

BAB III

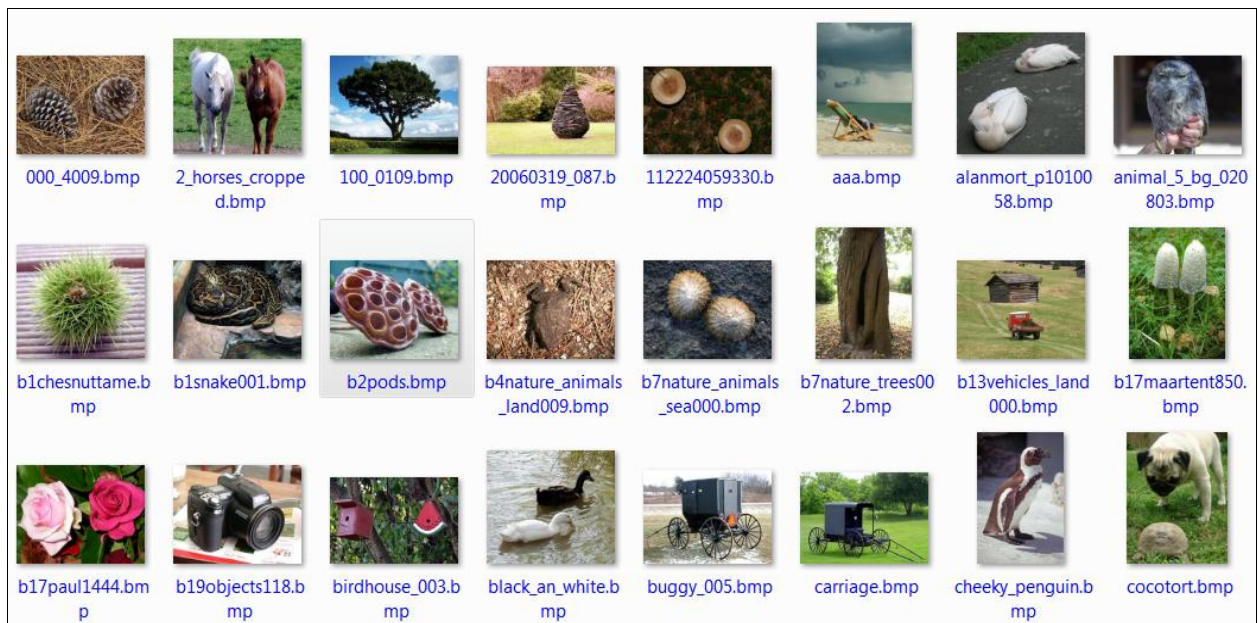
METODE PENELITIAN

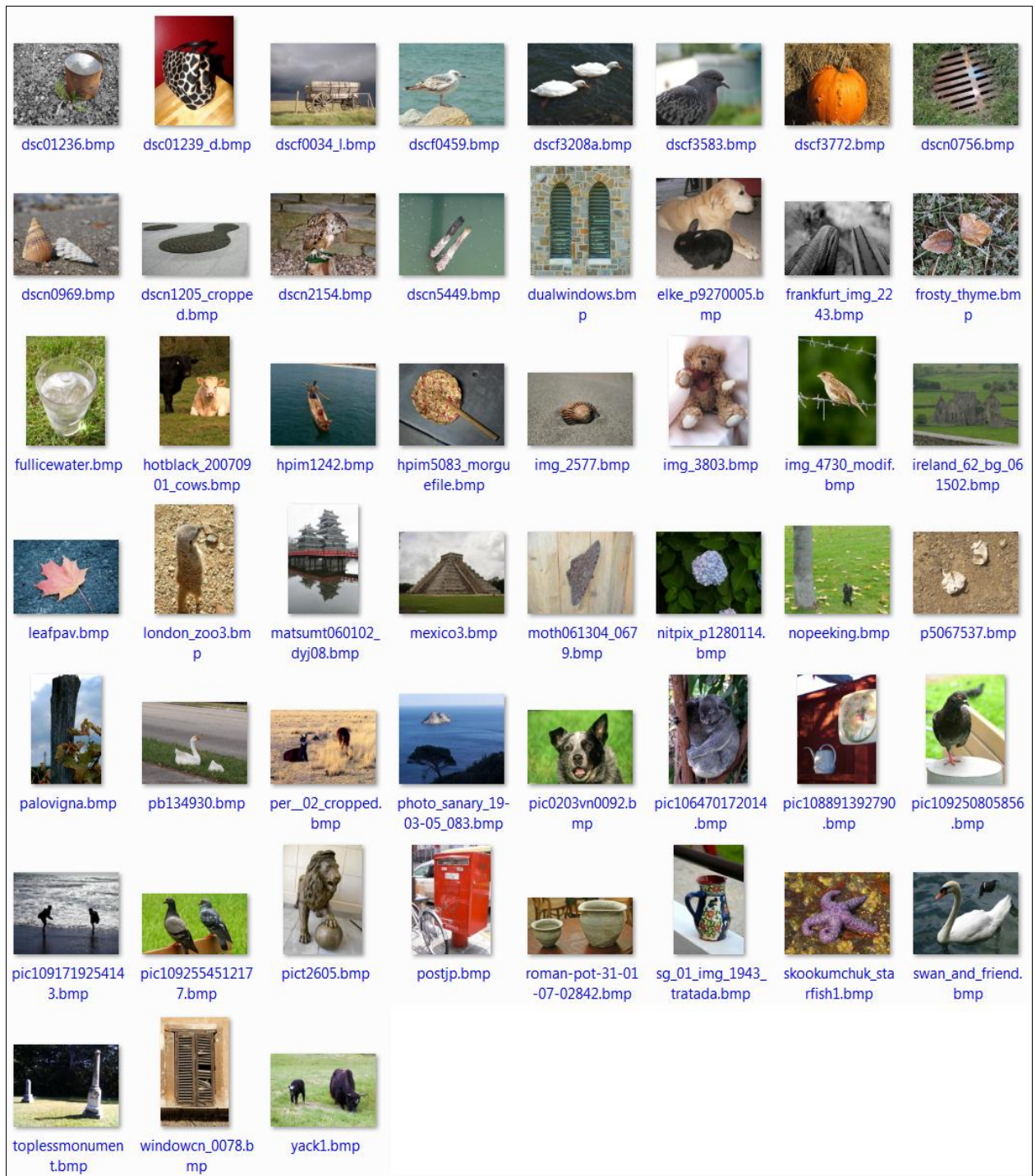
Penelitian ini menggunakan jenis penelitian eksperimen, dengan tahapan penelitian sebagai berikut:

3.1 Pengumpulan Data

Tahap ini merupakan langkah awal dari penelitian. *Dataset* citra yang akan digunakan dalam eksperimen diambil dataset publik yang ada pada laman *Segmentation Evaluation Benchmark* dari *Department of Computer Science and Applied Mathematics, Weizmann Institute of Science*. Dataset pada laman tersebut berjumlah 200 buah citra. Pada penelitian ini akan diambil 75 citra yaitu hanya citra yang mempunyai background yang kompleks. *Dataset* telah dilengkapi dengan citra hasil *human segmentation* yang berguna untuk pengukuran akurasi. Citra untuk *benchmark* juga dipilih sejumlah 75 buah citra, sesuai dengan citra *dataset* yang dipilih. *Dataset* dapat diakses pada alamat http://www.wisdom.weizmann.ac.il/~vision/Seg_Evaluation_DB/dl.html.

Citra *dataset* yang dipilih adalah sebagai berikut:

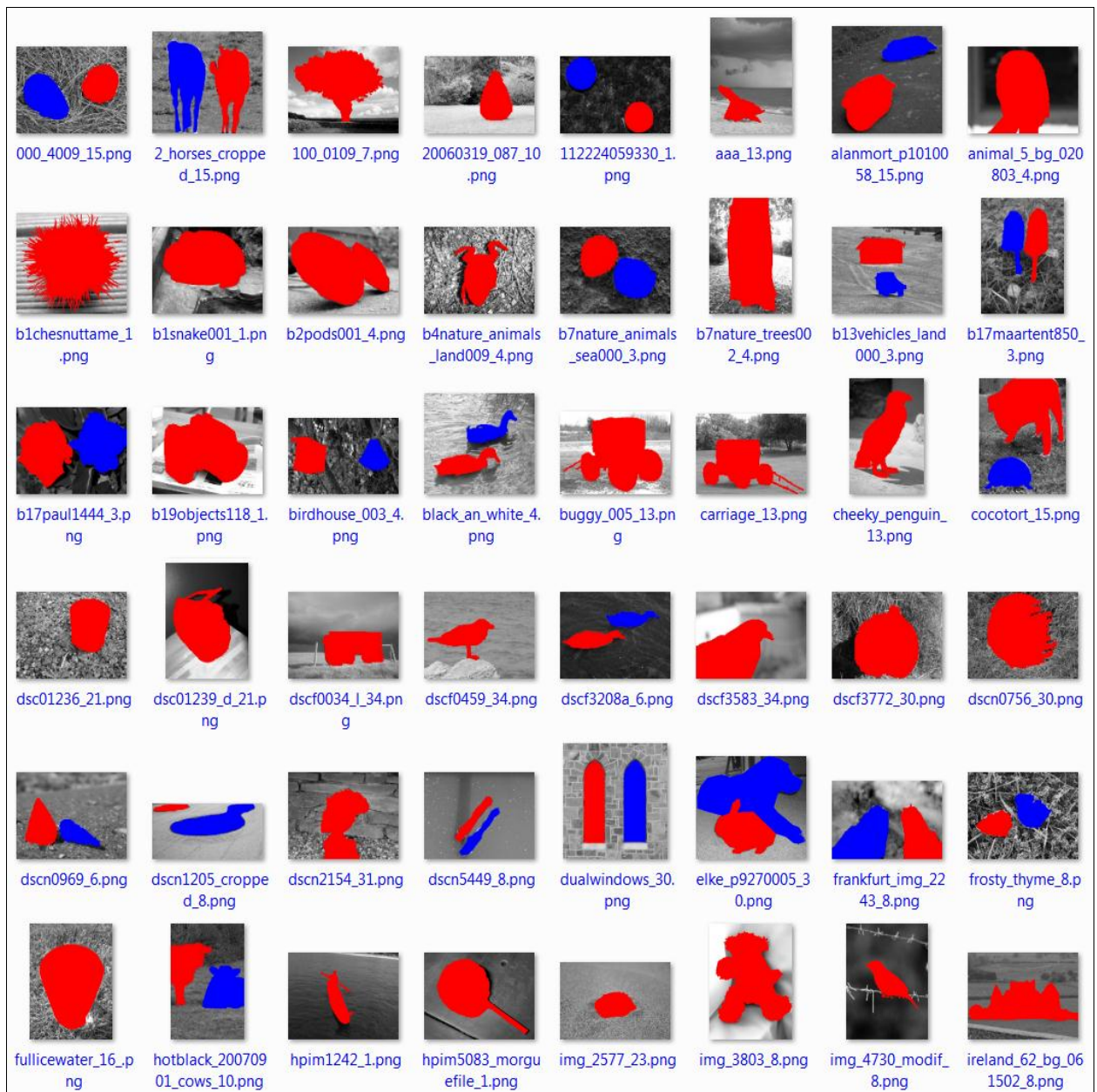


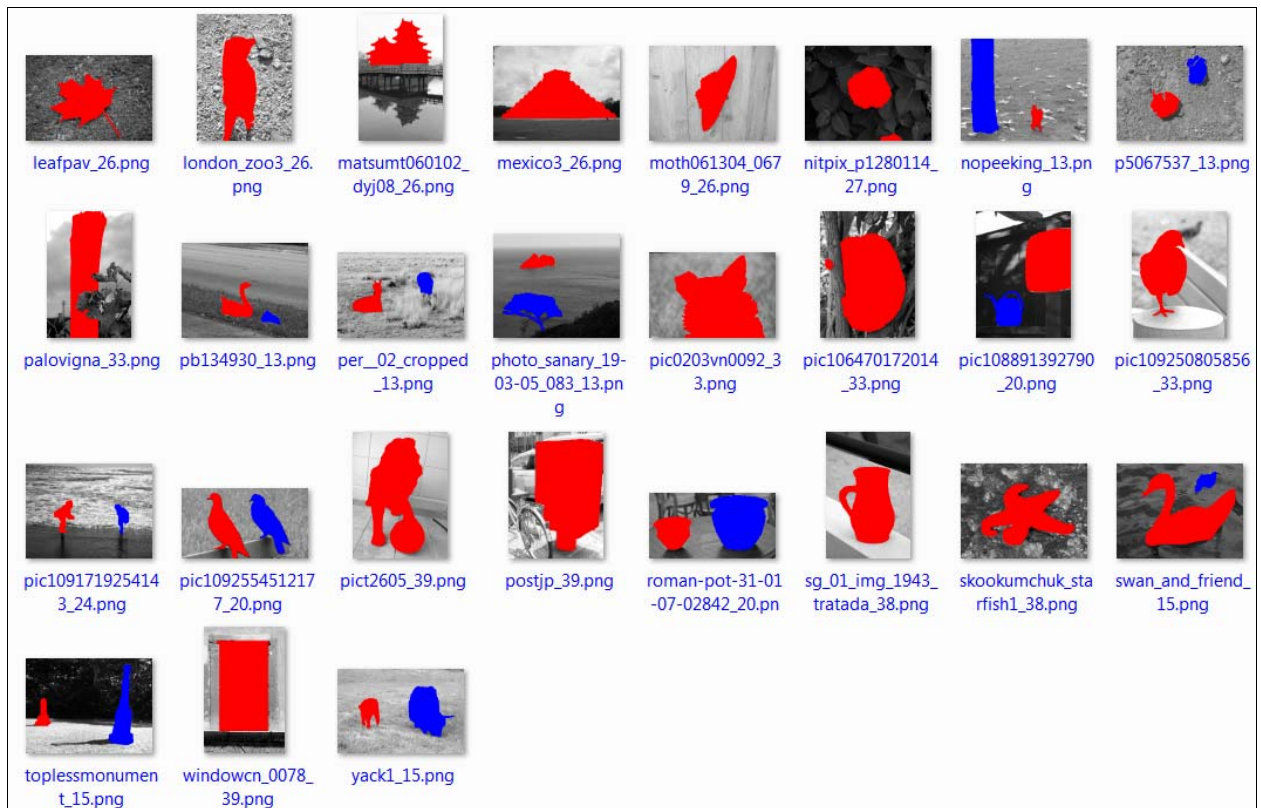


Gambar 3. 1 Citra *dataset*

Citra pembandingan sebagai *benchmark* untuk pengukuran akurasi akan digunakan citra hasil *manual human segmentation*. Citra ini adalah hasil segmentasi yang dilakukan oleh manusia secara manual. Obyek pada citra ditandai dan selanjutnya diberi warna berbeda mengikuti tepi dari obyek.

Citra *benchmark* adalah sebagai berikut:





Gambar 3. 2. Citra *benchmark* untuk pengukuran akurasi

3.2 Pengolahan Data

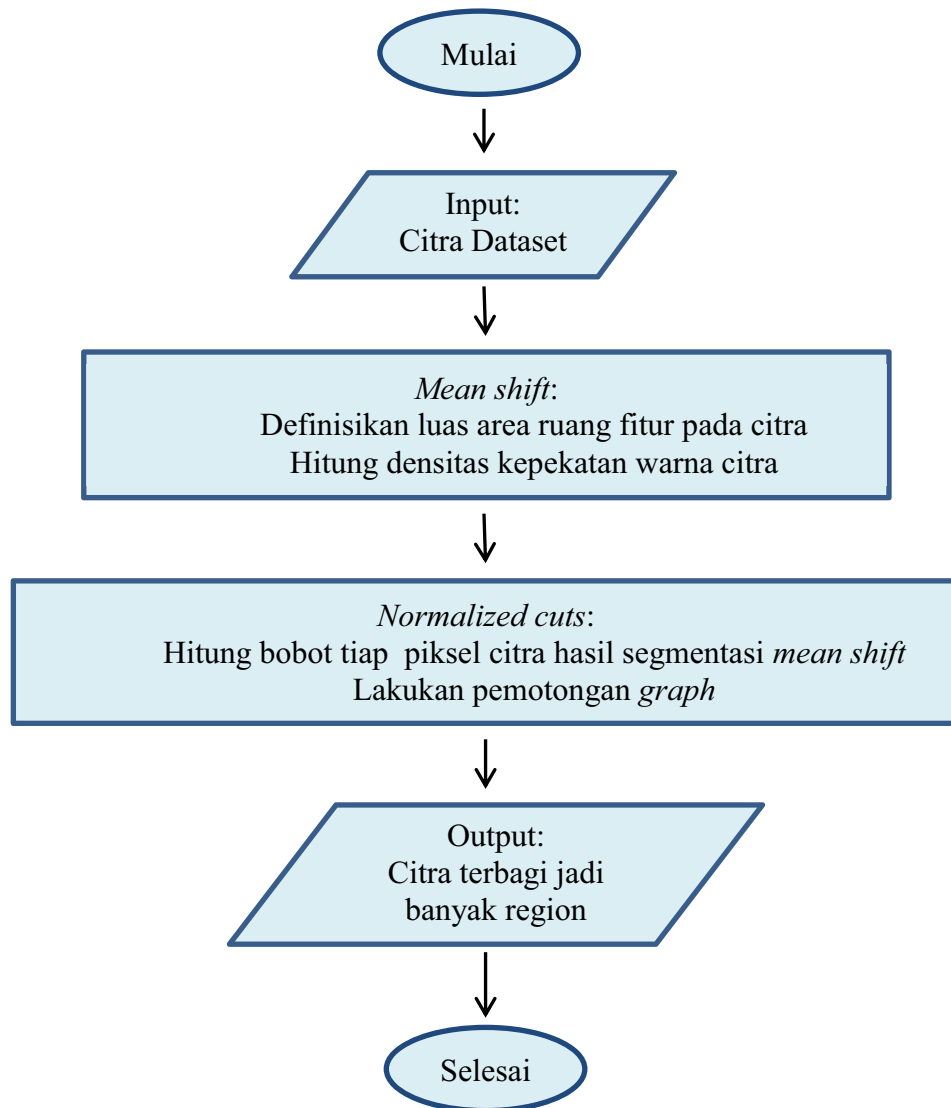
Pada data yang telah dikumpulkan akan dilakukan pengolahan data yaitu berupa eksperimen menggunakan model yang diusulkan yaitu segmentasi obyek secara *semi*-otomatis dengan metode *region merging maximal similarity* berbasis integrasi algoritma *mean shift* dan *normalized cuts*. Perangkat atau *tools* yang akan digunakan adalah MATLAB R2010.

Eksperimen dalam penelitian ini dibagi menjadi 3 tahap, yaitu:

1. Proses segmentasi awal atau *low level* dengan algoritma *mean shift + normalized cuts*.
2. Penambahan deteksi tepi *prewitt* dan operasi *invers* terhadap hasil segmentasi *low level*.
3. Proses *region merging* citra hasil segmentasi *low level* untuk memisahkan obyek dari *background* pada citra.

3.2.1 Segmentasi Awal

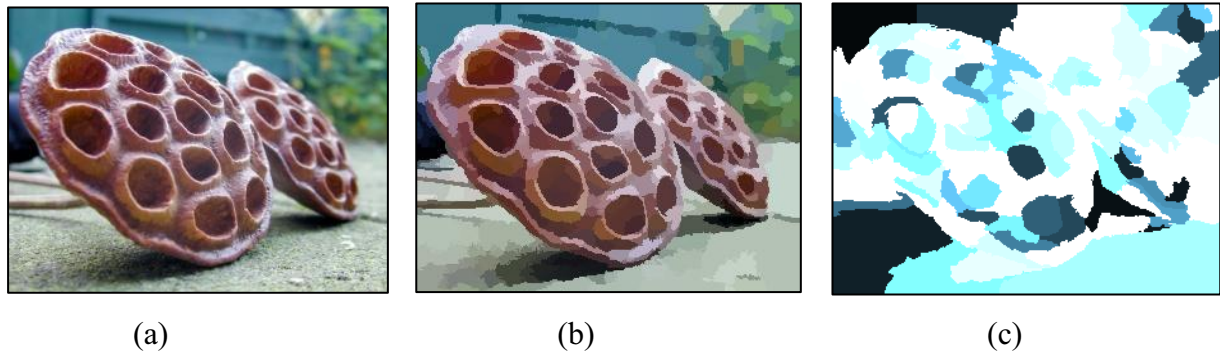
Diagram alir proses segmentasi awal atau *low level* adalah sebagai berikut:



Gambar 3. 3 Diagram alir proses segmentasi awal atau *low level*

Pada langkah pertama citra akan diolah menggunakan segmentasi *low level* untuk mendapatkan hasil berupa citra yang tersegmentasi menjadi *region-region* kecil. Citra yang terbagi menjadi *region-region* tersebut akan dikomputasi lagi pada tahap selanjutnya menggunakan metode *region merging* untuk memisahkan obyek dari *background*. Untuk membandingkan hasil segmentasi *low level* antara algoritma integrasi atau gabungan dengan algoritma asli, maka tiap citra akan disegmentasi sejumlah dua kali yaitu dengan algoritma *mean shift* dan dengan metode

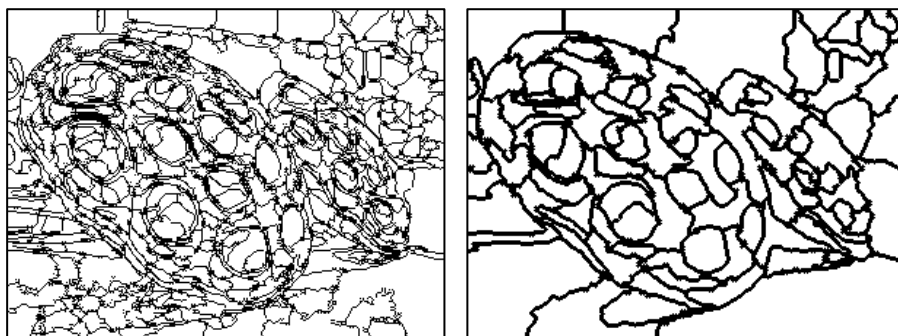
gabungan *mean shift + normalized cuts*. Hasil yang didapatkan dari proses ini adalah sebagai berikut:



Gambar 3.4 (a) citra *dataset*, (b) *mean shift*, (c) *mean shift + normalized cuts*

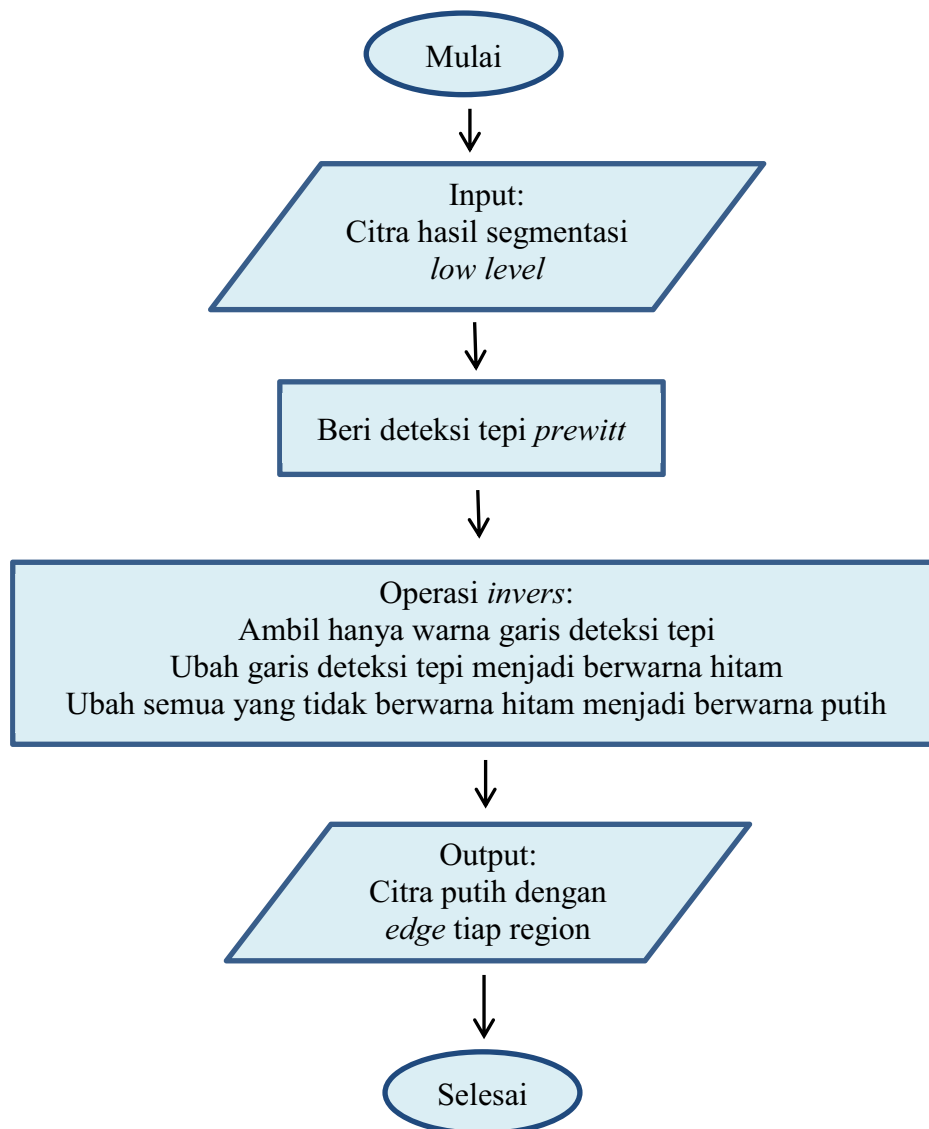
3.2.2 Deteksi Tepi *Prewitt* dan Operasi *Invers*

Selanjutnya hasil segmentasi *low level* akan diberi deteksi tepi *prewit* dan operasi *invers* untuk memperlihatkan tepi dari area segmentasi. Deteksi tepi *prewit* akan memberikan garis hitam pada tepi area. Operasi *invers* dimaksudkan untuk mengambil hanya garis hitam tepi area, sedangkan warna pada area tersegmentasi akan dihilangkan dan diganti dengan warna putih. Hasilnya adalah citra putih dengan tepi area tersegmentasi berwarna hitam dan mengandung informasi tentang tepi obyek di dalam citra. Hasil dari proses deteksi tepi *prewit* dan operasi *invers* citra segmentasi awal adalah sebagai berikut:



Gambar 3.5 Citra *edge prewitt+invers*, (a) *mean shift*, (b) *mean shift + normalized cuts*

Diagram alir proses deteksi tepi *prewitt* dan operasi *invers* adalah sebagai berikut:

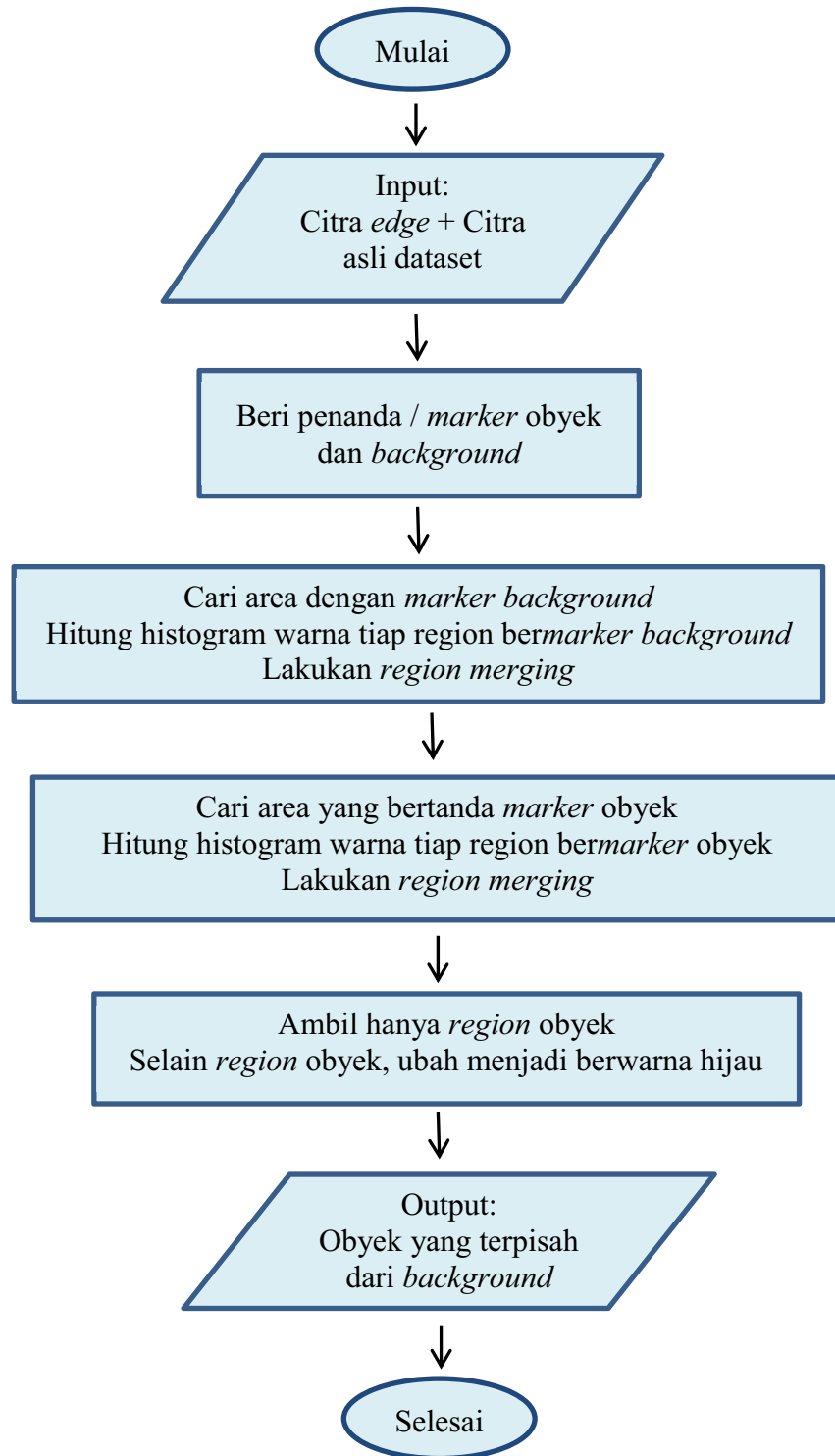


Gambar 3. 6 Diagram alir deteksi tepi *prewitt* dan operasi *invers*

3.2.3 Proses *Region Merging*

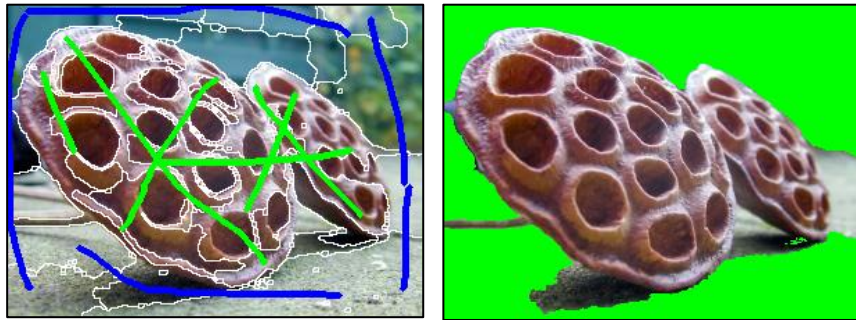
Hasil segmentasi awal yang berupa *region-region* tersegmen akan dikomputasi dengan metode *region merging* dengan pengukuran *maximal similaritas* pada masing-masing *region*. Pertama-tama citra asli akan digabungkan dengan citra deteksi tepi *prewit* plus *invers* setelah itu proses *region merging* akan dipandu dengan garis bantu dari *user* yang menandai bagian obyek dan bagian *background* pada citra.

Diagram alir proses *region merging* adalah sebagai berikut:



Gambar 3. 7 Diagram alir proses *region merging*

Hasil akhir segmentasi obyek dengan metode *region merging maximal similarity* adalah sebagai berikut:



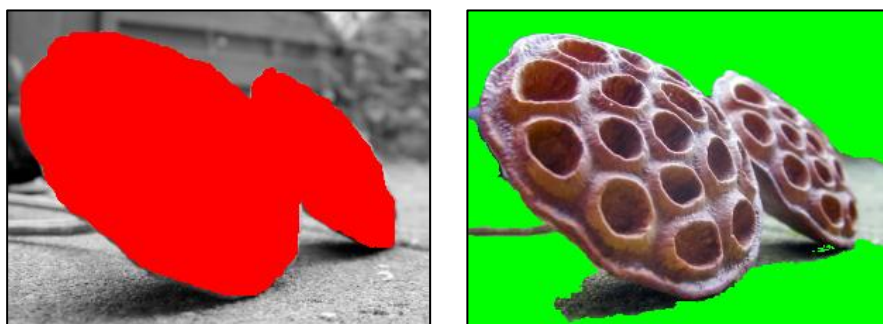
Gambar 3.8 Proses *region merging*, (a) *marker*, (b) hasil segmentasi obyek

3.3 Pengukuran Akurasi Hasil Segmentasi

Pengukuran akurasi dilakukan dengan membandingkan antara hasil segmentasi dengan citra *benchmark* dari *dataset*. Pengukuran dilakukan dengan menghitung nilai *bit error rate (BER)* dari citra hasil segmentasi. Penghitungan nilai *bit error rate* dilakukan cara membandingkan tiap-tiap piksel citra *head to head* dibagi dengan keseluruhan dimensi matriks.

Untuk mengukur akurasi dari hasil segmentasi citra maka metode yang digunakan adalah membandingkan antara citra *benchmark* dengan citra hasil segmentasi. Pada penelitian kali ini yang dibandingkan adalah citra *dataset* hasil *human segmentation* dengan citra hasil segmentasi metode yang diusulkan. Yang diukur adalah tepi dari obyek, semakin mendekati dengan tepi obyek pada *benchmark* maka hasil segmentasi semakin akurat.

Contoh citra yang akan dibandingkan adalah sebagai berikut:



Gambar 3.9 Pengukuran akurasi, (a) citra *benchmark*, (b) citra hasil segmentasi

Ubah citra hasil segmentasi dan citra *benchmark* ke dalam citra hitam putih atau citra *biner*. Hal ini karena yang diukur adalah potongan hasil segmentasi, sehingga warna dari citra dapat diabaikan. Saat citra diubah menjadi citra *biner* maka matriks warnanya hanya menjadi 2, yaitu nilai 1 untuk warna putih dan nilai 0 untuk warna hitam. Untuk menyamakan pengukuran maka pada kedua citra yang akan dibandingkan yaitu citra *benchmark* dan citra hasil segmentasi, bagian obyek akan diubah menjadi berwarna putih, sedangkan bagian *background* akan diubah menjadi berwarna hitam untuk selanjutnya akan dilakukan perhitungan perbedaan *absolut* pada masing-masing piksel citra. Hasil perubahan warna citra *benchmark* dan citra hasil segmentasi menjadi citra biner adalah sebagai berikut:



Gambar 3. 10 Citra *biner*, (a) *benchmark*, (b) hasil segmentasi

Nilai pada piksel pada baris pertama, kolom pertama pada citra 1 dibandingkan dengan baris pertama kolom pertama pada citra 2. Demikian seterusnya sampai seluruh titik piksel dalam citra dibandingkan. Setelah berhasil dibandingkan maka jumlah total dari bit data yang berbeda akan dibagi dengan dimensi piksel dari citra, dalam hal ini adalah **panjang x lebar** dari citra sehingga didapatkan nilai *bit rate error*.

Contoh perhitungan nilai *bit rate error* untuk citra hitam putih adalah sebagai berikut

Citra *benchmark*

1	0	1
0	1	0
1	0	1

Citra hasil segmentasi

1	1	0
0	1	0
1	0	0

Dari matriks citra biner diatas, titik piksel yang mempunyai nilai berbeda adalah pada titik (2,1) , (3,1) dan titik piksel (3,3) sehingga jumlah keseluruhan error adalah 3. Sedangkan dimensi dari citra adalah $3 \times 3 = 9$. Dari contoh diatas didapatkan nilai *bit rate error* = 0.33. Nilai *bit rate error* = 0 mempunyai arti bahwa citra hasil segmentasi identik 100% dengan citra *benchmark*, karena penyimpangannya adalah 0 dan dapat diartikan bahwa hasil segmentasi sangat akurat karena ama persis dengan citra *benchmark*. Demikian juga sebaliknya, apabila nilai *bit rate error* semakin besar, dalam hal perbandingan citra hitam putih atau *biner*, nilai terbesar dari *bit rate error* pada adalah 1 mempunyai arti bahwa citra hasil segmentasi semakin berbeda dengan citra *benchmark* sehingga disebutkan bahwa hasil segmentasi tidak akurat.