tulisan yang tersembunyi atau terselubung”. Secara umum steganografi adalah ilmu dan seni menyembunyikan pesan rahasia sedemikian sehingga keberadaan pesan tidak terdeteksi oleh indera manusia (Hafiz, 2019)

Steganografi adalah seni dan ilmu menulis pesan tersembunyi atau menyembunyikan pesan dengan suatu cara sehingga selain si pengirim dan si penerima, tidak ada seorangpun yang mengetahui atau menyadari bahwa ada suatu pesan rahasia (Simbolon, 2021). Contoh singkat yang dapat dijelaskan adalah pengirim memilih gambar yang ingin dititipi informasi, kemudian mengatur setiap warna *pixel* ke-100 untuk menyesuaikan huruf dalam alphabet. Setiap perubahan yang terjadi diamati dan dicatat, perubahan yang begitu halus atau tanpa cacat dalam kondisi fisik gambar tersebut sehingga tidak ada pihak lain yang menyadarinya bahwa didalam gambar tersebut terdapat sebuah informasi yang bersifat rahasia jika pihak lain tidak benar-benar memperhatikannya menggunakan aplikasi untuk mendeteksi informasi tersembunyi dalam gambar digital

Steganografi sudah dikenal oleh bangsa Yunani. Penguasa Yunani dalam mengirimkan pesan rahasia menggunakan kepala budak atau prajurit sebagai media. Dalam hal ini, rambut budak dicukur habis, lalu pesan

rahasia ditulis di kulit kepala budak tersebut. Ketika rambut budak tumbuh, budak tersebut diutus untuk membawa pesan rahasia di kepalanya. Lain halnya dengan bangsa Yunani, bangsa Romawi mengenal steganografi dengan menggunakan tinta tak tampak untuk menuliskan pesan. Tinta tersebut dibuat dari campuran sari buah, susu dan cuka. Jika tinta digunakan untuk menulis maka tulisannya tidak tampak. Tulisan di atas kertas tersebut dapat dibaca dengan cara memanaskan kertas tersebut (Ichsan & Windarto, 2017)

2.2.1 Kriteria Steganografi

Adapun kriteria steganografi adalah sebagai berikut :

1. *Imperceptibility*, keberadaan pesan rahasia tidak dapat dipersepsi oleh indera. Misalnya, jika *covertext* berupa citra, maka penyisipan pesan membuat citra *stegotext* sukar dibedakan oleh mata dengan citra *covertext*-nya. Jika *covertext* berupa audio, maka indera telinga tidak dapat mendeteksi perubahan pada audio *stegotext*-nya.
2. *Fidelity*, mutu stegomedium tidak berubah banyak akibat penyisipan. Perubahan tersebut tidak dapat dipersepsi oleh inderawi. Misalnya, jika *covertext* berupa citra, maka penyisipan pesan membuat citra *stegotext* sukar dibedakan oleh mata dengan citra *covertext*-nya. Jika *covertext* berupa audio, maka audio *stegotext* tidak rusak dan indera telinga tidak dapat mendeteksi perubahan tersebut.

3. *Recovery*, pesan yang disembunyikan harus dapat diungkapkan kembali. Karena tujuan steganografi adalah data *hiding*, maka sewaktu-waktu pesan rahasia di dalam stegotext harus dapat diambil kembali untuk digunakan lebih lanjut.

2.2.2 Teknik Steganografi

Ada tujuh teknik dasar yang digunakan dalam steganografi (Nurmaesah,dkk, 2017) yaitu :

1. *Injection*, merupakan suatu teknik menanamkan pesan rahasia secara langsung ke suatu media. Salah satu masalah dari teknik ini adalah ukuran media yang diinjeksi menjadi lebih besar dari ukuran normalnya sehingga mudah dideteksi. Teknik ini sering juga disebut *embedding*.
2. Substitusi, data normal digantikan dengan data rahasia. Biasanya, hasil teknik ini tidak terlalu mengubah ukuran data asli, tetapi tergantung pada file media dan data yang akan disembunyikan. Teknik substitusi bisa menurunkan kualitas media yang ditumpangi
3. Transform Domain, teknik ini sangat efektif. Pada dasarnya, transformasi domain menyembunyikan data pada transform space. Akan sangat lebih efektif teknik ini diterapkan pada file berekstensi JPG
4. *Spread Spectrum*, sebuah teknik pengtransmisian menggunakan pseudo-noise code, yang independen terhadap data informasi sebagai modulator bentuk gelombang untuk menyebarkan energi

sinyal dalam sebuah jalur komunikasi (bandwidth) yang lebih besar daripada sinyal jalur komunikasi informasi. Oleh penerima, sinyal dikumpulkan kembali menggunakan replika *pseudo-noise code* tersinkronisasi

5. *Statistical Method*, teknik ini disebut juga skema steganographic 1bit. Skema tersebut menanamkan satu bit informasi pada media tumpangan dan mengubah statistik walaupun hanya 1 bit. Perubahan statistik ditunjukkan dengan indikasi 1 dan jika tidak ada perubahan, terlihat indikasi 0. Sistem ini bekerja berdasarkan kemampuan penerima dalam membedakan antara informasi yang dimodifikasi dan yang belum
6. *Distortion*, metode ini menciptakan perubahan atas benda yang ditumpangi oleh data rahasia
7. *Cover Generation*, metode ini lebih unik daripada metode lainnya karena cover object dipilih untuk menyembunyikan pesan. Contoh dari metode ini adalah Spam Mimic.

2.3 Pengertian *Pixel Value Differencing (PVD)*

Metode Pixel Value Differencing merupakan salah satu metode yang dapat digunakan dalam pembuatan steganografi. Metode ini menawarkan kapasitas penyimpanan pesan yang lebih besar, dengan kualitas gambar yang lebih baik dibandingkan dengan metode lainnya. (Nofriansyah & Rahim, 2016). Metode *Pixel Value Differencing* membagi citra menjadi blok-blok, dimana setiap blok terdiri dari dua *pixel* yang bertetangga secara horizontal.

Proses penyisipan pada metode ini dilakukan dengan cara membandingkan dua *pixel* yang bertetangga dengan persamaan sebagai berikut :

$$d = |P_i + P_{i+1}| \dots \dots \dots (1)$$

Hasil dari perbandingan tersebut digunakan untuk mengetahui berapa banyak bit yang dapat disisipkan kedalam dua *pixel* yang dibandingkan.

Metode ini menggunakan skema Wu dan Tsai untuk mengetahui nilai *Continues Range* dari perbandingan pixel sebelumnya. Skema Wu dan Tsai yang digunakan dapat dilihat pada tabel berikut :

Tabel 2.1 Nilai Continues Range

Kuantitasi ke-k	Batas bawah – Batas Atas (l_k-u_k)	Rentang Nilai	Jumlah Bit n
1	0-7	8	3
2	8-15	8	3
3	16-31	16	4
4	32-63	32	5
5	64-127	64	6
7	128-255	127	7

Skema ini digunakan untuk mengetahui terdapat di range mana selisih dari dua *pixel* tersebut, jika telah diketahui dimana letak range nya, maka jumlah bit pesan yang disisipkan dapat diketahui dengan persamaan berikut

$$t = \lfloor \log_2 w_i \rfloor \dots \dots \dots (2)$$

w_i = Nilai terkecil dari skema wu dan tsai, letak range selisih perbandingan dua *pixel*

Penyisipan pesan dapat dilakukan dengan mengambil sebanyak t bit dari pesan yang akan disisipkan. Selanjutnya dihitung nilai *difference value* yang baru untuk penyisipan kedalam citra menggunakan persamaan sebagai

berikut :

$$d'_i = l_i + b \dots \dots \dots (3)$$

Untuk menyisipkan pesan ada beberapa aturan yang harus dipenuhi yaitu :

1. Jika $P_i \geq P_{i+1}$ dan $d'_i > d_i$, maka $(P'_i + [m/2], P_{i+1} - [m/2])$
2. Jika $P_i < P_{i+1}$ dan $d'_i > d_i$, maka $(P'_i - [m/2], P_{i+1} - [m/2])$
3. Jika $P_i \geq P_{i+1}$ dan $d'_i \leq d_i$, maka $(P'_i - [m/2], P_{i+1} - [m/2])$
4. Jika $P_i < P_{i+1}$ dan $d'_i \leq d_i$, maka $(P'_i + [m/2], P_{i+1} - [m/2])$

Dimana m didapat dari selisih d'_i dengan d_i , dengan menggunakan persamaan sebagai berikut :

$$m = |d'_i \leq d_i| \dots \dots \dots (4)$$

Proses-proses tersebut dilakukan terus hingga bit pesan tersisipi semuanya kedalam citra. Proses ekstraksi pesan dari citra stego menggunakan metode ini dimulai dengan menghitung nilai *difference value* (di) antara dua *pixel* yang bertetangga. Nilai *difference value* tersebut digunakan untuk mengetahui nilai *continuous ranges* (R) yang sudah didefinisikan menggunakan skema wu dan tsai.

Berdasarkan informasi tersebut dapat diketahui ukuran data rahasia yang disisipkan pada kedua *pixel* menggunakan persamaan (2), sehingga pesan rahasia yang telah disisipkan didapatkan kembali. Proses ekstraksi ini dilakukan sampai semua data rahasia yang telah disisipkan didapatkan kembali.

2.4 Citra Digital

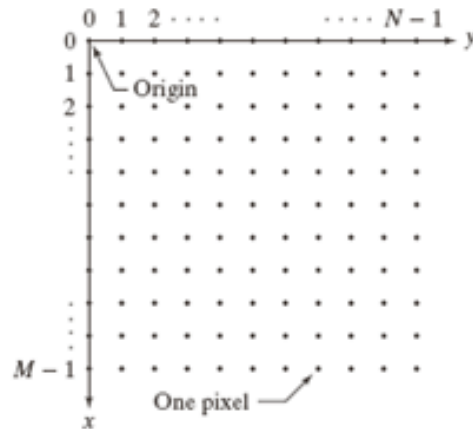
2.4.1 Pengertian Citra Digital

Citra atau gambar dapat didefinisikan sebagai fungsi dua dimensi $f(x,y)$, dimana x dan y adalah kordinat bidang datar; dan harga fungsi f di setiap pasangan kordinat (x,y) disebut intensitas atau level keabuan (greylevel) dari gambar di titik itu. Apabila nilai x,y dan f secara keseluruhan berhingga (*finite*) dan bernilai diskrit maka dapat dikatakan bahwa citra tersebut adalah citra digital (Jumadi, dkk 2020).

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 pemilihan ciri citra (*feature extraction*) yang optimal untuk bertujuan analisis, melakukan proses penarikan (Sindar & Sinaga, 2017).

Pengolahan citra merupakan cabang ilmudalam *Artificial Intelegence* 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 (Jumadi, dkk, 2021)

Citra digital dibentuk oleh kumpulan titik yang dinamakan piksel (*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



Gambar 2.1 Ilustrasi Citra Digital

Sebuah citra digital dapat diwakili oleh sebuah matriks yang terdiri dari M kolom dan N baris, di mana perpotongan antara kolom dan baris disebut piksel, yaitu elemen terkecil dari sebuah citra. Piksel mempunyai dua parameter, yaitu koordinat dan intensitas atau warna. Nilai yang terdapat pada koordinat (x,y) adalah $f(x,y)$, yaitu besar intensitas atau warna dari piksel di titik itu. Oleh sebab itu, sebuah citra digital dapat ditulis dalam bentuk matriks berikut

$$f(x,y) = \begin{bmatrix} f(0,0) & \cdots & f(0,M-1) \\ \vdots & \ddots & \vdots \\ f(N-1,0) & \cdots & f(N-1,M-1) \end{bmatrix}$$

Berdasarkan gambaran tersebut, secara matematis citra digital dapat dituliskan sebagai fungsi intensitas $f(x,y)$, di mana harga x (baris) dan y (kolom) merupakan koordinat posisi dan $f(x,y)$ adalah

nilai fungsi pada setiap titik (x,y) yang menyatakan besar intensitas citra atau tingkat keabuan atau warna dari piksel di titik tersebut. Pada proses digitalisasi (sampling dan kuantisasi) diperoleh besar baris M dan kolom N hingga citra membentuk matriks $M \times N$ dan jumlah tingkat keabuan piksel. Pada kebanyakan kasus, terutama untuk keperluan penampilan secara visual, nilai data visual tersebut merepresentasikan warna dari citra yang diolah, dengan demikian format data citra digital berhubungan erat dengan warna.

2.4.2 Jenis Citra

Nilai suatu pixel memiliki nilai dalam rentang tertentu, dari nilai minimum sampai nilai maksimum. Jangkauan yang digunakan berbeda-beda tergantung dari jenis warnanya. Namun secara umum jangkauannya adalah $0 - 255$. Citra dengan penggambarannya seperti ini digolongkan kedalam citra integer. Berikut adalah jenis-jenis citra berdasarkan nilai pixelnya.

1. Citra Biner

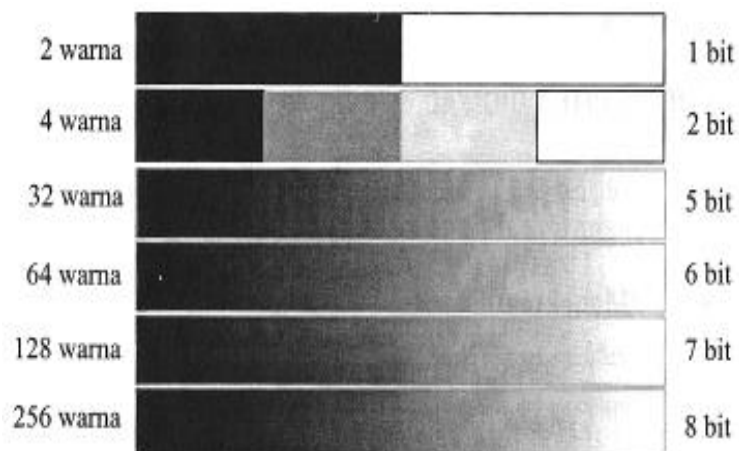
Citra biner adalah citra digital yang hanya memiliki dua kemungkinan nilai pixel yaitu hitam dan putih. Citra biner juga disebut sebagai citra *B&W (black and white)* atau citra monokrom. Hanya dibutuhkan 1 bit untuk mewakili nilai setiap pixel dari citra biner.



Gambar 2.2 Citra Biner

2. 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 2.5 menunjukkan perbandingan gradasi warna untuk jumlah bit tertentu.



Gambar. 2.3 Citra Perbandingan Gradasi Warna 1 bit, 2 bit, 5 bit, 6

3. 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 bir



Gambar 2.4. 8 Bit true color

2.5 Tinjauan Umum Visual Basic.Net 2010

Visual Studio 2010 merupakan suatu perangkat lunak yang dapat digunakan untuk pengembangan berbagai macam aplikasi yang memiliki

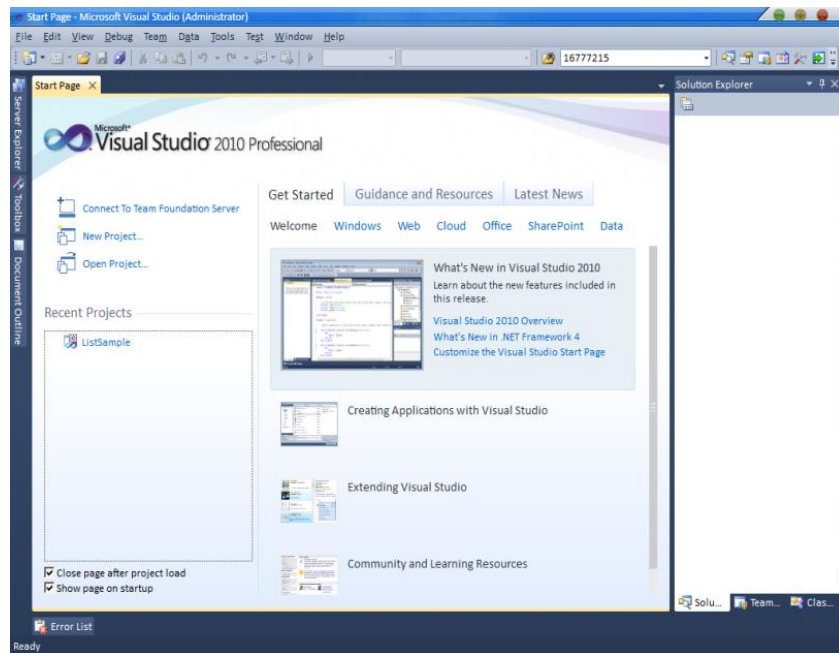
berbagai macam tipe antara lain aplikasi desktop (*Windows Form*, *Command Line (Console)*), Aplikasi Web, *Windows Mobile (Paket PC)* (Yesputra, 2017).

Pada VB.Net mempunyai fasilitas *Real Time Background Compiler* yaitu sebagai penanganan dalam *error* atau *bug* . Ketika terdapat kesalahan waktu menuliskan kode (*coding*), *VB.Net* langsung menuliskan pesan *error* pada bagian *Message Windows*, sehingga kesalahan *coding* dapat langsung diperbaiki tanpa harus eksekusi program terlebih dahulu. Selain itu lebih cepat dalam pembuatan aplikasi berbasis desktop dan mudah dipahami (*user friendly*)

Visual Studio 2010 memiliki lebih dari satu kompiler, SDK (*Software Development Kit*), dan Dokumentasi Tutorial (*MSDN Library*). Kompiler yang dimasukkan kedalam Visual Studio 2010 antara lain Visual Basic, *Visual C#*, *Visual C++*, *Visual Inter Dev*, *Visual J++*, *Visual F#*, dan *Visual Source Safe*, dan banyak yang lainnya. Dan semua itu sudah terpaket dan diperuntukkan kedalam *platform .Net Framework 4.0* atau versi yang lebih tinggi. Visual studio ini dapat digunakan untuk membuat aplikasi yang berbasis desktop yang merupakan platform *windows*, namun juga dapat dijalankan dalam bentuk *Microsoft Intermediate Language* diatas *.Net Framework*. Selain itu Visual Studio juga dapat digunakan untuk membuat aplikasi yang dapat dijalankan diatas *windows mobile* yang berjalan diatas *.Net Compact Framework*.

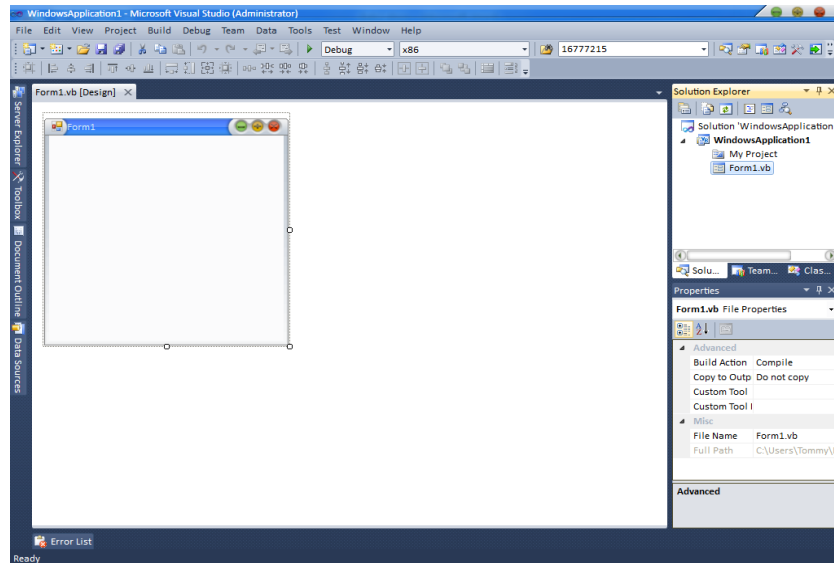
2.5.1 Lingkungan Kerja Visual Studio 2010

Secara garis besar lingkungan kerja Visual Basic 2010 tidaklah berbeda jauh dengan Visual Basic versi sebelumnya. Ketika pertama sekali dieksekusi tampilan Visual Basic 2010 akan terlihat seperti gambar berikut.



Gambar 2.5 Tampilan Halaman Muka Visual Basic 2010.

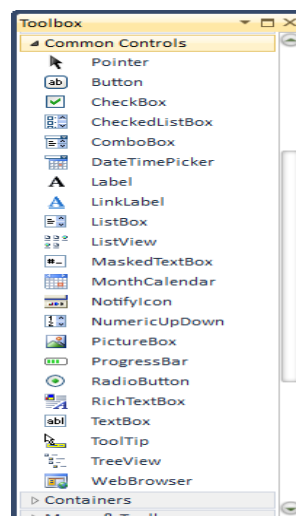
Setelah tampilan utama tampil, maka kita dapat membuat sebuah proyek aplikasi baru dan akan tampil tampilan desain dari proyek aplikasi baru seperti gambar berikut.



Gambar 2.6 Lingkungan kerja Visual Basic 2010

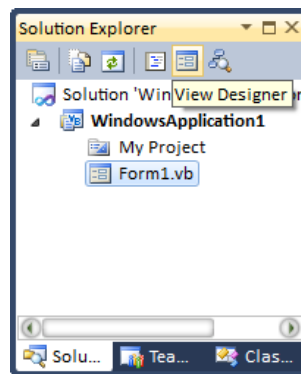
Berikut penjelasan komponen-komponen lingkungan kerja Visual Basic 2010.

1. *Toolbox* adalah bagian yang berguna sebagai tempat meletakkan objek-objek yang menjadi alat untuk membangun aplikasi. Alat yang dimaksud adalah sebuah *Class*. Objek-objek yang tersedia sangat banyak seperti TextBox, ComboBox, ListBox, dan lain sebagainya.



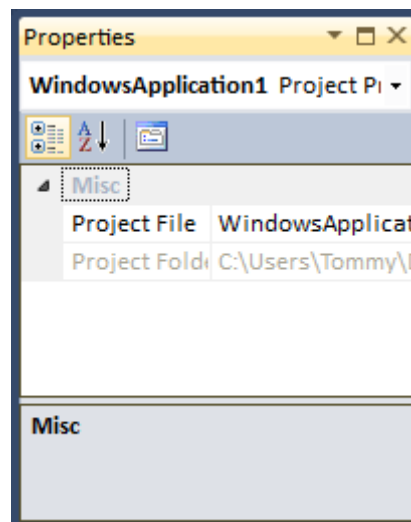
Gambar 2.7 Toolbox

2. *Solution Explorer* adalah bagian yang berfungsi melihat item-item penyusun sebuah proyek atau *solution*. Sebuah proyek aplikasi dapat disusun oleh berbagai macam item seperti form, class, module, laporan, dan lain sebagainya.



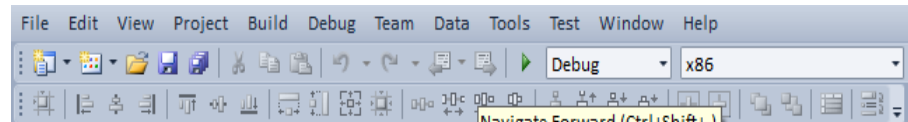
Gambar 2.8 Solution Explorer.

3. *Properties* berguna untuk melihat nilai property dari sebuah objek/*class*. Misalnya property Name, Text dan lain sebagainya.



Gambar 2.9 Properties

4. *Toolbar* berguna sebagai jalan pintas bagi fungsi-fungsi yang sering diakses di menu utama.



Gambar 2.10 Toolbar

2.6 Pengertian UML

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:




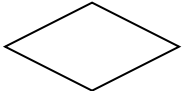

1. Merancang perangkat lunak.
2. Sarana komunikasi antara perangkat lunak dengan proses bisnis.
3. Menjabarkan sistem secara rinci untuk analisa dan mencari apa yang diperlukan sistem.
4. Mendokumentasi sistem yang ada, proses-proses dan organisasinya

2.6.1 Activity Diagram

Activity diagram menggambarkan berbagai aliran aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir

berawal, keputusan yang mungkin terjadi, dan bagaimana mereka berakhir (Andrianto dan Softwan, 2018). 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 2.2 Notasi Activity Diagram

Gambar	Nama Simbol	Keterangan
	Status Awal	Sebuah diagram aktivitas memiliki sebuah status awal
	Status Akhir	Status akhir yang dilakukan sistem sebuah diagram aktivitas memiliki sebuah status akhir
	Aktivitas	Aktivitas yang dilakukan sistem biasanya diawali dengan kata kerja
	Decision / Percabangan	Percabangan dimana ada pilihan aktivitas yang lebih dari satu
	<i>Fork</i>	Digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu


2.6.2 Use Case Diagram

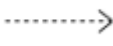




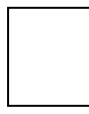


Use case diagram digunakan untuk memodelkan bisnis proses berdasarkan perspektif pengguna (Andrianto dan Softwan, 2018). *Use case* merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, meng-*create* sebuah daftar belanja, dan sebagainya.


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 2.3 Simbol Use Case Diagram

Gambar	Nama Simbol	Keterangan
	<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>Use</i>

Gambar	Nama Simbol	Keterangan
		<i>Case.</i>
	<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>)
	<i>Generalization</i>	Hubungan dimana objek anak(<i>Descended</i>) berbagi perilaku dan struktur data dari objek yang di atasnya objek induk.
	<i>Include</i>	Menspesifikasikan bahwa use case sumber secara explicit.
	<i>Extend</i>	Menspesifikasikan bahwa use case target memperluas perilaku pada use case sumber pada sebuah titik diberikan.
	<i>Assosiation</i>	Apa yang menghubungkan objek satu dengan objek yang lainnya.
	<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
	<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur dari sebuah <i>actor</i> .
	<i>Colaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan

Gambar	Nama Simbol	Keterangan
		elemen-elemennya.
	<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.



2.6.3 Class Diagram

Class diagram menggambarkan struktur statis dari kelas dalam sistem anda dan menggambarkan atribut, operasi dan hubungan antara kelas (Haviluddin, 2017). *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.

Class diagram memiliki tiga area pokok:

1. Nama (dan *stereotype*)
2. Atribut
3. Metoda

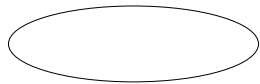
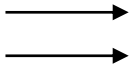
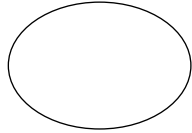

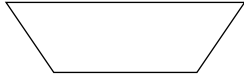
Tabel 2.4 Simbol Class Diagram

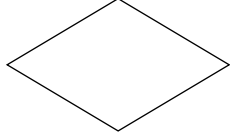



Gambar	Nama Simbol	Keterangan
	<i>Class</i>	Himpunan dari objek- objek yang berbagi atribut serta operasi yang sama.
	<i>Associatiom</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya

2.7 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
	<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

Gambar	Nama Simbol	Keterangan
	<p><i>Decision</i></p>	<p>Simbol <i>decision</i>, menunjukkan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan</p>
	<p><i>Keying Operation</i></p>	<p>Simbol <i>keying operation</i> menyatakan segala jenis operasi yang diproses dengan menggunakan suatu mesin yang mempunyai keyboard</p>
	<p><i>Input/Output</i></p>	<p>Simbol <i>input/output</i> menyatakan proses input/output</p>
	<p><i>Document</i></p>	<p>Simbol dokumen mencetak keluaran dalam bentuk dokumen</p>

BAB III

METODOLOGI PENELITIAN

3.1 Waktu dan Tempat Penelitian

Waktu penelitian dimulai dari 01 Desember 2022 sampai dengan 31 Mei 2023. Penelitian dilakukan di secara mandiri dengan cara melakukan ujicoba terhadap berbagai macam jenis citra.

3.2 Metode Penelitian

Dalam melakukan penelitian ini, penulis menggunakan metode terapan (*applied research*). Penelitian terapan atau *applied research* 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 maupun untuk keperluan industri atau politik dan bukan untuk wawasan keilmuan semata.

Dalam melaksanakan penelitian terapan ini terdapat 5 (lima) langkah, diantaranya :

- a. Melakukan sesuatu yang sedang diperlukan, dipelajari, diukur, dan diperiksa kelemahannya.
- b. Mencari satu dari kelemahan-kelemahan yang diperoleh dipilih untuk penelitian.
- c. Mencari dan memberikan solusi dalam melakukan pemecahan masalah

- d. Kemudian dilakukan modifikasi sehingga penyelesaian dapat dilakukan untuk diterapkan.
- e. Pemecahan dipertahankan dan menempatkannya dalam suatu kesatuan sehingga jadi bagian permanen dalam satu sistem.

3.3 Perangkat Keras dan Perangkat Lunak

Tabel 3.1 Perangkat Keras dan Perangkat Lunak

Perangkat Keras	Perangkat Lunak
Processor Intel Core i3, RAM 4GB, Hardisk 500GB	OS Windows 10 Visual Basic.Net 2010

3.4 Metode Pengumpulan Data

Metode Pengumpulan data digunakan untuk memperoleh data yang dapat mendukung permasalahan yang akan dibahas. Sehubungan dengan hal ini maka digunakan metode pengumpulan data yang meliputi :

a. Observasi

Dalam pengumpulan data melalui observasi, penulis mengamati dan menganalisa bagaimana steganografi Pada Citra Digital Menggunakan Metode *Pixel Value Differencing* (PVD).

b. Studi Pustaka

Studi pustaka adalah suatu metode pengumpulan data yang diambil dari perpustakaan, jurnal, buku-buku serta dari *internet* yang berhubungan dengan penulisan

ini. Tujuan dari studi pustaka ini adalah untuk mendalami dan memperoleh keterangan yang lengkap terhadap objek yang diteliti.

3.5 Analisa Perancangan Sistem

3.5.1 Analisa Sistem Aktual

Masalah yang diangkat dari penelitian tugas akhir ini adalah perancangan aplikasi steganografi dengan algoritma *Pixel Value Differencing* pada citra digital. Metode *Pixel Value Differencing* merupakan metode yang memanfaatkan perbedaan nilai piksel pada area tertentu sebagai kriteria dan bit data yang akan disisipkan.

3.5.2 Analisa Kebutuhan Sistem

Analisis kebutuhan sistem meliputi analisis kebutuhan fungsional dan analisis kebutuhan *non-fungsional* sistem

1. Analisis Kebutuhan Fungsional Sistem

Kebutuhan fungsional yang harus dimiliki oleh sistem steganografi menggunakan *pixel value differencing* pada citra digital adalah :

- a. Sistem dapat melakukan penyisipan pada citra digital menggunakan teknik steganografi *pixel value differencing*
- b. Sistem dapat melakukan ekstraksi kembali pesan yang disisipkan pada citra digital menggunakan teknik *pixel value differencing*.

2. Analisis Kebutuhan Non-Fungsional Sistem

Kebutuhan *non-fungsional* yang harus dimiliki oleh sistem steganografi pada citra digital menggunakan teknik *pixel value differencing* adalah :

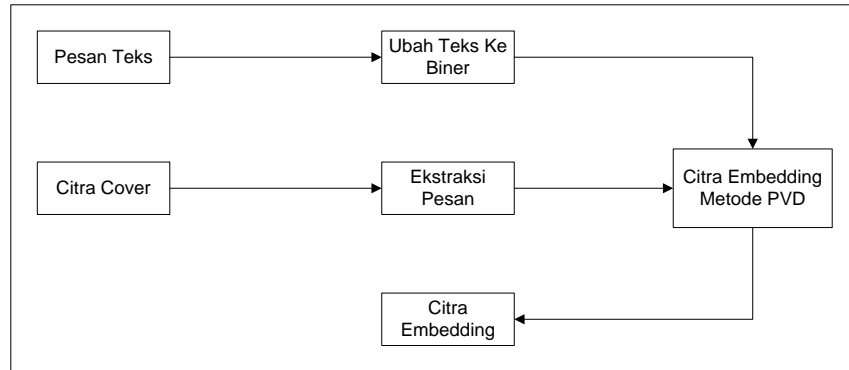
- a. Sistem dapat melakukan penyisipan yang tepat menggunakan teknik *pixel value differencing*
- b. Tampilan antarmuka sistem menarik dan dapat dimengerti oleh pengguna sistem

3.5.3 Analisa Steganografi Pada Citra Digital Menggunakan Metode *Pixel Value Differencing* (PVD)

Metode Pixel Value Differencing(PVD) merupakan salah satu metode steganografi pada citra digital yang beroperasi pada ranah spasial. Konsep dari metode ini yaitu dengan menyisipkan pesan kedalam dua pixel yang bertetangga, dengan memanfaatkan perbedaan intensitas warna dari kedua pixel yang bertetangga tersebut. Seperti halnya metode steganografi lainnya, pada metode ini terdapat dua proses yaitu proses penyisipan/embedding dan pengungkapan/ekstraksi

1. Analisis Proses Penyisipan/Embedding

Proses penyisipan yaitu proses menyembunyikan informasi kedalam media penampung, dalam hal ini media penampung berupa citra digital. Proses ini akan menghasilkan citra yang telah disisipkan pesan (stego-object) yang menyerupai dengan citra sebelum disisipkan pesan. Proses penyisipan pada metode pixel value differencing terlihat pada gambar berikut :



Gambar 3.1 Proses Penyisipan Pesan

Tabel 3.2 Nilai Continues Range Dari Nilai Difference Value

Kuantitasi ke-k	Batas bawah – Batas Atas (l_k - u_k)	Rentang Nilai	Jumlah Bit n
1	0-7	8	3
2	8-15	8	3
3	16-31	16	4
4	32-63	32	5
5	64-127	64	6
7	128-255	127	7

Contoh proses penyisipan, jika diketahui pesan yang akan disisipkan “Universitas Dehasen”, dengan wadah penampung citra ukuran 4 x 4 pixel.

130	156	142	239
149	243	147	240
153	209	151	240
155	188	159	221

Gambar 3.2 Nilai Pixel Citra Penampung

Adapun tahapan penyelesaian penyisipan pesan adalah sebagai berikut :

- a. Ubah pesan ke dalam bentuk biner

PESAN	BILANGAN ASCII	BINER
U	85	01010101
n	110	01101110
i	105	01110101
v	118	01110110
e	101	01100101
r	114	01110010
s	115	01110011
i	105	01110101
t	117	01110101
a	97	01100001
s	115	01110011
Spasi	32	00100000
D	68	01000100
e	101	01100101
h	104	01101000
a	97	01100001
s	115	01110011
e	101	01100101
n	110	01101110

b. Satukan semua biner pesan di atas:

010101010110111001110101011101100110010101110010
011100110111010101110101011000010111001100100000
010001000110010101101000011000010111001101100101
01101110

c. $d = |130 - 156| = 26$

d. Mencari letak nilai *continues range* dari nilai *difference value* berdasarkan tabel 3.2 diatas, maka letak nilai $d= 26$ yaitu $[16,31]$ dimana $l_k = 16$ dan $u_k = 31$

e. Menghitung berapa banyak bit dari pesanyang dapat disisipkan kedalam kedua pixel yang dibandingkan menggunakan persamaan

$$t = \lceil \log_2 w_i \rceil$$

$$t = |\log_2(16 - 31)| = 4$$

Maka diambil bit dari pesan sebanyak $t = 0101$

- f. Mengubah nilai bit t ke dalam nilai *desimal*. Bit informasi yang disisipkan yaitu 0101, maka nilai desimalnya yaitu 5 atau $b=5$
- g. Menghitung nilai *differencing value* menggunakan persamaan $d'_i = l_i + b$, sehingga diperoleh $d'_i = 16 + 5 = 21$, maka didapatkan nilai $d'_i = 21$ yang merupakan nilai *differencing value* yang baru
5. Melakukan penyisipan dengan mengubah nilai dari *pixel* yang dibandingkan dengan nilai *pixel* yang baru sesuai dengan aturan-aturan yang ditetapkan, dimana $m=5$, didapat menggunakan persamaan $m = |d'_i \leq d_i|$, $m = |21 \leq 26|$. Aturan terpenuhi yaitu

$$\text{Jika } P_i < P_{i+1} \text{ dan } d'_i \leq d_i, \text{ maka } \left(P'_i + \left\lceil \frac{m}{2} \right\rceil, P_{i+1} - \left\lfloor \frac{m}{2} \right\rfloor \right)$$

Sehingga diperoleh hasil :

$$\text{Jika } P_i < P_{i+1} \text{ dan } d'_i \leq d_i, \text{ maka } \left(130 + \left\lceil \frac{5}{2} \right\rceil, 156 - \left\lfloor \frac{5}{2} \right\rfloor \right)$$

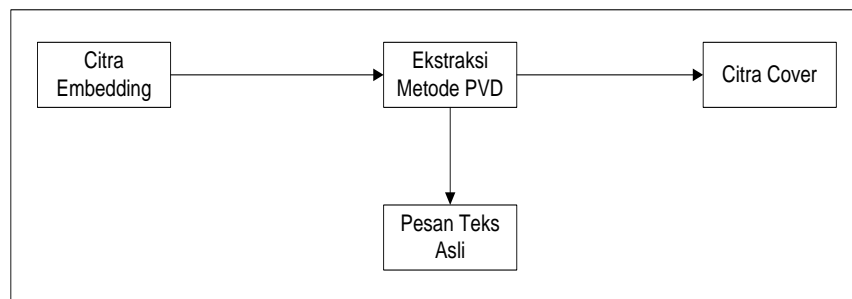
6. Menyimpan nilai *pixel* yang baru yaitu : $P'_i = 132$ dan $P_i = 153$ kedalam citra. Tahapan ini dilakukan sampai semua pesan tersisipi, sehingga menjadi seperti berikut :

132	153	142	239
149	243	147	240
153	209	151	240
155	188	159	221

Gambar 3.3 Nilai Pixel Citra Setelah Disisipi Pesan

2. Analisa Proses Ekstraksi Pesan

Proses ekstraksi yaitu proses pengambilan informasi yang tersembunyi pada citra digital. Proses ini akan menghasilkan *file* informasi yang disembunyikan, dengan masukan berupa citra *stego-object*. Proses ekstraksi pada metode *pixel value differencing* terlihat pada gambar berikut :



Gambar 3.4 Proses Ekstraksi Pesan

Tahap awal pada proses ekstraksi pesan yaitu mengambil nilai *pixel* dari citra yang telah disisipkan pesan.

132	153	142	239
149	243	147	240
153	209	151	240
155	188	159	221

Gambar 3.5 Nilai *Pixel* Citra yang Telah Disisipi Pesan

Maka tahap selanjutnya yaitu melakukan proses ekstraksi menggunakan metode *pixel value differencing* dengan tahapan-tahapan yaitu sebagai berikut :

- a. Mengambil *pixel* yang bertetangga dari citra. Contoh *pixel* yang bertetangga yaitu *pixel* (0,0) dengan *pixel* (0,1) seperti pada gambar 3.4. Nilai dari *pixel* yang bertetangga adalah $P_i = 132$ dan $P_{i+1} = 153$
- b. Menghitung nilai *differencing value* dari kedua *pixel* tersebut

$$d = |132 - 153| = 21$$

c. Mencari letak nilai *continues range* dari nilai *difference value* berdasarkan tabel 3.2 diatas, maka letak nilai $d = 21$ yaitu $[16,31]$ dimana $l_k = 16$ dan $u_k = 31$

d. Menghitung berapa banyak bit dari pesanyang dapat disisipkan kedalam kedua *pixel* yang dibandingkan menggunakan persamaan

$$t = |\log_2 w_i|$$

$$t = |\log_2(16 - 31)| = 4$$

Maka diambil bit dari pesan sebanyak $t = 0101$

e. Mengubah nilai bit t ke dalam nilai *desimal*. Bit informasi yang disisipkan yaitu 0101, maka nilai desimalnya yaitu 5 atau $b=5$

f. Tahapan-tahapan pada metode *pixel value differencing* tersebut diulang hingga semua pesan yang terdapat di dalam citra terekstrak. Tahap selanjutnya setelah semua pesan terekstrak yaitu merubah pesan dalam bentuk biner ke bentuk semula. Jika pesan yang disisipkan berupa pesan teks, maka diubah kedalam bentuk teks seperti pada

010101010110111001110101011101100110010101110010

011100110111010101110101011000010111001100100000

010001000110010101101000011000010111001101100101

01101110



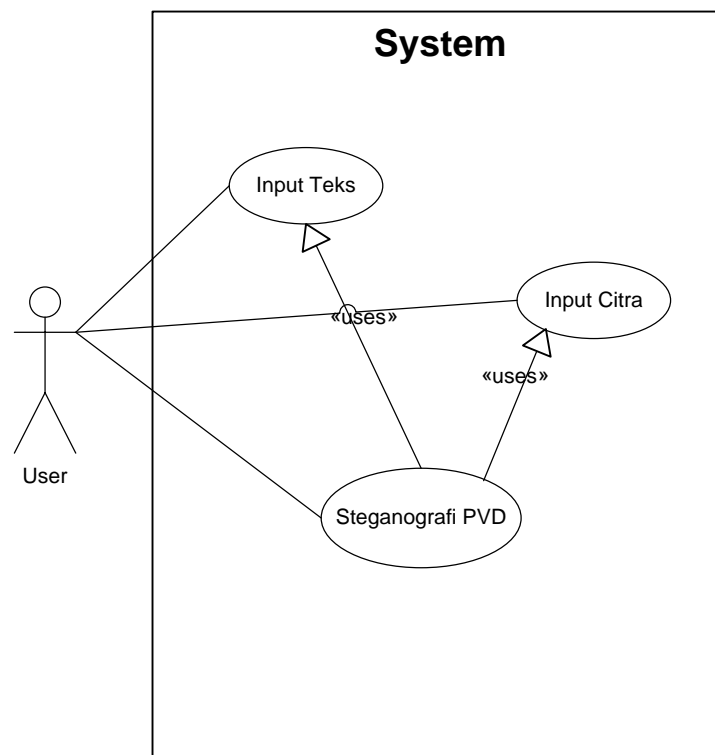
Universitas Dehasen

3.5.4 Analisa Perancangan Sistem Baru

Perancangan sistem merupakan salah satu proses yang menggambarkan bagaimana suatu sistem dibangun untuk memenuhi kebutuhan atau keinginan pada tahapan analisis. Perancangan sistem dilakukan dengan merancang *use case diagram*, *activity diagram*, *flowchart* dan perancangan *interface* (antarmuka) aplikasi.

A. Use Case Diagram

Use Case Diagram seperti yang terlihat pada gambar 3.5 dapat dilihat bahwa pengguna berinteraksi dengan aplikasi melalui empat *use case* yang berbeda yaitu *use case* Input Teks Rahasia, *use case* Input Citra, *use case* Steganografi PVD.



Gambar 3.6 Use Case Diagram

Keterangan :

1. *Use Case Input Teks Rahasia*, *use case* ini mendeskripsikan bagaimana pengguna berinteraksi dengan melakukan input teks rahasia yang akan

disisipkan kedalam citra penampung. Pengguna memilih menu yang tersedia untuk membuka file teks atau dengan mengisi langsung teks rahasia pada kolom yang disediakan.

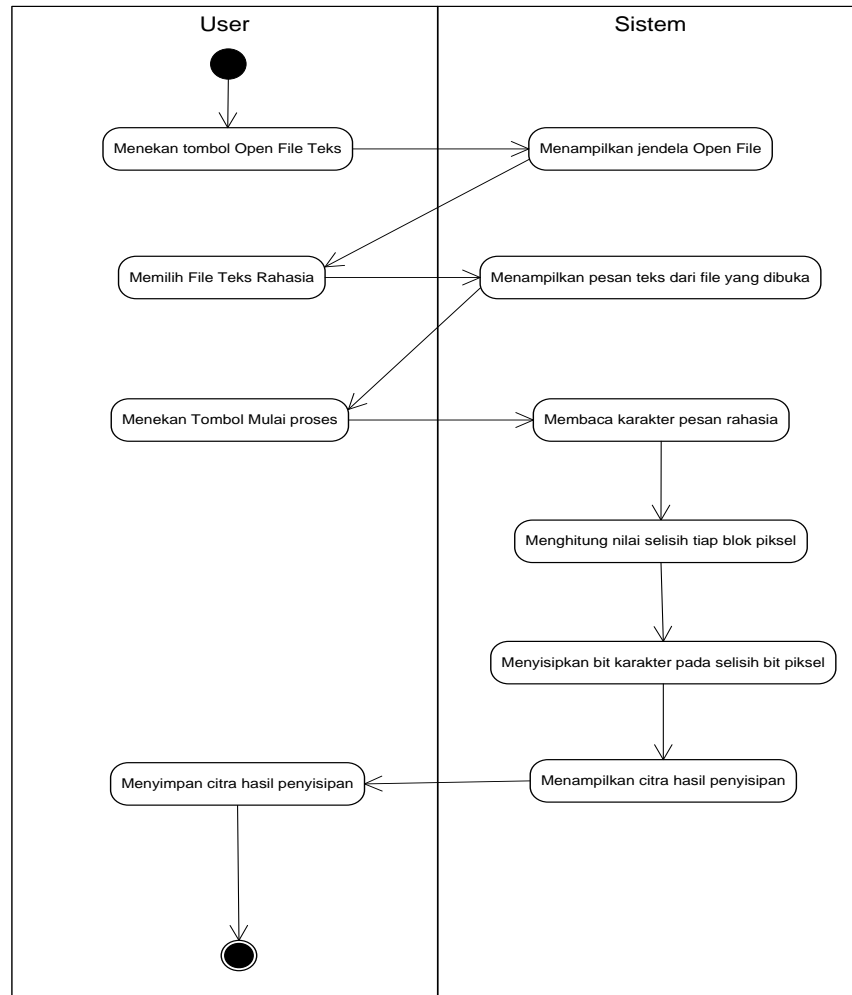
2. *Use Case Input Citra*, *use case* ini mendeskripsikan bagaimana pengguna berinteraksi dengan melakukan input file citra digital yang akan disisipkan dengan pesan rahasia. Pengguna memilih menu yang tersedia untuk membuka file citra digital yang akan digunakan.
3. *Use Case Steganografi PVD*, *use case* ini mendeskripsikan bagaimana pengguna melakukan proses penyisipan dan ekstraksi terhadap citra penampung. Pengguna dapat menekan menu atau tombol yang telah disediakan untuk memulai proses penyisipan maupun ekstraks pesan rahasia dari file citra penampung.

B. Acitivity Diagram

Acitivity Diagram merupakan diagram yang memperlihatkan urutan – urutan aktivitas yang dilakukan pengguna selama menggunakan sistem serta proses dan respon sistem yang dilakukan atas interaksi pengguna. Pada proposal skripsi ini terdapat dua activity diagram utama, yaitu activity diagram penyisipan dan acitivity diagram ekstraksi

1. Activity Diagram Penyisipan

Activity Diagram penyisipan digunakan untuk memperlihatkan alur aktivitas pengguna dalam menjalankan proses penyisipan pada aplikasi yang dibangun. Adapaun diagram dari *Activity Diagram Penyisipan* dapat dilihat pada gambar 3.7 dibawah ini

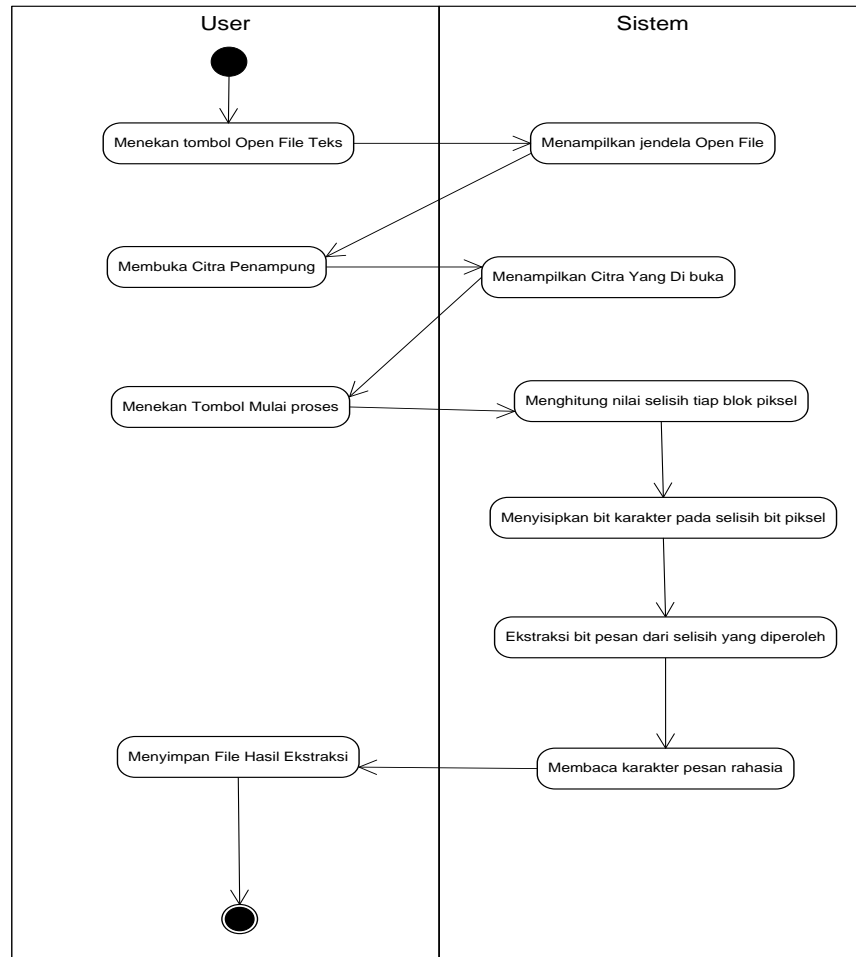


Gambar 3.7 Activity Diagram Penyisipan

2. Activity Diagram Ekstraksi

Activity Diagram ekstraksi digunakan untuk memperlihatkan alur aktivitas pengguna dalam menjalankan proses ekstraksi pada aplikasi yang dibangun.

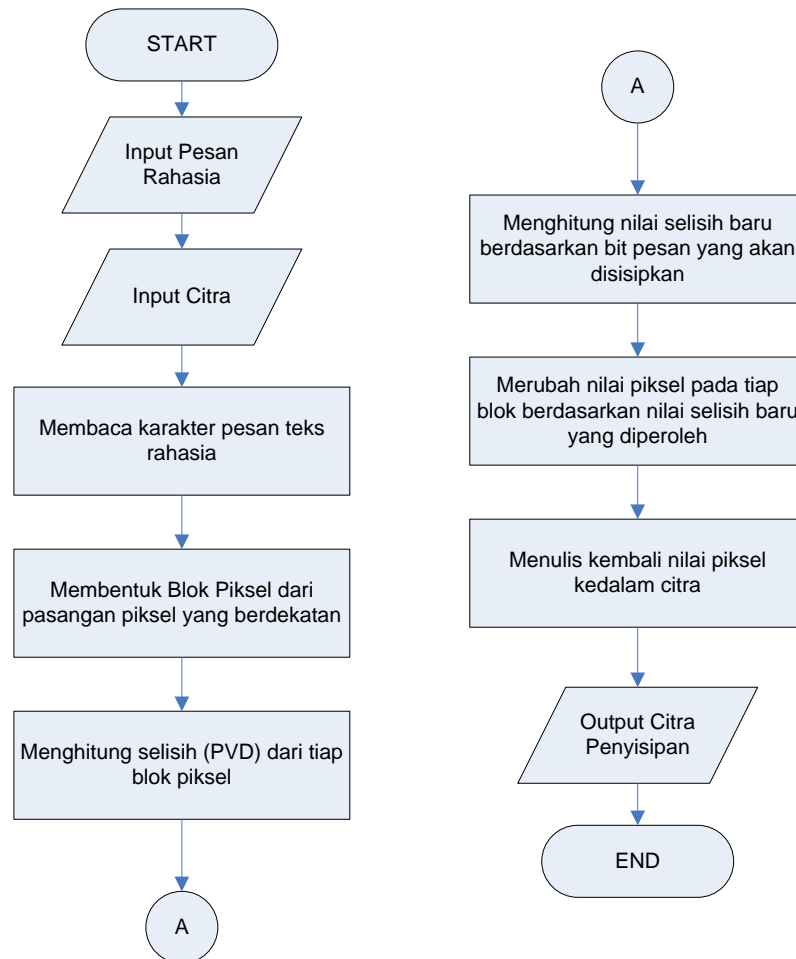
Adapaun diagram dari *Activity Diagram Ekstraksi* dapat dilihat pada gambar 3.8 dibawah ini



Gambar 3.8 Activity Diagram Ekstraksi

C. Flowchart Sistem

Flowchar (diagram alir) yang dirancang pada penelitian tugas akhir ini membahas alur dari proses yang digunakan pada aplikasi yang dibangun. Adapun diagram *flowchart* sistem dapat dilihat pada gambar 3.9 berikut



Gambar 3.9 Flowchart Sistem

Keterangan :

Dari Gambar 3.9 dapat dilihat gambaran umum aplikasi yang dibangun pada proposal skripsi ini. Pengguna menginput data citra dan pesan yang akan disisipkan. Pesan pertama sekali dibaca dan konversi menjadi kode ASCII dan

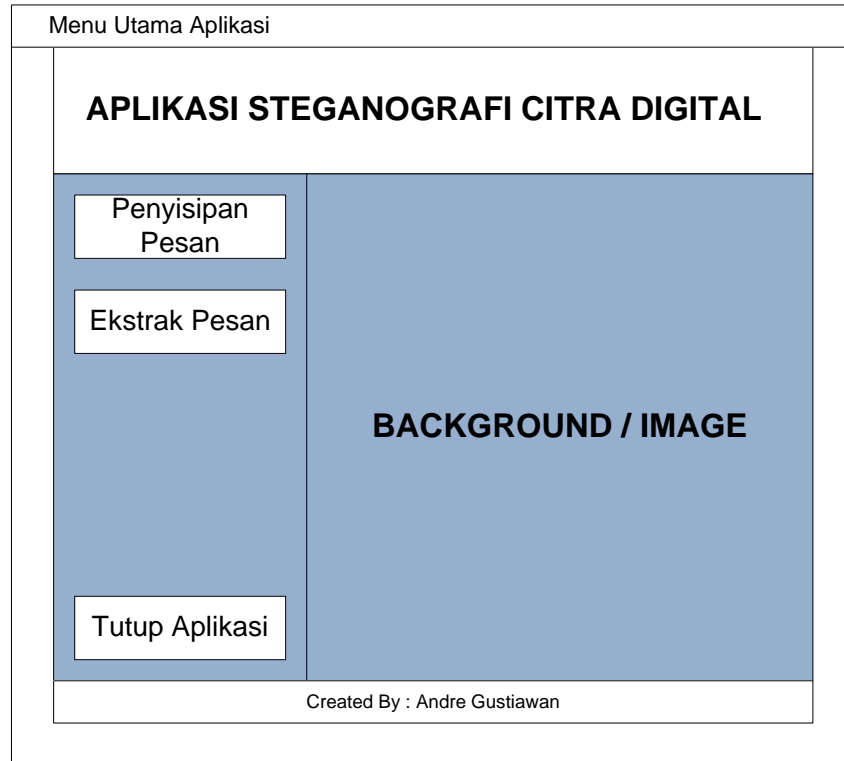
biner, hasil transformasi kemudian digunakan untuk menghitung nilai PVD dan penyisipan pada piksel citra penampung. *Output* citra yang telah disisipkan kemudian ditampilkan kepada pengguna.

D. Rancangan *Interface* (Antarmuka) Aplikasi

Sistem akan dibangun menggunakan bahasa pemrograman Visual Basic dengan menggunakan *software Microsoft Visual Studio 2010*. Rancangan antar muka disesuaikan dengan kebutuhan dan *software* yang digunakan. Antar muka 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 *Interface* Menu Utama

Rancangan *interface* menu utama merupakan tahapan rancangan antarmuka dari awal aplikasi ketika dijalankan. Perancangan antarmuka menu utama dari aplikasi steganografi ini terlihat pada gambar berikut :



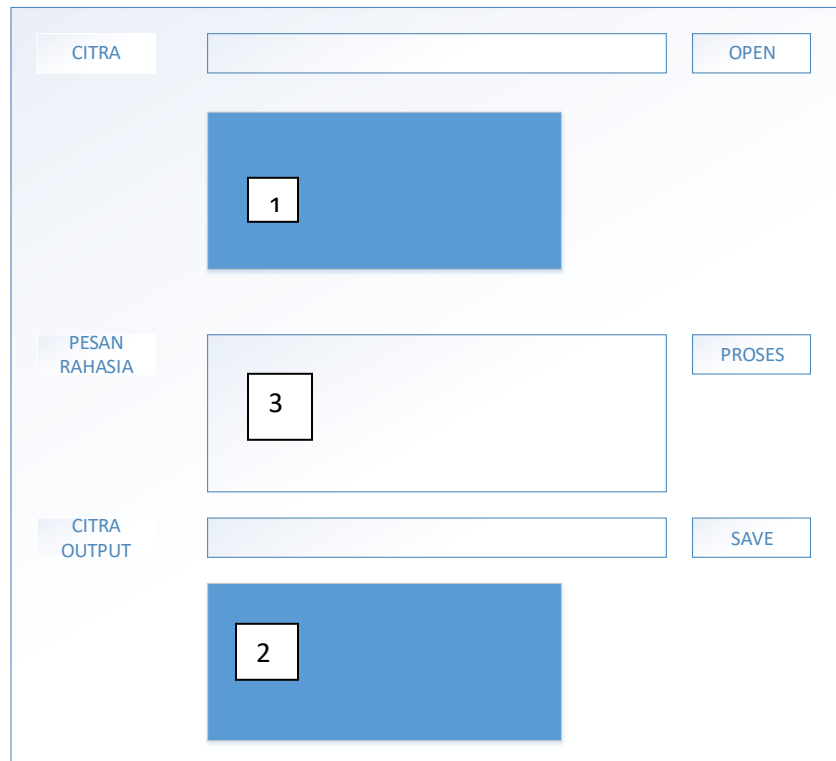
Gambar 3.10 Rancangan *Interface* Menu Utama

Keterangan :

- a. Penyisipan Pesan : Merupakan tombol yang digunakan untuk menuju ke *Form Penyisipan Pesan*
- b. Ekstrak Pesan : Merupakan tombol yang digunakan untuk menuju *Form Ekstrak Pesan*
- c. Tutup Aplikasi : Merupakan tombol yang digunakan untuk menutup atau keluar dari aplikasi

2. Rancangan *Interface* Penyisipan Pesan

Pada *form* Penyisipan terdapat *Interface* yang bertujuan untuk menerima pesan input dan file citra digital. Pada *form penyisipan* akan ditampilkan pesan hasil kompresi dan file citra digital hasil penyisipan

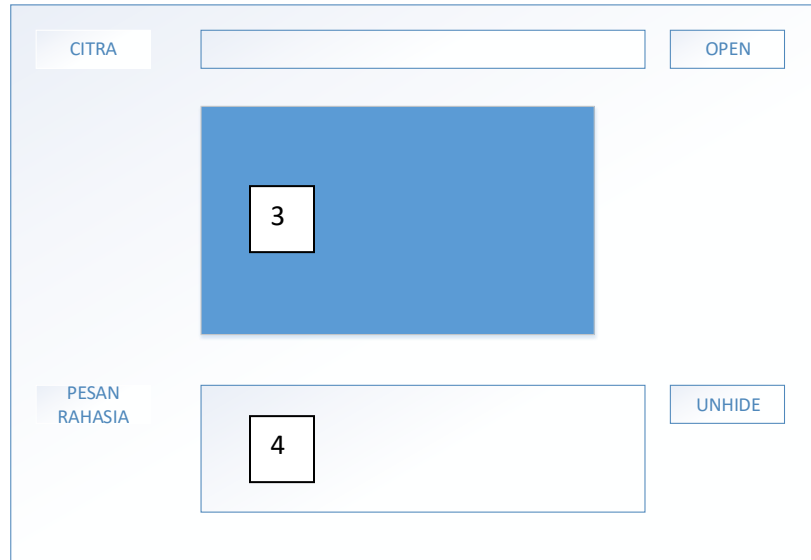


Gambar 3.11 Rancangan *Interface* Penyisipan Pesan

Komponen yang dipakai untuk membangun antar muka *form* penyisipan pada gambar 3.11 adalah sebagai berikut :

- a. Image 1 : control ini berfungsi untuk menampilkan citra penampung
- b. Text1 : control ini menampung teks pesan yang akan disisipkan pada citra penampung
- c. Image 2 : control ini berfungsi untuk menampilkan citra hasil penyisipan
- d. Tombol “Open” : tombol yang berfungsi untuk membuka dialog untuk memilih citra penampung
- e. Tombol “Proses” : tombol yang berfungsi untuk memulai proses
- f. Tombol “Save” : tombol yang berfungsi untuk menyimpan citra hasil penyisipan

3. Rancangan *Interface* Ekstrak Pesan



Gambar 3.12 Rancangan *Interface* Ekstrak Pesan

3.6 Perancangan Pengujian

Pengujian juga dilakukan dengan cara memberikan masukan (input) berupa bermacam tipe file citra digital, kemudian hasil dari stegano akan di rotasi, untuk mengetahui apakah proses PVD dapat bekerja dengan baik. Adapun rencana pengujian dapat dilihat pada tabel berikut ini :

Nama File	Size Sebelum Disisipi Pesan	Size Setelah Disisipi Pesan	Rotasi 180	Rotasi Kanan 90°	Rotasi Kiri 90°