

BAB III METODOLOGI PENELITIAN

3.1 Metode Pengembangan Perangkat Lunak

Metode pengembangan perangkat lunak yang digunakan dalam penelitian ini adalah *prototype*. Metode ini sangat baik digunakan untuk menyelesaikan masalah kesalah pahaman antara *user* dan analis yang timbul akibat *user* tidak mampu mendefinisikan secara jelas kebutuhannya. *Prototyping* adalah pengembangan yang cepat dan pengujian terhadap model kerja (*prototype*) dari aplikasi baru melalui proses interaksi dan berulang-ulang yang biasa digunakan ahli sistem informasi dan ahli bisnis.

3.1.1 Pengumpulan Kebutuhan

Metode pengumpulan kebutuhan yang dipakai dalam penelitian ini untuk memperoleh data-data penelitian yaitu :

A) Studi Literatur

Mencari literatur yang berkaitan dengan penelitian, baik buku, jurnal, penelitian sebelumnya dan informasi yang terdapat di *internet* yang bisa menunjang penelitian.

B) Penelitian Lapangan.

Penelitian ini dilakukan secara langsung pada objek yang bersangkutan yaitu di PT Sekawan Chandra Abadi Bandar Jaya

C) Metode Wawancara.

Teknik wawancara merupakan teknik pengumpulan data/fakta yang dilakukan dengan cara menanyakan langsung kepada bagian yang terkait sesuai yang dibutuhkan dalam kegiatan penelitian.

Kebutuhan lainnya dalam membangun aplikasi penentuan kelayakan kaleng minuman bersoda adalah sebagai berikut:

1. Kebutuhan Perangkat Lunak

Pada tahap ini akan dijelaskan mengenai spesifikasi perangkat lunak yang digunakan dalam perancangan Membangun Aplikasi, yaitu:

- A) *Sistem Operasi Windows 7*
- B) *Matlab 2008a*

2. Perangkat Keras

Pada tahap ini merupakan tahapan yang menjelaskan mengenai spesifikasi perangkat keras yang digunakan dalam rancang bangun Aplikasi dan detail dari spesifikasi perangkat keras tersebut adalah sebagai berikut :

- a) *Laptop/PC Acer Aspire 4752*
- b) *Intel(R) Core(TM) i3-2330M CPU @ 2.30GHz, 3MB L3 cache*
- c) *Intel(R) HD Graphics 3000*
- d) *2 GB DDR3 Memory*
- e) *500 GB HDD*
- f) *Mouse*
- g) *Printer canon 2770*
- h) *Layar monitor led 14 inch*

3.1.2. Perancangan (*Design*)

Perancangan sistem berfungsi mengimplementasikan kebutuhan-kebutuhan sistem yang diusulkan berdasarkan hasil analisis sistem yang berjalan. Gambaran umum sistem penentuan kelayakan kaleng minuman bersoda diusulkan dijelaskan melalui tahapan rancangan sebagai berikut :

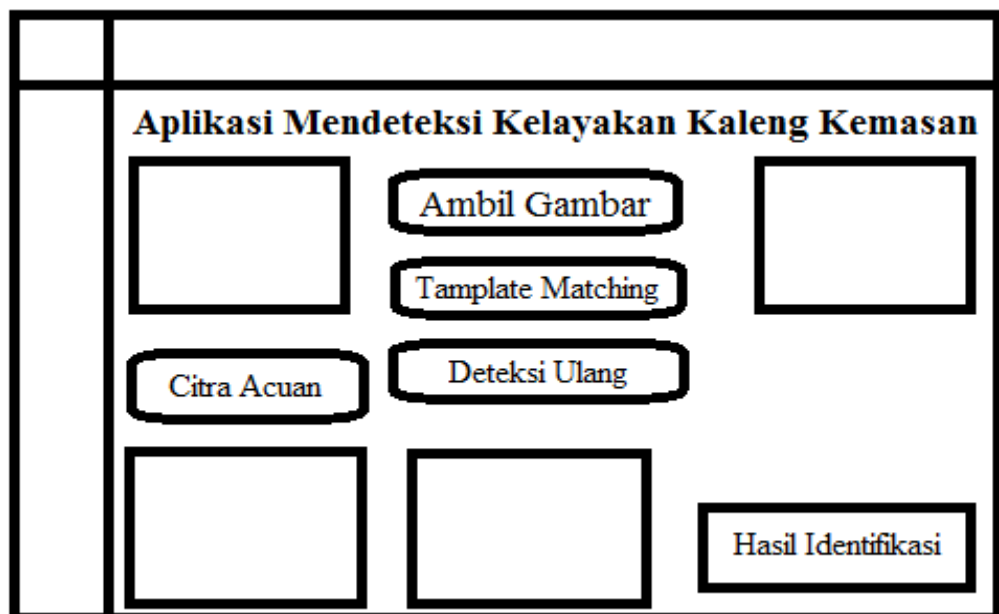
- a) *Input*
- b) *Output*
- c) *Algoritma UML (Unifed Modelling language)*

3.1.3 Perancangan Sistem *GUIDE* Matlab

GUIDE atau *GUI* builder merupakan sebuah *graphical user interface (GUI)* yang dibangun dengan obyek grafik seperti tombol (*button*), kotak teks,

slider, menu dan lain-lain. Aplikasi yang menggunakan *GUI* umumnya lebih mudah dipelajari dan digunakan menggunakan *GUI* umumnya lebih mudah dipelajari dan digunakan karena orang yang menjalankannya tidak perlu mengetahui perintah yang ada dan bagaimana kerjanya.

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



Gambar 3.1 Rancangan Antar Muka

Penjelasan tentang kegunaan dari rancangan aplikasi diatas adalah :

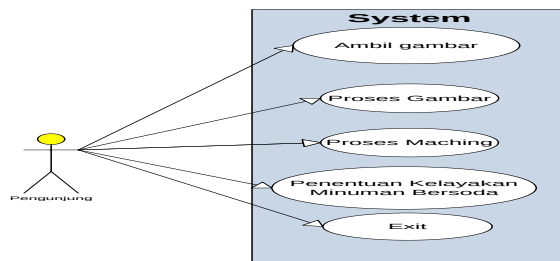
- a) Menu Ambil gambar adalah untuk pengambilan gambar kaleng kemasan secara langsung/*real time*
- b) Menu *Tampalte Matching* adalah metode untuk mendeteksi suatu citra untuk menentukan gambar citra tersebut berdasarkan tepinya.
- c) Menu hasil Kelayakan adalah perintah untuk menentukan kelayakan dari objek citra yang di inputan.
- d) Menu Deteksi ulang adalah untuk mengulang kembali pendeteksian kelayakan kaleng kemasan yang akan di deteksi kembali.

3.1.4 Algoritma UML (Unified Modelling Language)

Algoritma *uml* pada bab ini merupakan algoritma *uml* yang akan menjelaskan alur program *matlab* yang akan dibangun untuk sesuai dengan kebutuhan. Dalam rancang bangun sistem penentuan kelayakan minuman bersoda, penulis menjelaskan dengan beberapa bentuk algoritma *uml* antara lain adalah menggunakan *usecase* diagram, dan aktivitas diagram.

3.1.4.1 Usecase Diagram Pengguna

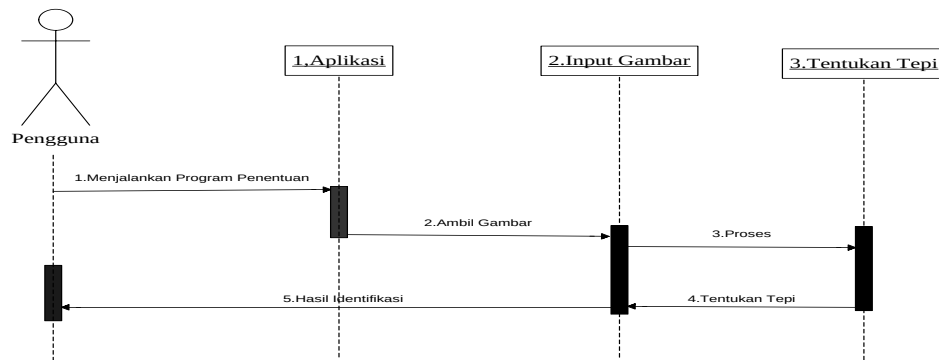
Usecase diagram merupakan *Usecase* diagram yang menjelaskan beberapa proses yang ada pada program *matlab* yang nantinya akan dibangun. *Usecase* ini menjelaskan beberapa langkah yang dapat dilakukan oleh *user*. Untuk melihat bagaimana dan apa saja proses yang dapat diakses oleh pengguna. dapat dilihat pada gambar 3.8 berikut ini:



Gambar 3.2 *Usecase* Diagram pengguna

3.1.4.2 Sequence Diagram

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa pesan yang digambarkan terhadap waktu. *Sequence diagram* terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait). *Sequence diagram* dapat dilihat pada gambar 3.3 dibawah ini:



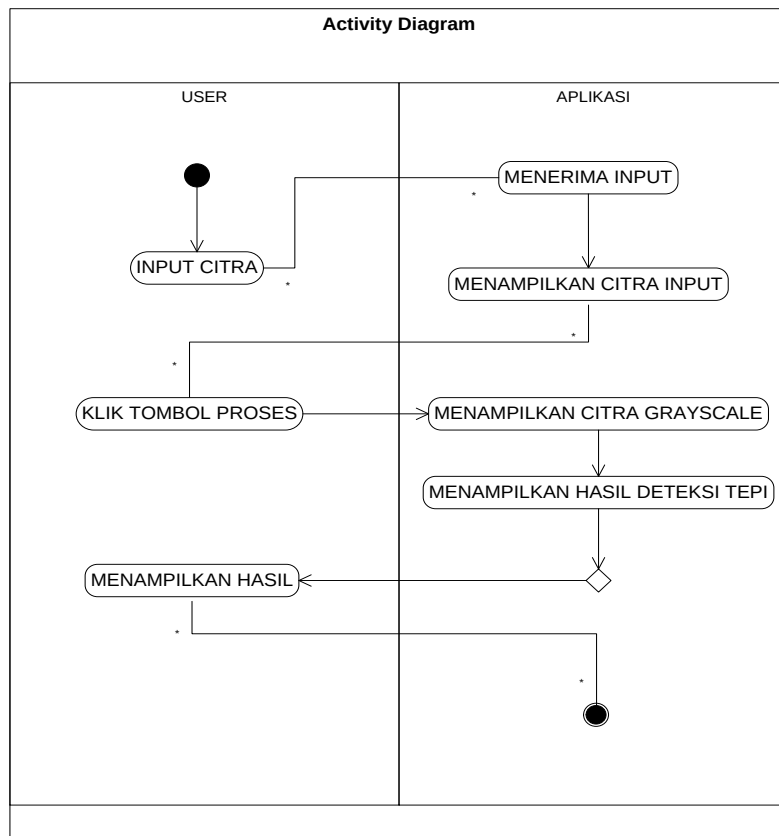
Gamabr 3.3 *Sequence diagram*

3.1.4.3 Activity Diagram

Activity diagrams menggambarkan berbagai alir aktifitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. *Activity diagram* merupakan *state diagram* khusus, di mana sebagian besar *state* adalah aksi dan sebagian besar transisi di-*trigger* oleh selesainya *state* sebelumnya (*internal processing*).

Sebuah aktifitas dapat direalisasikan oleh satu *use case* atau lebih. Aktifitas menggambarkan proses yang berjalan, sementara *use case* menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktifitas. Sama seperti *state*, standar *UML* menggunakan segiempat dengan sudut membulat untuk menggambarkan aktifitas. *Decision* digunakan

untuk menggambarkan *behaviour* pada kondisi tertentu. Untuk mengilustrasikan proses-proses paralel digunakan titik sinkronisasi yang dapat berupa titik, garis horizontal atau vertical. *Activity diagram* dapat dilihat pada gambar 3.4 berikut:



Gamabr 3.4 *Activity diagram*

3.1.5 Evaluasi

Penelitian ini dilakukan di PT. Sekawan Chandra Abadi Bandar Jaya Tepatnya adalah analisis ini menjelaskan tentang hasil penelitian.

3.1.6 Analisis Sistem Yang Berjalan

Dalam bab ini berisi analisis kebutuhan dari sistem dan masalah-masalah yang ada didalamnya. Hasil analisis ini dituangkan ke dalam suatu sistem pemodelan secara terstruktur. Dari tahap analisis kemudian dilanjutkan ketahap perancangan dan implementasi.

3.1.7 Kelemahan Sistem Yang Berjalan

Masalah yang sering muncul dalam pengembangan aplikasi ini yaitu banyak terjadi pada tahap pengambilan gambar/vidio atau capturing image. Masalah yang umum dan sering ada yaitu:

Resolusi citra input yang tidak sesuai, biasanya disebabkan pengambilan citra digital dalam menyeleksi kaleng kemasan yang terlalu jauh dari kameranya atau dikarenakan spesifikasi kamera yang kurang.

3.2 Proses kerja Aplikasi Kelayakan Kaleng Minuman

Proses kerja Aplikasi Penentuan Kelayakan kaleng minuman sebagai berikut:

1) Mengambil citra input

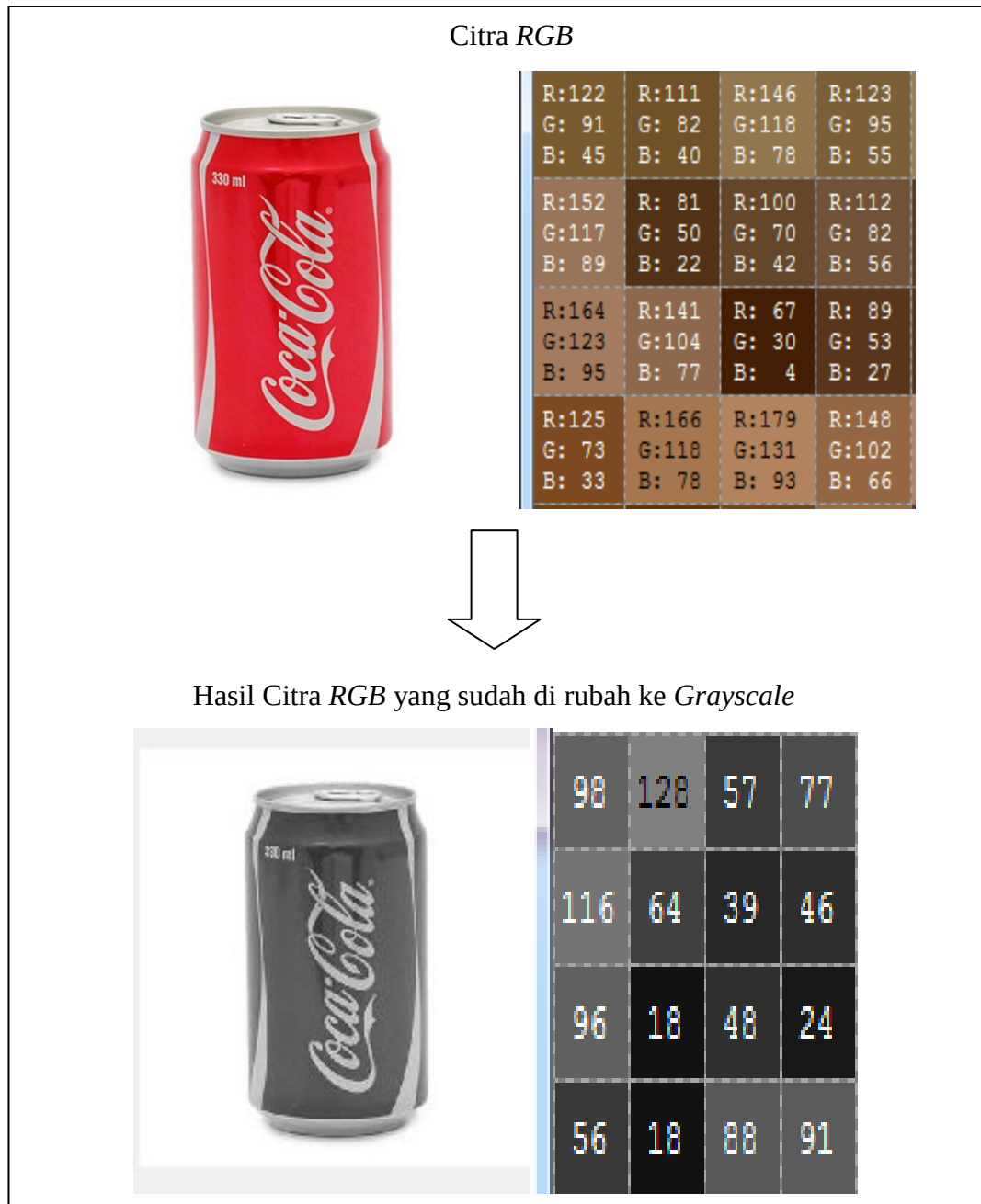
Citra input berupa citra kaleng kemasan yang diambil dengan menggunakan peangkat berupa kamera digital, Citra ini berupa citra *RGB*.

2) Mengubah citra warna menjadi *grayscale*

Proses ini termasuk dalam tahapan *preprocessing*. Citra *RGB* diubah warnanya menjadi abu-abu. Hal ini dimaksudkan untuk mempermudah perhitungan nilai piksel. Pada citra *RGB* setiap *pixel* mempunyai tiga nilai, masing-masing untuk nilai warna merah (*red*), hijau (*green*), dan biru (*blue*). Sedangkan pada citra keabuan, tiap *pixel* hanya memiliki satu nilai yang mewakili skala keabuannya.

dengan R menyatakan nilai komponen merah, G menyatakan nilai komponen hijau, dan B menyatakan komponen biru.

Berikut ini proses Citra *RGB* diubah menjadi citra *grayscale* Dapat dilihat pada gambar 3.5 di bawah ini:

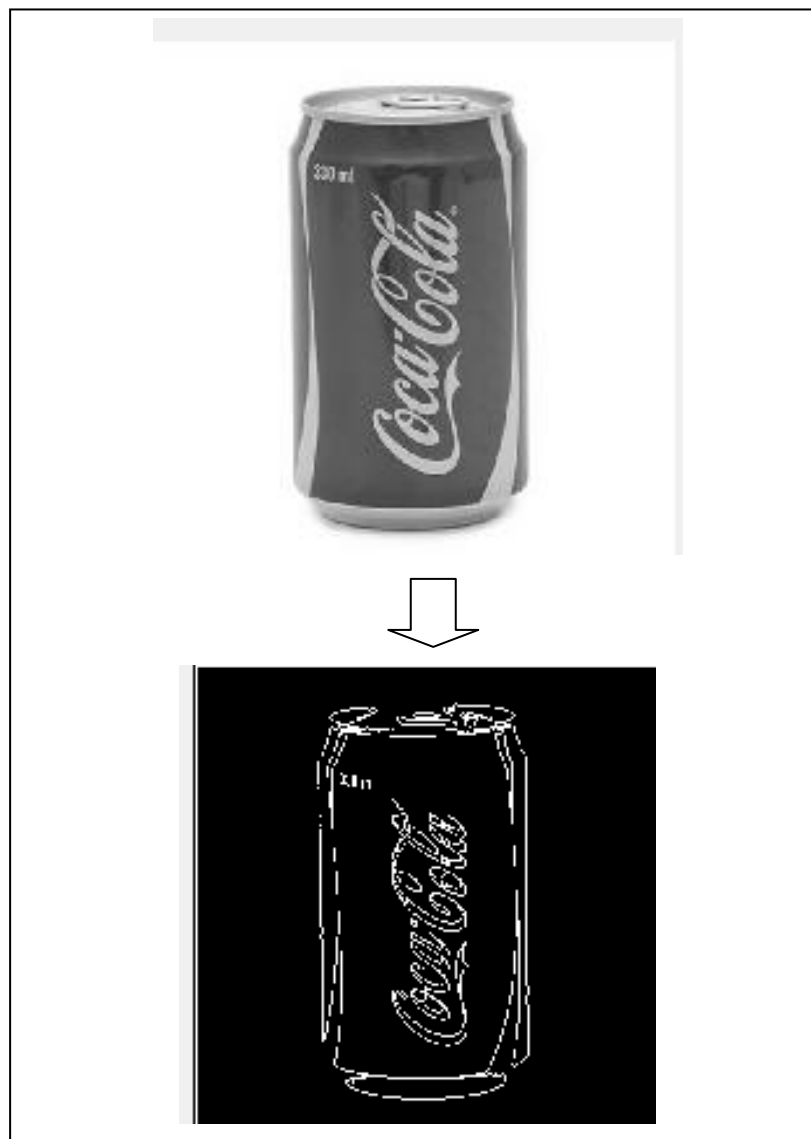


Gambar 3.5 Citra yang telah di *Grayscale*

3) Melakukan Deteksi *Matching*

Sebelum melakukan segmentasi citra, citra terlebih dahulu harus dideteksi tepinya. Tujuannya adalah untuk mendapatkan bagian citra yang akan disegmentasi.

Penjelasan tentang hasil dari deteksi konvolusi citra dapat dilihat pada gambar 3.6 berikut ini :



Gambar 3.6 Citra Hasil Konvolusi

4) Setelah objek di deteksi konvolusinya, objek dihitung jumlah *pixel* putihnya. Area A adalah jumlah *pixel-pixel* penyusun objek dan satuannya adalah *pixel* karena *pixel-pixel* inilah yang membentuk suatu luasan. Area dapat mencerminkan ukuran atau berat objek sesungguhnya pada beberapa benda pejal dengan bentuk yang hampir seragam. Perhitungan nilai A diperoleh dengan cara membaca *pixel-pixel* milik objek. Bila ditemukan *pixel* milik objek nilai A bertambah 1. Bila bukan *pixel* milik objek maka nilai A tidak berubah.

Penjelasan tentang perhitungan jumlah *pixel* pada citra adalah sebagai berikut:

$$A = \text{jumlah } \textit{pixel} \text{ dibaris ke-1} + \text{baris ke-2} + \dots + \text{baris ke-36}$$

$$A = 98+128+57+77+116+64+39+46+96+\dots+ 91$$

$$= 512625 \textit{ pixel}$$

Dari citra yang telah di proses, dapat diambil kesimpulan penentuan kelayakan kaleng kemasan tersebut berdasarkan citra deteksi konvelusi dan *template matching* yang telah di disesuaikan dengan objek acuannya. Hasil perhitungan dari tepi tersebut menghasilkan angka. Angka yang dihasilkan akan menentukan jenis tapis apakah yang telah diinputkan dalam bentuk gambar. Proses bisa terus dilakukan dengan cara menutup citra sebelumnya dan mengambil citra baru, namun jika pemeriksaan telah selesai *user* dapat menutup aplikasi.