

## PERANCANGAN DAN IMPLEMENTASI PENDETEKSI LETAK BARCODE DENGAN PENGOLAHAN CITRA DIGITAL BERBASIS ANDROID

*Design And Implementation Barcode Recognition With Image Processing Based On Android*

Anna Siwi Ramadhani , Dr. Ida Wahidah, ST, MT. , Desti Madya Saputri, ST, MT.

<sup>1,2,3</sup>Fakultas Teknik Elektro – Universitas Telkom

Jln. Telekomunikasi Dayeuhkolot Bandung 40257 Indonesia

### ABSTRAK

Seorang tunanetra memiliki kekurangan tidak dapat melihat sehingga mengalami kesulitan untuk melakukan transaksi belanja di supermarket. Pada penelitian [1] telah dibuat sebuah sistem untuk membantu tunanetra dalam berbelanja. Sistem tersebut adalah sebuah aplikasi pembaca barcode dengan keluaran suara berbasis android. Sistem ini sudah dapat berjalan dengan baik. Namun masih mengalami kekurangan yaitu tidak dapat mendeteksi letak barcode. Sehingga tunanetra masih kesulitan untuk mengambil gambar barcode yang akan di scan.

Pada penelitian tugas akhir kali ini, dibuat sebuah sistem yang dapat mendeteksi letak barcode. Sistem ini dibuat untuk menyempurnakan penelitian sebelumnya. Citra yang diambil mengalami proses greyscaling dan ekualisasi histogram untuk mempertegas perbedaan nilai 0 dan satu. Kemudian citra diambil nilai threshold dengan metode otsu. Nilai threshold digunakan untuk mengantisipasi noise yang terlalu banyak. Setelah mendapatkan citra yang baik, maka deteksi letak barcode terjadi dengan membuat angle sebagai ciri barcode. Citra selain barcode akan terdeteksi sebagai noise. Setelah itu citra dicrop dan hasilnya adalah barcode saja.

Hasil dari penelitian ini adalah untuk menyempurnakan pembuatan aplikasi sebelumnya. Sistem ini dapat digunakan secara optimal pengambilan gambar pada jarak 7 cm dengan tingkat akurasi 83.3%

**Kata kunci :** *Android, Ekualisasi Histogram, Thresholding Otsu.*

### ABSTRACT

A blind people cannot see and get some difficulties to shop in supermarket. In the previous research [1] it has been made an application which help blind people to know price of some products. The application is a system to read barcode with voice interface based on android system. This system can be well applied. But it can't detect barcode location. So the blind people still get some difficulties to take barcode image.

In this final project was made a system which can detect barcode location. This system will fulfill the previous research. The image get greyscaling and histogram equalization for making sure 0 dan 1 value. Then, system took threshold value from the picture by using otsu method. Threshold value is used to prevent image from noise. After getting the best image, barcode locator system detect location by making angle as a barcode characteristic. All image which not detect as a barcode is noise. Then image is crooped and the result is just a barcode image.

The result from this final project fulfill the previous research. The system can be used 7 cm long for taking the image by 83.3% accuracy system.

**Keywords:** *Android, Histogram Equalization, Otsu Thresholding.*

## 1. PENDAHULUAN

### 1.1 Latar Belakang

Penyandang tuna netra memiliki kesulitan untuk melihat sehingga sulit untuk melakukan beberapa hal seperti menentukan arah, memilih barang, dan membaca. Namun, penyandang tuna netra juga ingin hidup mandiri. Untuk mewujudkan hal tersebut penyandang tuna netra biasanya mengikuti sekolah SLB-A, macam-macam kursus keterampilan untuk tuna netra, dan pelatihan kemandirian untuk tuna netra. Selain kegiatan softskill, sarana dan prasarana di tempat umum juga harus menunjang kemandirian tuna netra.

Hal yang masih sulit dilakukan oleh tuna netra adalah saat melakukan transaksi jual beli di minimarket maupun supermarket. Oleh karena itu, untuk membantu penyandang tuna netra mandiri dalam berbelanja dengan memanfaatkan pendengaran mereka. Maka penelitian sebelumnya telah dibuat sebuah alat yang dapat membaca barcode dengan keluaran suara. Aplikasi ini diinstal pada smartphone berbasis android. Pada penelitian tersebut masih ditemukan kekurangan. Salah satunya adalah tunanetra masih kesulitan untuk mencari dan menentukan letak barcode pada produk yang akan di scan. Sehingga alat ini tidak dapat digunakan dengan optimal.

Melalui permasalahan di atas, maka akan dilakukan penelitian lanjutan untuk menyempurnakan aplikasi yang telah ada. Pada penelitian kali ini akan dilakukan proses identifikasi untuk mendeteksi letak barcode pada produk yang akan di scan. Sehingga tunanetra tidak mengalami kesulitan dalam mencari letak barcode.

## 2. Dasar Teori

### 2.1 Barcode

Barcode adalah bentuk kode mesin yang dapat dibaca. Bentuknya berupa bar dan spasi berwarna hitam dan putih dalam jarak tertentu yang merepresentasikan karakter alfanumerik. Barcode digambarkan dalam bentuk baris hitam dan tipis yang disusun berderet sejajar secara horizontal. Pada produk-produk dipasaran barcode dilengkapi dengan kode angka yang dimaksud untuk memudahkan pembacaan. Namun kode angka tersebut tidak berpengaruh pada pola kode baris yang tercantum.

### 2.3 Pengolahan Citra Digital

Citra merupakan bentuk visual yang diterima oleh indera penglihatan. Pada ilmu komputer citra disebut dengan image yang merepresentasikan bentuk visual dari sebuah objek. Bentuk visual ini telah mengalami transformasi data numerik sehingga dapat teridentifikasi sebagai sebuah objek. Untuk mengidentifikasi sebuah image adalah objek tertentu maka harus mengalami serangkaian proses. Proses tersebut adalah akuisisi data, manipulasi data, visualisasi data, dan penyimpanan data.

Sebuah citra digital dapat dinyatakan sebagai matriks dengan indeks berupa baris dan kolom yang menyatakan koordinat titik dari objek tersebut. Citra digital juga memiliki elemen yang menyatakan intensitas cahaya pada titik tertentu. Titik tersebut biasa disebut dengan image element atau picture element atau lebih biasa disebut pixel.

Pengolahan citra digital digunakan untuk mengolah citra sehingga menghasilkan spesifikasi gambar yang diinginkan. Operasi ini biasa digunakan untuk mendapatkan kualitas gambar yang lebih baik. Atau operasi yang membuat gambar menjadi pixel yang lebih tajam.

#### 2.3.1 Citra RGB

Sebuah warna merupakan kombinasi dari warna dasar yaitu merah, hijau, dan biru (Red, Green, Blue). Citra ini juga disebut sebagai truecolor yang direpresentasikan dalam nilai heksadesimal. Nilai paling kecilnya adalah 0 dan paling besar adalah 256 yang terdiri dari 8 digit bilangan biner. Sehingga total warna yang dapat dihasilkan dari kombinasi RGB adalah 26 juta warna.

Bagian terpenting dalam mengolah data pada citra adalah informasi warna di setiap pixel. Bagian ini memiliki informasi yang terdiri dari 3 kombinasi warna dasar yaitu RGB. Untuk menentukan suatu warna yang bukan warna dasar digunakan penggabungan skala kecerahan satu setiap warna dasar yang ada.

#### 2.3.2 Citra Grayscale

Citra digital grayscale adalah citra yang setiap pixel di dalamnya berupa sampel tunggal. Hasil citranya terdiri dari warna kelabu dengan rentang intensitas hitam paling lemah hingga putih paling kuat. Grayscale disebut juga citra black-and-white dan monokromatik.

Citra grayscale memiliki struktur warna yang lebih sederhana dari RGB. Sehingga dalam proses komputasinya lebih cepat dan lebih efisien. Intensitas keabuan yang disimpan sebagai data citra adalah 8 bit per sampel piksel atau 256 intensitas warna abu dari nilai 0 (hitam) hingga 255 (putih).

### 2.4 Histogram Equalization

Histogram Equalization adalah suatu proses untuk meratakan derajat keabuan dari nilai yang paling rendah (0) sampai yang paling tinggi (255). Citra pada umumnya memiliki nilai histogram yang tidak merata. Hal ini karena citra biasanya memiliki kecenderungan warna tertentu. Untuk citra yang memiliki warna cerah keputihan akan cenderung berkumpul pada posisi mendekati nilai 255. Sedangkan citra yang memiliki warna agak gelap keabuan akan cenderung berkumpul pada nilai mendekati 0.

Proses histogram equalization akan membuat nilai tersebut tersebar secara merata dari 0 hingga 255. Sehingga terdapat perbedaan nilai hitam dan putih yang tegas dari citra yang akan diekstraksi. Tujuan utama histogram equalization adalah untuk memperoleh penyebaran histogram yang merata sehingga derajat keabuan memiliki jumlah piksel yang relative sama.

#### 2.4 Otsu Thresholding

Pengambilan citra dengan hasil ideal adalah pada kondisi pencahayaan yang baik dan kondisi kamera yang konstan. Namun pada realitanya kondisi ideal seperti hal itu sulit untuk di dapatkan. Variasi kondisi pencahayaan dan kondisi kamera membuat citra yang dihasilkan sangat rentan terhadap noise. Hal ini mengakibatkan level segmentasi yang bervariasi. Sehingga diperlukan metode untuk mengatur threshold secara otomatis.

Metode otsu merupakan cara untuk menentukan nilai threshold dari penyebaran latarbelakang (background) dan latar depan (foreground). Sehingga parameter yang masuk itu sesuai dengan karakteristik inputan yang ada. Dasar dari metode otsu adalah perbedaan intensitas dari setiap piksel yang dipisahkan dalam kelas tertentu (background dan foreground).

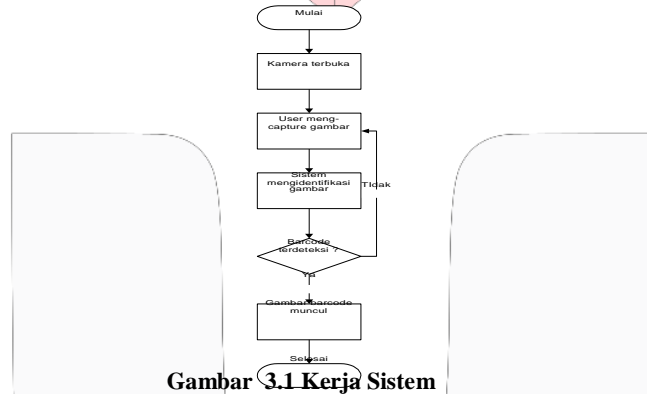
**2.5 Android**

Android merupakan sistem operasi berbasis linux yang ditanam di handphone, sekarang lebih dikenal dengan sebutan smartphome. Karena android menggunakan OS linux yang notabene adalah open source, hal ini memungkinkan pengguna android mengembangkan dan menciptakan aplikasi mereka sendiri. Perkembangan android sendiri sangat cepat karena adroid menggunakan sistem operasi open source yang dapat dipakai dan di distribusikan oleh perorangan maupun vendor manapun. Perkembangan android tidak hanya di operasi smartphome namun juga sistem operasi tablet PC.

**3. PERANCANGAN DAN IMPLEMENTASI SISTEM**

**3.1 Desain Model Sistem**

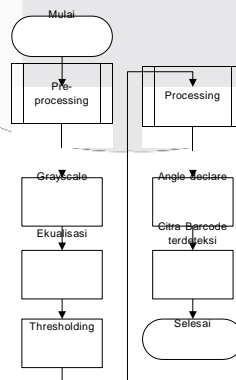
Aplikasi ini berkerja untuk mendeteksi letak barcode yang ada pada sebuah produk. Ketika membuka aplikasi ini tuna netra langsung masuk ke sistem kamera untuk mengambil gambar produk yang akan discan barcodenya. Tuna netra bisa mengambil gambar pada seluruh bagian produk tanpa harus mengetahui letak barcode secara tepat. Setelah gambar masuk kedalam sistem, maka sistem akan mengidentifikasi letak barcode. Jika barcode teridentifikasi maka sistem akan melanjutkan proses ke aplikasi selanjutnya yaitu aplikasi pembaca barcode dengan keluaran suara. Namun jika sistem tidak dapat mengidentifikasi letak barcode maka harus dilakukan pengambilan gambar ulang.



**Gambar 3.1 Kerja Sistem**

**3.2 Diagram Blok Sistem**

Pada diagram blok sistem ini akan menjelaskan tentang tahapan dalam pembuatan tugas akhir ini yaitu:



**Gambar 3.2 Diagram Blok Sistem**

Pada gambar diatas sistem dimulai dengan proses *pre-processing*. Proses ini bertujuan untuk membuat gambar yang berkualitas baik sehingga saat dilakukan ekstrasi ciri dapat menghasilkan citra yang bagus.

### 3.2.1 Histogram Equalization

Proses dimulai dari merubah tiap pixel citra menjadi *grayscale*. Hal ini bertujuan agar nanti citra lebih mudah dicirikan karena citra yang masuk tidak terlalu kompleks. Citra bersifat *grayscale* ini kemudian diekualisasi nilai histogramnya. Ekualisasi disesuaikan dengan penyebaran nilai piksel pada citra yang masuk. Sehingga nilai yang dihasilkan lebih merata. Berikut ilustrasi penghitungan nilai histogram dari sampel yang telah diuji.

Dari sampel gambar diatas, didapatkan nilai piksel yang dicuplik dengan dimensi 8 x 8 sebagai berikut:

	0	1	2	3	4	5	6	7
0	80	121	84	150	169	185	186	191
1	126	162	165	155	154	171	129	144
2	162	157	175	155	147	169	128	163
3	147	158	169	135	92	167	169	128
4	156	165	155	141	172	167	168	130
5	164	163	100	157	164	166	127	92
6	156	169	126	98	170	164	128	49
7	157	186	144	166	160	165	167	161

Dari nilai piksel citra diatas, dapat ditentukan nilai distribusi kumulatifnya

Nilai Piksel	Frek	Kum	Nilai Piksel	Frek	Kum
49	1	1	156	2	29
80	1	2	157	3	32
84	1	3	158	1	33
92	2	5	160	1	34
98	1	6	161	1	35
100	1	7	162	2	37
121	1	8	163	2	39
126	2	10	164	3	42
127	1	11	165	3	45
128	3	14	166	2	47
129	1	15	167	3	50
130	1	16	168	1	51
135	1	17	169	5	56
141	1	18	170	1	57
144	2	20	171	1	58
147	2	22	172	1	59
150	1	23	175	1	60
154	1	24	185	1	61
155	3	27	186	2	63
			191	1	64

Melalui tabel diatas, dapat ditentukan nilai piksel yang baru hasil dari proses histogram equalization.

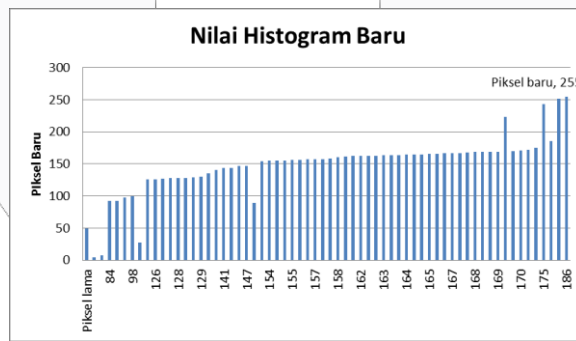
$$h(v) = \text{round} \left( \frac{cdf(v) - cdf_{min}}{(M \times N) - cdf_{min}} \times (L - 1) \right)$$

Piksel Lama	Frek	Piksel Baru	Nilai Piksel	Frek	Piksel Baru
49	1	0	156	2	113
80	1	4	157	3	125

84	1	8	158	1	130
92	2	16	160	1	134
98	1	20	161	1	138
100	1	24	162	2	146
121	1	28	163	2	154
126	2	36	164	3	166
127	1	40	165	3	178
128	3	53	166	2	186
129	1	57	167	3	198
130	1	61	168	1	202
135	1	65	169	5	223
141	1	69	170	1	227
144	2	77	171	1	231
147	2	85	172	1	235
150	1	89	175	1	239
154	1	93	185	1	243
155	3	105	186	2	251
			191	1	255

Maka dihasilkan nilai piksel yang baru adalah

	0	1	2	3	4	5	6	7
0	4	28	8	89	223	243	251	255
1	126	162	165	155	154	171	129	144
2	162	157	175	155	147	169	128	163
3	147	158	169	135	92	167	169	128
4	156	165	155	141	172	167	168	130
5	164	163	100	157	164	166	127	92
6	156	169	126	98	170	164	128	49
7	157	186	144	166	160	165	167	161



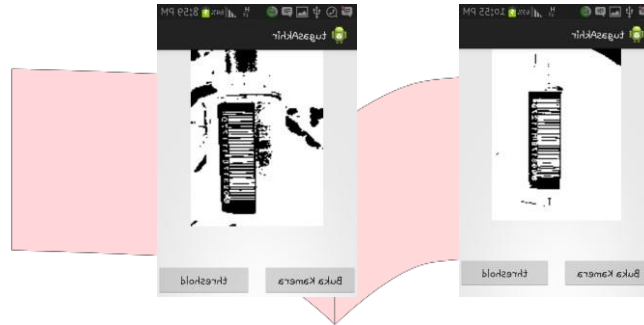
3.4 Grafik Histogram Baru

2.3.2 Thresholding

Setelah mengalami proses ekualisasi, citra masih harus mendapatkan perbaikan piksel. Hal ini dikarenakan kemungkinan pencahayaan dan efek kamera yang tidak sempurna. Pada penelitian kali ini telah dicoba dua buah metode thresholding, yaitu metode manual dan metode otsu.

Pada percobaan pertama digunakan metode manual yaitu memasukkan kemungkinan nilai terbaik. Nilai ini yang akan mengacu untuk dijadikan sebagai nilai *thresholding*. Percobaan manual dimulai dengan memasukkan angka 0 sampai 255.

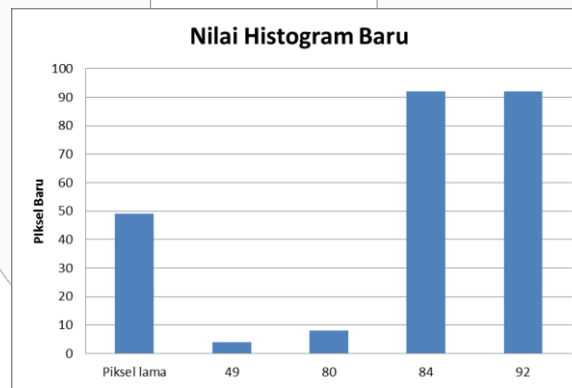
Pada percobaan pertama, digunakan metode manual. Caranya adalah dengan mencoba memasukkan nilai *threshold* dari 0 sampai 255 dengan percobaan masing-masing tiga kali untuk satu barang. Nilai mendekati nol mengindikasikan sistem menyaring warna hitam. Sedangkan nilai mendekati 255 menunjukkan sistem menyaring warna putih. Kemudian, diamati nilai *threshold* yang menghasilkan citra barcode paling baik hasil *thresholding*. Melalui percobaan di atas ditemukan nilai yang cocok pada range nilai 106-110 dengan pencahayaan normal. Namun ternyata nilai tersebut tidak adaptif pada produk yang lain. Ketika diganti produk ditemukan nilai *threshold* lain yang cocok. Nilainya berkisar 60-70 pada cahaya normal.



**Gambar 3.6 Threshold Manual 106 Gambar 3.7 Threshold Manual 60**

Pada percobaan kedua, digunakan metode otsu. Otsu *thresholding* akan memisahkan setiap piksel *foreground* dan *background*. Proses pemisahan ini berdasarkan nilai variansi yang terkecil. Sehingga bagian hitam dan putih lebih mudah untuk dibedakan.

Pada penelitian kali ini, akan dihitung nilai otsu *threshold*, dengan melihat grafik histogram. Perhitungan dengan mengambil sampel dari grafik histogram tersebut.



**Gambar 3.9 Cuplikan Otsu**

Cuplikan gambar diatas merupakan hasil dari histogram ekualisasi. Maka dapat dihitung nilai otsu *thresholding*nya.

Komponen Otsu		0	1	2	3	4
Backgr ound	Weight	0	0.125	0.375	0.875	1.5
	Mean	0	1	1.6666 67	2.4285 71	3.083 333
	Varian	0	0	0.1111 11	0.3119 53	0.638 455

Foregr ound	Weight	2.25	2.125	1.875	1.375	0.75
	Mean	3.72 222 2	3.8823 53	4.1333 33	4.5454 55	5
	Varian	1.11 406 9	1.0449 83	0.6488 89	6.0631 1	0.206 612
Whitin Class Varian		2.50 665 5	2.2205 88	1.2583 33	8.6097 36	1.112 641

Dari hasil perhitungan *background* dan *foreground* dipilih nilai terkecil dari whitin class varian yaitu pada nilai threshold 4. Pemilihan nilai terkecil ini dikarenakan semakin kecil-variansi sebuah citra maka semakin kecil pula perbedaan piksel citra tersebut. Hal ini mengindikasikan bahwa citra tersebut berada pada segmen yang sama.

**2.3.4 Menentukan Letak Angle**

Jika citra yang masuk sudah memiliki karakteristik yang baik maka proses segmentasi dimulai. Proses segmentasi yang digunakan adalah proses yang sederhana, yaitu proses *edge detection* yang mencari nilai selisih *angle*. Nilai sudut yang dicari adalah sudut dengan selisih nilai terkecil. Jika sudut terdeteksi per segmen memiliki nilai rata-rata besar maka terdeteksi sebagai bukan barcode. Namun jika nilai rata-rata per segmen sudut kecil maka terdeteksi sebagai barcode. Dikarenakan proses segmentasi yang dipilih adalah proses yang sangat sederhana. Maka citra masuknya harus mengalami *pre-processing* dahulu sehingga menghasilkan citra yang sempurna.

Melalui sebuah percobaan, dapat dicari komponen piksel dari sebuah citra seperti gambar di atas. Setiap segmen pengambilan dapat dilustrasikan sebagai berikut:

Segemen	Sudut
1	209
2	14
3	10
4	120
5	80
6	112
Rata-rata	90.83 = 91

Segemen	Sudut
1	100
2	102
3	112
4	126
5	110
6	122
Rata-rata	112

Tabel di atas menunjukkan kombinasi segemen yang terdeteksi sebagai bukan barcode. Hal ini dapat dilihat dari selisih antara rata-rata dan nilai sudut tiap segmen yang terlalu besar. Artinya perbedaan antar piksel sangat signifikan sehingga menghasilkan nilai piksel yang berbeda-beda.

Tabel di atas menunjukkan kombinasi segmen yang telah mendeteksi adanya letak barcode. Hal ini dilihat dari selisih antara rata-rata dan nilai sudut tiap segemen relatif kecil. Artinya perbedaan piksel tidak terlalu besar sehingga terindikasikan sebagai barcode.

**3.3 Analisis Performansi**

Pada tugas akhir ini dilakukan analisis terhadap performansi sistem aplikasi yang telah dibuat. Adapun parameter dari performansi adalah sebagai berikut:

1. Parameter *Input*:
  - a. Rotasi pengambilan citra



Rotasi pengambilan citra adalah posisi melingkar sesuai sudut pada saat pengambilan citra. Hal ini dimaksudkan untuk melihat performansi sistem dengan input citra yang berbeda.

b. Jarak pengambilan gambar

Jarak pengambilan gambar adalah posisi jauh dekatnya kamera ketika mengambil gambar. Hal ini akan berpengaruh pada gambar yang dihasilkan, terdapat banyak noise atau tidak.

2. Parameter *Output*:

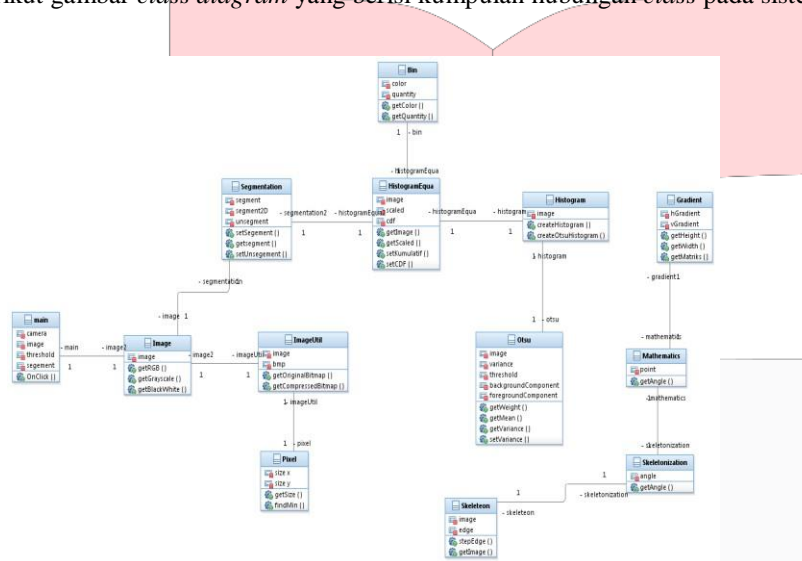
a. Tingkat Akurasi pendeteksian letak

Tingkat akurasi adalah tingkat ukuran ketepatan sistem dalam mengenali objek sesuai letaknya. Adapun rumus matematis tingkat akurasi dapat dinyatakan sebagai berikut:

$$Akurasi = \frac{Data\ Benar}{Data\ Keseluruhan} \times 100\%$$

3.5 Class Diagram

Berikut gambar *class diagram* yang berisi kumpulan hubungan *class* pada sistem aplikasi ini.



4. PENGUJIAN SISTEM DAN ANALISIS

4.1 Pengujian Sistem

Untuk mengetahui hasil performansi perlu melakukan pengujian aplikasi yang telah dirancang. Adapun tujuan dari pengujian aplikasi ini adalah sebagai berikut:

- a. Menganalisis performansi berdasarkan dua parameter *input* yaitu rotasi dan jarak saat pengambilan citra.
- b. Menganalisis performansi berdasarkan dua parameter *output* yaitu akurasi pendeteksian letak barcode.

4.2 Skenario Pengujian Sistem

Proses pengujian aplikasi dilakukan dengan mengambil citra pada sebuah produk pada jarak jauh (10 cm), sedang (7 cm), dan dekat (5 cm). Kemudian dilihat dari posisi kemiringan sudut 0°, 45°, 90°, 135°. Kedua hal diatas adalah parameter *input*. Sedangkan untuk parameter *output* dilakukan dengan mengambil gambar sampel 30 kali kemudian menghitung keberhasilannya. Hal ini dilihat dari akurasi pendeteksi letak berhasil menemukan letak dengan benar.

4.3 Hasil Pengujian

Berikut hasil pengujian yang telah dilakukan dengan 30 produk. Barcode terbaca didefinisikan sebagai kondisi saat barcode yang terdeteksi dapat tertangkap secara penuh maupun sebagian hilang. Dengan catatan hilangnya bagian barcode hanya pada bagian atas atau bawah saja. Hal ini dikarenakan barcode tetap akan bisa terbaca dengan benar jika yang hilang hanya bagian atas atau bawah saja. Komponen yang paling penting dan tidak boleh hilang pada barcode adalah digit *bar*. Hal penting pada barcode adalah komponen bar dan jarak setiap bar nya. Jika komponen bar ini yang hilang maka barcode tidak akan terbaca.



4.3.1 Pengujian Parameter Input

Hasil	Jarak (cm)					Seluruh permukaan
	5	6	7	8	10	
Terbaca	22	24	25	25	26	21
Tidak Terbaca	8	6	5	5	4	5
Jumlah	30	30	30	30	30	30

Pada percobaan menggunakan parameter input berupa jarak, ditemukan bahwa sistem dapat bekerja pada jarak 5 sampai 7 cm. Hal ini ditunjukkan dari banyaknya sampel yang terbaca benar pada jarak ini. Namun jarak 7 cm lebih baik dilihat dari jumlah sampel yang tidak terbaca yang lebih sedikit dari pada sampel pada jarak 5 cm.

Melalui percobaan di atas juga dapat dilihat bahwa sistem tetap dapat bekerja dengan mengambil keseluruhan bagian dari sampel citra tanpa mempertimbangkan jarak pengambilan. Pernyataan tersebut ditunjukkan dari banyaknya citra yang terdeteksi dibandingkan yang tidak terdeteksi.

Jarak 10 cm melalui percobaan di atas terdefiniskan sebagai jarak yang terlalu jauh. Jika jarak pengambilan citra terlalu jauh maka gambar barcode menjadi kecil sehingga sistem tidak dapat menjangkaunya. Hal ini ditunjukkan dari jumlah sampel yang terdeteksi benar bernilai kecil sedangkan yang tidak terdeteksi bernilai besar. Artinya kegagalan pendeteksian pada jarak ini dan jarak yang lebih jauh dari ini berpeluang besar.

Hasil	Sudut Besar (derajat)			
	0	45	90	135
Terbaca	30	1	2	0
Tidak Terbaca	0	29	28	28
Jumlah	30	30	30	30

Hasil	Sudut Kecil (derajat)						
	0	1	2	3	4	5	10
Terbaca	30	29	28	28	26	25	20
Tidak Terbaca	0	1	2	2	4	5	10
Jumlah	30	30	30	30	30	30	30

Pada percobaan menggunakan parameter input berupa sudut ditemukan bahwa sistem bekerja optimal pada sudut 0 sedangkan pada sudut yang lain tidak dapat bekerja. Posisi pengambilan sampel citra harus sejajar untuk memastikan bahwa masukan citra bernilai benar. Derajat kemiringan yang signifikan belum dapat ditoleransi pada sistem ini. Hal ini dikarenakan *screening* dilakukan secara horizontal.

4.3.2 Pengujian Parameter Output

Setelah melakukan pengujian dengan parameter *input*, maka dapat dihitung parameter *output*nya. Parameter output dapat dilihat dari nilai akurasi yang dihasilkan oleh sistem.

Detection rate untuk jarak 5 cm

$$Akurasi = \frac{22}{30} \times 100\% = 73\%$$

$$Akurasi = \frac{24}{30} \times 100\% = 80\%$$

Detection rate untuk jarak 7 cm

$$Akurasi = \frac{25}{30} \times 100\% = 83.3\%$$

Detection rate untuk jarak 8 cm

$$Akurasi = \frac{25}{30} \times 100\% = 83.3\%$$

Detection rate untuk jarak 10 cm

$$Akurasi = \frac{26}{30} \times 100\% = 86.67\%$$

Detection rate untuk pengambilan citra seluruh permukaan

$$Akurasi = \frac{21}{30} \times 100\% = 70\%$$

Berdasarkan data di atas, *detection rate* yang paling tinggi ada pada jarak 7 cm, 8 cm, dan kondisi saat seluruh permukaan produk terambil gambarnya dengan nilai *detection rate* 83.3% dan 70%. Melalui percobaan yang telah dilakukan, dapat dilihat bahwa semakin jauh jarak pengambilan gambar, maka nilai *detection rate* semakin besar.

Detection rate untuk sudut 0°

$$Akurasi = \frac{30}{30} \times 100\% = 100\%$$

Detection rate untuk sudut 1°

$$Akurasi = \frac{29}{30} \times 100\% = 96.67\%$$

Detection rate untuk sudut 2°

$$Akurasi = \frac{28}{30} \times 100\% = 93.33\%$$

Detection rate untuk sudut 3°

$$Akurasi = \frac{28}{30} \times 100\% = 93.33\%$$

Detection rate untuk sudut 4°

$$Akurasi = \frac{26}{30} \times 100\% = 86.67\%$$

Detection rate untuk sudut 5°

$$Akurasi = \frac{25}{30} \times 100\% = 83.33\%$$

Detection rate untuk sudut 10°

$$Akurasi = \frac{20}{30} \times 100\% = 66.67\%$$

Detection rate untuk sudut 45°

$$Akurasi = \frac{1}{30} \times 100\% = 3.33\%$$

Detection rate untuk sudut 90°

$$Akurasi = \frac{2}{30} \times 100\% = 6.67\%$$

Detection rate untuk sudut 135°

$$Akurasi = \frac{2}{30} \times 100\% = 6.67\%$$

Berdasarkan data di atas, *detection rate* yang paling tinggi ada pada sudut 0°. Semakin tinggi tingkat kemiringan maka *detection rate* semakin kecil. Namun sistem ini masih memiliki toleransi pengambilan gambar berkisar pada sudut maksimal 10° dengan *detection rate* 66.67%.

Hal ini dikarenakan sistem belum dapat mendeteksi perubahan sudut yang signifikan. Jika sudut terlalu miring, maka nilai piksel letak sudut akan berbeda. Sehingga pada proses penentuan letak *barcode* sistem tidak dapat menemukan piksel yang berkesesuaian. Oleh karena itu, sistem yang dibuat belum dapat mentoleransi perubahan sudut yang terlalu signifikan.

## 5. KESIMPULAN DAN SARAN

### 5.1 Kesimpulan

Adapun beberapa kesimpulan dari hasil pengujian dan analisis yang telah dilakukan pada aplikasi ini adalah sebagai berikut:

5.2 Rancangan sistem pendeteksi letak *barcode* ini dapat diimplementasikan.

5.3 Sistem ini dapat bekerja optimal diberbagai jarak namun jarak paling baik berdasarkan penelitian ada pada jarak 7 cm dengan *detection rate* 83.3% dan sudut 0° dengan *detection rate* 100%

5.4 Komponen *barcode* yang paling penting adalah nilai *bar*. Jika ada komponen yang dibagian atas atau bawah yang tidak terambil gambarnya maka *barcode* masih bisa terbaca.

### 5.5 Saran

Adapun saran untuk mengembangkan tugas akhir ini agar menjadi lebih baik, yaitu sebagai berikut:

1. Aplikasi ini digabungkan dengan aplikasi pembaca *barcode* dengan keluaran suara yang telah dibuat sebelumnya.
2. Sistem dikembangkan dengan fleksibilitas sudut pengambilan citra.
3. Sistem dianalisis dengan sistem komputasi yang lebih cepat.

## 6. DAFTAR PUSTAKA

- [1] Hariyanto, Bambang. (2011). *Esensi-Esensi Bahasa Pemrograman JAVA Revisi Keempat*. Bandung: Informatika.
- [2] I. C. (2007). *Essentials of IBM Rational Software Architect v7 RD541/DEV396 February 2007 Student Manual*. California: IBM Rational Software.
- [3] Irawan. (2011). *Java untuk Orang Awam*. Palembang: Irawan
- [4] Safaat, Nazuddin. 2011. *Android Pemrograman Aplikasi Mobile Smartphone dan Tablet PC berbasis Android*. Bandung: Informatika
- [5] Pengolahan Gambar Secara Digital. (1998). Agustinus Nalwan. Jakarta: Gramedia
- [6] Pengolahan Citra Digital Menggunakan Matlab. Marvin Wijaya, Agus Prijono. (2007). Bandung: Informatika
- [6] Matlab Programming. (2010). Gunaidi Abdi Away. Bandung: Informatika
- [7] [labbookpages.co.uk/software/imgProc/otsuThreshold.html](http://labbookpages.co.uk/software/imgProc/otsuThreshold.html) diakses 5 Juni 2015