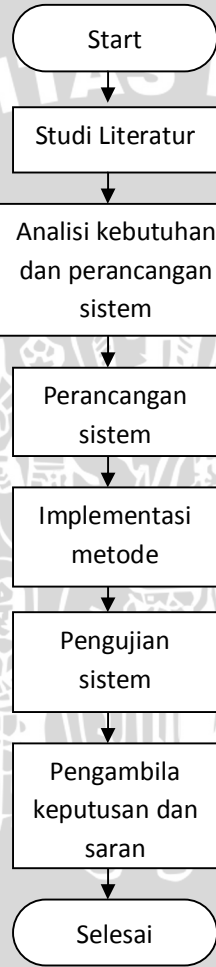


### BAB III

#### METODE PENELITIAN

Dalam penyusunan skripsi ini, dirancang suatu perangkat aplikasi segmentasi citra digital dengan menggunakan metode *watershed* dan *low pass filter* sebagai proses awal. Metode penelitian yang digunakan pada penyusunan skripsi ini dapat digambarkan dalam bentuk diagram seperti Gambar 3.1



**Gambar 3.1** Diagram Alir Metodologi Penelitian

### 3.1 Studi Literatur

Studi literatur menjelaskan dasar teori yang digunakan untuk menunjang penulisan skripsi. Teori-teori pendukung tersebut meliputi:

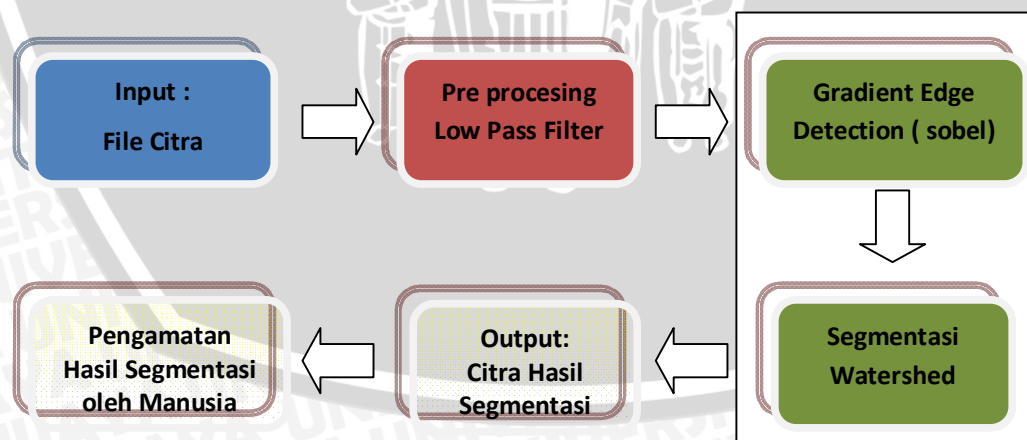
1. Membaca dan mempelajari buku – buku yang berhubungan dengan *image processing* .
2. Mempelajari algoritma *watershed*.
3. Mempelajari metode - metode yang mendukung segmentasi citra digital.
4. Mempelajari teknik – teknik dasar pemrograman dengan menggunakan MATLAB.

### 3.2 Analisa Kebutuhan

Untuk menentukan sistem yang dapat digunakan untuk menjalankan aplikasi segmentasi citra digital menggunakan algoritma watershed, antara lain:

- a. Perangkat Keras :
  1. PC atau Laptop
- b. Perangkat Lunak :
  1. *Operating system* Windows 7
  2. Software MATLAB (matrix laboratory)

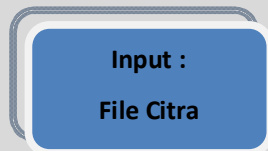
### 3.3 Blok Diagram Sistem



Gambar 3.2. Blok diagram sistem

## Keterangan :

1. Input: Citra *Gray Scale* : Citra yang akan diproses dan berada pada *gray level*
2. Pre Processing *Low Pass Filter* : Proses yang menyaring noise yang ada di citra (Lihat *Pseudocode*)
3. Gradient *Edge Detection (sobel)* : Proses Deteksi tepi pada citra sebagai bentuk proses pembuatan Dam atau batas.
4. Segmentasi *Watershed* : Proses pemisahan bagian objek dalam citra yang mengandung informasi dari latar atau *background* ( lihat Gambar Diagram Alir proses segmentasi *watershed*)
5. *Output* : Citra yang sudah ter segmentasi.
6. Pengamatan Manusia : Pengamatan dengan Mata manusia untuk menentukan hasil segmentasi yang terbaik.



Input file citra adalah citra yang nantinya akan disegmentasi. Setelah file citra di masukan, gambar di persiapkan untuk memasuki tahap pre prosesi. File citra yang akan di proses adalah citra *grayscale* atau abu- abu, karena dalam pengolahan citra digital jika tidak dalam bentuk *grayscale* maka proses mengolah akan menjadi lama dan memakan resouse yang banyak. File citra yang akan di proses adalah citra yang mempunyai ukuran citra (256 x 256 pixel), (512 x 512 pixel), dan (680 x 480 pixel). Ukuran citra tersebut merupakan ukuran citra uji yang biasa digunakan dalam pengolahan citra digital.



**Pre processing  
Low Pass Filter**

Pre processing merupakan proses awal untuk menyaring file citra sebelum dilakukan proses utama, dalam hal ini adalah proses segmentasi. Dalam proses pre processing, citra akan disaring menggunakan metode low pass filter. Metode ini dimaksudkan untuk menyaring atau menghilangkan noise dalam citra. Konsep metode ini adalah dengan menkonvolusikan matrik citra dengan matrik lowpass filter. Ini adalah *pseudocode* untuk metode *low pass filter*:

*Awal Modul*

*dari nilai (y) yang ke-0 sampai (tinggi dari gambar -1)*

*mulai*

*dari nilai (x) yang ke-0 sampai (panjang dari gambar -1)*

*mulai*

*nilai pixel saat ini((x), (y)) =*

*nilai pixel ((x)-1, (y)+1)\*1/9 + nilai pixel ((x), (y)+1)\*1/9 + nilai pixel*

*((x)+1, (y)+1)\*1/9 +*

*nilai pixel((x)-1, (y))\*1/9 + nilai pixel((x), (y))\*1/9 + nilai pixel*

*((x)+1, (y))\*1/9 +*

*nilai pixel((x)-1, (y)-1)\*1/9 + nilai pixel ((x), (y)-1)\*1/9 + nilai pixel*

*((x)+1, (y)-1)1/9*

*selesai*

*selesai*

*akhir modul*

matrik low pass filter yang akan di gunakan adalah, matrik yang berukuran 3x3, karena matrik citra yang akan dikonvolusi menggunakan potongan matrik 3x3. Penggunaan matrik 3x3 diharapkan mampu mengurangi noise karena tidak membuat informasi yang ada di citra tidak tergerus atau terbuang karena proses pre processing.

### Gradient Edge Detection ( sobel)

Sobel *edge detection* merupakan metode *edge detection* yang termasuk dalam Gradient edge detection. Pixel dari sebuah citra yang akan dilakukan pendeteksian batas (*edge*) akan menjadi sebuah *edge* jika pixel tersebut melewati batas (*threshold*) tertentu. Sehingga apabila *threshold* telah ditetapkan maka nilai dari setiap pixel dapat ditentukan tersebut merupakan batas (*edge*) atau bukan, dengan cara mengalikan matrik sebuah citra dengan matrik sobel yang telah ditentukan.

$$O[i,j] = I[i-1, j-1] * G_x[i-1, j-1] + I[i, j-1] * G_x[i, j-1] + I[i+1, j-1] * G_x[i+1, j-1] \\ + I[i-1, j] * G_x[i-1, j] + I[i, j] * G_x[i, j] + I[i+1, j] * G_x[i+1, j] \\ + I[i-1, j+1] * G_x[i-1, j+1] + I[i, j+1] * G_x[i, j+1] + I[i+1, j+1] * G_x[i+1, j+1]$$

Proses perkalian matrik citra dengan sobel

### Segmentasi Watershed

Konsep transformasi *Watershed* adalah dengan menganggap sebuah citra merupakan bentuk tiga dimensi yaitu posisi x dan y dengan masing-masing tingkatan warna yang dimilikinya. Posisi x dan y merupakan bidang dasar dan tingkat warna pixel, yang dalam hal ini adalah citra abu-abu (gray level) merupakan ketinggian. Berikut adalah diagram proses segmentasi dengan menggunakan Algoritma *Watershed* sebagai berikut

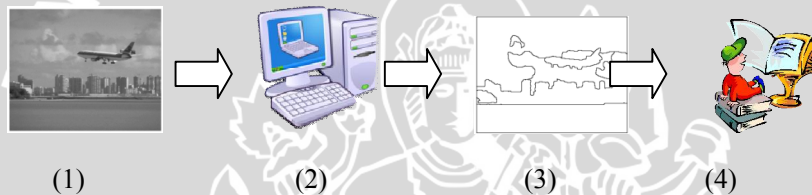
### Output: Citra Hasil Segmentasi

Output merupakan akhir dari proses segmentasi. Pada bagian akhir ini, hasil dari segmentasi citra akan di kumpulkan dan disimpan, kemudian dibandingkan dengan hasil dari segmentasi citra uji lainnya. Semua hasil segmentasi citra akan diperlihatkan sebagai perbandingan.

**Pengamatan Hasil Segmentasi oleh Manusia**

Setelah melakukan proses pre-procecing dan segmentasi citra menggunakan algoritma *watershed*, proses masih di lanjutkan dengan menilai hasil secara subjektif atau dengan pengamatan manusia. Hasil segmentasi sangat memerlukan campur tangan dari manusia, karena citra sangat bersifat relatif dan masing-masing pandangan manusia juga berbeda.

### 3.4 Perancangan dan Implementasi Sistem



**Gambar 3.3.** Rancangan sistem segmentasi secara keseluruhan

Secara garis besar desain yang akan dibuat untuk aplikasi segmentasi menggunakan algoritma *watershed* terdiri dari beberapa langkah. Berikut ini desain rancangan segmentasi citra menggunakan algoritma *watershed*:

1. Citra uji *grayscale* dipilih untuk di proses di aplikasi segmentasi *watershed*
2. Citra uji di proses ke dalam perangkat komputer yang berisi aplikasi segmentasi *watershed*
3. Hasil segmentasi keluar dan disimpan
4. Hasil citra uji aplikasi segmentasi *watershed* di persiksa manusia untuk menentukan hasil segmentasi yang terbaik.

### 3.5 Pengujian Sistem

Pengujian dilakukan untuk menjamin dan memastikan bahwa aplikasi yang telah dirancang memiliki tingkat kesalahan yang kecil. Untuk mengetahui apakah aplikasi bekerja dengan baik dan sesuai dengan perancangan, maka



diperlukan serangkaian pengujian. Pengujian aplikasi segmentasi dilakukan terhadap beberapa gambar yang berasal dari citra uji.

Hasil pengujian termasuk jumlah segmentasi dan waktu yang dibutuhkan mulai dari pre-processing hingga selesainya segmentasi. Perbandingan juga dilakukan terhadap besarnya ukuran gambar yang digunakan dan apakah objek yang dihasilkan sama baik jumlah maupun bentuknya.

Selain citra yang berbeda ukuran, pengujian juga dilakukan untuk segmentasi tanpa melalui pre-processing. Sehingga dapat membandingkan hasil citra yang melalui pre-processing dan tanpa melalui pre-processing.

Pengujian juga dilakukan dengan bantuan mata telanjang manusia untuk menentukan hasil segmentasi citra mana yang terbaik. Karena citra merupakan objek yang relatif dan dalam skripsi ini hanya bertujuan untuk menciptakan aplikasi segmentasi citra dengan menggunakan metode *watershed*, oleh karena itu penilaian hasil segmentasi citra sangat membutuhkan bantuan manusia untuk menentukannya.

### 3.6 Kesimpulan dan Saran

Pada tahap ini, diambil dari hasil pengujian dan analisa terhadap Aplikasi Segmentasi citra *digital*. Tahap Selanjutnya adalah membuat saran untuk perbaikan terhadap penelitian selanjutnya sehingga dapat menyempurnakan kekurangan-kekurangan yang ada.