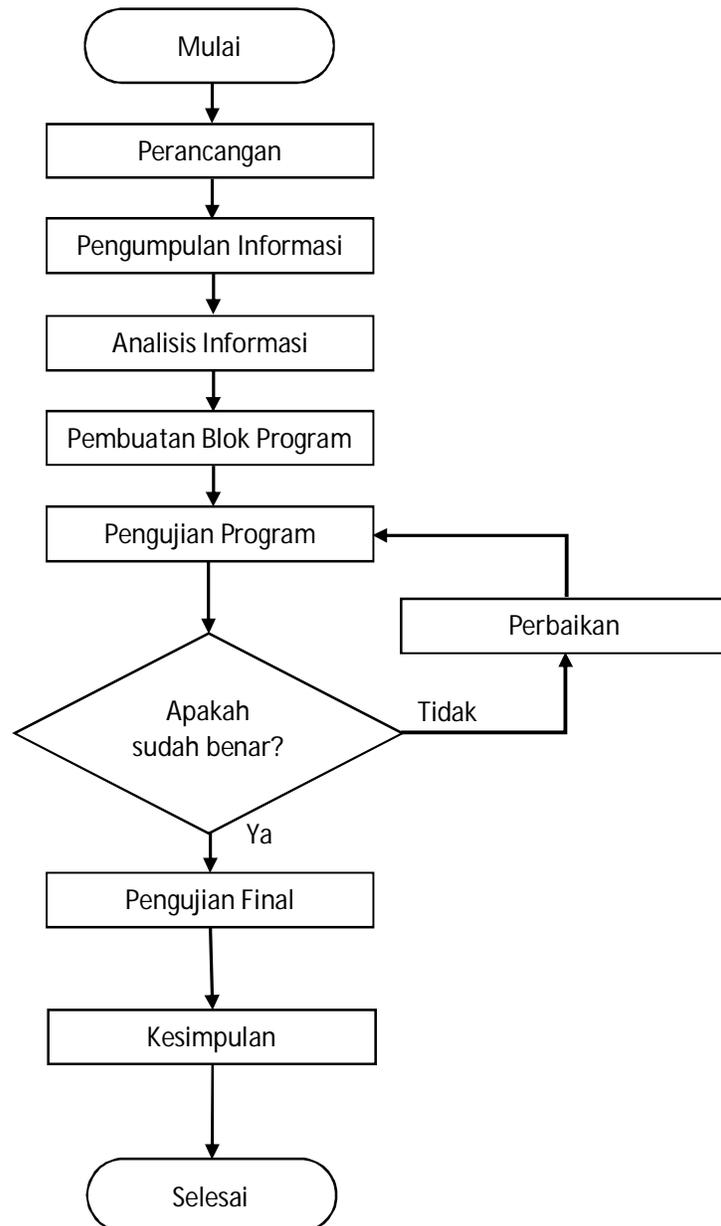


BAB III

Metode Penelitian

3.1 Diagram Alir Penelitian

Diagram blok penelitian yang akan dilakukan dapat digambarkan sebagai berikut:



Gambar 3.1 Flowchart Penelitian

3.1.1 Penjelasan Blok Diagram

a) Perancangan

Tahap perancangan yaitu merancang model program sesuai dengan permasalahan yang dihadapi. Dalam hal ini yaitu bagaimana cara mengukur luas citra foto satelit dari danau Aral menggunakan pengukuran tidak langsung (perangkat lunak)

b) Pengumpulan Informasi

Tahap pengumpulan informasi yaitu mengumpulkan segala macam informasi mengenai permasalahan yang dihadapi. Informasi yang dicari mulai dari ketersediaan foto/citra, cara penggunaan matlab, cara melakukan pemrograman, sampai informasi mengenai pengukuran. Informasi-informasi tersebut diperoleh melalui buku-buku, *ebook* dan artikel dari internet.

c) Analisis Informasi

Tahap analisis informasi yaitu menganalisis semua informasi yang sudah didapat. Tujuan dari tahap ini adalah memisahkan informasi-informasi yang diperlukan dalam penelitian dari semua informasi yang sudah didapat.

d) Pembuatan blok program

Tahap pembuatan blok program yaitu membuat dan merencanakan program secara modular/per bagian. Dengan

membuat program secara modular maka proses pengujian dapat dilakukan dengan mudah tanpa harus merubah struktur program secara keseluruhan. Pembuatan blok-blok program juga dapat membuat lebih mudah dalam mengidentifikasi *error* yang mungkin terjadi.

e) Pengujian program

Tahap pengujian program yaitu menguji blok-blok program yang telah dibuat pada tahap sebelumnya. Blok program diuji satu per satu untuk kemudian digabungkan menjadi satu program utuh.

f) Perbaikan

Tahap perbaikan yaitu memperbaiki blok-blok program yang telah diuji pada tahap sebelumnya. Blok program diperbaiki dengan menggabungkan berbagai macam informasi yang telah didapat berdasarkan kebutuhan.

g) Pengujian Final

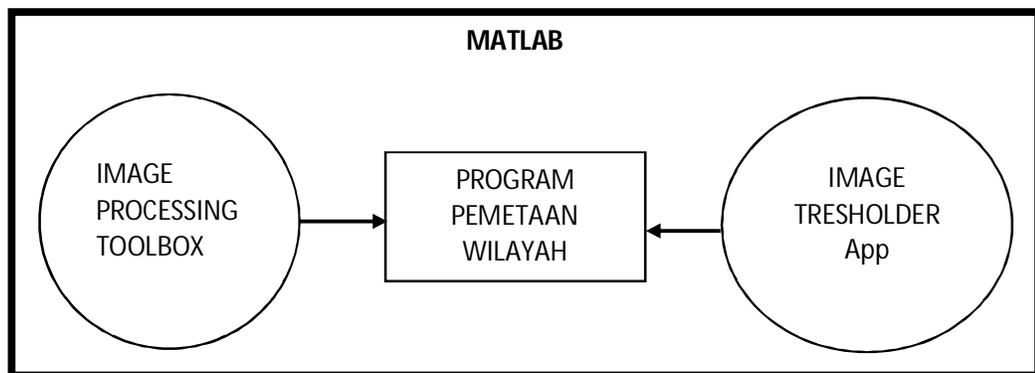
Tahap pengujian final yaitu pengujian akhir dari blok-blok program yang telah diperbaiki. Pengujian kedua ini sekaligus menggabungkan blok program ke dalam suatu program yang utuh. Pada pengujian, jika tidak ditemukan *error* pada program maka program akan diujikan pada berkas gambar/citra yang diinginkan untuk dilihat hasilnya.

h) Kesimpulan

Tahap kesimpulan yaitu membuat laporan mengenai program yang telah diuji pada tahap sebelumnya. Pada tahap ini akan dikemukakan kelebihan dan kekurangan dari program yang telah dibuat serta apa yang bisa dilakukan untuk membuat program menjadi lebih baik.

3.2 Rancangan

Program pemetaan wilayah ini secara umum dapat dijabarkan dalam diagram berikut yaitu:



Gambar 3.2 Blok Diagram Perancangan Program Penentuan Luas Wilayah

Program memerlukan dua hal penting yaitu *Image Processing Tool* dan *Image Threshold app*. *Image Processing Tool* merupakan komponen penting pada MATLAB untuk melakukan manipulasi pada foto/citra. *Image Threshold App* berfungsi untuk mencari *threshold* dari foto/citra yang akan diproses dan mengekspor hasilnya dalam bentuk *pseudocode*.

Berkas *masukan* yang digunakan pada penelitian ini adalah sebelas gambar MODIS (*Moderate Tesolution Imaging Spectroradiometer*) Laut Aral yang diambil dari satelit NASA. Berkas citra diambil pada bulan Agustus dari tahun 2000 sampai tahun 2010. Citra juga menampilkan garis yang merepresentasikan garis pantai dari laut Aral. Sedangkan output dari Penelitian ini adalah gambar *binary* dan hasil perhitungan luas laut Aral.

Secara garis besar program pemetaan memiliki cara kerja sebagai berikut, yaitu pertama program akan meminta user untuk memasukkan berkas gambar (harus lebih dari satu), kemudian program akan meminta kalibrasi jarak spasial pada citra dengan jarak yang sebenarnya untuk keperluan penghitungan luas citra. Program akan memproses citra dari ruang warna RGB ke HSV kemudian ke *binary* (*BW Image*) menggunakan seleksi warna tertentu, yang telah dicari menggunakan *image threshold app*, sehingga dapat dihitung jumlah piksel. Program akan menampilkan hasil *threshold* citra pada tampilan selanjutnya. Kemudian program menghitung jumlah piksel pada bagian putih menggunakan algoritma NNZ (*Number of Non Zero Matrix Element*) dan melakukan perhitungan luas, setelah itu menampilkannya pada MATLAB.

3.2.1 Struktur Program

Program secara umum mengandung tiga *function*, yaitu:

- a. Menu
- b. Kalibrasi
- c. Menampilkan semua citra

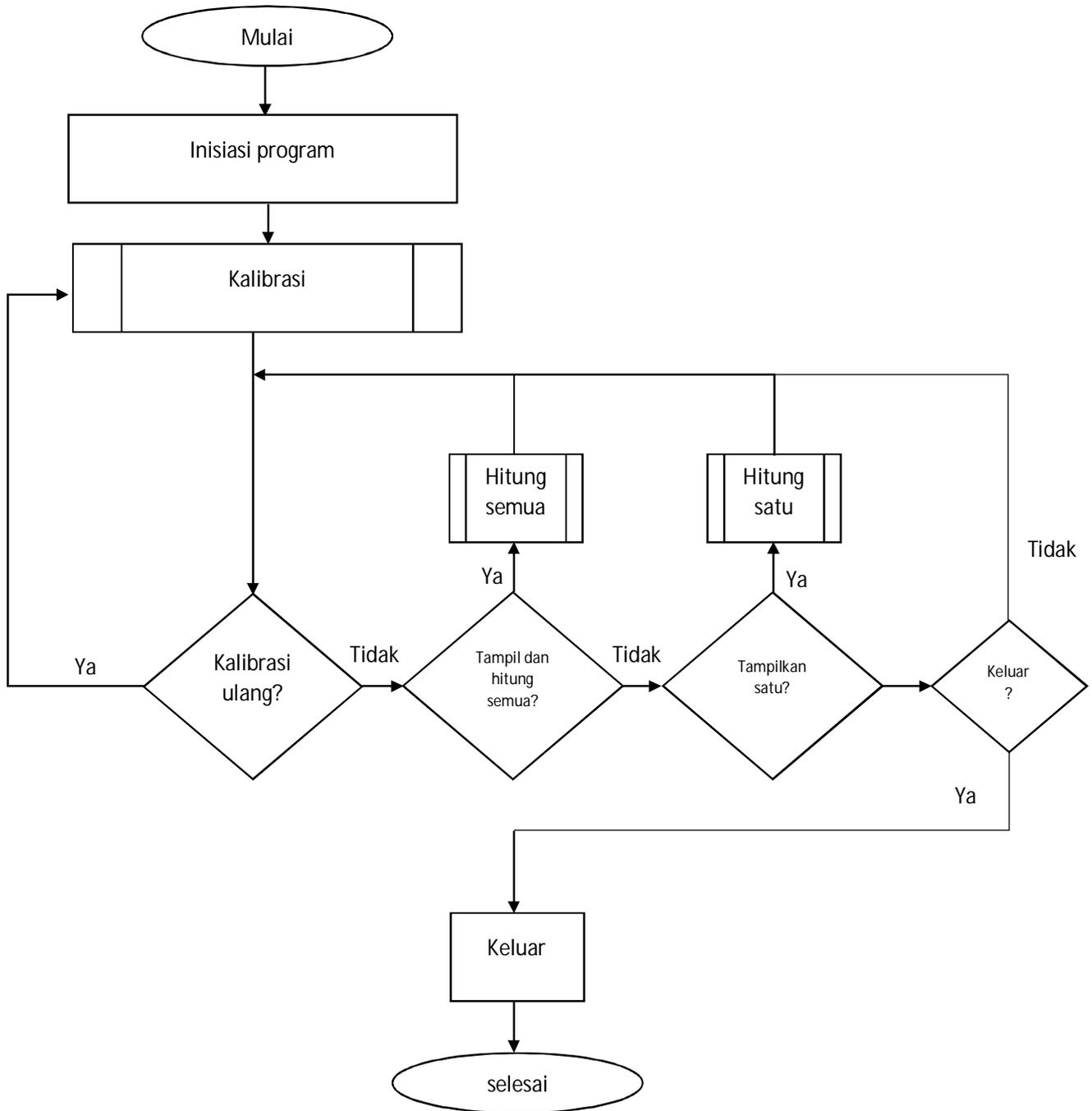
d. Menampilkan satu citra

Ketiga *function* tersebut dijelaskan sebagai berikut:

A. Menu

Menu mengandung semua pilihan dari *function* secara umum yang dimiliki oleh program. Menu menggunakan fungsi *While* dan *Case*, dengan 4 pilihan yaitu kalibrasi ulang, tampilkan semua, tampilkan satu hasil, dan keluar. *Flowchart* blok menu adalah seperti dibawah ini:

Flowchart Menu:



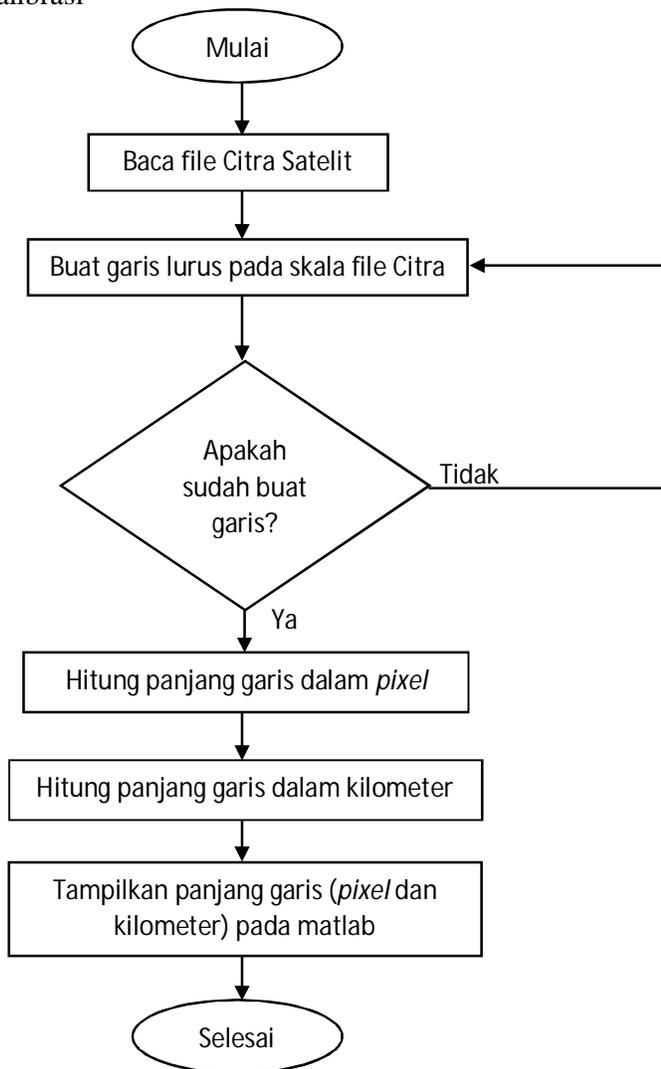
Gambar 3.3 Flowchart menu

Blok Menu berfungsi untuk memberikan pilihan pada pengguna untuk menentukan jalannya program. Program berisi empat subfungsi yang dapat dipilih dari blok menu, yaitu kalibrasi(), tampilsemua(), tampilsatu(), dan keluar.

Pseudocode blok menu adalah sebagai berikut:

```
While i<4
    If i>1
        Menu(judul, pilihan1, pilihan2, pilihan 3, pilihan4)
    End
Switch menu
    Case 1
        Argumen1
    Case 2
        Argumen2
    Case_n
        Argumen_n
end
```

B. Kalibrasi



Gambar 3.4 Flowchart Kalibrasi Spasial Citra

Kalibrasi dilakukan dengan mengukur jumlah piksel pada suatu garis lurus yang ditentukan, kemudian menghitung dengan rumus:

$$\text{Jarak dalam piksel} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Dimana x adalah koordinat X dan y adalah koordinat Y pada citra. Setelah jarak diketahui maka dihitung jarak per piksel dan jumlah piksel per jarak, dengan perhitungan sebagai berikut:

$$\text{Jarak per Piksel} = \frac{\text{Jarak Sebenarnya}}{\text{Jarak dalam pixel}},$$

Pada pemrograman MATLAB, setiap variabel bersifat lokal, yaitu hanya dapat digunakan di dalam satu fungsi. Jika kita ingin menggunakan variabel dari suatu fungsi tertentu maka variabel harus dideklarasikan sebagai variabel global yaitu:

Global variable_n

Setelah dideklarasikan maka variable dapat digunakan disetiap fungsi yang memerlukan variable tersebut.

Blok rumus perhitungan jarak garis adalah:

```
distanceInPixels = sqrt( (xi(2)-xi(1)).^2 + (yi(2)-yi(1)).^2);
```

dimana x dan y merupakan posisi X,Y diagram kartesian citra pada figure. Hasil dari perhitungan tersebut adalah dalam unit piksel. Blok konversi dari piksel ke unit yang telah kita pilih adalah:

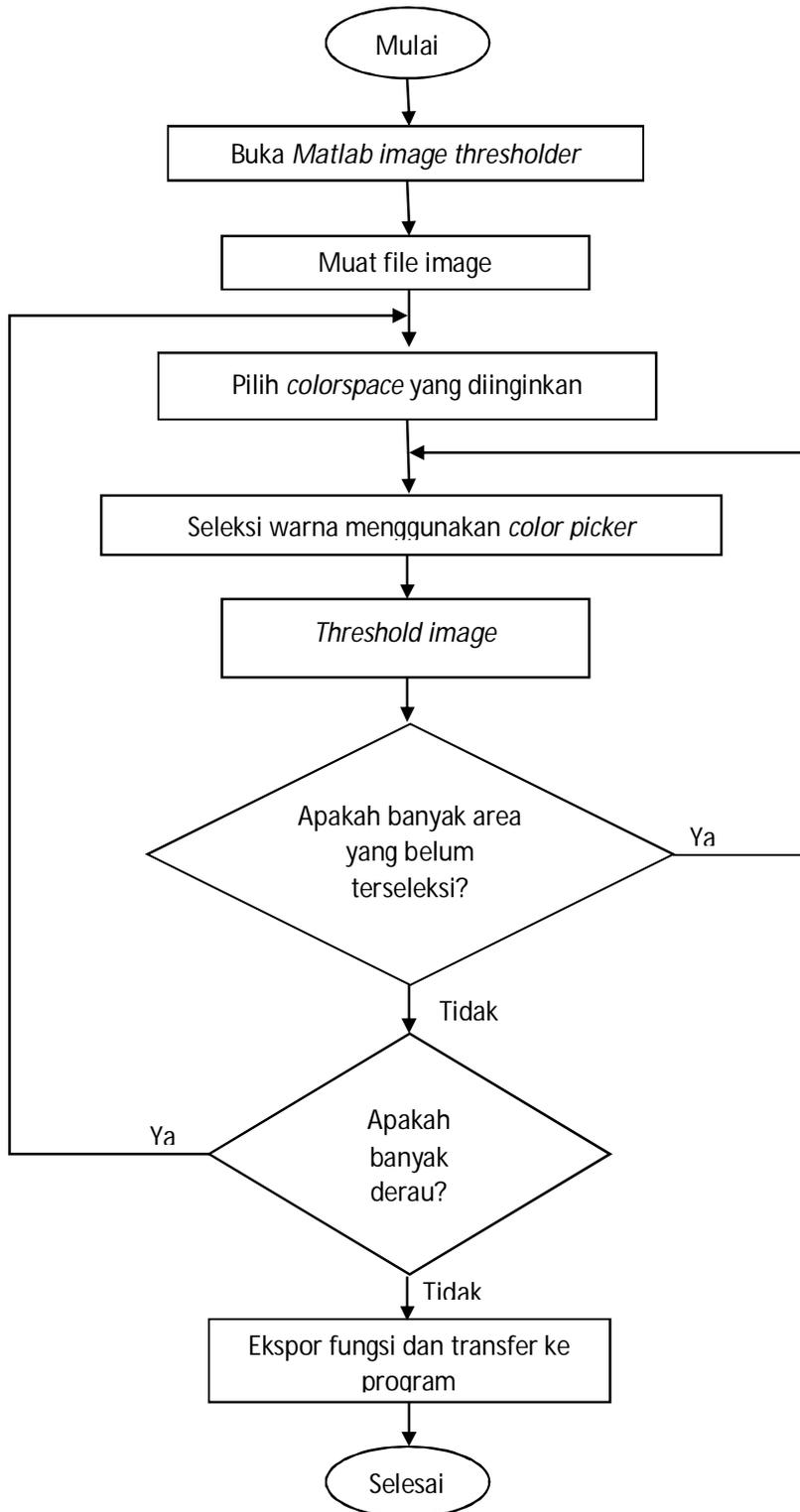
dimana piksel ke jarak = 1/jarak dalam piksel, dengan hasil jumlah jarak per piksel.

3.2.2 Algoritma *Image Processing*

Image processing yang digunakan pada penelitian ini ada dua, yaitu konversi RGB ke L*a*b dan konversi RGB ke HSV. Penelitian ini menggunakan dua macam ruang warna tersebut karena HSV dan L*a*b yang paling cocok dengan citra yang digunakan pada penelitian.

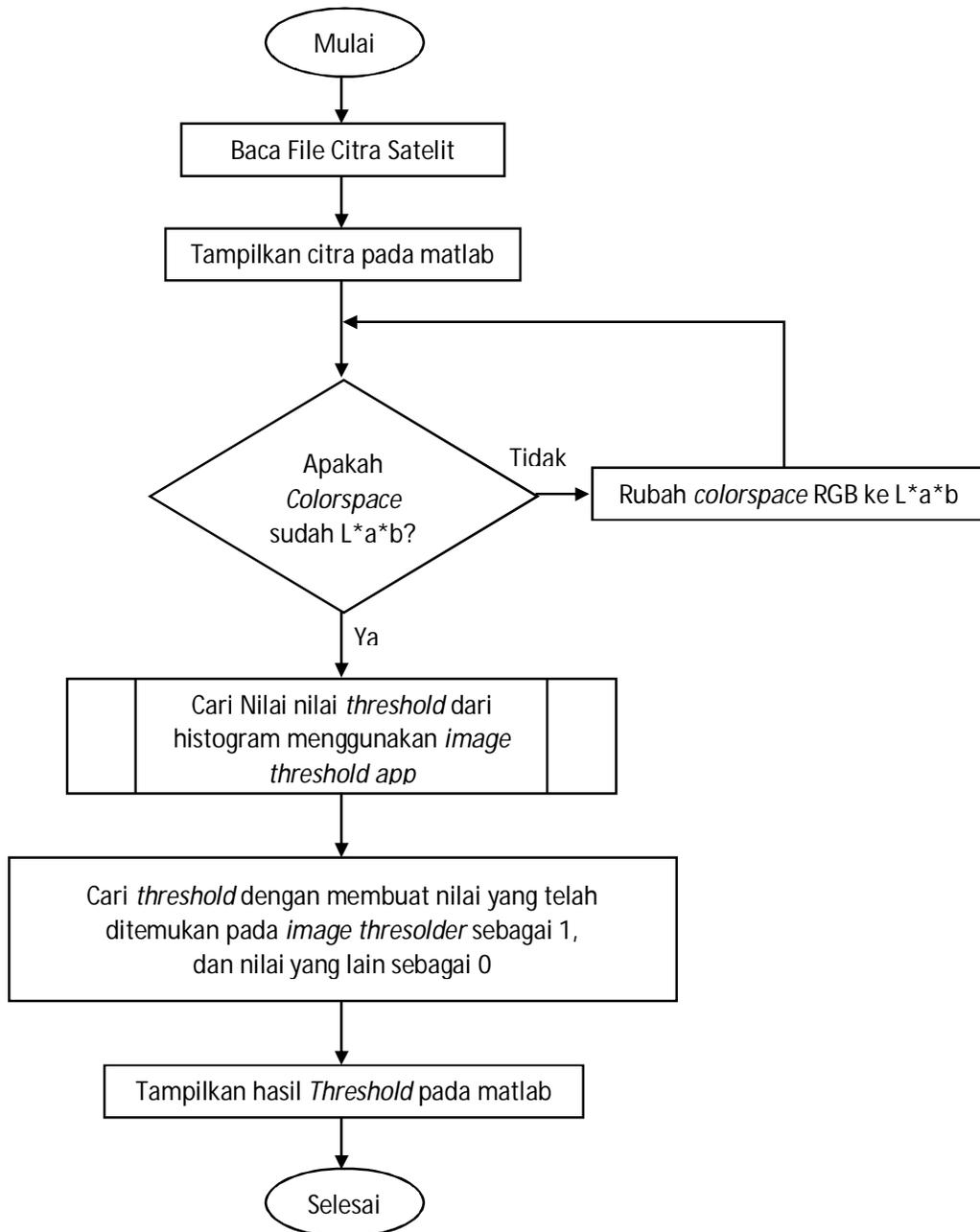
Pada *color thresholder app*, gambar akan dimuat pada aplikasi, kemudian pengguna memilih ruang warna yang diinginkan. Setelah ruang warna ditentukan pengguna akan menyeleksi warna menggunakan *slider* pada *channel-channel* di *thresholder app* atau menggunakan *lasso tool*. Setelah diseleksi maka *image* akan dirubah menjadi *binary image*.

Pengguna dapat memilih ruang warna yang lain jika *binary image* tidak seperti yang diinginkan. Flowchart cara kerja *color thresholder app* dan konversi ruang warna RGB ke HSV dan L*a*b adalah sebagai berikut:



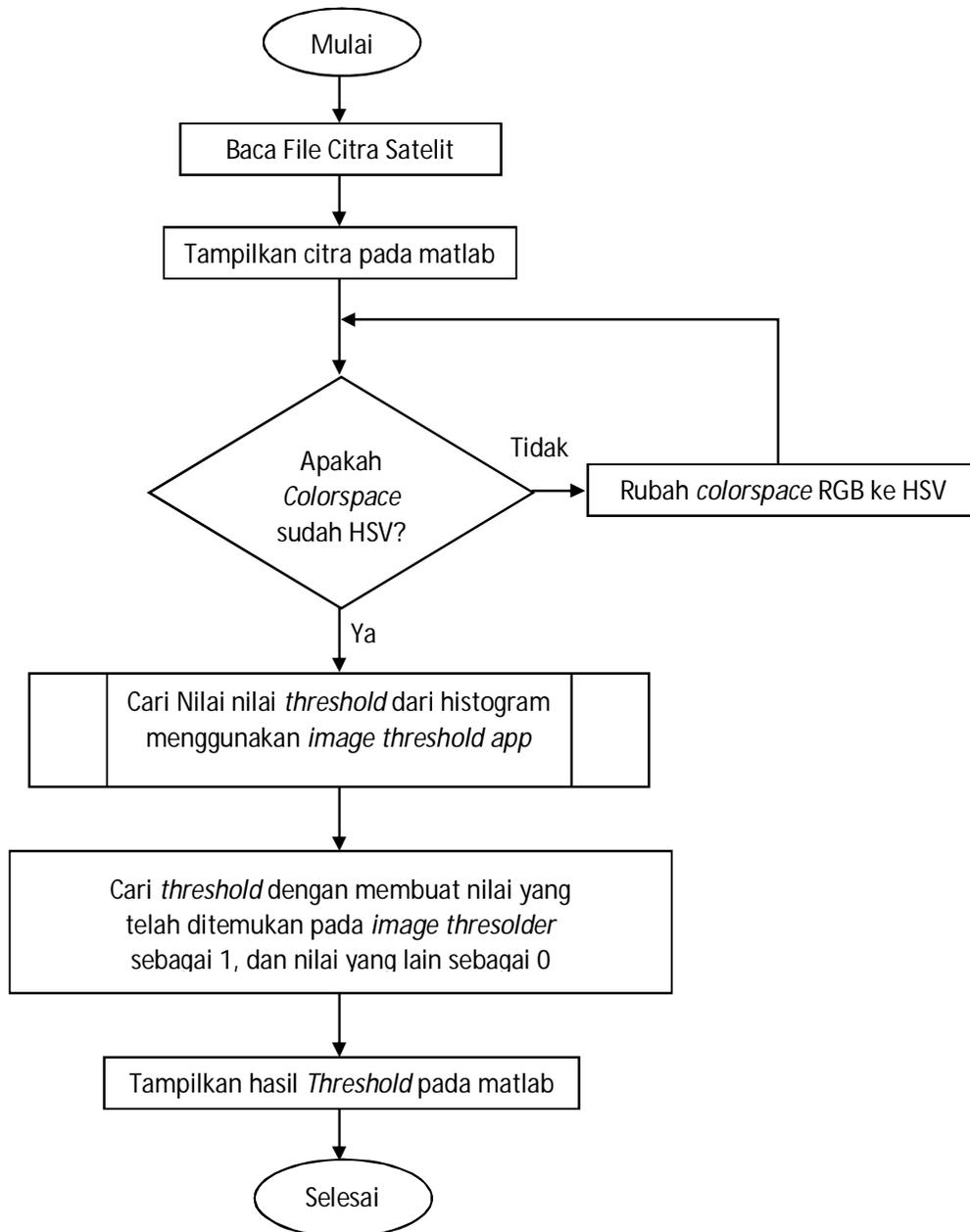
Gambar 3.5 Flowchart Matlab Color Thresholder App

Konversi ruang warna RGB ke L*a*b dapat dilihat pada flowchart berikut:



Gambar 3.6 Flowchart Image Processing RGB2CIELAB

Sedangkan konversi RGB ke HSV dapat dilihat pada flowchart berikut:



Gambar 3.7 Flowchart Image Processing RGB2HSV

Pada ruang warna L^*a^*b , *channel* 1 merupakan *channel* L, *channel* 2 merupakan *channel* a, dan *channel* 3 merupakan *channel* b. Sedangkan pada ruang warna HSV, *channel* 1 merupakan komponen H, *channel* 2 merupakan komponen S, dan *channel* 3 merupakan komponen V. setiap *channel* pada blok program mewakili satu komponen dari citra. Nilai dari *channel* diketahui berdasarkan histogram per *channel*. Setelah diketahui nilai masing-masing, maka dihitung untuk dicari *threshold* dari Citra dengan cara mencari nilai diantara *channel* min dan *channel* max. Hasil dari *threshold* adalah warna yang diseleksi akan bernilai “1” dan warna yang tidak diseleksi akan bernilai “0”.

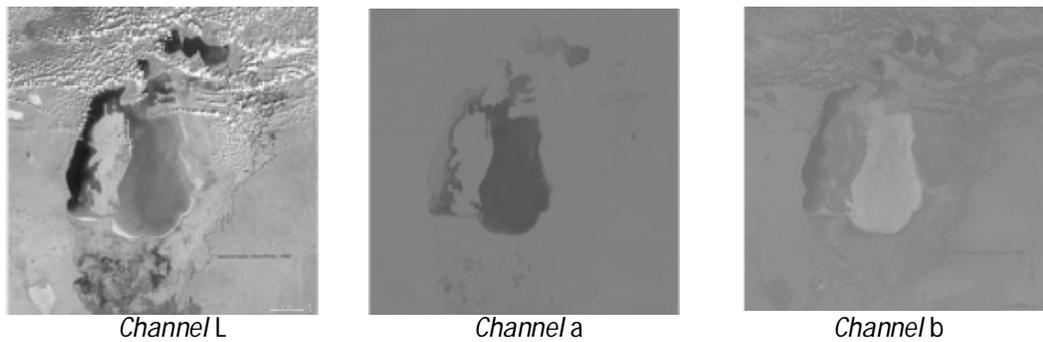
Setelah *threshold* citra diketahui, maka langkah selanjutnya adalah menghitung jumlah piksel yang ada pada bagian yang terseleksi (bernilai 1). Kita menggunakan sintaks *nnz* untuk menghitung jumlah piksel pada bagian image yang diseleksi, yaitu:

$$ARAL1 = nnz(ARAL);$$

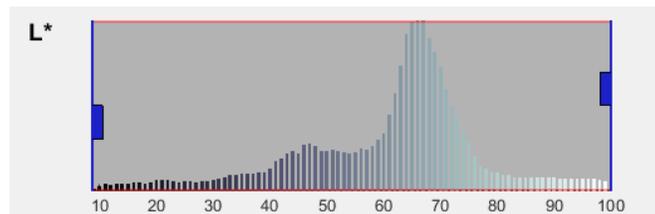
yaitu fungsi “*Number of nonzero matrix elements*” untuk menghitung jumlah piksel yang terdapat pada bagian warna yang diseleksi. Setiap elemen pada citra yang telah di-*threshold* mewakili satu piksel. Luas pada bagian yang diseleksi dihitung dengan rumus:

$$\text{luas wilayah} = \text{jumlah piksel} \times (\text{jarak per piksel})^2$$

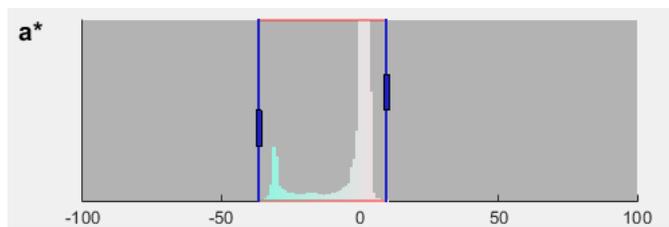
Jarak per piksel didapat dari pengukuran kalibrasi spasial dan jumlah piksel didapat dari menghitung jumlah komponen matriks citra yang bernilai 1.



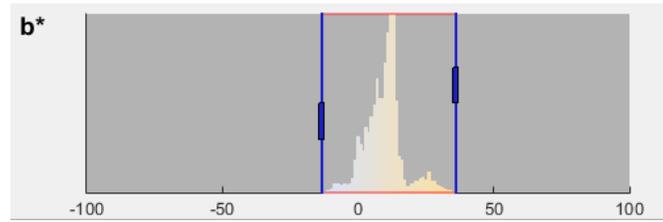
Gambar 3.8 Channel L*a*b Citra Laut Aral



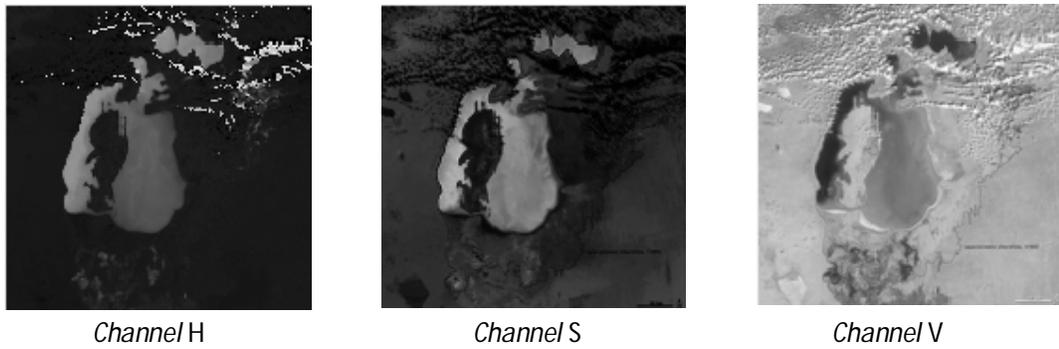
Gambar 3.9 Histogram *channel L*



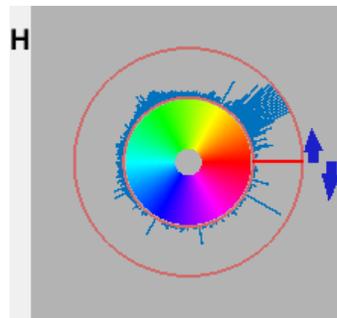
Gambar 3.10 Histogram *channel a*



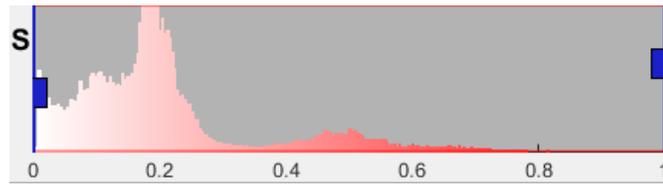
Gambar 3.11 Histogram *channel b*



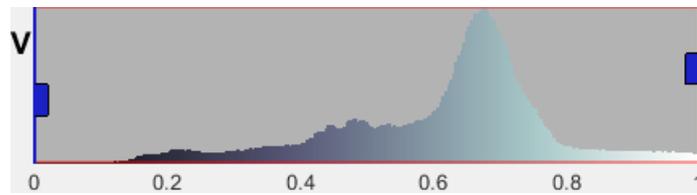
Gambar 3.12 Channel L^*a^*b Citra Laut Aral



Gambar 3.13 Histogram *channel H (Hue)*



Gambar 3.14 Histogram *channel S (Saturation)*



Gambar 3.15 Histogram *channel V (Value)*