BAB III

ANALISIS DAN PERANCANGAN SISTEM

III.1 Analisis Sistem

Analisis sistem merupakan tahap yang bertujuan untuk memahami sistem, mengetahui keunggulan dan kelemahan dari sistem di tinjau dari sisi pengguna. Dengan menganalisis prosedur sistem yang sering digunakan, maka sistem yang sering dipakai dapat dievaluasi sehingga dapat dijadikan sebagai acuan untuk membangun suatu sistem yang baru dari hasil evaluasi tersebut.

Dalam penelitian ini yang akan dianalisis adalah tahapan membangun aplikasi pengamanan hak cipta untuk data gambar digital (*digital image*) dan bahasan mengenai teknik *watermarking* yang akan digunakan yaitu metode SVD (*Singular Value Decomposition*)

Sistem yang akan dibangun adalah sistem yang implementasikan menggunakan teknik *watermarking* metode SVD untuk pengamanan hak cipta untuk data gambar digital, sehingga dari sistem tersebut harus dibangun sesuai dengan kebutuhan-kebutuhan sebagai berikut :

1. Membangun aplikasi yang dapat melakukan pengamanan hak cipta untuk data gambar digital.
2. Mengimplementasikan teknik *watermarking* dengan menggunakan metode SVD sesuai tahapan-tahapan didalamnya.
3. Melakukan pengujian aplikasi untuk diukur performansi kerjanya dalam pengamanan hak cipta untuk data gambar digital.

III.1.1 Analisis Masalah

Langkah selanjutnya adalah identifikasi masalah dari hasil analisis terhadap sistem. Maka dapat diidentifikasi masalah, yaitu sebagai berikut :

1. Aplikasi yang akan dibangun harus dapat digunakan untuk pengamanan hak cipta untuk gambar data digital.
2. Teknik *watermarking* dengan metode SVD harus dapat digunakan untuk pengamanan hak cipta untuk gambar data digital.

III.1.2 Analisis Kebutuhan Non-Fungsional

Analisis dan kebutuhan non-fungsional meliputi analisis dan kebutuhan pengguna, analisis dan kebutuhan perangkat keras, serta analisis dan kebutuhan perangkat lunak.

III.1.2.1 Analisis Kebutuhan Pengguna (*user*)

Pengguna diartikan sebagai orang yang mengakses dan menggunakan perangkat lunak aplikasi *Watermarking* metode *SVD*, dalam hal ini dibutuhkan pengguna yang memiliki kemampuan tingkat *average user* dimana *user* dapat mengoperasikan komputer dan memiliki kapabilitas dalam menggunakan aplikasi komputer serta pengguna yang membutuhkan aplikasi watermarking untuk melindungi hak cipta gambar digital.

III.1.2.2 Analisis Kebutuhan Perangkat Keras

Perangkat keras adalah seluruh komponen atau unsur peralatan yang digunakan untuk menunjang pembangunan suatu aplikasi, dalam kasus ini adalah aplikasi *watermarking* menggunakan metode *SVD (Singular Value*

Decomposition) dan metode *DWT (Discreete Wavelet Transform)*, untuk kebutuhan minimum perangkat keras dapat dilihat dalam tabel 3.1

Tabel 3.1 Tabel spesifikasi perangkat keras

Processor	***Memory (RAM)***	***Hard-disk***	***Monitor (Resolution)***
Intel Pentium 4	Kapasitas memory mulai dari 512 MB sampai 1 GByte	Kapasitas hard-disk mulai dari 40 GByte	Pada umumnya menggunakan monitor dengan resolusi 1024 x 768 (32 bit) 60 Hz

III.1.2.2 Analisis Kebutuhan Perangkat Lunak

Analisis kebutuhan perangkat lunak yang di perlukan adalah sebagai berikut :

1. Sistem operasi yang digunakan adalah Windows Xp Service Pack 3.
2. Perangkat lunak yang digunakan untuk membangun aplikasi ini adalah Microsoft Visual Studio 2008 (C#).

III.1.3 Analisis Kebutuhan Fungsional

Actor Identification

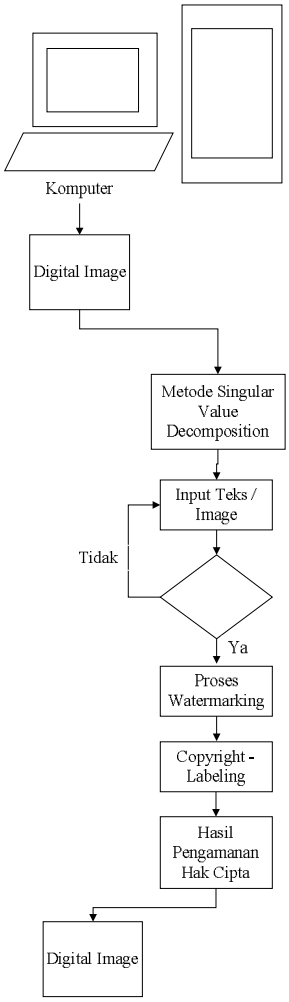
Tahap pertama yang dilakukan dalam melakukan analisis berorientasi objek menggunakan UML adalah menentukan actor atau pengguna sistem. Kata aktor dalam konteks UML, menampilkan peran (*roles*) yang pengguna (atau sesuatu di luar sistem yang dikembangkan yang dapat berupa perangkat keras, *end user*, sistem yang lain, dan sebagainya).

III.1.4 Analisis Metodologi SVD

Analisis ini dilakukan di setiap tahapan penelitian membangun aplikasi dengan ruang lingkup, sebagai berikut :

a. Tahap implementasi pada data gambar digital.
b. Teknik *watermarking*
c. Metode SVD (*Singular Value Decomposition*)

Ketiga ruang lingkup metodologi di atas dapat digambarkan seperti pada gambar 3.1



Gambar 3.1 Alur Proses Watermarking Pada Data Gambar Digital Menggunakan Metode SVD

III.1.5 Analisis Kode Warna Gambar

Dalam sebuah gambar digital kita sering melihat kode warna yang terdiri dari tanda ‘#’ dan 6 angka/huruf dibelakang tanfa tersebut, kode warna tersebut dapat diterjemahkan menjadi kode warna RGB (Red, green, Blue).

Misalkan kita mempunyai kode warna “#0088FF”, kita akan mencari beberapa porsi untuk warna merah (red), hijau (Green), dan biru (blue). Dengan begini kita bisa memperkirakan warna apa yang terbentuk untuk kode tersebut.

Pertama tanda “#” menunjukkan kalau itu adalah kode warna jadi tidak perlu terlalu diperhatikan, untuk 6 karakter disamping kanan tanda “#” digunakan bilangang berbasis 16 yang terdiri dari angka 0-9 dan A-F : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F (semakin kekanan nilainya semakin besar)dan setelah F ada angka 10 (bukan berarti bernilai “Sepuluh” tetapi 1 angka lebih besar dari F) 11, 12, ..., 1E, 1F, 20, 21, ..., 2E, 2F, ..., A0, A1, ..., AE, AF, ..., F9, FA, FB, FC, FD, FE, dan terakhir FF.

Yang kita dapatkan dari penjelasan diatas adalah bahwa “00” adalah angka terkecil sedangkan “FF” adalah yang terbesar. Lihat kembali kode warna”#0088FF” lihat 2 angka setelah tanda “#” : “00” manunjukkan porsi untuk warna merah (red), 2 angka berikutnya “88” menunjukkan porsi untuk warna hijau(green), dan dua huruf terakhir “FF” menunjukkan persi untuk warna biru (blue).

Dari sini kita lihat bahwa porsi untuk warna merah tidak ada, untuk warna hijau sebagian dan warna biru mendapatkan posri paling banyak, jadi kita bisa memperkirakan bahwa kode “#0088FF” berwarna agak kebiru-biruan.

Kadang ada juga yang cuma menuliskan kode warna 3 angka/huruf. Tapi ini cuma untuk menyingkat penulisan jika terdapat banyak angka/huruf yang sama. Contoh: "**#0088FF**" bisa disingkat jadi "**#08F**"

III.1.6 Analisis Perubahan Gambar Kedalam Matrik

Matriks adalah suatu kumpulan besaran (variabel dan konstanta) yang dapat dirujuk melalui indeknya, yang menyatakan posisinya dalam representasi umum yang digunakan, yaitu sebuah tabel persegipanjang. Matriks merupakan suatu cara visualisasi variabel yang merupakan kumpulan dari angka-angka atau variabel lain, misalnya vektor. Dengan representasi matriks, perhitungan dapat dilakukan dengan lebih terstruktur. Pemanfaatannya misalnya dalam menjelaskan persamaan linier, transformasi koordinat, dan lainnya. Matriks seperti halnya variabel biasa dapat dimanipulasi, seperti dikalikan, dijumlah, dikurangkan dan didekomposisikan.

Gambar yang penulis akan coba ubah adalah gambar di bawah ini :



Gambar 3.2 gambar yang akan diubah ke dalam matrik

Image di atas berukuran 107 x 105 pixel yang berarti image di atas terdiri dari 107 kolom dan 105 baris pixel. Total keseluruhan adalah 11235 pixel, dimana tiap pixel adalah perpaduan dari 3 warna yaitu merah (Red), hijau (Green) dan biru (Blue) atau yang biasa dikenal dengan RGB.

Untuk mempermudah merubah gambar menjadi matriks disini saya akan menggunakan aplikasi matlab untuk menerapkannya :

```
>>c=imread(‘lock.png’,'png’);
>> asci=uint8(c)
```

baris pertama berfungsi untuk menampilkan gambar yang sudah disimpan pada folder work dalam matlab :

C:\Program Files\MATLAB71\work

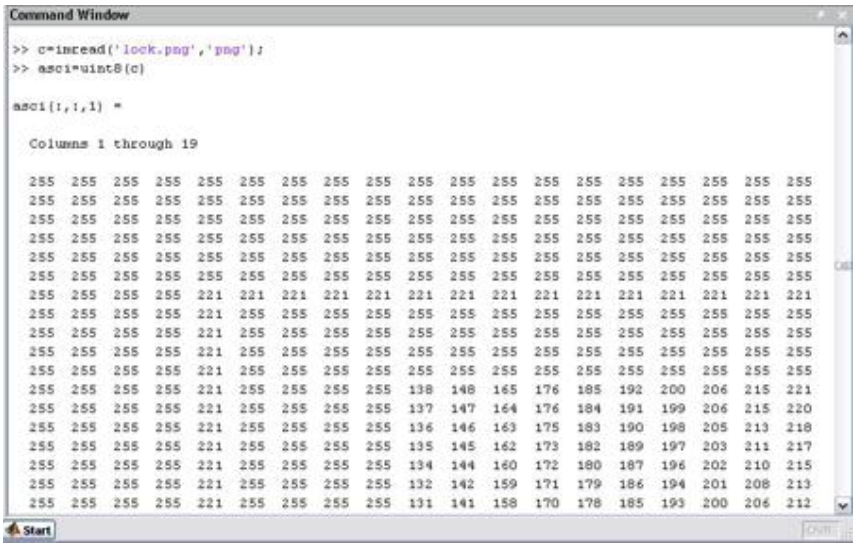
sedangkan pada baris kedua berfungsi untuk merubah gambar menjadi matriks dimana fungsi uint8() digunakan untuk mengubah string menjadi bilangan integer.



Gambar 3.3 Pengetikan kode didalam MATLAB

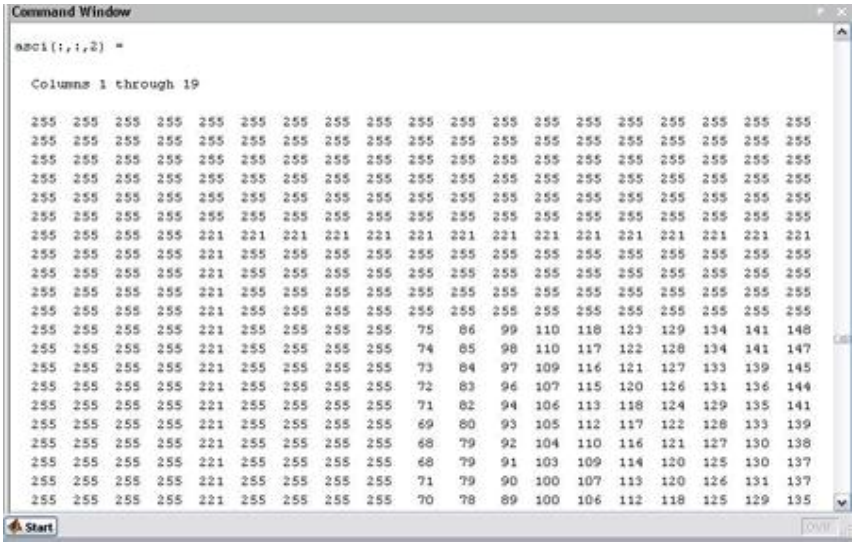
Hasilnya :

asci(: , : , 1) Menandakan bahwa matriks yang ditampilkan adalah komposisi matriks dari warna pertama yaitu merah (Red), Gambar 3.4



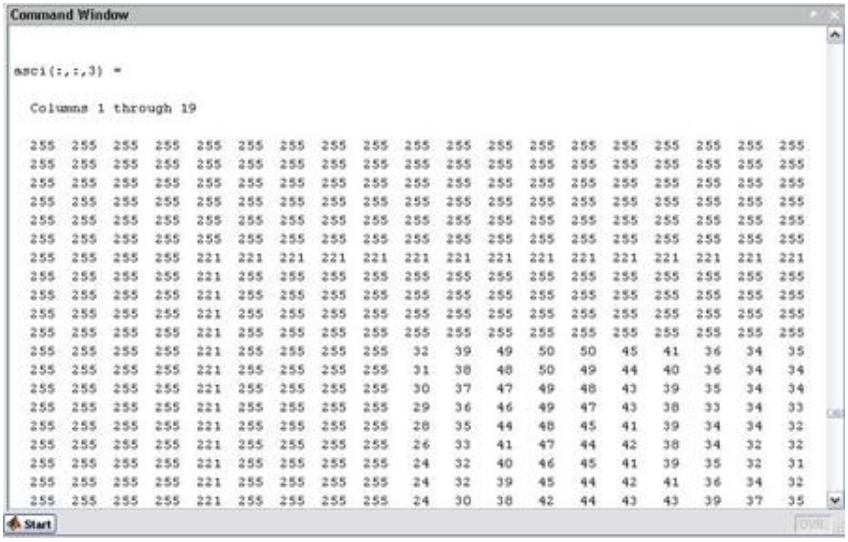
Gambar 3.4 Matriks Gambar untuk warna merah

asci(: , : , 2) Menandakan bahwa matriks yang ditampilkan adalah komposisi matriks dari warna kedua yaitu hijau (Green), Gambar 3.5.



Gambar 3.5 Matriks Gambar untuk warna hijau

asci(: , : , 3) Menandakan bahwa matriks yang ditampilkan adalah komposisi matriks dari warna ketiga yaitu biru (Blue), gambar 3.6.



Gambar 3.6 Matriks Gambar untuk warna biru

III.1.7 Metode SVD

Singular Value Decomposition (SVD) merupakan operator kompak T pada Hilbert Space yang didefinisikan sebagai eigenvalue dan operator akar pangkat dari T*T (T* merupakan adjoint dari T dan akar diambil dari operator sense)

Singular value merupakan bilangan nonnegative dan dinyatakan dengan urutan menurun, sebuah matrik dapat dikomposisi dalam bentuk *USV*, dengan U dan V merupakan matrik ortogonal dan S merupakan matrik diagonal, dengan singular value pada matrik diagonal nya maka hal ini disebut dengan singular value decomposition.

$$\mathbf{A}_{NXP} = \mathbf{U}_{NXN}\, \mathbf{S}_{NXP}\, \mathbf{V}_{PXP}$$

Sebagai contoh, misalkan sebuah matrik

$$A = \begin{bmatrix} 2 & 4 \\ 1 & 3 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

Maka

$$A.A^T = \begin{bmatrix} 2 & 4 \\ 1 & 3 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 2 & 1 & 0 & 0 \\ 4 & 3 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 20 & 14 & 0 & 0 \\ 14 & 10 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Dicari eigenvaluenya

$$\begin{bmatrix} 20-\lambda & 14 & 0 & 0 \\ 14 & 10-\lambda & 0 & 0 \\ 0 & 0 & -\lambda & 0 \\ 0 & 0 & 0 & -\lambda \end{bmatrix} \mathbf{x} = (W - \lambda I)\mathbf{x} = 0$$

Dari |W-αI didapatkan 4 buah eigenvalue

α=0,α=0

α=15+ akar pangkat dari 221.5~29.883

α=15- akar pangkat dari 221.5~0.117

kemudian didapat persamaan berikut dengan menstubtitusikan eigenvalue pertama

19.883x1 + 14x2 = 0

14x1 + 9.883x2 = 0

x3=0

x4=0

Solusi yang memenuhi persamaan diatas x1= -0.58, x2=0.82, x3 dan x4 = 0 (solusi ini menjadi kolom kedua matriks U)

Dari eigenvalue yang didapatkan

-9.883x1 + 14x2 = 0

14x1 + 19.883x2 = 0

x3 = 0

x4 = 0

solusi yang memenuhi persamaan tersebut x1=0.82, x2= -0.58 dan x3=x4=0 solusi ini menjadi kolom pertama matriks U

$$U = \begin{bmatrix} 0.82 & -0.58 & 0 & 0 \\ 0.58 & 0.82 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Demikian pula $A^T.A$ akan mengisi kolom dari matriks V maka dilakukan analisis serupa

$$A.A^T = \begin{bmatrix} 2 & 1 & 0 & 0 \\ 4 & 3 & 0 & 0 \end{bmatrix} \begin{bmatrix} 2 & 4 \\ 1 & 3 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

Didapatkanlah

$$V = \begin{bmatrix} 0.40 & -0.91 \\ 0.91 & 0.40 \end{bmatrix}$$

Akhirnya S merupakan akar kuadrat dari $A^T.A$ dan didapatkan

$$S = \begin{bmatrix} 5.47 & 0 \\ 0 & 0.37 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

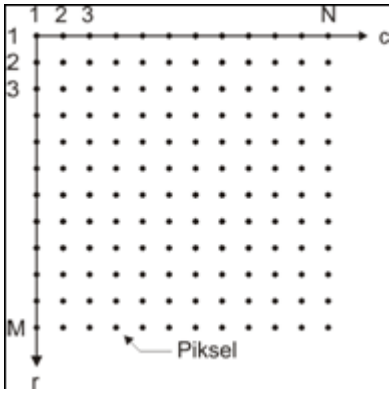
Matrik S ini merupakan singular value dari matriks A

III.1.8 Analisis Gambar Digital

Sebuah gambar didefinisikan sebagai fungsi dua dimensi, f(x,y) dimana x dan y adalah koordinat spasial. Amplitudo fungsi pada setiap pasangan titik (x,y) merupakan intensitas atau kecemerlangan gambar pada titik tersebut (Gonzales, 2004). Gray level digunakan untuk menyebut intensitas dari gambar monokrom atau satu warna. Gambar berwarna dibentuk oleh kombinasi dari beberapa gambar monokrom, misalnya RGB dibentuk oleh kombinasi tiga warna yaitu, merah, hijau dan biru. Karena itu banyak teknik pemrosesan gambar monokrom dapat digunakan untuk gambar berwarna dengan memproses tiga komponen gambar tersebut secara terpisah.

Sebuah gambar yang direkam oleh kamera merupakan gambar analog. Untuk dapat diolah oleh komputer, gambar analog tersebut harus dikonversi ke bentuk gambar digital. Proses konversi nilai koordinat (x, y) disebut sampling, dan proses konversi nilai amplitudo f disebut quantisasi. Ketika x, y dan amplitude f semuanya bernilai diskrit, maka gambar tersebut dapat dikatakan sebagai gambar digital.

Gambar digital dapat di kodekan dalam bentuk matriks dimana indeks baris dan kolomnya menyatakan suatu titik pada gambar tersebut dan elemen matriksnya (yang disebut sebagai piksel) menyatakan tingkat keabuan pada titik tersebut. Pada gambar berikut ini ditampilkan contoh koordinat suatu piksel dari gambar digital dengan x merepresentasikan baris dan y merepresentasikan kolom.



Gambar 3.7 Representasi Baris dan Kolom Pada Data Gambar Digital

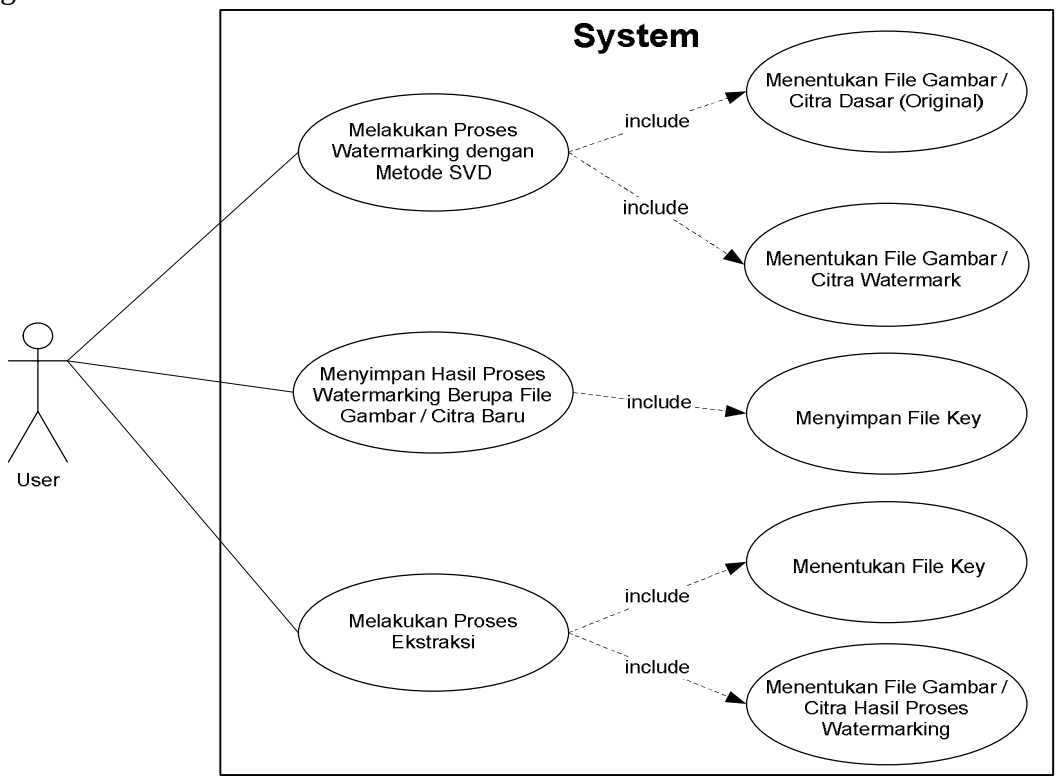
Data gambar digital yang akan digunakan dalam penelitian ini adalah gambar digital sesuai pendeskripsian di atas,

III.2 Use Case Diagram

Use case diagram digunakan untuk menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Sebuah use case merepresentasikan sebuah interaksi antara aktor dengan sistem. Use case diagram pada Gambar 3.8 yang menggambarkan bagaimana proses yang terjadi pada aplikasi *watermarking* transfer data.

III.2.1 Use Case Diagram *Watermarkingdan* Ekstraksi

Use case diagram proses *watermarking dan Ekstraksi* dapat dilihat pada gambar 3.8



Gambar 3.8 Use case diagram proses watermarking dan Ekstraksi citra

III.2.2 Skenario Use Case Watermarking dan Ekstraksi

Skenario setiap bagian pada use case menunjukkan proses apa yang terjadi pada setiap bagian didalam use case tersebut, dimana user memberikan perintah pada setiap bagian dan respon apa yang diberikan oleh sistem kepada user setelah user memberikan perintah pada setiap bagian – bagian use case.

III.2.2.1 Menentukan File Citra Original

Skenario proses menentukan file citra original dapat dilihat pada tabel 3.2

Tabel 3.2 Tabel skenario proses menentukan file citra original

Identifikasi		
Nomor	1	
Nama	*Menentukan File Citra Original*	
Tujuan	Melakukan proses *watermarking*	
Deskripsi	Pada saat aplikasi *watermarking* dibuka *user* dapat melakukan pemilihan *file* / input *file*	
Aktor	*User*	
Skenario Utama		
Kondisi awal	Aplikasi *watermarking* dalam keadaan terbuka	
Aksi Aktor		**Reaksi Sistem**
1. Memasukkan *file* yang akan di *watermarking*		2. Memberikan informasi status *file* yang akan di *watermarking*
3. Menekan tombol *Run*		4. Memberikan informasi status watermarking pada *user*.
Skenario Alternatif – Autentikasi Gagal		
Aksi Aktor		**Reaksi Sistem**
1. Memasukkan *file* yang akan di *watermarking*, jika citra dasar lebih kecil dari citra *watermark*		2. Memberikan informasi bahwa *watermarking* gagal.
Kondisi akhir	1. File yang *watermarking* akan berhasil jika citra dasar lebih besar ukurannya dari citra *watermark* 2. File yang watermarking akan gagal jika citra dasar lebih kecil ukurannya dari citra *watermark*	

III.2.2.2 Menentukan File Citra watermark

Skenario proses menentukan file citra original dapat dilihat pada tabel 3.3

Tabel 3.3 Tabel skenarioproses menentukan file citra watermark

Identifikasi	
Nomor	2
Nama	*Menentukan file Citra watermark*
Tujuan	User menentukan *file* yang akan di *watermarking*
Deskripsi	Sistem menerima *file* yang dipilih user dan melakukan

	watermarking
Aktor	User
Skenario Utama	
Kondisi awal	Aplikasi *watermarking* dalam keadaan baru dibuka
Aksi Aktor	**Reaksi Sistem**
1. Memilih dan menginputkan *file* yang akan di *watermarking*	2. Memberikan informasi status *file* yang akan di *watermarking*
Skenario Alternatif – Autentikasi Gagal	
Aksi Aktor	**Reaksi Sistem**
1. Memasukkan *file* yang akan di *watermarking,* jika citra dasar lebih kecil dari citra *watermark*	2. Memberikan informasi bahwa *watermarking* gagal.
Kondisi akhir	1. File yang *watermarking* akan berhasil jika citra dasar lebih besar ukurannya dari citra *watermark* *2.* File yang watermarking akan gagal jika citra dasar lebih kecil ukurannya dari citra *watermark*

III.2.2.3 Watermarking SVD method

Skenario proses *watermarking SVD method* dapat dilihat pada tabel 3.5

Tabel 3.5 Tabel skenario proses watermarking SVD method

Identifikasi	
Nomor	4
Nama	*Watermarking SVD method*
Tujuan	Melakukan proses *watermarking* dengan metode SVD
Deskripsi	Sistem telah menerima *file* yang dipilih *user*
Aktor	*User*
Skenario Utama	
Kondisi awal	Aplikasi *watermarking* dalam keadaan dibuka Terdapat file yang akan di *watermarking*
Aksi Aktor	**Reaksi Sistem**
1. Menekan Tombol *Run*	2. Mengkomposisi 2 buah *file* citra menjadi matrix U, S, V
Skenario Alternatif – Autentikasi Gagal	
Aksi Aktor	**Reaksi Sistem**
-	-
Kondisi akhir	1. 2 *file* citra yang akan di *watermark* sudah di komposisi kedalam matrik

III.2.2.4 Save file

Skenario proses *Save File* dapat dilihat pada tabel 3.6

Tabel 3.6 Tabel skenario proses Save file

Identifikasi		
Nomor	5	
Nama	*save file*	
Tujuan	menyimpan *file* yang telah di *watermark*	
Deskripsi	Sistem akan Menyimpan *file* yang telah di *watermark*	
Aktor	*User*	
Skenario Utama		
Kondisi awal		
Aksi Aktor		**Reaksi Sistem**
1. Menekan Tonbol *Save*		2. Sistem menyimpan *file* yang telah di *watermark* di tempat yang telah ditentukan
Skenario Alternatif – Autentikasi Gagal		
Aksi Aktor		**Reaksi Sistem**
-		-
Kondisi akhir	1. *File* citra asli dan *file* citra watermark telah melalui proses *watermarking*	

III.2.2.5 Menentukan File Key

Skenario proses menentukan file Key dapat dilihat pada tabel 3.7

Tabel 3.7 Tabel skenario proses menentukan File Key

Identifikasi		
Nomor	6	
Nama	*Menentukan file Key*	
Tujuan	Melakukan proses *Ekstraksi*	
Deskripsi	Pada saat aplikasi *watermarking* dibuka *user* dapat melakukan pemilihan *File Key, dan File* Citra hasil *watermarking*	
Aktor	*User*	
Skenario Utama		
Kondisi awal	Aplikasi *watermarking* dalam keadaan terbuka	
Aksi Aktor		**Reaksi Sistem**
1. Menentukan metode untuk proses ekstraksi		2. Memberikan informasi status *file* yang akan di *Ekstraksi*

3. Menekan tombol *Run*	4. Memberikan informasi status Ekstraksi pada *user*.
Skenario Alternatif – Autentikasi Gagal	
Aksi Aktor	**Reaksi Sistem**
-	-
Kondisi akhir	1. *File* kunci telah dipilih

III.2.2.6 Menentukan File Citra Hasil Watermarking

Skenario proses menentukan file citra original dapat dilihat pada tabel 3.8

Tabel 3.8 Tabel skenario proses menentukan file citra original

Identifikasi	
Nomor	7
Nama	*Menentukan File Citra Hasil Watermarking*
Tujuan	Melakukan proses *Ekstraksi*
Deskripsi	Pada saat aplikasi *watermarking* dibuka *user* dapat melakukan pemilihan *file* / input *file*
Aktor	*User*
Skenario Utama	
Kondisi awal	Aplikasi *watermarking* dalam keadaan terbuka
Aksi Aktor	**Reaksi Sistem**
1. Memasukkan *file* yang akan di *Ekstraksi*	2. Memberikan informasi status *file* yang akan di *Ekstraksi*
3. Menekan tombol *Run*	4. Memberikan informasi status Ekstraksi pada *user*.
Skenario Alternatif – Autentikasi Gagal	
Aksi Aktor	**Reaksi Sistem**
-	-
Kondisi akhir	1. Citra hasil watermarking sudah di *input*

III.2.2.7 Ekstraksi

Skenario proses *Ekstraksi* Citra *Watermark* dapat dilihat pada tabel 3.9

Tabel 3.9 Tabel skenario proses Ekstraksi Citra watermark

Identifikasi	
Nomor	8
Nama	*Ekstraksi*

Tujuan	Melakukan proses *Ekstraksi*	
Deskripsi	Pada saat aplikasi *watermarking* dibuka *user* dapat melakukan pemilihan *file* / input *file* yang telah di *watermarking*	
Aktor	*User*	
Skenario Utama		
Kondisi awal	Server Aktif	
Aksi Aktor		**Reaksi Sistem**
1. Memasukkan *file* yang akan di *Ekstraksi*		2. Memberikan informasi status *file* yang akan di *Ekstraksi*
3. Menekan Tombol *Run*		4. Memberikan informasi status *Ekstraksi* pada *user*.
Skenario Alternatif – Autentikasi Gagal		
Aksi Aktor		**Reaksi Sistem**
-		-
Kondisi akhir	1. *File* Citra asli dan *File* Citra *watermark* telah berhasil dipisahkan	

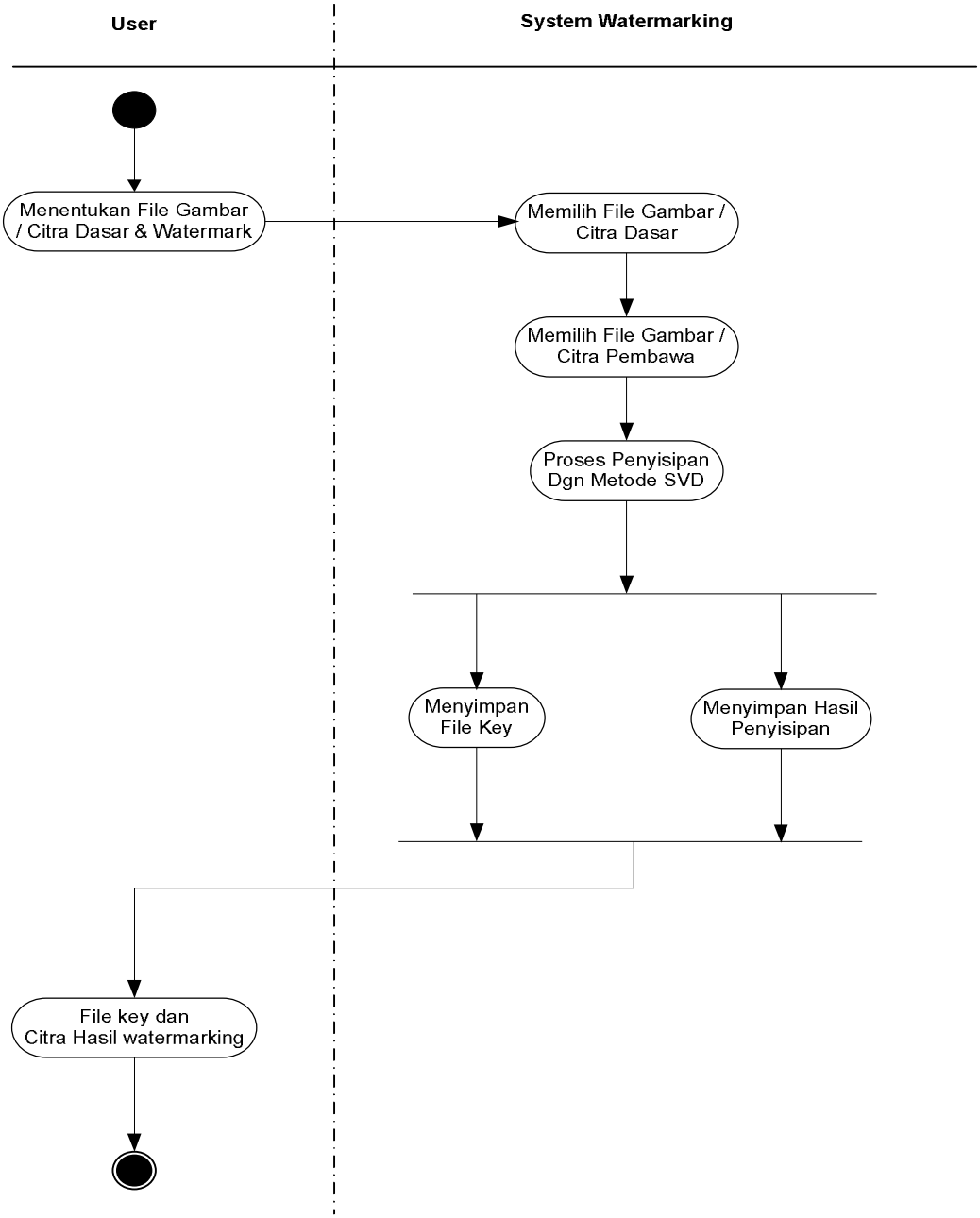
III.3 Activity Diagram

Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. *Activity diagram* merupakan *state diagram* khusus, di mana sebagian besar *state* adalah *action* dan sebagian besar transisi di-*trigger* oleh selesainya *state* sebelumnya (*internal processing*). Oleh karena itu *activity diagram* tidak menggambarkan behaviour internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum. Sebuah aktivitas dapat direalisasikan oleh satu *use case* atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara *use case* menggambarkan bagaimana aktor menggunakan sistem untuk melakukan

aktivitas. Gambar 3.9 dan gambar 3.10 menunjukan activity diagram yang dilakukan aplikasi.

III.2.3 Aktivity Diagram Watermarking

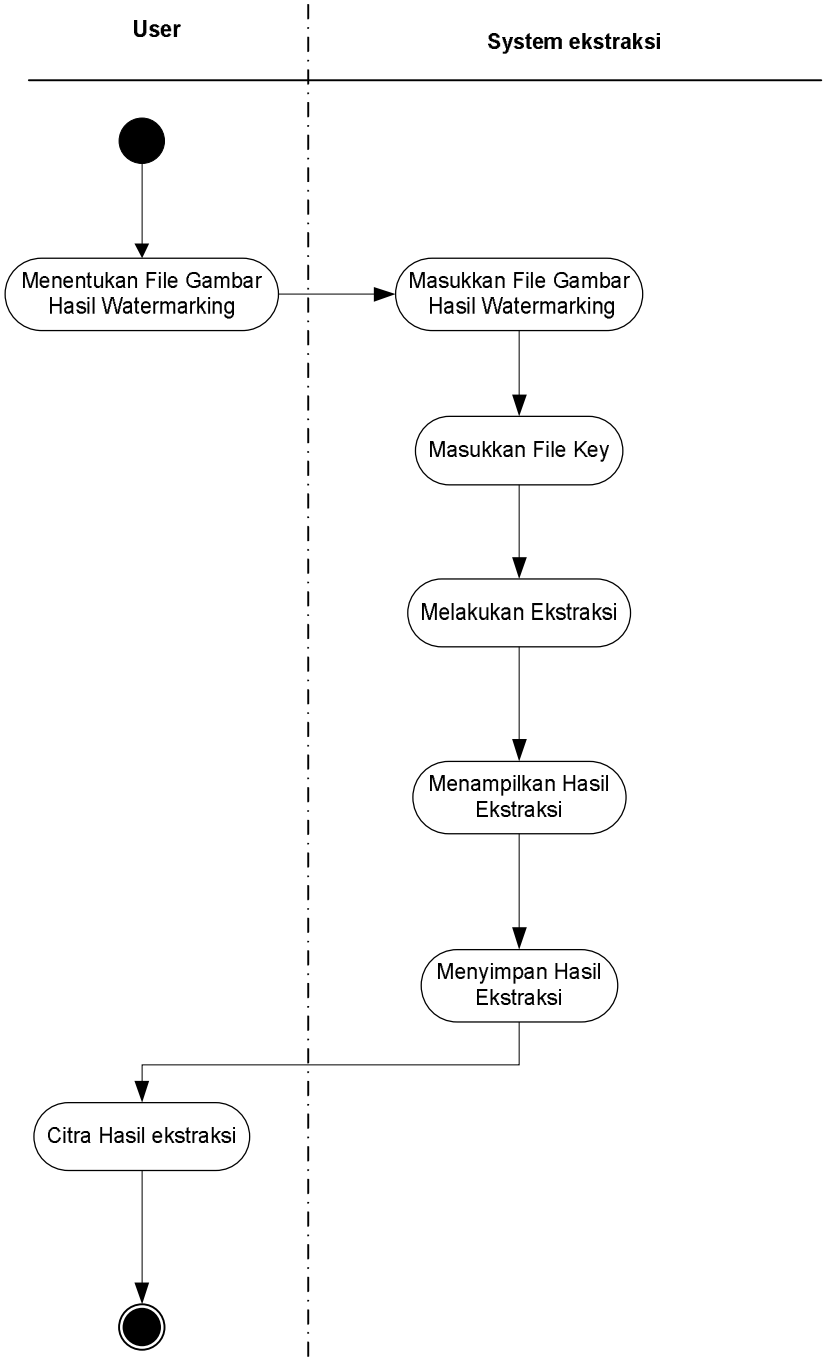
Activity diagram proses *watermarking* dapat dilihat pada gambar 3.9



Gambar 3.9 Activity diagram proses watermarking

III.2.2 Activity Diagram Ekstraksi

Activity diagram proses *Ekstraksi* dapat dilihat pada gambar 3.10



Gambar 3.10 Activity diagram proses Ekstraksi

III.4 Sequence Diagram

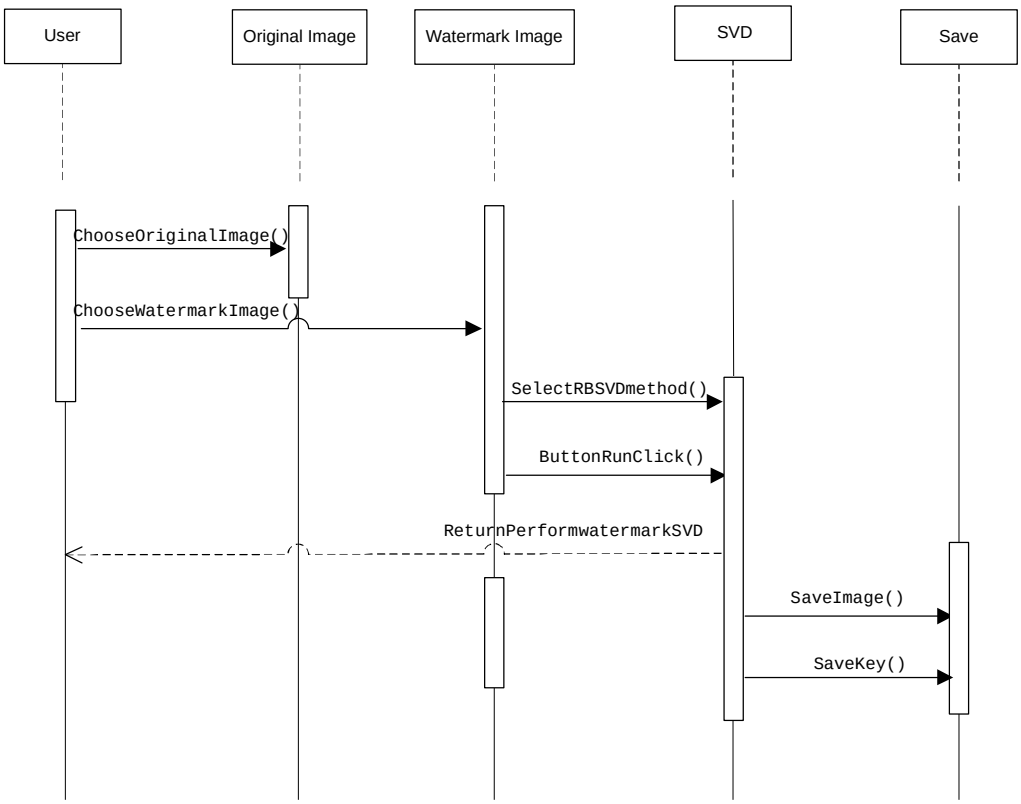
Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri atar dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait).

Sequence diagram biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah *event* untuk menghasilkan *output* tertentu. Diawali dariapa yang men-*trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan *output* apa yang dihasilkan.

Sequence Diagram yang digambarkan dalam perancangan sistem dapat dilihat pada gambar 3.11 dan gambar 3.12.

III.4.1 Sequence Diagram Watermarking

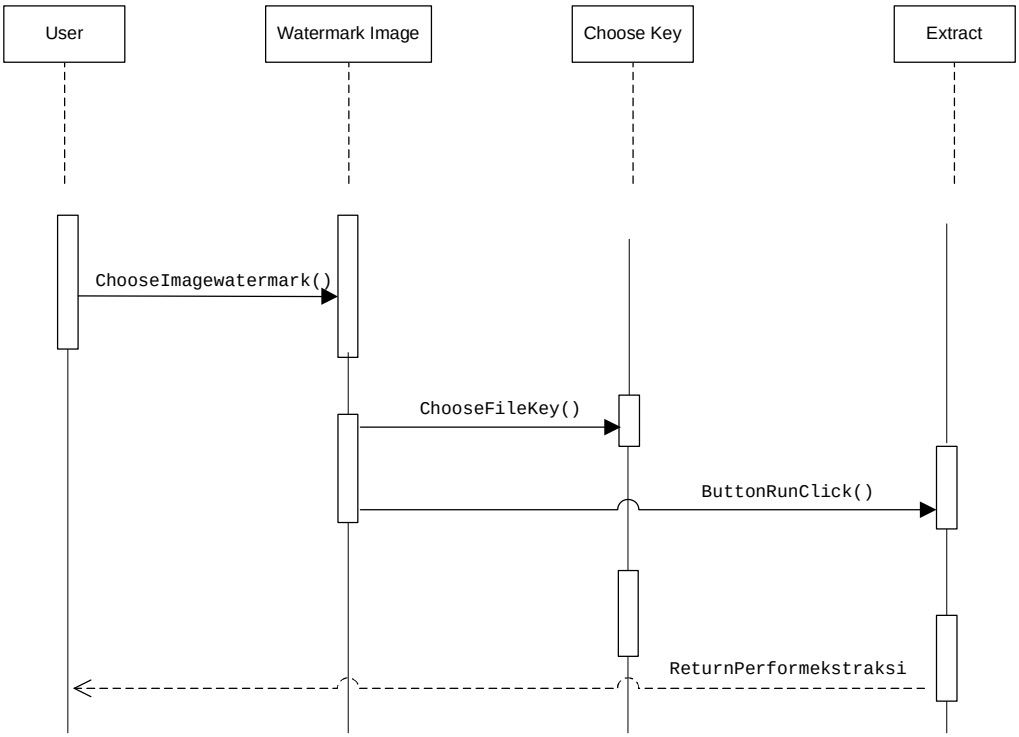
Sequence diagram proses *watermarking* secara umum dapat dilihat pada gambar 3.11



Gambar 3.11 Sequence diagram proses watermarking

III.4.2 Sequence Diagram Ekstraksi

Sequence diagram proses *Ekstraksi* secara umun dapat dilihat pada gambar 3.12



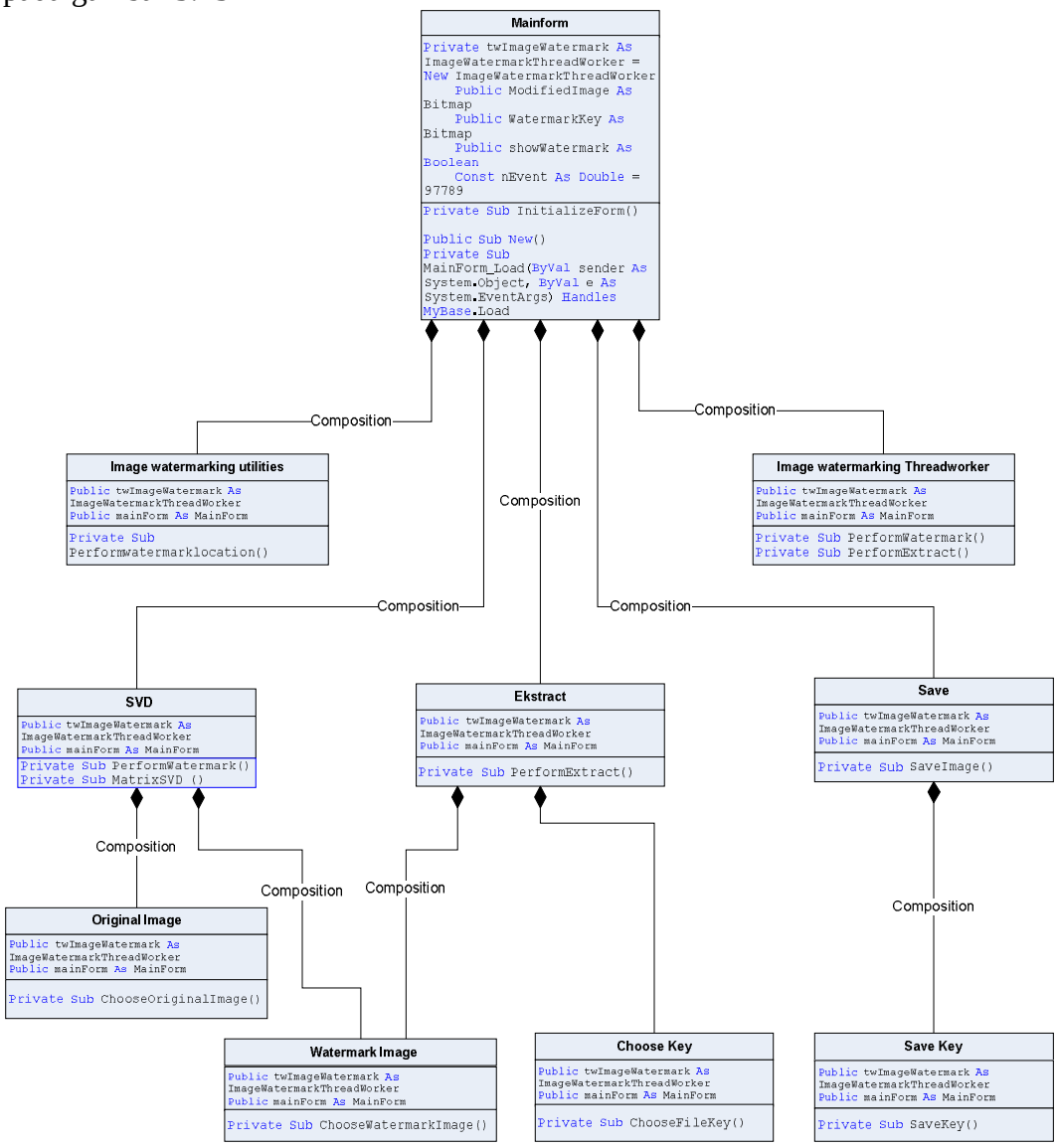
Gambar 3.12 Sequence diagram proses Ekstraksi

III.5 Class Diagram

Class Diagram menggambarkan keadaan suatu sistem (*atribut*), dan memberikan pelayanan untuk menyelesaikan keadaan tersebut (metoda). Class diagram dapat dilihat pada gambar 3.13.

III.5.1 Class Diagram Watermarking dan Ekstraksi

Class Diagram Proses Watermarking dan Proses Ekstraksi dapat dilihat pada gambar 3.13



Gambar 3.13 Perancangan Class

III.6 Perancangan Sistem

Perancangan sistem adalah suatu proses yang menggambarkan bagaimana suatu sistem dibangun untuk memenuhi kebutuhan pada fase analisis. Adapun tahapan yang dilakukan dalam perancangan sistem ini membahas mengenai tujuan perancangan sistem, dan perancangan antar muka.

III.6.1 Tujuan Perancangan Sistem

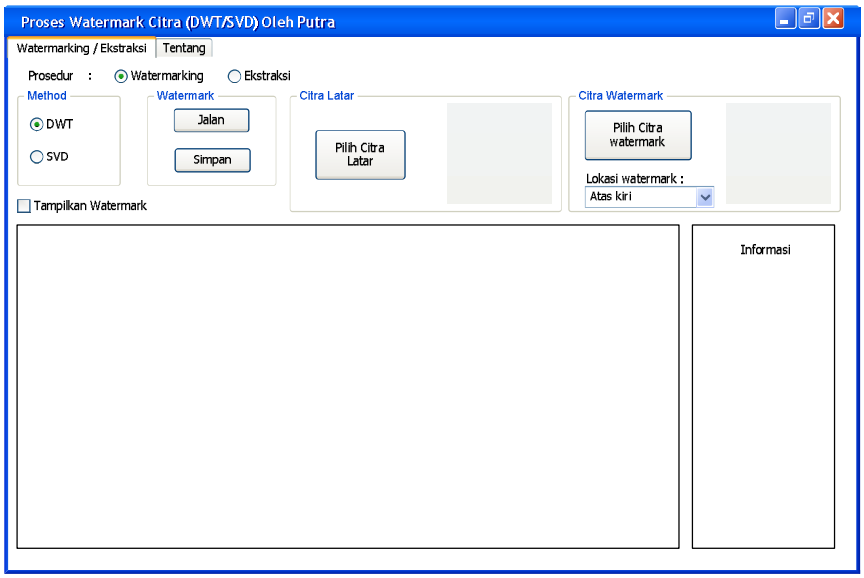
Perancangan sistem merupakan tindak lanjut dari tahap analisa. Perancangan sistem bertujuan untuk memberikan gambaran sistem yang akan dibuat. Dengan kata lain perancanagn sistem didefinisikan sebagai penggambaran atau pembuatan sketsa dari beberapa elemen yang terpisah kedalam satu kesatuan yang utuh dan berfungsi. Selain itu juga perancangan bertujuan untuk lebih mengarahkan sistem yang terinci, yaitu pembuatan perancangan yang jelas dan lengkap yang nantinya akan digunakan untuk pembuatan simulasi. Aplikasi *watermarking* ini dibuat dengan sederhana, sehingga diharapkan user dapat dengan mudah menggunakan aplikasi *watermarking* ini.

III.6.2 Perancangan Antar Muka

Perancangan antarmuka dibutuhkan untuk mewakili keadaan sebenarnya dari aplikasi yang akan dibangun, berikut akan disajikan perancangan antarmuka dari aplikasi yang akan dibangun:

III.6.3 Menu Watermarking

Antar muka menu *watermarking* digunakan untuk melakukan proses *watermarking* file. Gambar 3.14 menampilkan rancangan menu *watermarking*.



Gambar 3.14 Rancangan antar muka menu Watermarking

Keterangan tentang tombol-tombol yang digunakan dalam aplikasi *watermarking* dapat dilihat pada Tabel 3.8

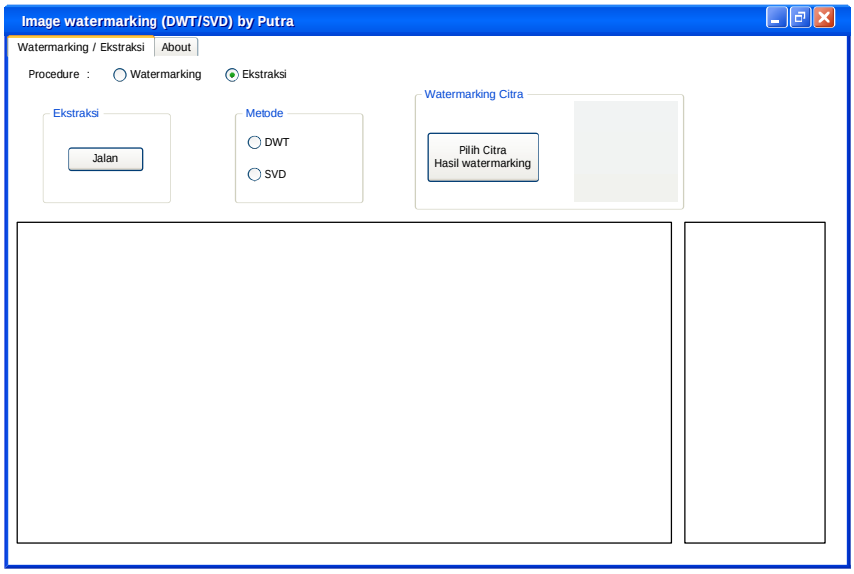
Tabel 3.8. Keterangan tampilan menu watermarking

No Objek	Jenis Objek	Fungsi Objek
1	*Watermarking*/Ekstraksi	Untuk masuk kehalaman *watermarking* / de*watermarking*
2	About	Untuk menampilkan halaman tentang *software* dan pembuat
3	Select original image	Untuk memilih gambar sebagai carrier image
4	Select watermark image	Untuk memilih gambar atau file yang kan disisipkan
5	Procedure	Untuk memilih proses yang akan dilakukan (*watermarking*/de*watermarking*)
6	Status bar	Untuk mencantumkan informmasi proses yang telah dilakuakan
7.	*Run*	Untuk melakukan proses *watermarking* file

8	Save	Untuk Menyimpan File yang sudah diwatermarking

III.6.4 Menu Ekstraksi

Antar muka menu *ekstraksi* digunakan untuk melakukan proses ekstraksi file. Gambar 3.15 menampilkan rancangan menu Ekstraksi .



Gambar 3.15 Rancangan antar muka menu Ekstraksi

Keterangan tentang tombol-tombol yang digunakan dalam aplikasi *Ekstraksi* dapat dilihat pada Tabel 3.9

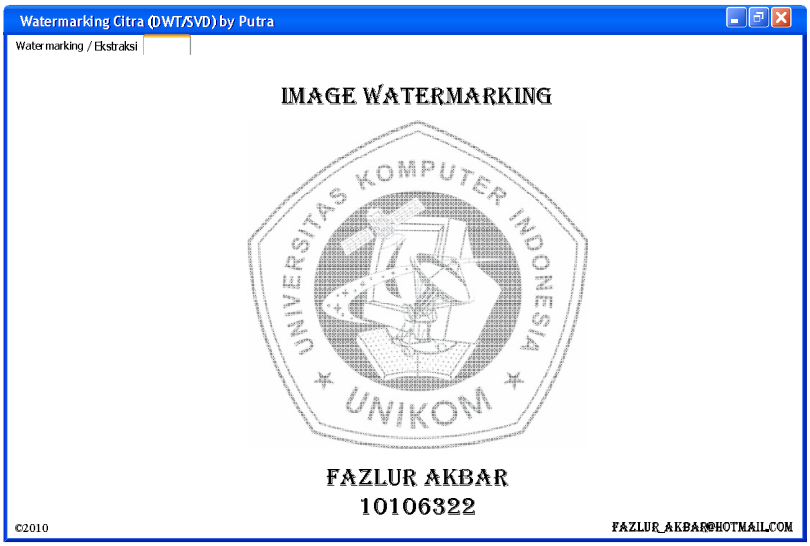
Tabel 3.9. Keterangan tampilan menu Ekstraksi

No Objek	Jenis Objek	Fungsi Objek
1	*Watermarking*/Ekstraksi	Untuk masuk kehalaman *watermarking* / de*watermarking*
2	About	Untuk menampilkan halaman tentang *software* dan pembuat
3	Select watermarking image	Untuk memilih gambar hasil proses watermarking
5	Procedure	Untuk memilih proses yang akan dilakukan (*watermarking*/Ekstraksi)

6	Status bar	Untuk mencantumkan informmasi proses yang telah dilakuakan
7.	*Run*	Untuk melakukan proses Ekstraksi file

III.6.5 Menu About

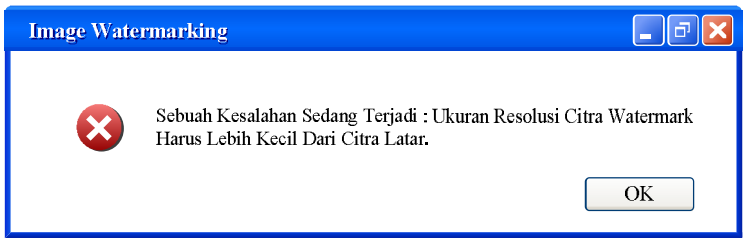
Antar muka menu about digunakan untuk menampilkan informasi mengenai *software* dan juga pembangun *software*. Gambar 3.16 menampilkan rancangan menu about.



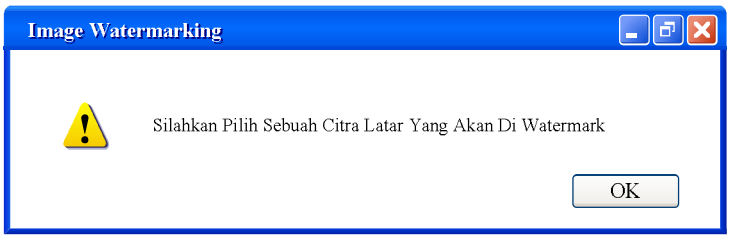
Gambar 3.16 Rancangan antar muka menu About

III.6.6 Pesan Kesalahan

Antar muka Pesan Kesalahan digunakan untuk menampilkan peringatan bahwa telah terjadi sebuah kesalahan dalam penggunaan program.Gambar 3.16 dan gambar 3.17 menampilkan Rancangan Pesan Kesalahan.



Gambar 3.16 Rancangan antar muka Pesan Kesalahan 1



Gambar 3.17 Rancangan antar muka Pesan Kesalahan 2