



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB III

METODOLOGI DAN PERANCANGAN SISTEM

3.1 Metodologi Penelitian

Metodologi yang digunakan dalam penelitian ini adalah sebagai berikut.

a. Studi Literatur

Bertujuan untuk mencari dan mempelajari teori-teori dasar dari *Digital Watermarking*, *Discrete Haar Wavelet Transform*, *Least Significant Bit*, *Peak Signal to Noise Ratio*, *Technology Acceptance Model*, dan Gambar Bitmap.

b. Perancangan Aplikasi

Pada tahap ini dibuat perancangan dari aplikasi seperti flowchart, DFD, dan tampilan antarmuka aplikasi. Perancangan aplikasi mengikuti alur sebagai berikut.

1. Pemilihan proses *embedding* atau ekstraksi *watermark*.
2. Pemilihan gambar terkait.
3. Proses penyisipan atau ekstraksi *watermark*.
4. Menampilkan hasil akhir.

Hasil akhir yang akan diuji pada penelitian ini adalah *fidelity* dengan menghitung nilai PSNR, *imperceptibility* dengan melihat secara visual hasil *embedding*, dan *recovery* melalui proses *extraction*.

c. Pembangunan Aplikasi

Pada tahap ini dilakukan pembangunan aplikasi berbasis *desktop* sesuai dengan spesifikasi yang telah ditentukan. Pembangunan aplikasi menggunakan bahasa pemrograman C# dan bahasa pemrograman XAML.

d. Pengujian dan Evaluasi

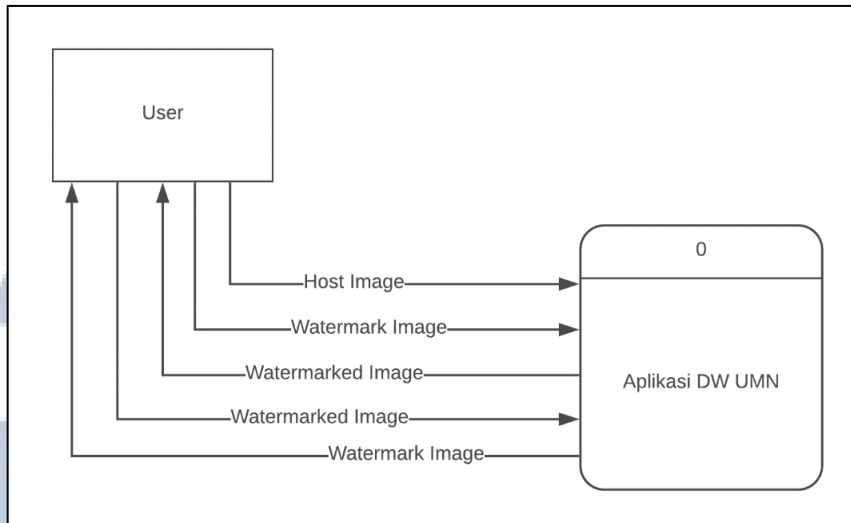
Pada tahap ini dilakukan pengujian aplikasi untuk menguji apakah aplikasi yang telah selesai dibangun sesuai dengan perancangan aplikasi tersebut. Evaluasi aplikasi dilakukan dengan meminta responden mahasiswa Universitas Multimedia Nusantara sebanyak 30 responden menggunakan metode *purposive sampling* untuk mencoba menggunakan aplikasi dan mengisi kuesioner menggunakan *Technology Acceptance Model* (TAM) yang diisi oleh responden setelah menggunakan aplikasi.

3.2 Perancangan Aplikasi

Perancangan sistem dibagi menjadi pembuatan Data Flow Diagram (DFD) pembuatan desain tampilan antarmuka, dan *flowchart* aplikasi. Aplikasi yang akan dibangun dibagi menjadi dua proses yaitu proses penyisipan *watermark* (*embedding*) dan proses ekstraksi *watermark* (*extraction*).

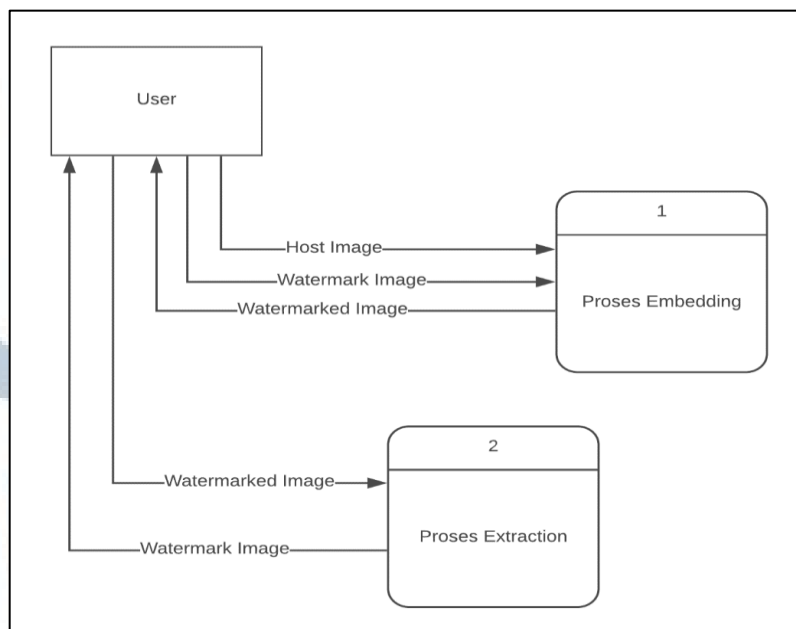
3.2.1 Data Flow Diagram

Aplikasi DW UMN memiliki *input* berupa *host image* dan *watermark image* dengan *output* *watermarked image* atau *input watermarked image* yang menghasilkan *output watermark image*. DFD aplikasi DW UMN dapat dilihat pada Gambar 3.1.



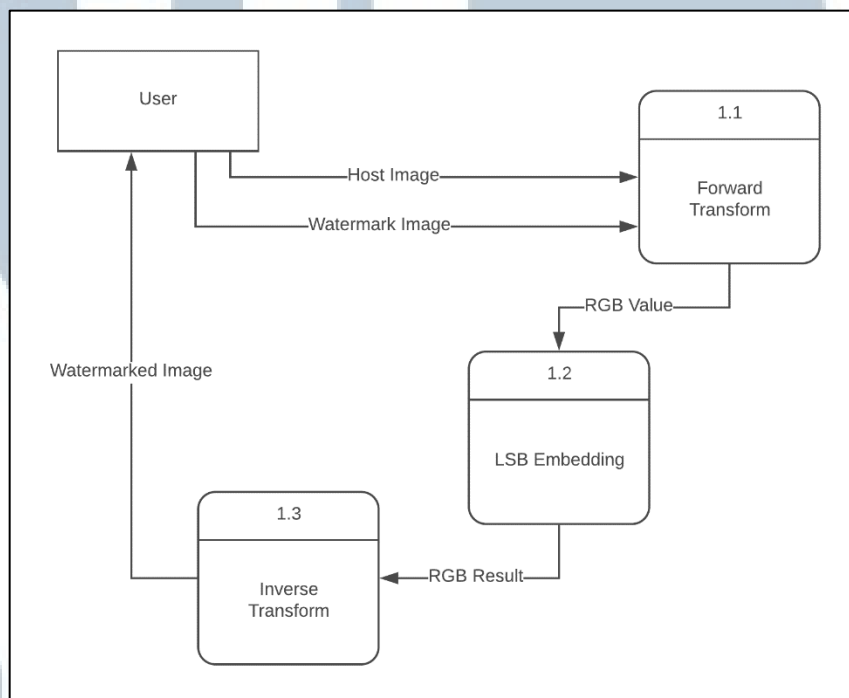
Gambar 3.1 Diagram Konteks Aplikasi DW UMN

Aplikasi terbagi dua proses yaitu proses *embedding* untuk menyisipkan *watermark* dengan *input* berupa *host image* dan *watermark image* dan proses *extraction* untuk mendapatkan *watermark* dari gambar dengan *input* berupa *watermarked image*. DFD Level 1 aplikasi DW UMN dapat dilihat pada Gambar 3.2.



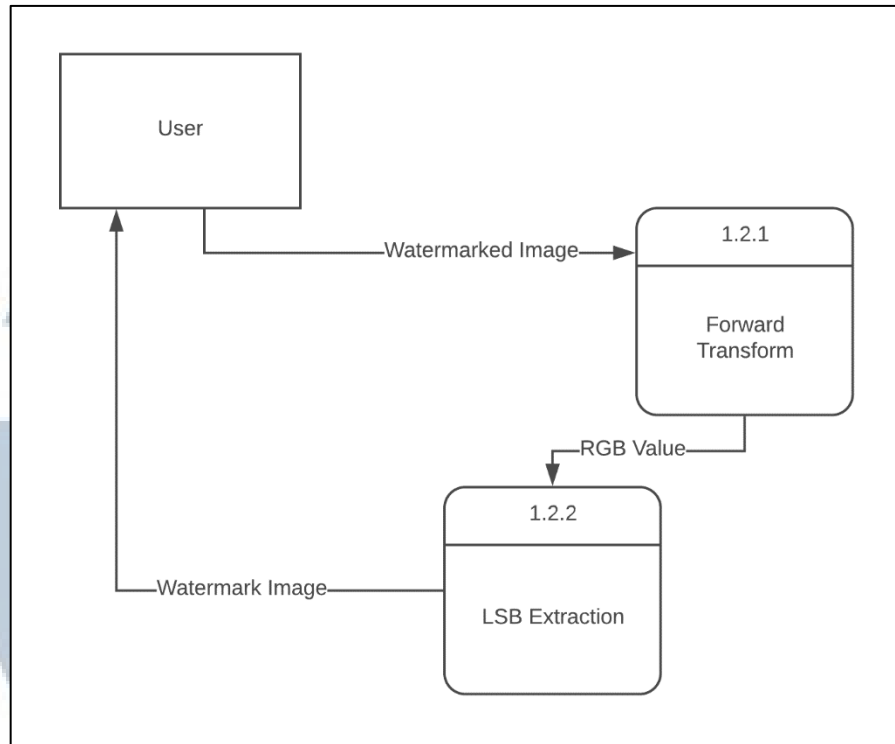
Gambar 3.2 DFD Level 0 Aplikasi DW UMN

Proses *embedding* terdiri dari proses *forward transform* yang menggunakan metode DHWT yang menghasilkan nilai frekuensi RGB gambar. Nilai RGB tersebut kemudian dilakukan proses *embedding* menggunakan metode LSB dengan nilai *watermark*. Setelah *embedding* selesai maka dikembalikan kembali dengan proses *inverse transform* menjadi gambar. DFD proses *embedding* dapat dilihat pada Gambar 3.3.



Gambar 3.3 DFD Level 1 Proses Embedding

Proses *extraction* terdiri dari proses *forward transform* yang menggunakan metode DHWT pada gambar kemudian dilakukan proses *extraction* menggunakan metode LSB. Setelah *extraction* selesai gambar *watermark* yang didapatkan akan dikembalikan kepada *user*. DFD proses *extraction* dapat dilihat pada Gambar 3.4.



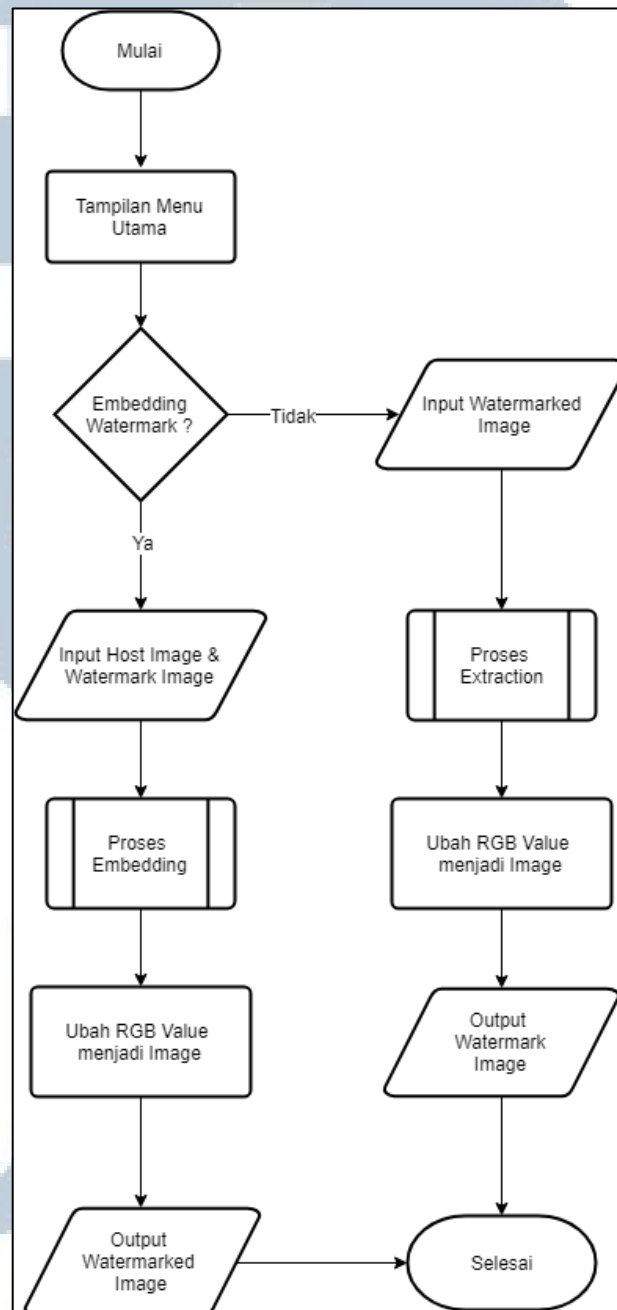
Gambar 3.4 DFD Level 1 Proses Extraction

3.2.2 Flowchart Aplikasi

Ketika aplikasi pertama kali dijalankan *user* diminta untuk memilih proses antara proses *embedding* atau proses *extraction*. Jika *user* memilih proses *embedding* maka *user* akan diminta *input* berupa dua gambar yaitu gambar *host* dan gambar *watermark* yang ingin disisipkan. Gambar *host* kemudian akan diproses dengan metode DHWT kemudian dilakukan penyisipan *watermark* melalui proses *embedding* dengan metode LSB. Hasil metode LSB kemudian dikembalikan menjadi bentuk gambar dan dikembalikan kepada *user* hasil proses *embedding*. Jika *user* memilih proses *extraction* maka *user* diminta *input* berupa gambar yang memiliki *watermark*. Gambar tersebut akan kemudian dilakukan proses *extraction* untuk mendapatkan *watermark* yang disisipkan. Nilai *watermark* yang didapatkan

kemudian diubah kembali menjadi gambar dan dikembalikan kepada *user*.

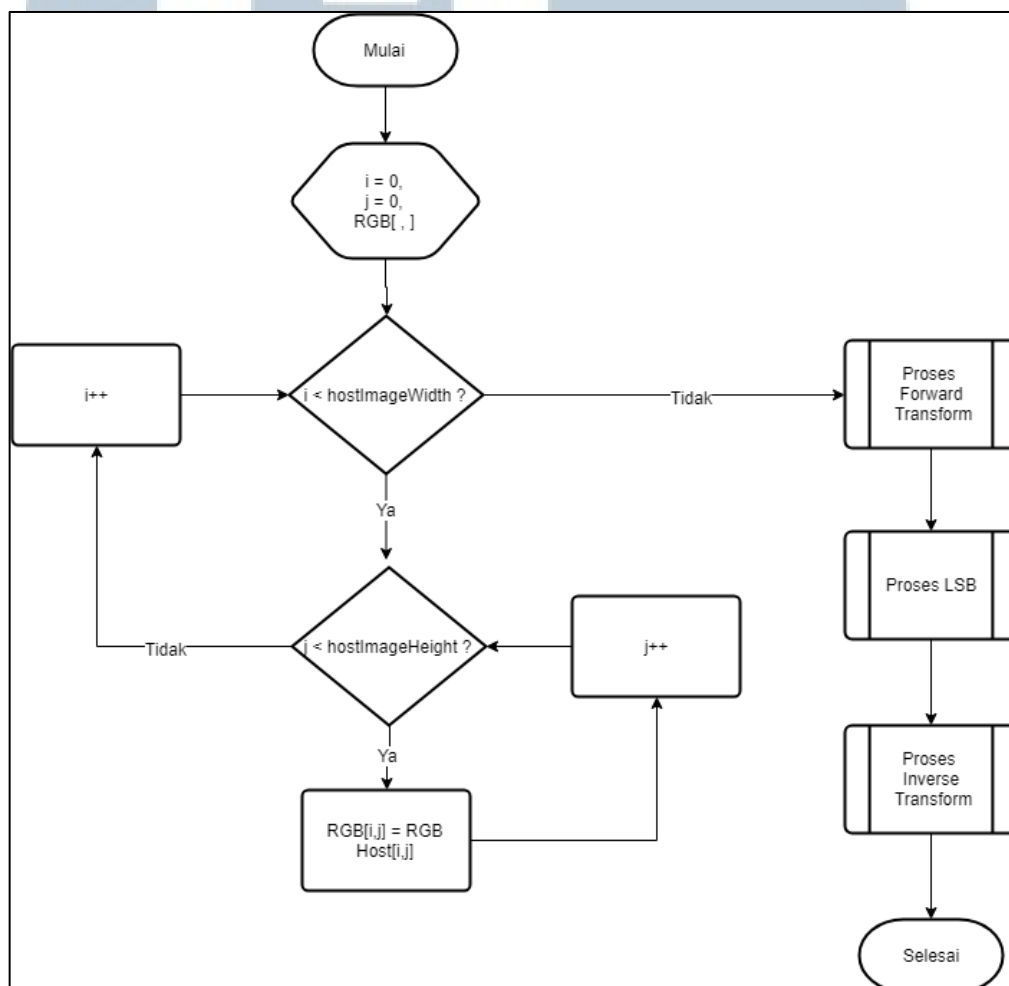
Flowchart aplikasi dapat dilihat pada Gambar 3.5.



Gambar 3.5 Flowchart Aplikasi DW UMN

Proses *embedding* dimulai dengan mengambil nilai RGB tiap *pixel* dimana variabel *hostImagewidth* didapatkan dari panjang *pixel* gambar *host* sedangkan

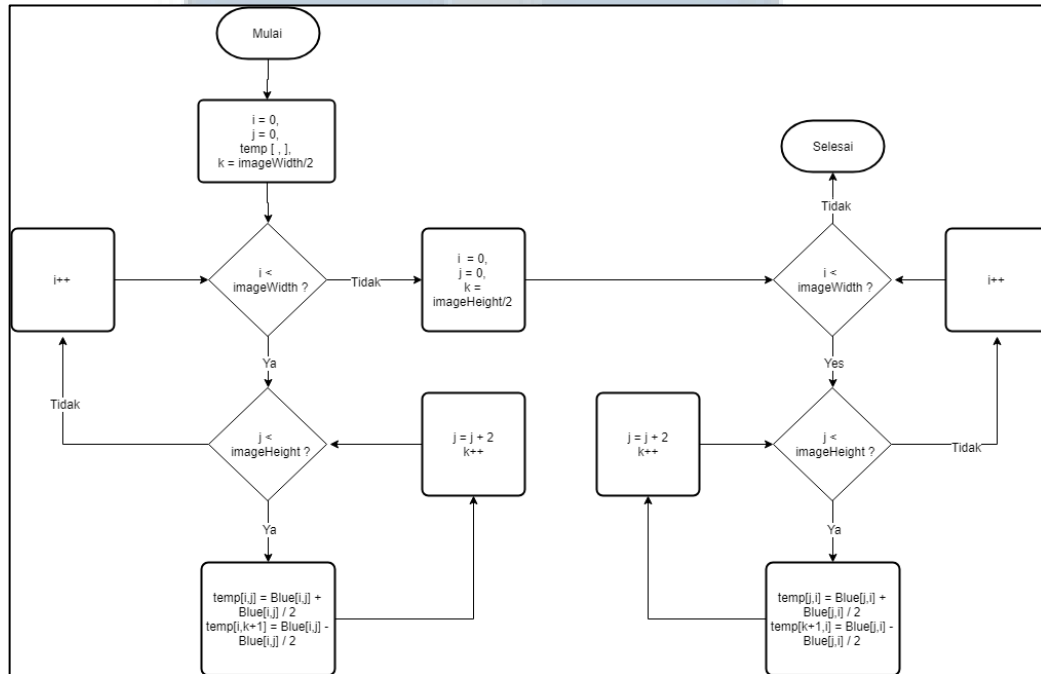
variabel *hostImageHeight* didapatkan dari lebar *pixel* gambar *host*. Nilai RGB tersebut kemudian akan dilakukan proses *forward transform* untuk menjadi nilai frekuensi yang dapat diubah-ubah. Nilai frekuensi tersebut kemudian dilakukan proses LSB untuk menyisipkan *watermark* ke dalamnya. Setelah *watermark* disisipkan maka nilai frekuensi dikembalikan dengan proses *inverse transform*. Flowchart proses Embedding dapat dilihat pada Gambar 3.6.



Gambar 3.6 Flowchart Proses Embedding

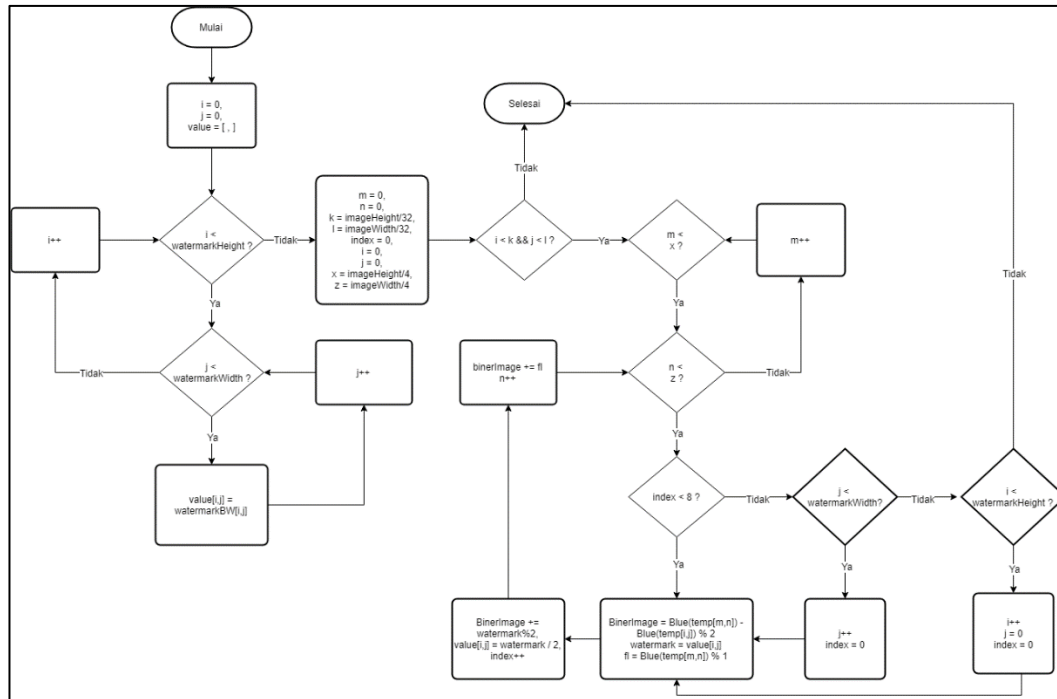
Proses *Forward Transform* dilakukan pada setiap baris dan kolom dari gambar dimana *imageWidth* merupakan nilai panjang gambar dan *imageHeight* merupakan nilai lebar gambar dan menggunakan Rumus 2.1 dan 2.2 dan dilakukan berulang

sampai seluruh baris dan kolom telah selesai. Flowchart proses Forward Transform dapat dilihat pada Gambar 3.7.



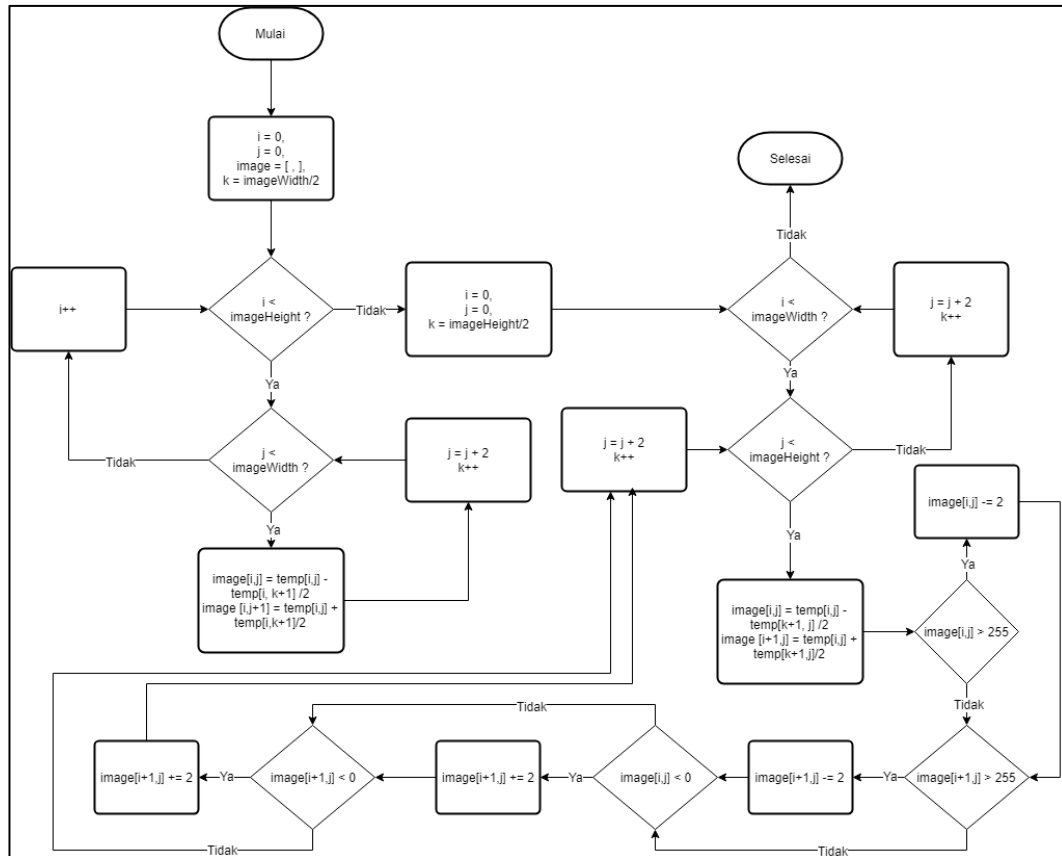
Gambar 3.7 Flowchart Proses Forward Transform

Proses LSB dimulai dengan mengambil nilai hitam putih setiap *pixel* dari gambar *watermark* yang ingin disisipkan dimana *watermarkHeight* adalah nilai lebar dari gambar *watermark* dan *watermarkWidth* adalah nilai panjang dari gambar *watermark*. Jika jumlah nilai *watermark* lebih sedikit dari jumlah nilai yang dapat disisipkan maka nilai *watermark* dapat disisipkan dengan proses LSB dengan mengubah *bit* terakhir dari nilai gambar menjadi *bit* dari *watermark* yang diulang. Variabel *index* digunakan untuk menghitung jumlah *biner* tiap nilai *watermark* yang disisipkan. Jika nilai *index* telah mencapai delapan maka dapat dilanjutkan pada nilai *watermark* berikutnya. Setelah seluruh nilai *watermark* disisipkan maka proses LSB selesai dan dapat dilanjutkan dengan proses *inverse transform*. Flowchart Proses LSB dapat dilihat pada Gambar 3.8.



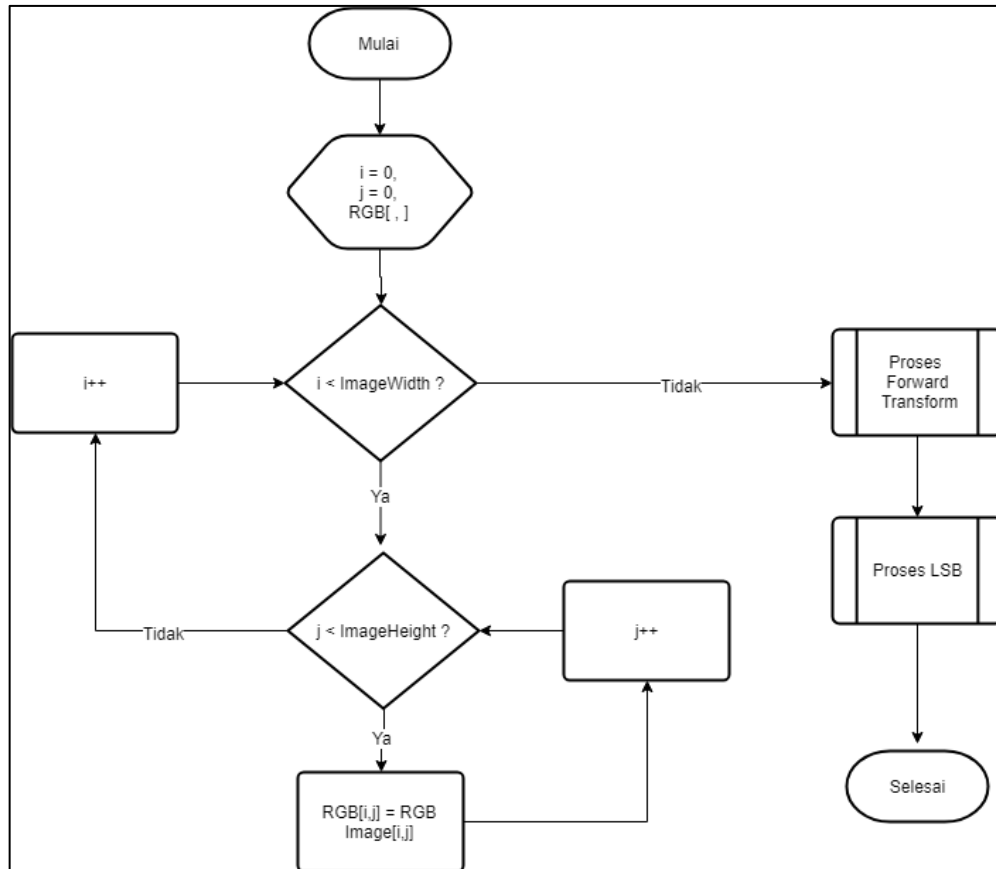
Gambar 3.8 Flowchart Proses LSB

Proses *Inverse Transform* dilakukan pada setiap baris dan kolom dari gambar dimana imageWidth merupakan nilai panjang gambar dan imageHeight merupakan nilai lebar gambar dan menggunakan Rumus 2.3 dan 2.4 dan dilakukan berulang sampai seluruh baris dan kolom telah selesai. Pada kasus spesifik dapat terjadi nilai warna biru -1 atau 256 yang melewati nilai RGB yaitu antara 0 – 255 sehingga dilakukan optimalisasi dengan mengurangi atau menambah nilai tersebut dengan dua. Flowchart Proses *Inverse Transform* dapat dilihat pada Gambar 3.9.



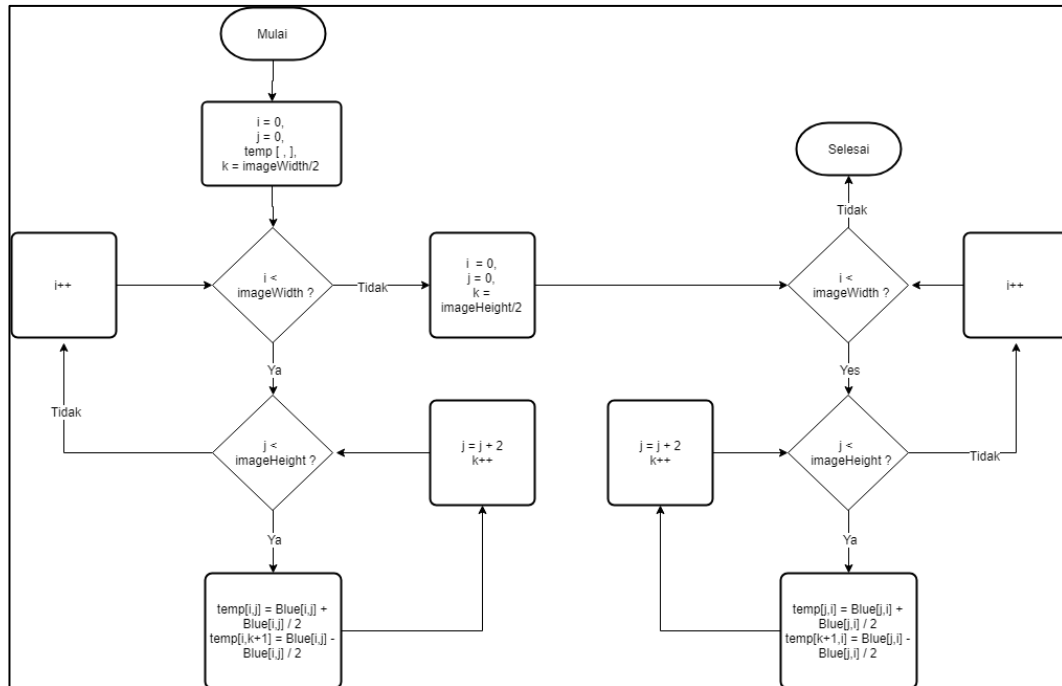
Gambar 3.9 Flowchart Proses Inverse Transform

Proses *extraction* dimulai dengan mengambil nilai RGB tiap *pixel* dimana variabel `hostImagewidth` didapatkan dari panjang *pixel* gambar *host* sedangkan variabel `hostImageHeight` didapatkan dari lebar *pixel* gambar *host*. Nilai RGB tersebut kemudian akan dilakukan proses *forward transform* untuk menjadi nilai frekuensi yang dapat diubah-ubah. Nilai frekuensi tersebut kemudian dilakukan proses LSB untuk mendapatkan *watermark* di dalamnya. Setelah *watermark* didapatkan maka nilai *watermark* dikembalikan untuk diubah menjadi gambar. Flowchart proses *extraction* dapat dilihat pada Gambar 3.10.



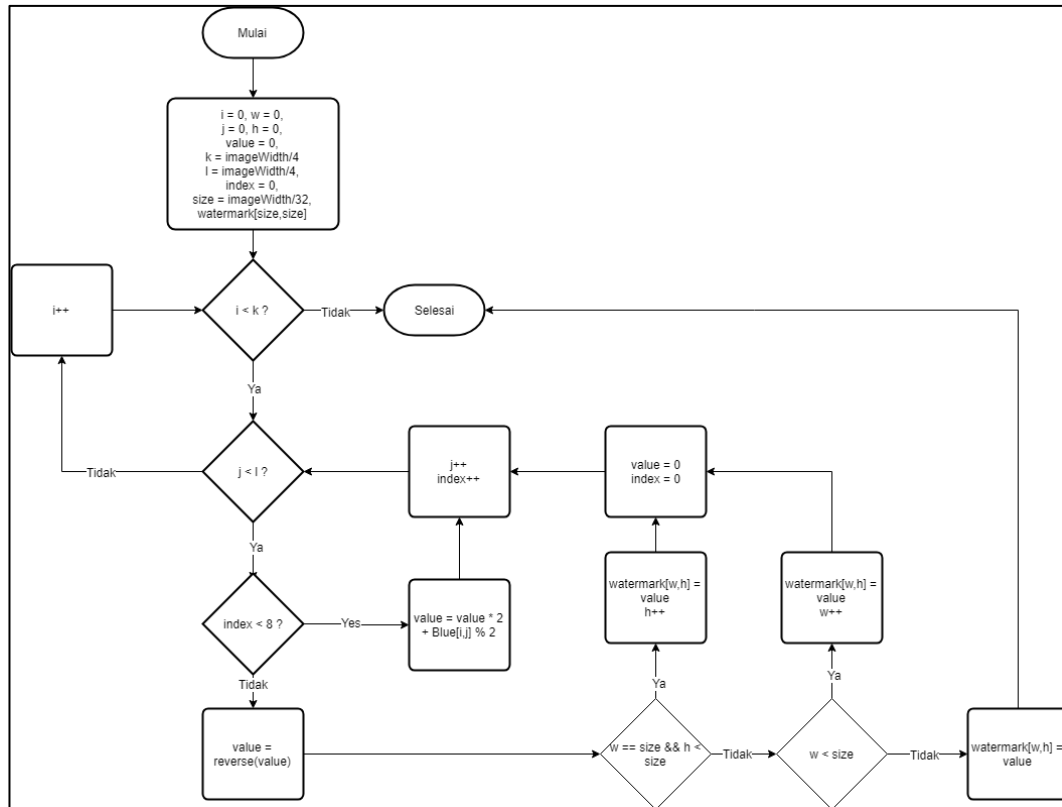
Gambar 3.10 Flowchart Proses Extraction

Proses *Forward Transform* dilakukan pada setiap baris dan kolom dari gambar dimana *imageWidth* merupakan nilai panjang gambar dan *imageHeight* merupakan nilai lebar gambar dan menggunakan Rumus 2.1 dan 2.2 dan dilakukan berulang sampai seluruh baris dan kolom telah selesai. Flowchart proses *Forward Transform* dapat dilihat pada Gambar 3.11.



Gambar 3.11 Flowchart Proses Forward Transform

Proses LSB diambil nilai biru dari hasil *forward transform* untuk mendapatkan nilai *watermark* yang disisipkan. Jumlah *pixel* didapatkan dari nilai *imageWidth* atau panjang gambar dibagi empat dan nilai *imageHeight* atau lebar gambar dibagi empat. Nilai panjang dan lebar dibagi empat karena hanya satu bagian saja yang dapat disisipkan *watermark*. *Bit* terakhir dari warna biru yang didapat kemudian disimpan ke dalam bentuk value nilai *watermark*. Nilai *index* digunakan untuk menentukan jumlah *biner* yang telah diambil. Jika nilai *index* mencapai delapan maka berarti telah didapatkan satu nilai *watermark*. Nilai *watermark* yang telah didapatkan kemudian diputarbalik karena pada proses penyisipan *watermark*, nilai yang disisipkan dimulai dari nilai *biner* besar. Flowchart proses LSB dapat dilihat pada Gambar 3.12.



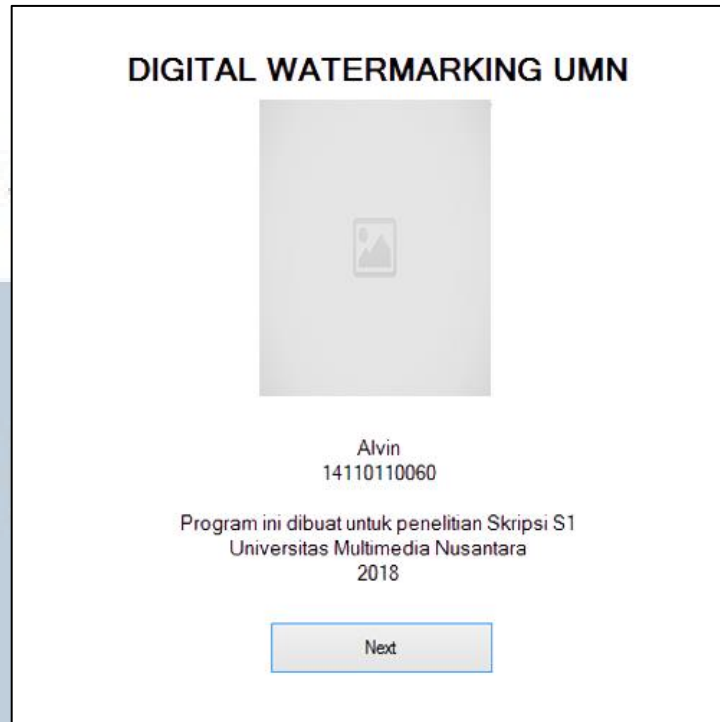
Gambar 3.12 Flowchart Proses LSB

3.2.3 Tampilan Antarmuka

Tampilan antarmuka terdiri dari tampilan awal aplikasi, tampilan halaman informasi, tampilan utama aplikasi, tampilan halaman proses *embedding*, dan tampilan halaman proses *extracting*.

Tampilan awal aplikasi berupa informasi mengenai tujuan pembuatan aplikasi. Tampilan awal ini hanya tersedia tombol *next* untuk masuk ke halaman berikutnya. Tampilan awal aplikasi dapat dilihat pada Gambar 3.13.

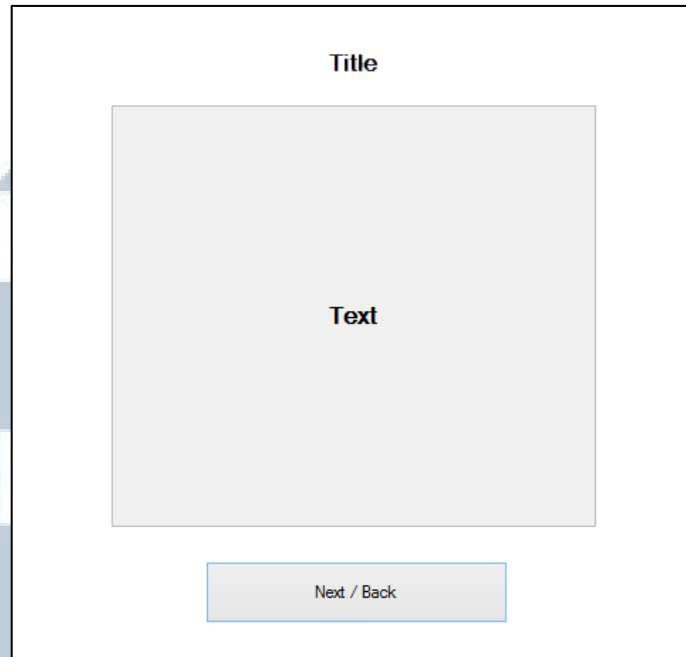
UNIVERSITAS
MULTIMEDIA
NUSANTARA



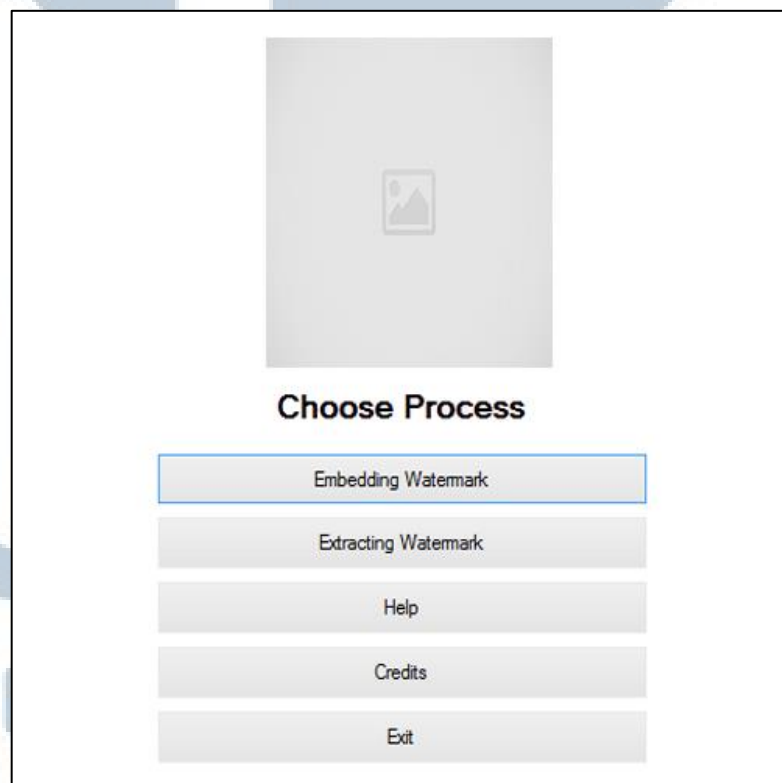
Gambar 3.13 Desain Tampilan Awal

Tampilan halaman informasi digunakan untuk halaman *help*, *about*, dan *credits* memiliki desain tampilan yang sama dengan judul dan isi *text* yang ditampilkan. Tombol *next/back* digunakan untuk kembali atau menuju halaman berikutnya. Tampilan halaman informasi dapat dilihat pada Gambar 3.14.

Tampilan utama aplikasi dimana *user* dapat memilih antara dua proses *embedding* atau *extraction*. Jika *user* menekan tombol *embedding watermark* maka akan ditampilkan halaman *embedding* sedangkan jika menekan tombol *extracting watermark* maka akan ditampilkan halaman *extraction*. Selain itu terdapat dua tombol untuk menuju halaman *help* dan *credits*. Tombol *exit* digunakan untuk menutup aplikasi. Tampilan utama aplikasi dapat dilihat pada Gambar 3.15.

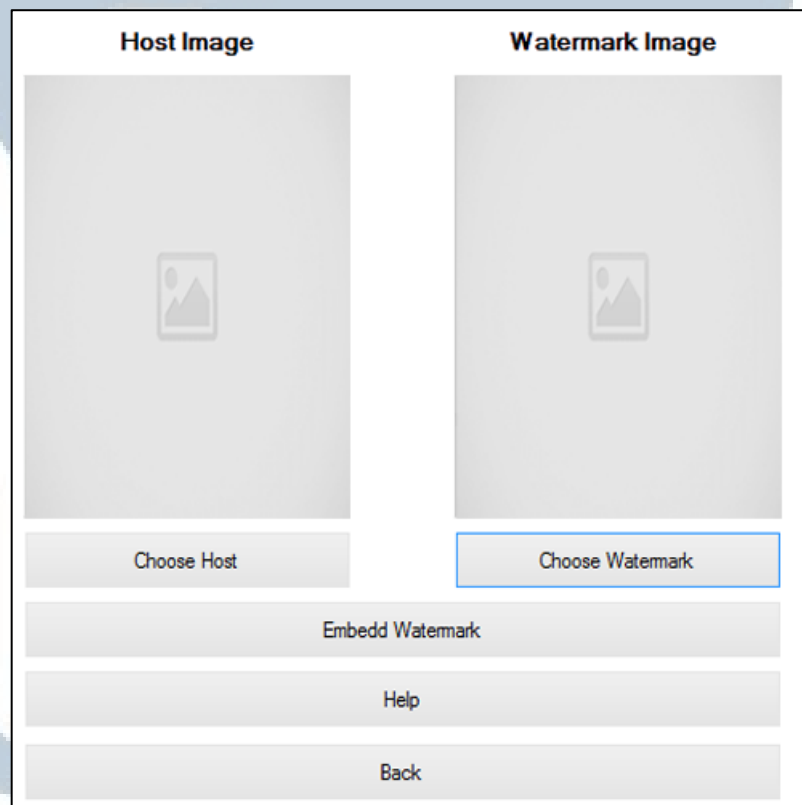


Gambar 3.14 Desain Tampilan Informasi



Gambar 3.15 Desain Tampilan Utama Aplikasi

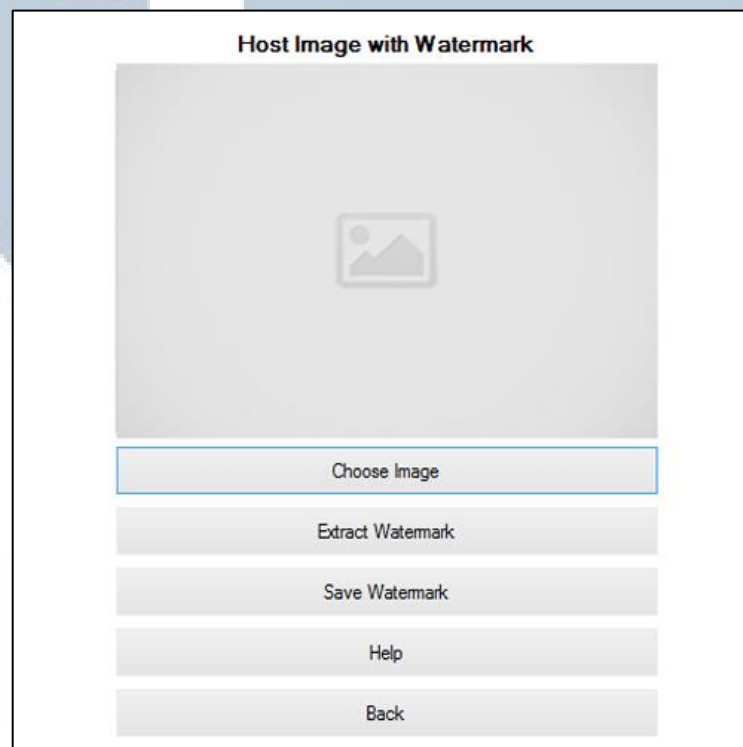
Tampilan halaman proses *embedding* untuk melakukan proses penyisipan *watermark* ke dalam gambar *host*. *User* akan diminta untuk memasukkan *input* berupa gambar *host* dan gambar *watermark* kemudian menekan tombol *embedd watermark* untuk menjalankannya. Gambar hasil proses penyisipan akan langsung disimpan ke dalam komputer *user*. Tombol *help* dapat digunakan untuk menampilkan informasi cara menggunakan aplikasi dan tombol *back* digunakan untuk kembali ke halaman utama aplikasi. Tampilan proses *embedding* dapat dilihat pada Gambar 3.16.

The image shows a web application interface for the embedding process. It features two side-by-side image selection areas. The left area is titled 'Host Image' and contains a large grey rectangle with a small image icon in the center, and a 'Choose Host' button below it. The right area is titled 'Watermark Image' and contains a similar large grey rectangle with an image icon and a 'Choose Watermark' button below it. Below these two areas are four horizontal buttons: 'Embedd Watermark', 'Help', and 'Back'. The 'Choose Watermark' button is highlighted with a blue border.

Gambar 3.16 Desain Tampilan Proses Embedding

Tampilan halaman proses *extraction* untuk melakukan proses ekstraksi *watermark* dari dalam gambar *host* yang telah disisipkan *watermark* pada proses *embedding*. *User* akan diminta untuk memasukkan *input* berupa gambar yang

memiliki *watermark* kemudian menekan tombol *extract watermark* untuk menjalankannya. Gambar hasil proses berupa gambar *watermark* ditampilkan pada halaman *extraction*. *User* dapat menyimpan *watermark* yang diekstraksi dengan menekan tombol *save watermark*. Tombol *help* dapat digunakan untuk menampilkan informasi cara menggunakan aplikasi dan tombol *back* digunakan untuk kembali ke halaman utama aplikasi. Tampilan proses *extraction* dapat dilihat pada Gambar 3.17.



Gambar 3.17 Desain Tampilan Proses Extracting

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A