BAB III
METODE PENELITIAN

3.1 Metode Pengumpulan Data

Penelitian ini melakukan metode pengumpulan data sehingga dapat melengkapi data yang ada. Adapun metode yang digunakan sebagai berikut :

3.1.1 Studi Literatur

Melakukan penelitian dengan cara mempelajari buku-buku dan literatur-literatur yang ada pada perpustakaan, akademi atau dari tempat lain yang berhubungan langsung maupun yang tidak langsung.

3.1.2 Observasi

Penulis mendapatkan data-data dan fakta dari pengamatan langsung di Kebun Karet Desa Sidomulyo Kecamatan Negeri Katon Pesawaran.

3.1.3 Wawancara

Penulis melakukan wawancara terhadap petani karet di desa sidomulyo kecamatan negeri katon pesawaran.

3.2 Pre-procesing Citra

Preprocessing dilakukan sebelum data tersebut diolah dengan Matlab. Pada penelitian ini metode preprocessing yang digunakan adalah *Cropping*, *Cropping* adalah penghapusan bagian sudut dari suatu gambar untuk memotong sebagian isi dari gambar guna memperoleh hasil yang diinginkan.

Perangkat keras dan perangkat lunak yang digunakan adalah Processor Intel Atom, harddisk 500GB dan RAM 1 GB. Adapun perangkat lunak yang digunakan dalam proses penentuan kualitas getah karet yaitu menggunakan Sistem Operasi Windows 7 dan MATLAB 8.5.0 (R2008b)serta paint.

3.3 Metode Pengembangan Perangkat Lunak

Metode yang digunakan dalam perancangan perangkat aplikasi dalam penelitian ini adalah *K-Means Clustering.* Metode ini memiliki 2 tahapan yaitu sebagai berikut :

3.3.1 Pegumpulan Kebutuhan

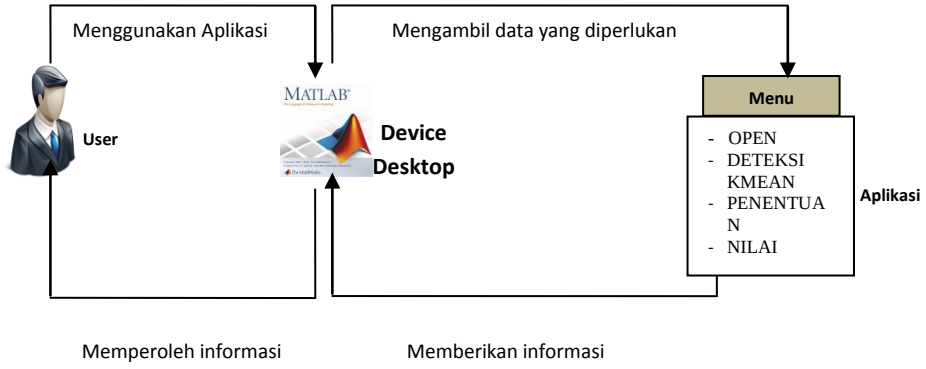
Tahap pengumpulan kebutuhan pada penelitian ini diantaranya melakukan pengumpulan citra jenis getah karet mentah yang diambil langsung di perkebunan karet di desa sidomulyo kecamatan negeri katon pesawaran.

3.3.2 Perancangan

Tahap perancangan pada aplikasi yang akan dibangun adalah rancangan *activity diagram*, *sequence diagram*, dan rancangan antarmuka.

3.3.2.1 Sistem Yang Diusulkan

Arsitektur system digunakan untuk mendefinisikan masing-masing komponen pendukung sistem yang lebih spesifik secara terstruktur. Untuk lebih jelasnya dapat dilihat pada gambar 3.1 berikut.



Gambar 3.1 Arsitektur Sistem

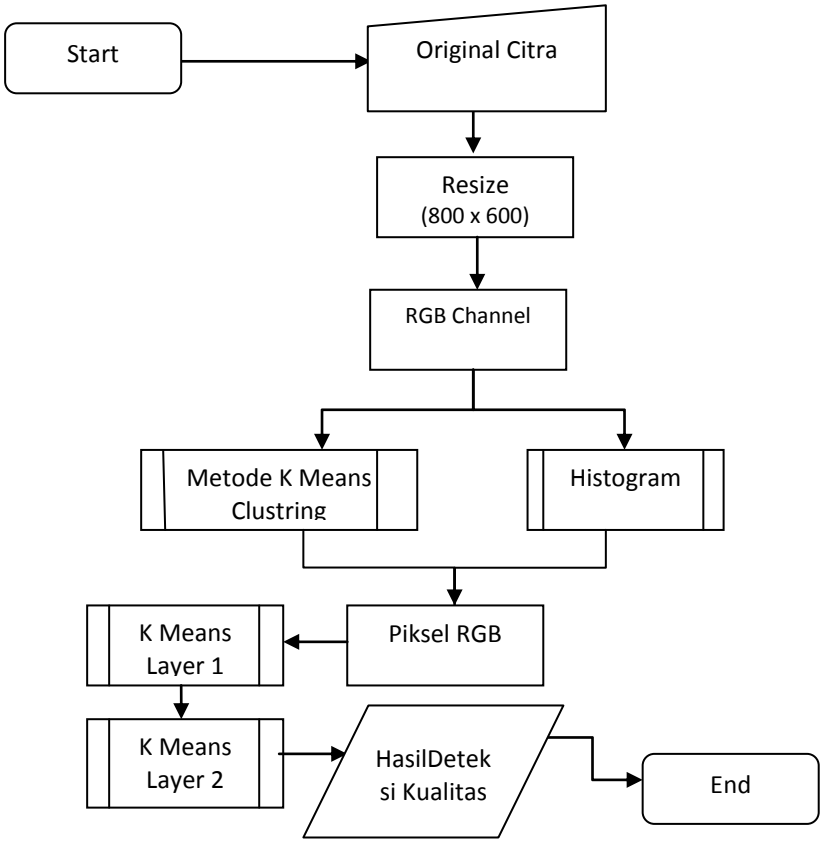
Berdasarkan arsitektur tersebut, *user* dapat menggunakan aplikasi untuk memperoleh informasi yang dibutuhkan dengan menggunakan *device Pc atau laptop.* Selanjutnya, *device* meminta data yang dibutuhkan *user* pada aplikasi yang tersedia.

Keterangan :

a. Citra Inputan adalah letak dari citra yang di masukan.
b. Citra Hasil adalah letak dari citra yang telah di proses dengan kmeans Clustering.
c. Menu open adalah menu untuk memanggil citra yang akan diproses.
d. Menu *Kmeans* adalah menu untuk memproses gambar dengan menggunakan metode *k means clustering.*
e. Menu Penentuan adalah menu untuk mendapatkan angka pixel dari gambar yang di hasilkan.
f. Menu Nilai adalah untuk melihat nilai yang di hasilkan program dengan nilai yang sudah di simpan sebelumnya.

3.3.2.2 Flowchart Aplikasi

Gambar berikut merupakan tahapan- tahapan citra penentuan kualitas getah karet yang dilakukan dalam perangkat lunak ini



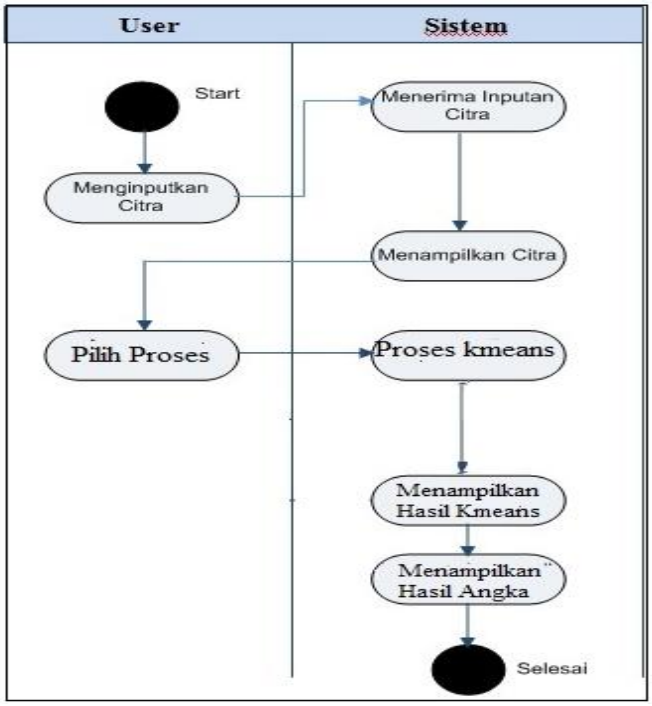
Gambar 3.2 Flowchart Aplikasi

Keterangan :

a. Citra Original adalah citra yang ingin di deteksi.
b. Resize 800*600 adalah proses meresize ukuran citra yang akan di deteksi.
c. RGB Chanel adalah gambar yang di masukan dalam bentuk berwarna.
d. Metode *Kmeans* dan histogram adalah menu untuk memproses gambar dengan menggunakan metode *k means clustering* dan *histogram.*
e. Pixel RGB menghitung pixel dari gambar.
f. K Means layer 1dan 2 adalah layer untuk memunculkan citra yang telah di olah dengan metode kmeans.
g. Deteksi Kualitas adalah menu untuk menentukan kualitas dari gambar getah tersebut. Output yang di hasilan berupa angka.

3.3.2.3 *Activity Diagram*

Gambar 3.3 di bawah ini merupakan *activity diagram* dari aplikasi penentuan kualitas getah karet.



Gambar 3.3 *Activity* diagram deteksi penentuan kualitas getah karet

1. **Menjalankan Program**

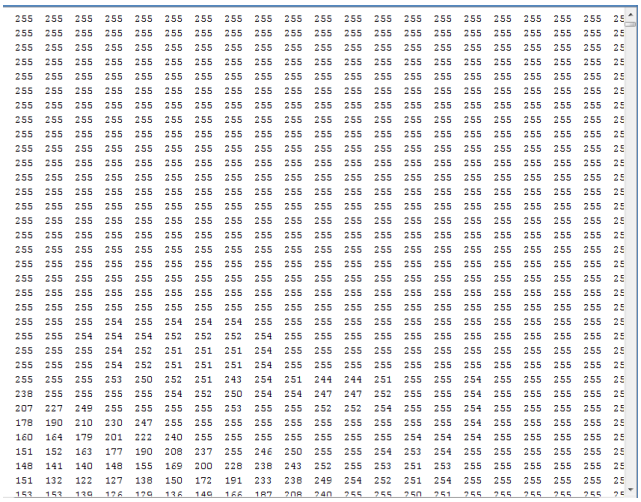
User menjalankan program penentuan kualitas getah karet guna untuk menentukan kualitas dari getah karet tersebut apakah kadar berkualitas bagus atau berkualitas buruk.

2. **Menginputkan Citra**

Citra input berupa citra buah yang didapatkan dari hasil pengambilan menggunakan kamera, citra ini berupa citra RGB dengan format JPEG

3. **Proses Deteksi Citra Menggunakan *K-Means Clustering***

Sebelum melakukan pendeteksian yang harus dilakukan terlebih dahulu melakukan proses mencari matrik citra dengan menggunakan perintah a = imread ('bagus.jpg').



255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	25
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	25
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	25
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	25
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	25
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	25
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	25
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	25
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	25
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	25
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	25
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	25
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	25
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	25
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	25
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	25
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	25
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	25
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	25
255	255	255	254	255	254	254	254	255	255	255	255	255	255	255	255	255	255	255	25
255	255	254	254	254	252	252	252	254	255	255	255	255	255	255	255	255	255	255	25
255	255	255	254	252	251	251	251	254	255	255	255	255	255	255	255	255	255	255	25
255	255	255	254	252	251	251	251	254	255	255	255	255	255	255	255	255	255	255	25
255	255	255	253	250	252	251	243	254	251	244	244	251	255	255	254	255	255	255	25
238	255	255	255	255	254	252	250	254	254	247	247	252	255	255	254	255	255	255	25
207	227	249	255	255	255	255	253	255	255	252	252	254	255	255	254	255	255	255	25
178	190	210	230	247	255	255	255	255	255	255	255	255	255	255	254	255	255	255	25
160	164	179	201	222	240	255	255	255	255	255	255	255	254	254	254	255	255	255	25
151	152	163	177	190	208	237	255	246	250	255	255	254	253	254	255	255	255	255	25
148	141	140	148	155	169	200	228	238	243	252	255	253	251	253	255	255	255	255	25
151	132	122	127	138	150	172	191	233	238	249	254	252	251	254	255	255	255	255	25
153	153	139	126	129	136	149	166	187	208	240	255	255	250	251	255	255	255	255	25

Gambar 3.4 Proses mencari matrik sebuah citra

Setelah perhitungan matrik sebuah citra secara otomatis sudah diketahui, maka selanjutnya merubah citra mendeteksi citra menggunakan metode *K-Means Clustering*. Proses dapat di lihat pada gambar berikut.

Gambar 3.5 Citra Input awal metode K-Means Clustering

4. Proses Deteksi Citra RGB Ke Cfrom

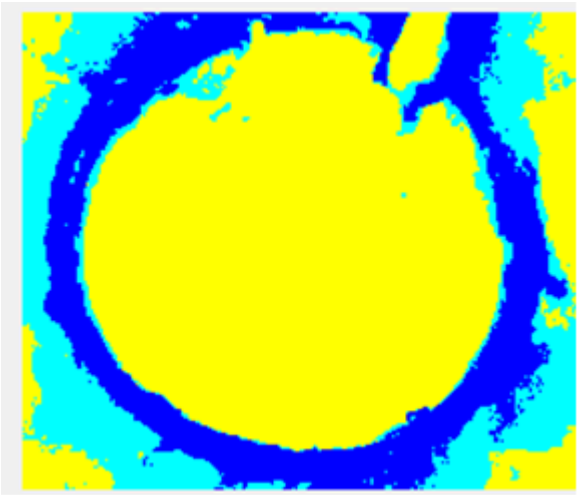
Pada proses ini citra awal RGB di ubah dalam bentuk Citra Cform yang dimana warna pada gambar di ubah guna untuk mencari warna getah yang berkualitas bagus. Koding yang digunakan adalah cform = makecform('srgb2lab');

Gambar 3.6 Proses Citra RGB Diubah Menjadi Citra Cform

5. Proses Deteksi Citra RGB Ke *Image Labeled by Cluster Index*

Pada proses ini citra awal RGB di ubah dalam bentuk Citra *Image Labeled by Cluster Index* yang dimana warna pada gambar di ubah menjadi warna biru dan kuning dengan koding [cluster_idx,cluster_center]=kmeans(ab,nColors,'distance','sqEuclidean'.'Replicates',3);.

:



Gambar 3.7 Proses Citra metode *K-Means Clustering*

Kemudian melakukan pendeteksian dengan melihat gambar –gambar dan pola-pola dari gambar tersebut.

Setelah di deteksi mencari ukuran citra dengan perintah

```
e=conv2(l,m);
s=imfill(e,'holes');
n=imclearborder(s,18);
seD = strel ('diamond',1);
d = imerode (n, seD);
res=imresize(d,6);
pix=bwarea(res)%/10^4
hasil=pix;
```

kemudian ukuran sudah diketahui maka hasil citra tersebut dapat diketahui apa nama citra tersebut, maka mencari sebuah besarnya citra dengan menggunakan rumus sebagai berikut :

$$[pix = bwarea(res)]$$

Keterangan :

pix = hasil pixel area

size = menghitung jumlah pixel

(res) = variabel

Adapun proses pendeteksian kualitas getah karet yang akan menggunakan perintah yang digunakan untuk mendeteksi kualitas getah karet dengan membandingkan gambar getah karet kualitas baik dengan gambar yang di inputkan yaitu:

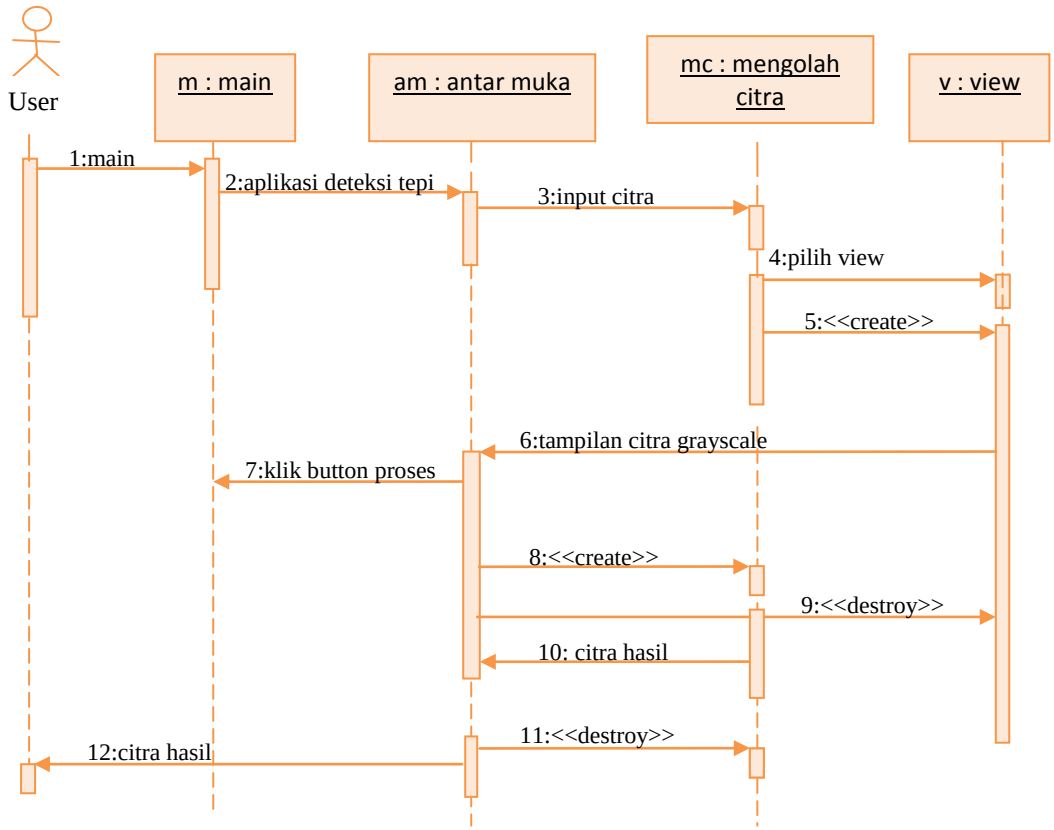
```
pix2=bwarea(B1)%/10^4
errorkolerasi=(pix2*0.05)%/10^4
pix1=(pix2-errorkolerasi);
```

6. Citra Hasil

Setelah dilakukan pendeteksian penentuan kualitas getah karet menggunakan metode *K-Means Clustering*, maka akan dihasilkan berapa besar nilai citra yang sudah di deteksi.

3.3.2.4 *Sequence Diagram*

Gambar 3.9 dibawah merupakan *sequence diagram* sistem penentuan kualitas getah karet



Gambar 3.9 Sequence diagram sistem penentuan kualitas getah karet

Keterangan:

1.User membuka aplikasi penentuan kualitas getah karet.

2.User menginputkan citra getah karet yang akan di deteksi.

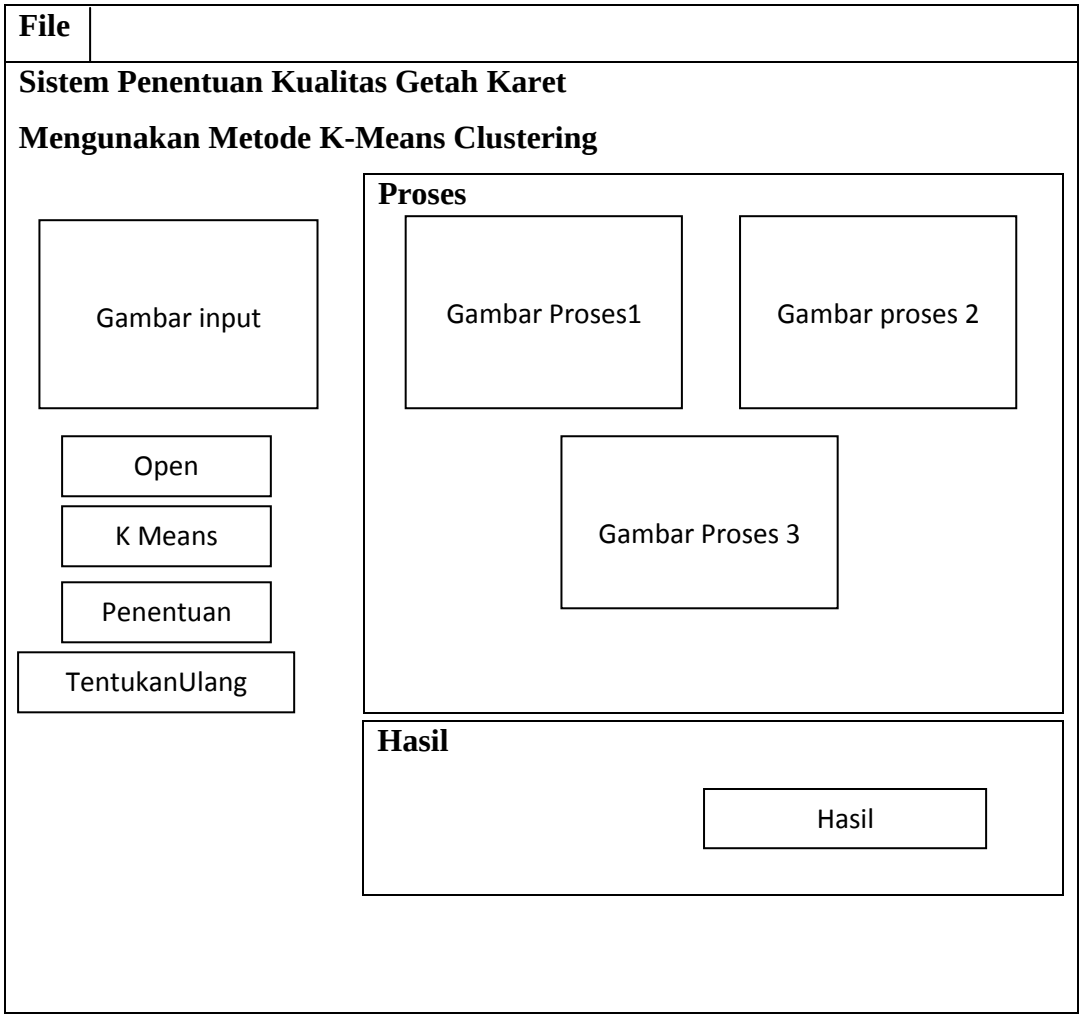
3.Aplikasi akan memberikan gambar yang telah di ubah dengan metode *k-mean Clustering* yang berubah *grayscale* (hitam putih).

4.User mengklik buton proses untuk menjalan program aplikasi penentuan kualitas getah karet.

5.Aplikasi akan memberikan hasil berupa citra getah karet yang telah di deteksi.

3.3.2.5 Rancangan Tampilan Antar Muka

Gambar berikut ini merupakan rancangan tampilan antar muka deteksi *K-Means Clustering* untuk menentukan kualitas getah karet.



Gambar 3.10 Rancangan antar muka Aplikasi penentuan kualitas getah karet

3.4 Pengujian Aplikasi Penentuan Getah Karet

Metode ujicoba *blackbox* memfokuskan pada keperluan fungsional dari *software*. Karna itu ujicoba *blackbox* memungkinkan pengembang *software* untuk membuat

himpunan kondisi input yang akan melatih seluruh syarat-syarat fungsional suatu program.

Uji coba *blackbox* berusaha untuk menemukan kesalahan dalam beberapa kategori, diantaranya :

1. Fungsi yang tidak benar atau tidak ada.
2. Kesalahan antarmuka (*interface errors*).
3. Kesalahan performansi (*performance errors*).
4. Kesalahan inisialisasi dan terminasi.