

## BAB IV

### PERANCANGAN DAN IMPLEMENTASI

Pada bab ini akan menjelaskan perancangan dan implementasi aplikasi segmentasi citra digital dan terdapat beberapa tahap. Tahap pertama merupakan tahap pre proses dimana proses ini untuk menyiapkan citra yang akan diolah dan tahap kedua merupakan proses segmentasi citra digital menggunakan algoritma *watershed*.

#### 4.1 Perancangan Secara Umum

Tahap awal yang dibutuhkan adalah membuat perancangan aplikasi secara global dimana tahap awal ini berfungsi sebagai acuan dalam proses pembuatan aplikasi yang akan dibuat. Perancangan ini didahului dengan pendefinisian kegiatan pelaku atau *user* dalam menggunakan program segmentasi citra digital dengan pengolahan citra serta perangkat yang digunakan meliputi blok sistem diagram dan cara kerja aplikasi.

##### 4.1.1 Blok Sistem Diagram

Pada segmentasi citra digital terdiri dari beberapa langkah yang dapat digambarkan menjadi blok diagram dengan model seperti pada gambar 4.1



**Gambar 4.1** Diagram Blok Sistem Secara Keseluruhan

**Sumber :** Perancangan

Fungsi masing-masing bagian dalam diagram blok ini adalah sebagai berikut:

1. Citra fotografi digunakan sebagai input sistem.
2. Melakukan *image processing* pada citra fotografi, yaitu proses awal dan proses segmentasi.
3. Citra fotografi yang telah melewati proses pengolahan citra pada sistem akan menghasilkan suatu citra fotografi yang telah tersegmentasi untuk

selanjutnya dapat dilakukan proses yang lain sesuai dengan kebutuhan.

#### 4.1.2 Cara Kerja Aplikasi

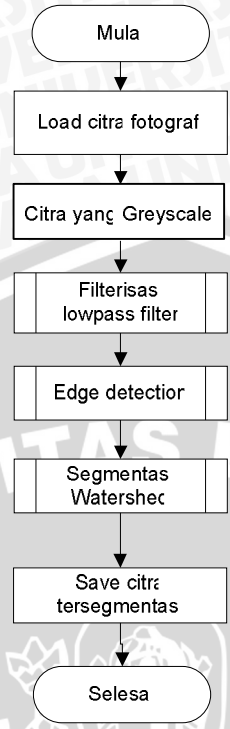
Aplikasi segmentasi citra digital menggunakan algoritma *watershed* memiliki cara kerja yang dimulai dengan pengambilan atau pemilihan citra digital yang telah dalam skala keabu-abuan atau *grayscale*. Citra *digital* yang telah dipilih tersebut, kemudian dilakukan proses awal atau *pre-processing* dengan menggunakan *lowpass filter*. Selanjutnya citra digital yang telah melalui tahap *pre-processing* kemudian dilakukan tahap segmentasi menggunakan algoritma *watershed*.

#### 4.2 Perancangan Perangkat Lunak

Pada bagian perancangan ini perangkat lunak yang akan dibuat menggunakan bahasa pemrograman MATLAB *software* R2012b dan sistem yang digunakan untuk membangun perangkat lunak ini dirancang dengan spesifikasi mampu melakukan hal-hal sebagai berikut:

1. Mengakses data citra yang tersimpan didalam *harddisk* komputer.
2. Melakukan proses filtering dengan metode *lowpass filter* pada citra *digital* sebagai proses awal atau *pre-processing*
3. Melakukan proses *edge detection* pada citra secara otomatis.
4. Melakukan proses segmentasi pada citra *digital*.

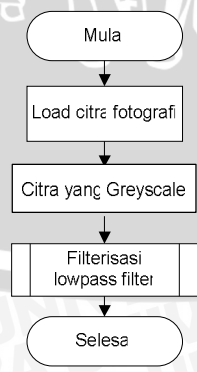
Sedangkan untuk detail desain aplikasi secara umum akan ditunjukkan pada gambar 4.2



**Gambar 4.2** Detail Desain Aplikasi  
**Sumber :** Perancangan

**4.2.1 Perancangan *Preprocess***

Citra fotografi yang terdapat di dalam harddisk komputer di *load*, kemudian citra fotografi yang di pilih adalah citra yang sudah dalam bentuk grayscale. Setelah itu dilakukan proses filterisasi dengan menggunakan *lowpass filter*. Flowchart perancangan *preprocess* seperti pada gambar 4.3.



**Gambar 4.3** Flowchart *Preprocess*  
**Sumber :** Perancangan

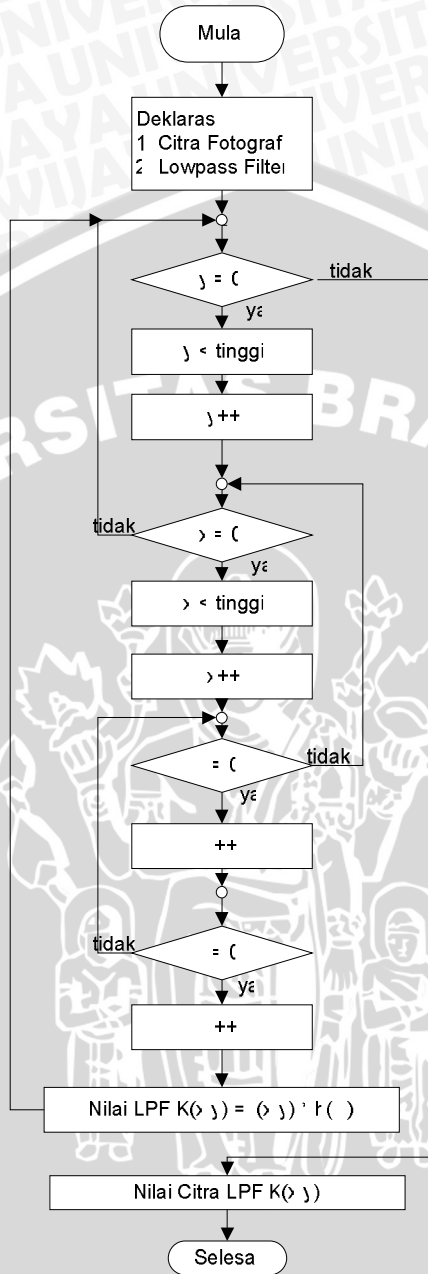
Berikut penjelasan flowchart preprocess :

1. Citra yang akan diproses merupakan citra fotografi yang sudah dalam bentuk *grayscale* untuk mempermudah proses selanjutnya.
2. Setelah citra fotografi dipilih, proses selanjutnya adalah filterisasi menggunakan lowpass filter dimana proses ini dilakukan untuk mengurangi noise atau gangguan pada citra fotografi agar saat proses proses segmentasi dapat mengurangi over - segmentation atau segmentasi yang berlebihan.

UNIVERSITAS BRAWIJAYA



4.2.1.1 Perancangan Sub Proses Filterisasi Low Pass Filter



Gambar 4.4 Flowchart proses bagian Low Pass Filter

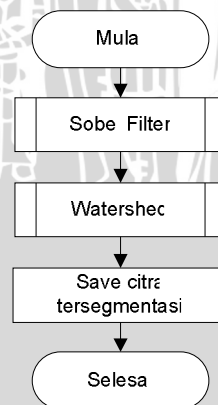
Sumber : Perancangan

Penjelasan Flowchart sebagai berikut:

1. Melakukan inialisasi pada citra fotografi dan lowpass filter menjadi bentuk matrik atau array bitmap pixel untuk memanipulasi secara langsung pada memori.
2. Deklarasi (  $I$  ) dimana nilai yang ditunjuk oleh (  $I$  ) yaitu alamat dari elemen pertama yaitu nilai *pixel* citra.
3. Deklarai (  $h$  ) dimana nilai yang ditunjuk oleh (  $h$  ) yaitu alamat dari elemen pertama yaitu nilai *pixel lowpass filter*.
4. Setelah mendapat nilai *pixel* pada kanal citra dan *lowpass filter*, kemudian dilakukan proses konvolusi. Jumlah nilai matrik konvolusi antara citra dan filter diperiksa apakah jumlahnya sama dengan citra.
5. Mendapat nilai *lowpass filter* dari hasil konvolusi.

#### 4.2.2 Perancangan Proses

Setelah citra melewati preprocess maka dilanjutkan pemrosesan citra untuk menghasilkan citra yang tersegmentasi. Proses yang dilakukan pada tahap ini yaitu citra yang telah melalui *lowpass filter* diteruskan untuk di deteksi tepi atau *edge detection* menggunakan *filter sobel*. Setelah melalui proses deteksi tepi dilanjutkan dengan proses segmentasi menggunakan *watershed* agar mendapatkan satu citra *image processing* yang tersegmentasi.



Gambar 4.5 Flowchart process

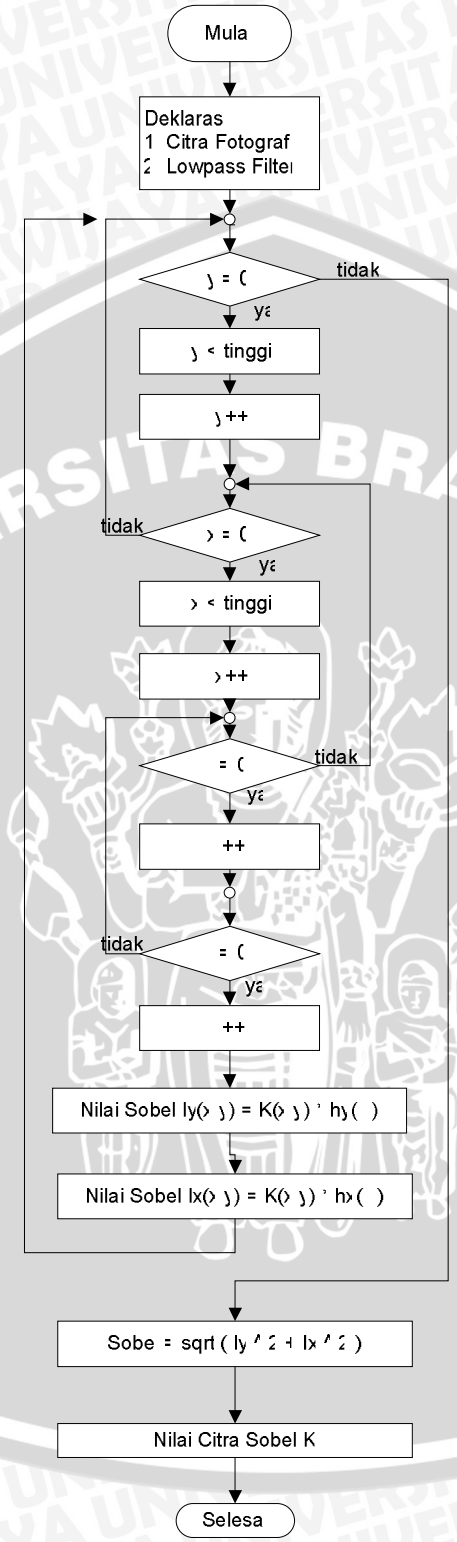
Sumber : Perancangan

Penjelasan flowchart sebagai berikut :

1. Citra hasil *lowpass filter* digunakan sebagai citra *input*. Selanjutnya dilakukan proses pendeteksi tepi menggunakan *filter sobel*.
2. Hasil yang diperoleh dari proses *filter sobel* ini selanjutnya akan dilakukan proses segmentasi menggunakan *watershed* untuk mendapatkan satu citra hasil segmentasi.
3. Citra hasil proses segmentasi menggunakan *watershed* merupakan hasil akhir yang kemudian disimpan sebagai data.

#### 4.2.2.1 Perancangan Sub Proses Bagian Sobel Filter





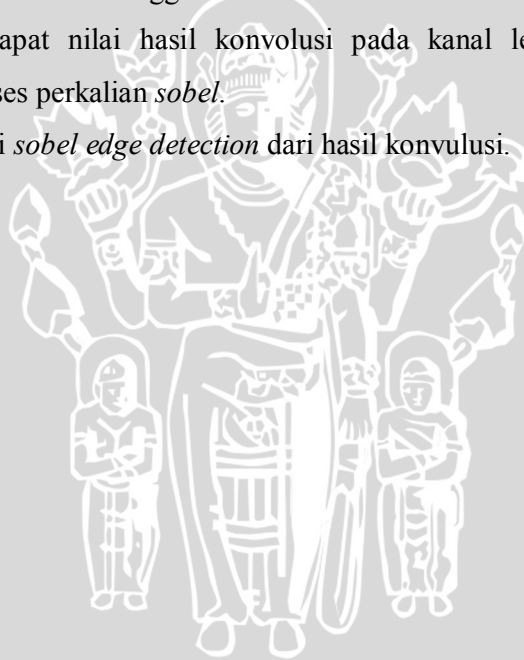
Gambar 4.6 Flowchart proses Sobel edge detection

Sumber : Perancangan

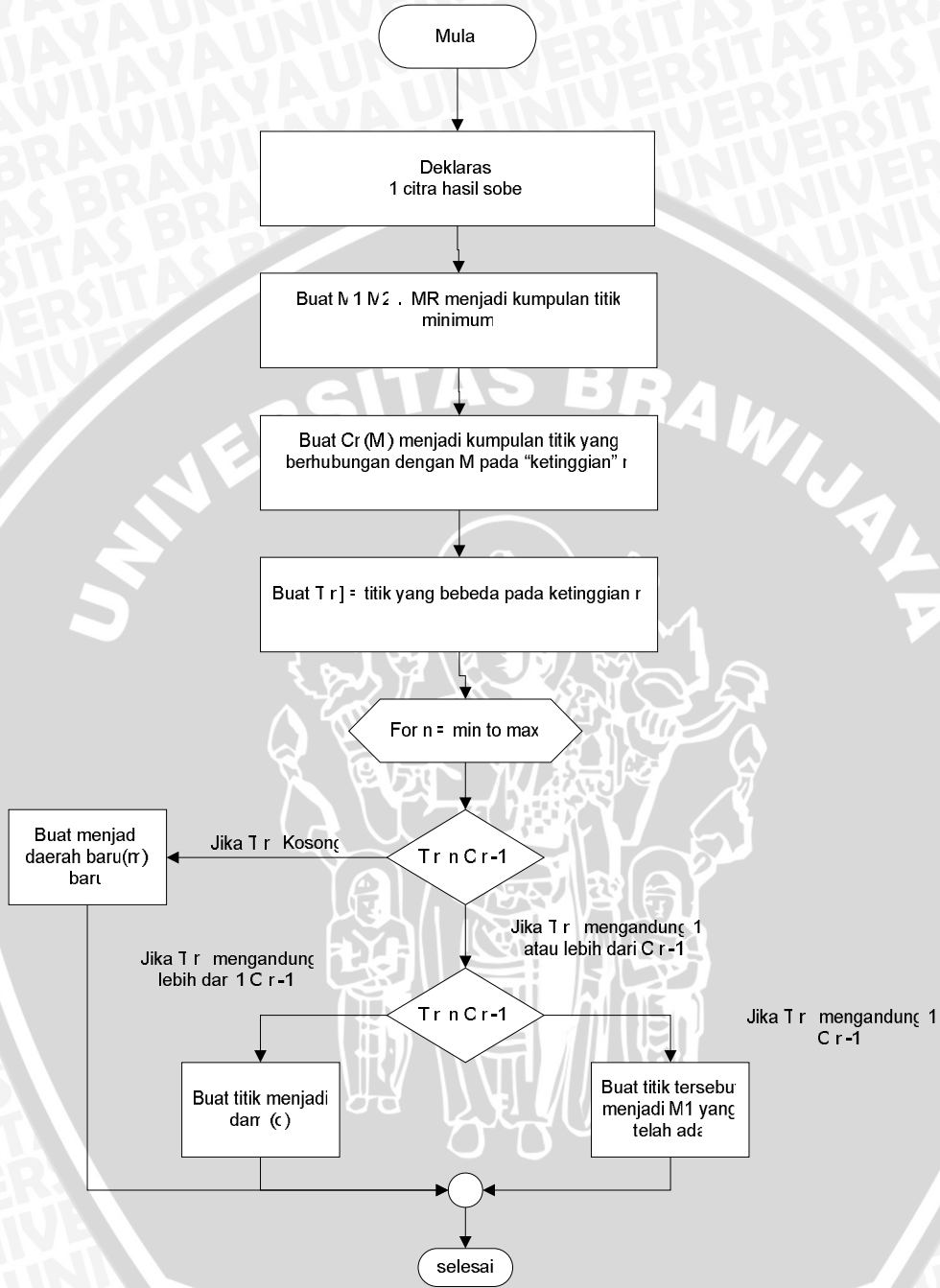


Penjelasan Flowchart :

1. Melakukan inialisasi pada citra *lowpass filter* dan *sobel edge detection* menjadi bentuk matrik atau *array bitmap pixel* untuk memanipulasi secara langsung pada memori.
2. Deklarasi ( K ) dimana nilai yang ditunjuk oleh ( K ) yaitu alamat dari elemen pertama yaitu nilai *pixel citra lowpass filter*.
3. Deklarai ( hx , hy ) dimana nilai yang ditunjuk oleh ( hx , hy ) yaitu alamat dari elemen pertama yaitu nilai *pixel sobel edge detection*.
4. Setelah mendapat nilai *pixel* pada kanal citra *lowpass* dan *sobel edge detection*, kemudian dilakukan proses konvolusi.
5. Proses konvolusi dibagi menjadi dua yaitu pada kanal lebar matrik citra *lowpass* dan pada kanal tinggi matrik citra.
6. Setelah mendapat nilai hasil konvolusi pada kanal lebar dan tinggi, dilakukan proses perkalian *sobel*.
7. Mendapat nilai *sobel edge detection* dari hasil konvolusi.



4.2.2.2 Perancangan Sub Proses Bagian Watershed



Gambar 4.7 Flowchart proses watershed

Sumber : Prancangan

Penjelasan flowchart:

1. Melakukan inialisai terhadap citra hasil deteksi *sobel* untuk mekakukan proses segmentasi dengan watershed.
2. Membuat Kumpulan Koordinat titik (  $M$  ) menjadi regional minima dalam sebuah citra.
3. mengumpulkan koordinat titik ( regional minima) menjadi *catchment basins* atau lembah ( $C_n ( M_i )$ ).
4. Membuat nilai *threshold* (  $T[n]$  ) dengan ketinggian  $n$ , dimana  $n$  adalah kedalaman dari *catchment basin*
5. Setelah *catchment basin* dan nilai *threshold* terbentuk maka koordinat yang terdapat didalam citra mulai dialiri dimulai dari ketinggian minimal  $n = \min + 1$  hingga ketinggian maksimal  $n = \max + 1$ .
6. Koordinat didalam citra diperiksa untuk membentuk dam atau pemisah *catchment basin*, jika didalam koordinat tidak terdapat *catchment basin* maka koordinat tersebut dibiarkan terbenam.
7. Jika didalam koordinat tersebut terdapat lebih dari satu buah *catchment basin* maka dibuat dam (  $q$  ) sebagai pemisah, tetatapi jika hanya terdapat sebuah *cactment basin* maka dibuat menjadi daerah regional minima.

### 4.3 Implementasi Sistem

Setelah tahap perancangan, selanjutnya adalah tahap implementasi. Tahap ini merupakan proses transformasi hasil perancangan perangkat lunak yang telah dibuat ke dalam kode (*coding*) sesuai dengan sintaks dari bahasa pemrograman yang digunakan.

#### 4.3.1 Lingkungan Implementasi

Aplikasi dibuat dengan menggunakan MATLAB software R2012b. Sistem diimplementasikan dengan menggunakan spesifikasi sebagai berikut:

1. Perangkat keras (Komputer)

Spesifikasi :

Processor : Intel Core i3-2330M CPU @ 2,20GHz

Memory : 8192 MB DDR3

OS : Windows 7

VGA : NVIDIA GeForce GT520M

## 2. Perangkat Lunak :

- Sistem operasi : Microsoft Windows 7
- Bahasa pemrograman : MATLAB software R2012b

### 4.3.2 Implementasi Algoritma

#### 4.3.2.1 Proses *Lowpass Filter*

Citra yang akan diproses pada aplikasi ini menggunakan citra *grayscale* dan untuk mengurangi gangguan atau *noise* maka citra melalui proses *lowpass filter* untuk mengurangi *noise*. Proses *lowpass filter* ini dilakukan dengan iterasi dimana syarat iterasi menggunakan lebar dan panjang citra berikut listing program dalam syntac MATLAB.

##### 1. Membaca file citra

```
gbr = imread ([path , gambar]);
```

##### 2. Membuat lowpass filter

```
h=[ 1 1 1;  
   1 1 1;  
   1 1 1]/9;
```

##### 3. Melakukan proses konvolusi citra dengan *lowpass filter*

```
K = imfilter (gbr,h);
```

#### 4.3.2.2 Proses *sobel edge detection*

*Sobel edge detection* digunakan untuk mendeteksi tepi yang membatasi dua wilayah citra homogen yang memiliki tingkat kecerahan yang berbeda. Proses konvolusi sobel dengan citra dilakukan secara otomatis dengan bantuan dari aplikasi MATLAB.

##### 1. Melakukan inisialisasi Sobel edge detection

```
hy = fspecial('sobel');  
hx = hy;
```

2. Melakukan proses konvolusi matrik citra pada komponen panjang dan lebar dengan matrik *sobel*

```
Iy = imfilter(double(K), hy);
```

```
Ix = imfilter(double(K), hx);
```

3. Mencari nilai mutlak dari *sobel edge detection*

```
gradmag = sqrt(Ix.^2 + Iy.^2);
```

```
figure, imshow(gradmag), title('3. Gradient magnitude  
(gradmag)')
```

#### 4.3.2.3 Proses Segmentasi *Watershed*

Proses segmentasi menggunakan *watershed* dilakukan untuk mendapatkan satu citra hasil image processing, berikut listing program dalam syntax MATLAB

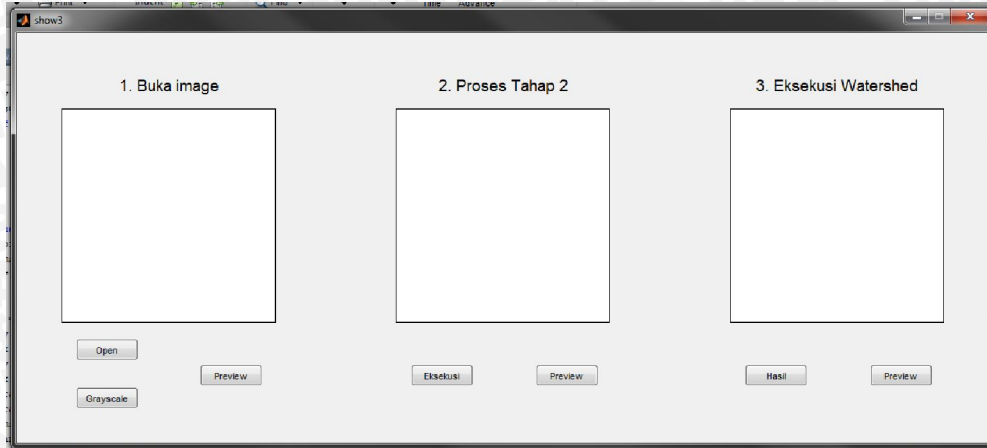
```
L = watershed(gradmag);
```

```
Lbw = bwlabel(L);
```

```
figure, imshow(Lbw), title('5. Watershed transform of  
gradient magnitude (Lrgb)')
```

#### 4.4 Implementasi Interface ( Antarmuka )

Program segmentasi citra menggunakan *watershed* ini dibuat dengan desain sederhana mungkin dan sesuai dengan langkah-langkah yang berurutan hal ini bertujuan agar pemakai tidak bingung dalam menggunakan program ini. Pada tampilan utama ini terdapat tujuh button yang mempunyai fungsi berbeda – beda. Selain itu pada tampilan utama terdapat tiga *picture box* yang berguna untuk menampilkan citra *input*, pre proses dan citra tersegmentasi.

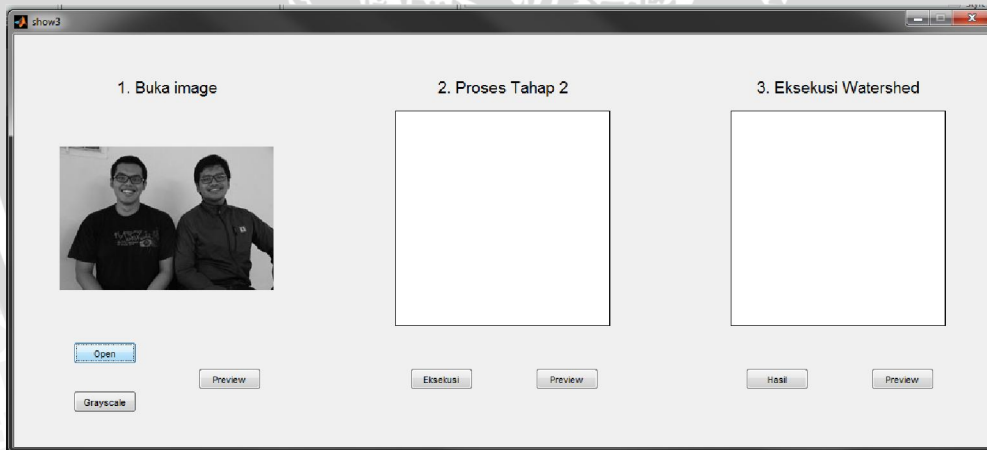


**Gambar 4.8** Tampilan utama program

Sumber : Perancangan

#### 4.4.1 Implementasi Antarmuka Untuk Load data

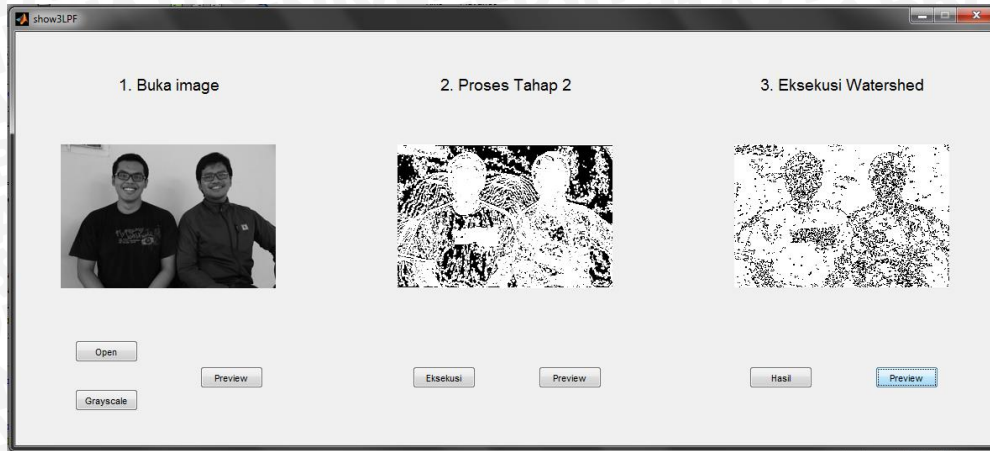
Pada tampilan antarmuka dari *load* data terlihat pada gambar dibawah ini dimana terdapat *picture box* untuk menampilkan data citra yang akan diproses.



**Gambar 4.9** Tampilan Load Citra

Sumber : Perancangan

### 4.4.2 Implementasi Antarmuka Proses Pengolahan Citra



**Gambar 4.10** Tampilan Proses citra hasil Watershed

Sumber : Perancangan

