BAB III
PERANCANGAN SISTEM

Perancangan adalah proses menuangkan ide dan gagasan berdasarkan teori-teori dasar yang mendukung. Pada bab ini, pembahasan materi difokuskan pada perancangan dan pembuatan sistem. Perancangan sistem merupakan tahap awal yang penting dalam proses pembuatan program. Perancangan bertujuan agar dalam pembuatannya dapat berjalan secara sistematis dan terstruktur, sehingga hasil program dapat berjalan sesuai dengan apa yang dikehendaki. Pada perancangan sistem terdapat beberapa komponen dalam proses perancangannya, baik perangkat keras maupun perangkat lunak.

3.1 Perangkat Keras

Pada bagian ini menjelaskan perangkat keras yang digunakan dalam perancangan sistem seperti dibawah ini.

Tabel 3.1 Spesifikasi Perangkat Keras

Perangkat keras	Jumlah	Spesifikasi
Laptop	1 Unit	➢ Intel Pentium Dual-Core 1.8 GHz ➢ Memory 1536 Mb/1.50 Gb ➢ Hardisk 160 Gb
WebCam USB 2.0	1 unit	*Interpolated to 16 MP Image Capture*

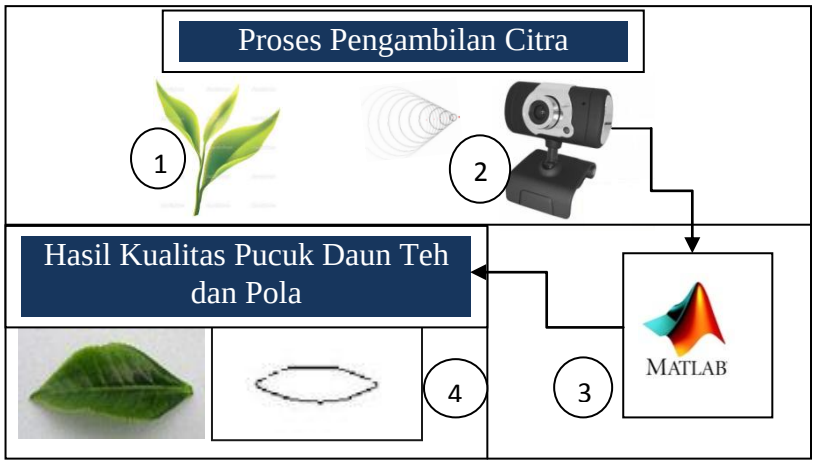
3.2 Perangkat Lunak

Pada bagian ini menjelaskan perangkat lunak yang digunakan sebagai berikut:

1. MatLab 7.8.0.347 (R2009a), 32-bit (win32)
2. Sistem operasi windows 7 ultimate 32-bit.

3.3 Diagram Blok

Pada Gambar 3.1 adalah diagram blok sistem secara keseluruhan. Kamera yang digunakan pada sistem ini sebagai alat untuk mengambil gambar adalah kamera jenis *MicroPack* C-16FT. Kamera jenis ini mampu untuk menangkap gambar yang baik, kontrol gambar saturasi, kontras dan tingkat kecerahan yang baik. Perangkat lunak yang digunakan untuk membangun aplikasi adalah MatLab versi 7 tahun 2009.



Gambar 3.1 Diagram Blok Keseluruhan Sistem.

Keterangan:

(1). Dimana pucuk daun teh yang akan diambil gambarnya adalah pucuk teh yang kedua dan yang ketiga, karena pucuk tersebut akan diolah sebagai teh hitam dan hijau. Sedangkan pucuk yang pertama atau paling ujung (pucuk peko) adalah diolah sebagai teh putih.

(2). Proses pengambilan gambar pucuk dengan kamera untuk proses tahapan selajutnya.

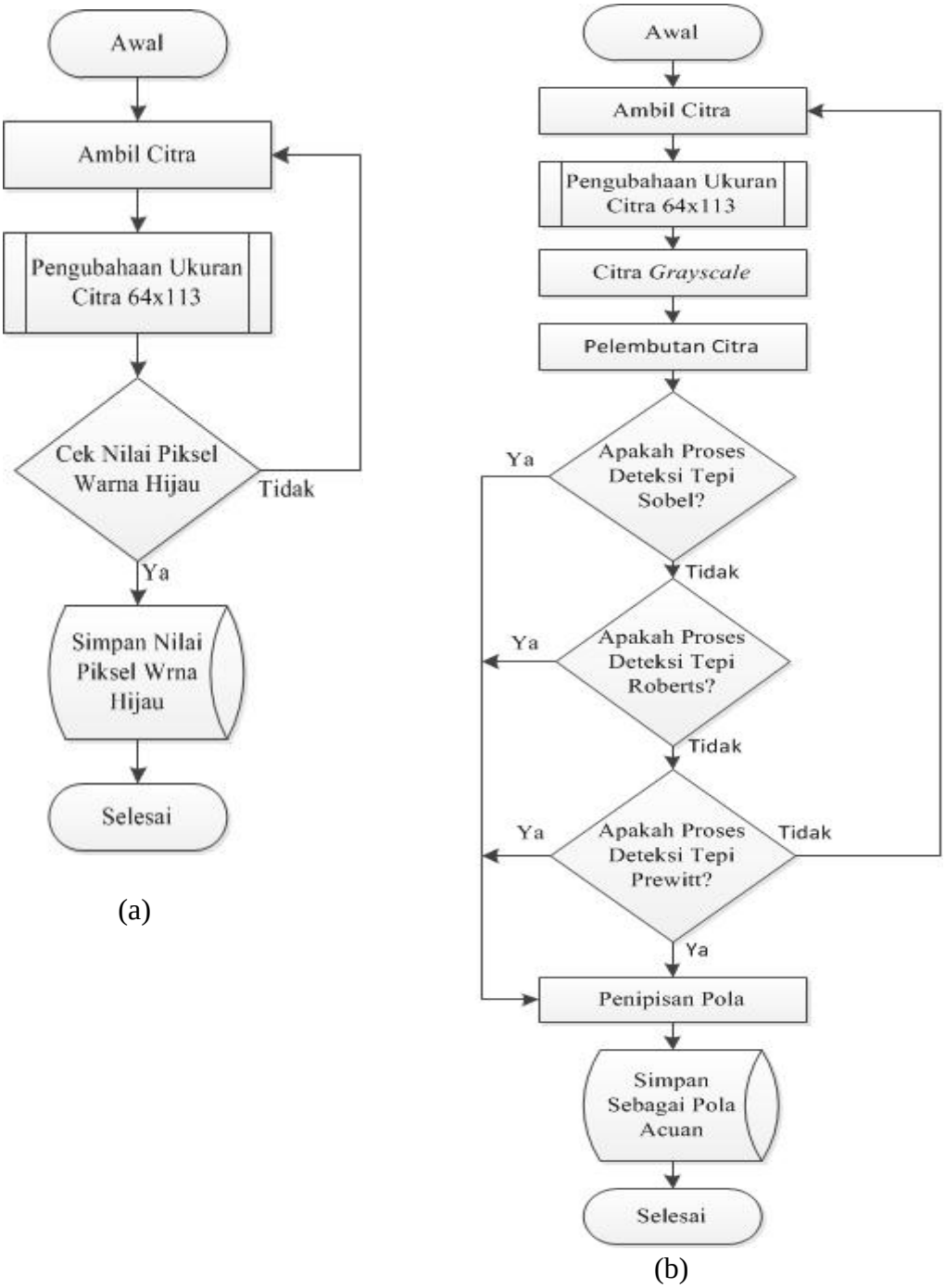
(3). MatLab berfungsi untuk mengolah citra dari hasil tahapan pengambilan gambar oleh kamera.

(4). Hasil deteksi tepi yang dirubah dalam bentuk pola dan kualitas pucuk daun teh.

3.4 Diagram Alir

Pembuatan sistem ini terdiri dari proses pengambilan gambar atau *capture* pada USB kamera, gambar yang dihasilkan kemudian diproses dengan deteksi tepi

turunan pertama kemudian dicari ciri dari bentuk pola daun dan nilai pixel tersebut sebagai gambar acuan untuk dikenali sebagai citra yang nantinya akan dibandingkan dalam bentuk pola daun dan nilai pixel. Gambar 3.2 menunjukkan diagram alir pengambilan sampel gambar acuan.

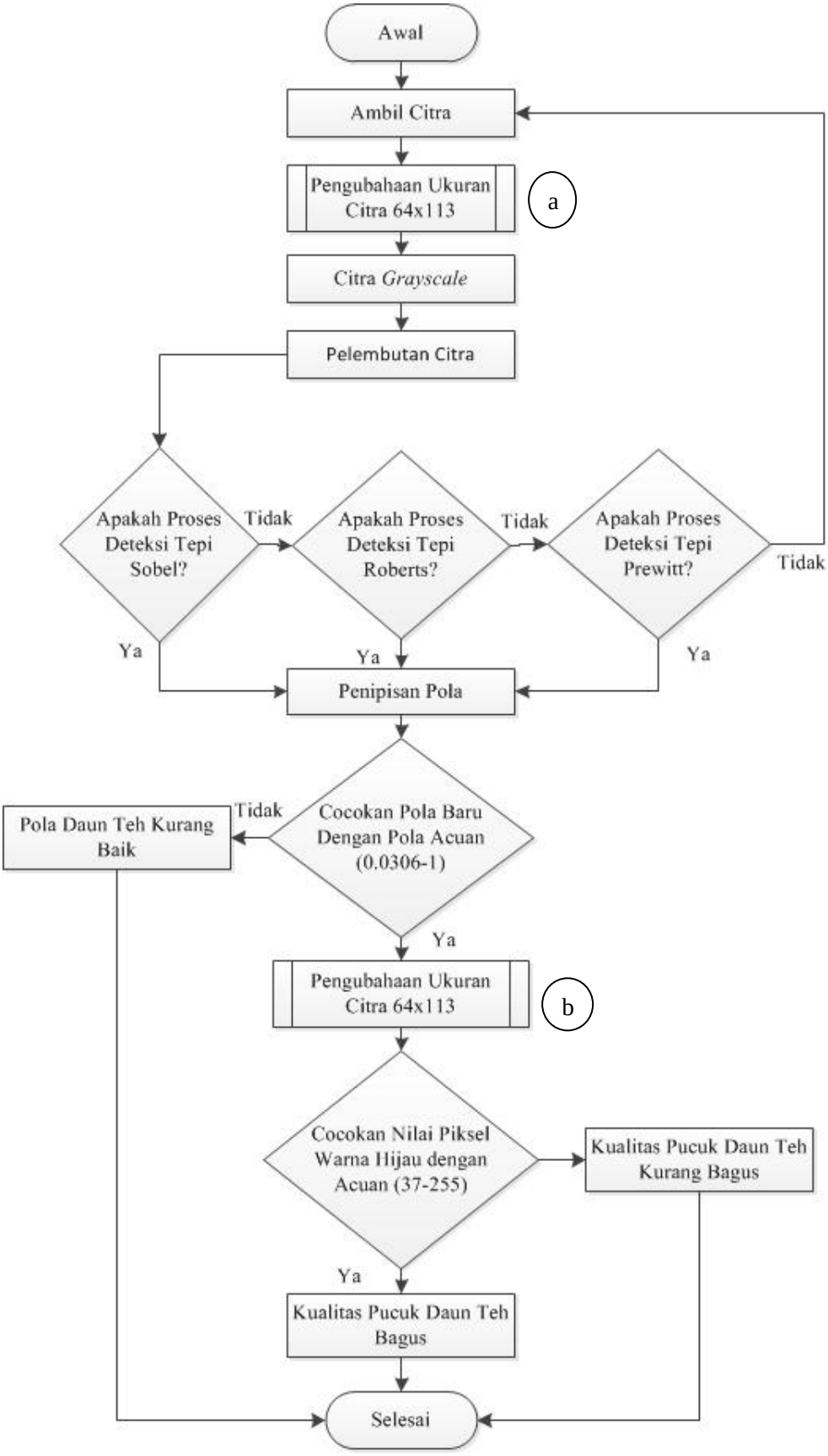


Gambar 3.2 Diagram Alir Proses Pelatihan Piksel Warna Hijau (a) dan Pelatihan Pengenalan Pola (b).

Tabel 3.2 (a).Penjelasan algoritma pada *flowchart* pelatihan nilai pixel warna hijau dan (b). Pelatihan Pengenalan Pola.

Nama	Keterangan
Awal	Awal program.
Ambil Citra	Memanggil Citra yang sudah di *capture* pada kamera untuk proses tahap selanjutnya.
Pengubahaan Ukuran Citra 64x113	Proses perubahaan ukuran citra untuk mempermudah pengoprasian pada pengecekan nilai piksel dan untuk menghemat memori.
Cek piksel Warna Hijau	Proses ini untuk menampilkan atau mengecek nilai piksel warna hijau, jika baik maka disimpan dan jika kurang baik maka proses ke awal program.
Simpan Nilai Piksel Warna Hijau	Proses penyimpanan piksel warna hijau yang telah di uji.
Selesai	Program selesai.

Nama	Keterangan
Awal	Awal program.
Ambil Citra	Memanggil Citra yang sudah di *capture* pada kamera untuk proses tahap selanjutnya.
Pengubahaan Ukuran Citra 64x113	Proses perubahaan ukuran citra untuk mempermudah pengoprasian pada pengecekan nilai piksel dan untuk menghemat memori.
Citra *Grayscale*	Tahapan ini untuk mendapatkan nilai batas ambang dan untuk proses tahapan citra biner.
Pelembutan Citra	Pelembutan citra berfungsi untuk menghilangkan adanya *nois* pada saat pengambilan gambar.
Apakah Deteksi Tepi *Sobel*?	Apakah proses pengujian menggunakan deteksi *sobel* baik atau tidak untuk disimpan sebagai pola acuan.
Apakah Deteksi Tepi *Roberts*?	Apakah proses pengujian menggunakan deteksi *roberts* baik atau tidak untuk disimpan sebagai pola acuan.
Apakah Deteksi Tepi *Prewitt*?	Apakah proses pengujian menggunakan deteksi *prewitt* baik atau tidak untuk disimpan sebagai pola acuan.
Penipisan Pola	Berfungsi untuk memperkecil pola.
Simpan Sebagai Pola Acuan	Tempat penyimpanan *file* pola acuan yang ada pada *file* Matlab.
Selesai	Program selesai.



Gambar 3.3 Diagram Alir Pengenalan Pola dan Nilai Pixel.

Tabel 3.3 Penjelasan algoritma pada *flowchart* Pengenalan Pola dan Nilai Pixel.

Nama	Keterangan
Awal	Awal program.
Ambil Citra	Memanggil Citra yang sudah di *capture* pada kamera untuk proses tahap selanjutnya.
Pengubahaan Ukuran Citra 64x113(a)	Proses perubahaan ukuran citra untuk mempermudah pengoprasian pada pengecekan nilai piksel dan untuk menghemat memori.
Citra *Grayscale*	Tahapan ini untuk mendapatkan nilai batas ambang dan untuk proses tahapan citra biner.
Pelembutan Citra	Pelembutan citra berfungsi untuk menghilangkan adanya nois pada saat pengambilan gambar.
Apakah Deteksi Tepi *Sobel*?	Apakah proses pengujian menggunakan deteksi *sobel* baik atau tidak untuk disimpan sebagai pola acuan.
Apakah Deteksi Tepi *Roberts*?	Apakah proses pengujian menggunakan deteksi *roberts* baik atau tidak untuk disimpan sebagai pola acuan.
Apakah Deteksi Tepi *Prewitt*?	Apakah proses pengujian menggunakan deteksi *prewitt* baik atau tidak untuk disimpan sebagai pola acuan.
Penipisan Pola	Berfungsi untuk memperkecil pola.
Cocokan Pola Baru dengan Pola Acuan (0.0306-1)	Proses ini adalah untuk mencocokan pola baru dengan pola acuan. Dengan pola acuan yaitu 0.0306 sampai 1.
Pola Daun Teh kurang Bagus	Jika pola yang dicocokan kurang bagus maka program selesai.
Pengubahaan Ukuran Citra 64x113(b)	Proses ini adalah jika proses pencocokan pola baik, maka prosedur pengubahan ukuran citra dipanggil untuk proses tahapan selanjutnya.
Cocokan Nilai Piksel Warna Hijau dengan Acuan(37-255)	Tahapan ini untuk mencocokan nilai piksel baru dengan piksel acuan (37 sampai 255).
Kualitas Pucuk Daun Teh kurang Bagus	Tampilan kualitas pucuk daun teh yang kurang bagus, karena tidak memenuhi nilai piksel acuan.
Kualitas Pucuk Daun Teh Bagus	Tampilan kualitas pucuk daun teh yang bagus, karena memenuhi nilai piksel acuan yang ada dalam database.
Selesai	Program selesai.

3.5 Pengambilan Gambar Pada *WebCam*

3.5.1 Deteksi Jenis *Video* Kamera

Sebelum memulai mengambil data atau gambar dari *video* kamera perlu dipastikan bahwa adaptor kamera kita sudah terbaca dengan baik oleh MatLab, untuk melakukan hal tersebut dapat digunakan perintah sebagai berikut.

```
>>imaqInfo = imaqhwinfo;
>>imaqInfo.InstalledAdaptors;
>>hwInfo = imaqhwinfo('winvideo');
>>hwInfo.DeviceInfo;
```

kemudian untuk melihat jenis spesifikasi informasi dari kamera, maka hal tersebut dapat digunakan perintah

```
>> device1 = hwInfo.DeviceInfo(2);
```

jika terdapat ada dua jenis kamera yang digunakan, maka pilih salah satu jenis kamera yang digunakan

contoh:

```
>> device1 = hwInfo.DeviceInfo(1);
```
atau
```
>> device1 = hwInfo.DeviceInfo(2);
```

kemudian untuk melihat jenis ukuran kamera maka dapat digunakan perintah

```
>> device1.DefaultFormat;
```

3.5.2 Mengambil Data Citra Pada *Video* Kamera (*WebCam*)

Proses selanjutnya dalam pemrosesan *video* yaitu mengambil data dengan menggunakan perintah *video input* dan menentukan beberapa parameter seperti resolusi, lebar, panjang dan band. Kemudian membuat sebuah *handle* yang menampung *image frame* tersebut. Setelah *handle* image didapatkan maka biasanya *video* dapat ditampilkan menggunakan perintah *preview*. Contoh sederhana dari menampilkan *video* pada MatLab seperti dibawah ini.

```
>>vid = videoinput('winvideo', 2, 'RGB24_640x480');
>>savevid;
>>vid.FramesPerTrigger = 1;
>>vid.ReturnedColorspace = 'rgb';
>>triggerconfig(vid, 'manual');
```

>>vidRes = get(vid, 'VideoResolution');

>>imWidth = vidRes(1);

>>imHeight = vidRes(2);

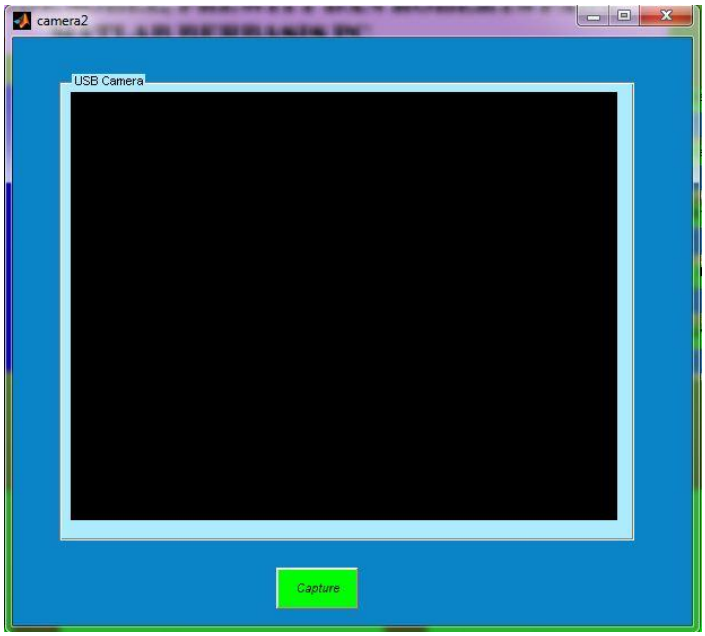
>>nBands = get(vid, 'NumberOfBands');

>>hImage = image(zeros(imHeight, imWidth, nBands), 'parent',

>>handles.axPreview);

>>savehImage;

>>preview(vid, hImage);



Gambar 3.4 Tampilan *Video* Kamera

3.6 Konversi Citra Aras Keabuan

Citra yang dibaca adalah citra warna (RGB) sehingga citra perlu dirubah kedalam citra aras keabuan. Pengubahan citra asli atau citra warna menjadi citra aras keabuan menggunakan perintah:

>>citra_keabuan = rgb2gray(citra);

Untuk merubah gambar RGB kegambar *grayscale* pada MatLab disediakan fungsi khusus yaitu "*rgb2gray (matrik_gambar)"*, tetapi kadangkala diinginkan untuk perubahan bentuk *grayscale* ini tidak menggunakan fungsi MatLab yang sudah ada yang merupakan nilai rata-rata piksel RGB tetapi masing-masing nilai RGB diberi nilai bobot yang berbeda-beda, hal ini dengan mudah

dilakukan dengan menggunakan pemisahan nilai seperti yang telah dilakukan diatas seperti contoh perintah program berikut:

>>gambar=imread('citra.png'); %--------membaca file gambar

>>red=gambar(:,:,1);

>>green=gambar(:,:,2);

>>blue=gambar(:,:,3);

*>>gray2=0.3*red+0.5*green+0.2*blue ;*



Gambar 3.5 Tampilan Menu Greyscale

3.7 Pelembutan Citra

Pelembutan citra meliputi pengaturan *intensitas* dan *penapisan* citra. *Intensitas* disini adalah sumber cahaya pada saat pengambilan gambar. *Penapisan* menggunakan metode *median* dan *wiener*. Perintah untuk proses pelembutan citra adalah:

>>median=medfilt2(adjust,[3 3]);

>>wiener=wiener2(median,[5 5]);

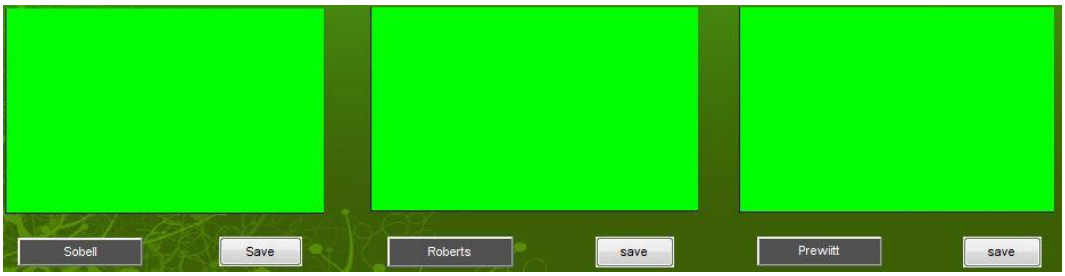
3.8 Deteksi Tepi

Seleksi objek biasanya dilakukan langkah-langkah deteksi tepi dalam proses pengolahan citra, pada MatLab proses pendeteksian tepi dilakukan dengan perintah atau fungsi "*edge"* dan hanya menggunakan gambar 2D saja. Deteksi Tepi (*Edge Detection*) suatu proses yang menghasilkan tepi-tepi dari objek citra, tujuannya menandai bagian yang menjadi detail citra dan memperbaiki citra yang

kabur, yang terjadi karena *error* atau adanya efek dari proses akuisisi citra. Ada beberapa metode yang dapat dilakukan pada deteksi tepi menggunakana MatLab yaitu metode *Sobel, Prewitt,* dan *Roberts*. Yang penting diperhatikan pada deteksi tepi bahwa hanya dapat dilakukan menggunakan citra *grayscale* atau citra 2-D.

Contoh penggunaan metode deteksi tepi:

```
>>I = imread('captured.png');
>>gray=rgb2gray(I);
>>BW1 = edge(gray,'prewitt');
>>BW2 = edge(gray,'canny');
>>BW3 = edge(gray,'sobel');
>>BW4 = edge(gray,'roberts');
>>figure, imshow(BW1);
>>figure, imshow(BW2);
>>figure, imshow(BW3);
>>figure, imshow(BW4);
```



Gambar 3.6 Tampilan Proses Deteksi Tepi

3.9 Penipisan Pola

Penipisan pola hanya bias dilakukan pada citra biner. Penipisan pola menggunakan perintah:

bwmorph.

```
>>thinning=bwmorph(edge,'skel',Inf);
```

3.10 Pengenalan Pola

Pengenalan pola terdiri dari dua tahap, yaitu pengenalan jenis bentuk dan jenis tepi daun. Pengenalan jenis bentuk merupakan pengenalan yang hanya mengenali bentuk gambar seperti bentuk segitiga, bulat dan lain-lain. Sedangkan

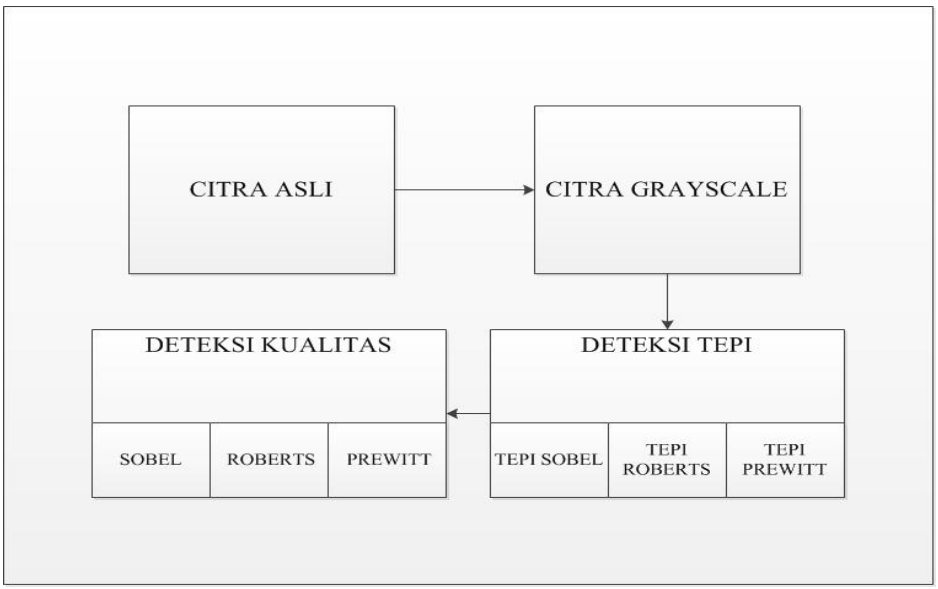
pengenalan pola adalah meliputi tepi atau pinggiran-pinggiran objek gambar bila titik tersebut mempunyai perbedaan yang tinggi dengan tetangganya. Untuk menghitung nilai koefisien korelasi antara citra masukan dengan citra acuan di dalam basis data. Pengenalan pola digunakan perintah *corr2*.

contoh:

```
>>re=imread('mask_sobel.jpg');
>>citra1=imread('bentuk_1.jpg');
>>y01=corr2(citra1,re);
```

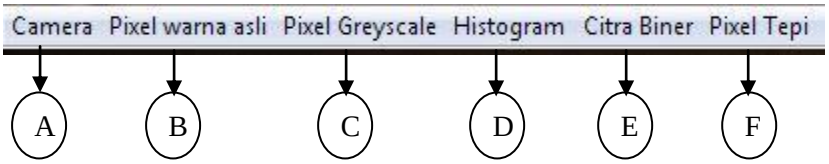
re adalah citra masukan, sedangkan citra 1 adalah citra acuan di dalam basis data.

Pada perangkat lunak MatLab dibuat sebuah aplikasi yang memiliki dua fungsi. Fungsi pertama adalah untuk pengambilan sampel data dan fungsi kedua adalah untuk pemrosesan perbandingan pucuk daun teh, yang dibandingkan citra masukan dengan citra sampel yang telah diuji kualitas pucuk daun tehnya.



Gambar 3.7 Rancangan Tampilan Aplikasi Penentuan Kualitas Pucuk Daun Teh

3.11 Fungsi – Fungsi Menu



Gambar 3.8 Fungsi Tombol Menu

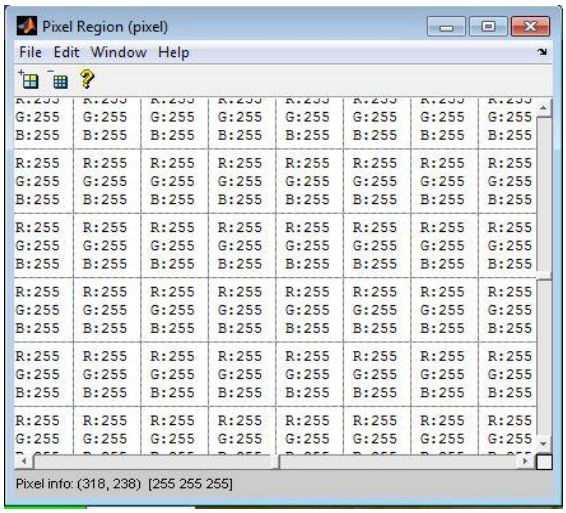
A. *Camera*

Fungsi tombol kamera untuk mengambil data atau *capture* sebuah gambar pada tampilan *video*.

B. Pixel Citra Asli

Untuk menampilkan sebuah informasi atau mengetahui piksel warna hijau pada pengujian sampel atau gambar asli berupa bit-bit pixel RGB yang dimulai nilai desimal dari 0-255 dengan tiga layer R=0-255, G=0-255, B=0-255..

Contoh gambar pixel RGB.

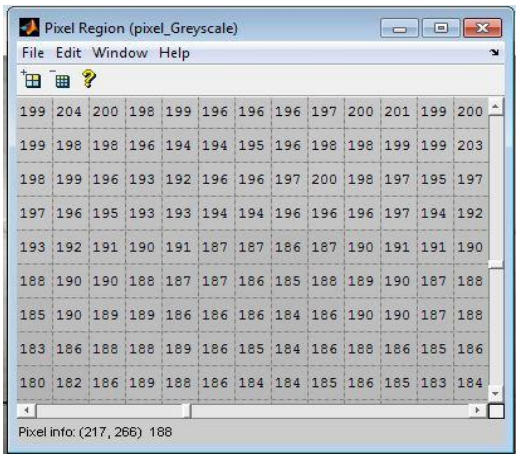


Gambar 3.9 Nilai Pixel Citra RGB

C. Pixel Citra *Grayscale*

Untuk menampilkan sebuah informasi gambar *grayscale* berupa bit-bit pixel yang dimulai nilai desimal dari 0-255 hanya satu *layer*.

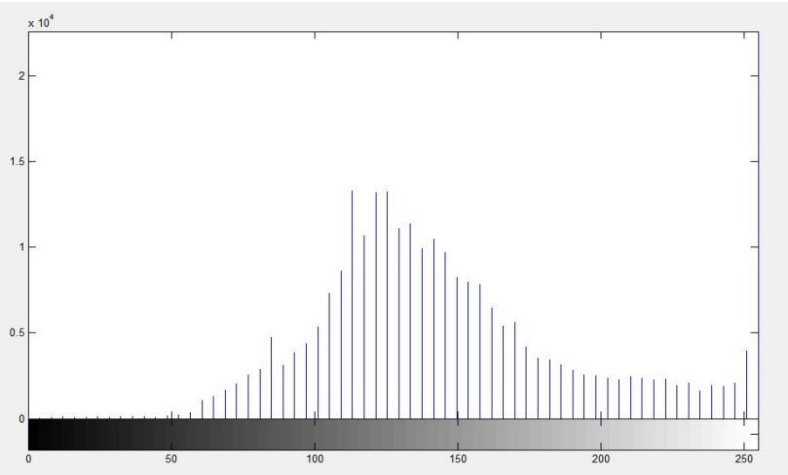
Contoh gambar pixel *grayscale*.



Gambar 3.10 Nilai Pixel Citra *Grayscale.*

D. Histogram

Fungsi dari tombol histogram sendiri adalah grafik yang menunjukan frekuensi kemunculan setiap nilai gradasi warna. Bila digambarkan pada koordinat kartesian maka sumbu *x* (*absis*) menunjukan tingkat warna dan sumbu *y* (*ordinat*) menunjukan frekuensi kemunculan.



Gambar 3.11 Tampilan Citra Bentuk Histogram

E. Citra Biner

Fungsi dari tombol citra biner adalah untuk menampilkan hasil pengambangan (*thresholding*) atau citra baru yang dirubah ke citra hitam dan putih (biner). Tombol dari citra biner menampilkan sebuah gambar hitam dan putih untuk menghemat kebutuhan memori, batas ambang yang digunakan adalah nilai piksel tertinggi *grayscale*, dan terendah dibagi 2.

Contoh perhitungan mencari nilai batas ambang T untuk citra biner.

Misalkan citra *grayscale* 4x4 piksel dengan kedalaman 8 bit seperti berikut.

200	230	150	75
240	50	170	90
210	100	120	80
100	90	200	230

$$T = \frac{fmax + fmin}{2} = \frac{240 + 50}{2} = 145$$

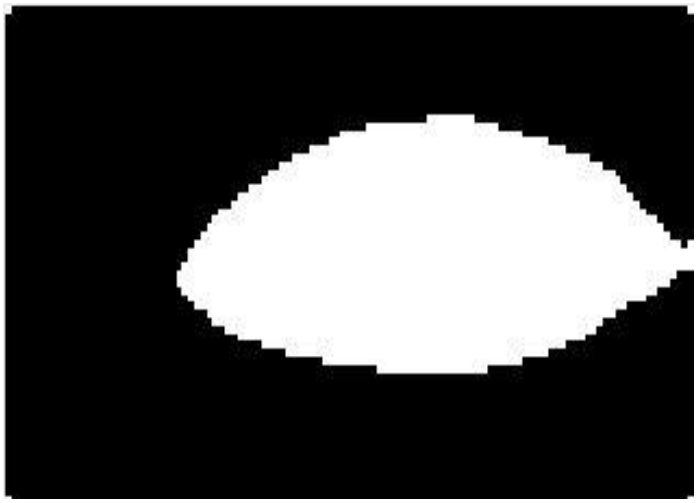
$$f(x,y) = 255, jika\ f(x,y) \geq T \text{.....(2.15)}$$

$$f(x,y) = 0\ jika\ f(x,y) < T \text{.....(2.16)}$$

maka untuk hasil citra biner dengan batas ambang 145 adalah:

255	255	255	0
255	0	255	0
255	0	0	0
0	0	255	255

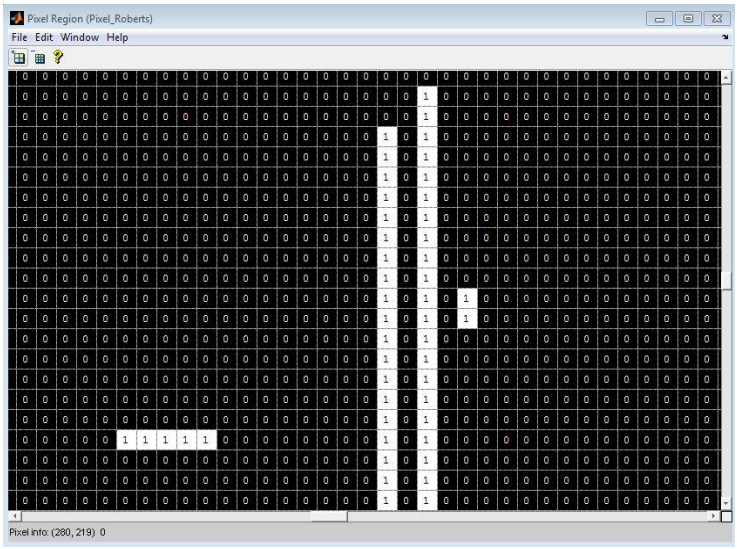
contoh gambar citra biner



Gambar 3.12 Citra Biner

F. Pixel Tepi

Pixel tepi terdapat beberapa pilihan yang merupakan informasi gambar yang sudah dirubah ke citra biner 0 dan 1. Hitam untuk nilai 0, dan nilai 1 untuk warna putih.



Gambar 3.13 Nilai Pixel Deteksi Tepi