

## BAB III

### METODE PENELITIAN

#### 3.1. Model Pengembangan

Tujuan dari tugas akhir ini adalah untuk membuat sebuah aplikasi untuk mengatur kontras pada gambar secara otomatis. Dan dapat meningkatkan kualitas citra yang ada pada video atau gambar yang akan digunakan untuk mendeteksi atau melacak badan manusia. Dimana hasil dari deteksi atau melacak manusia itu akan dimanfaatkan dalam bidang lain. Seperti untuk robot pelacak manusia yang akan mengikuti kemana manusia itu bergerak. Gambar atau video tersebut akan diolah menggunakan Microsoft Visual Studio C++ dan library openCV untuk bisa mendeteksi objek yang akan diharapkan yaitu badan manusia.

#### 3.2. Prosedur Penelitian

Prosedur penelitian yang dipakai dalam pengerjaan tugas akhir ini adalah

##### 1. Studi literatur

Pencarian data-data literature dari masing-masing fungsi pada library Microsoft Visual C++ dan openCV melalui pencarian dari internet, dan konsep-konsep teoritis dari buku-buku penunjang serta metode yang digunakan untuk melakukan pengolahan citra.

##### 2. Tahap perancangan dan pengembangan sistem

Dalam membuat pengembangan sistem, terdapat beberapa langkah rancangan sistem yang diambil antara lain:

- a. Membuat *flowchart* pada proses sistem secara keseluruhan proses pengolahan citra, proses deteksi badan manusia

menggunakan metode HAAR-like-feature, dan proses pengaturan kontras.

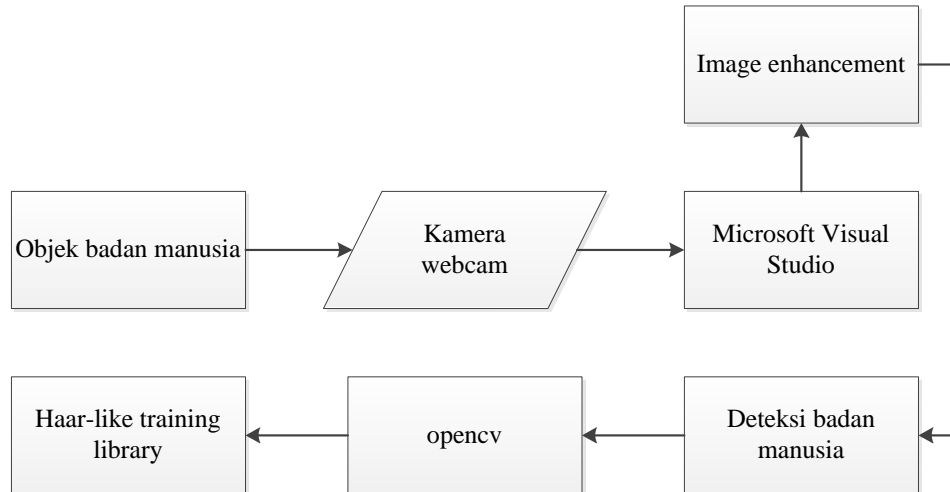
- b. Membuat aplikasi pendeteksi badan manusia bagian atas
- c. Mengatur fungsi-fungsi yang digunakan pada aplikasi ini dan mengelompokkan fungsi tersebut pada beberapa *class*.
- d. Pengambilan gambar secara *real-time*.
- e. Melakukan proses pengolahan citra yaitu konversi ke BGR dan *grayscale*.
- f. Menerapkan metode HAAR-like feature.
- g. Melakukan percobaan pada aplikasi ini untuk memastikan apakah aplikasi ini sudah dapat berjalan dengan baik atau belum.

### 3.3. Diagram Blok Sistem

Sistem ini terdiri 2 proses utama yaitu blok proses deteksi dan proses pengaturan kontras atau *image enhancement*. Aplikasi *image enhancement* akan melakukan perbaikan kontras pada gambar yang ditangkap oleh kamera. Kemudian hasil dari proses *image enhancement* ini akan digunakan pada proses deteksi.

Sedangkan pada proses deteksi terdiri dari haar *training*, aplikasi deteksi, dan objek badan manusia. Aplikasi deteksi akan melakukan proses pengolahan citra untuk mendeteksi objek yang sesuai dari hasil proses *image enhancement*.

Untuk mempermudah dalam memahami sistem yang akan dibuat dapat dijelaskan melalui blok diagram dibawah ini.

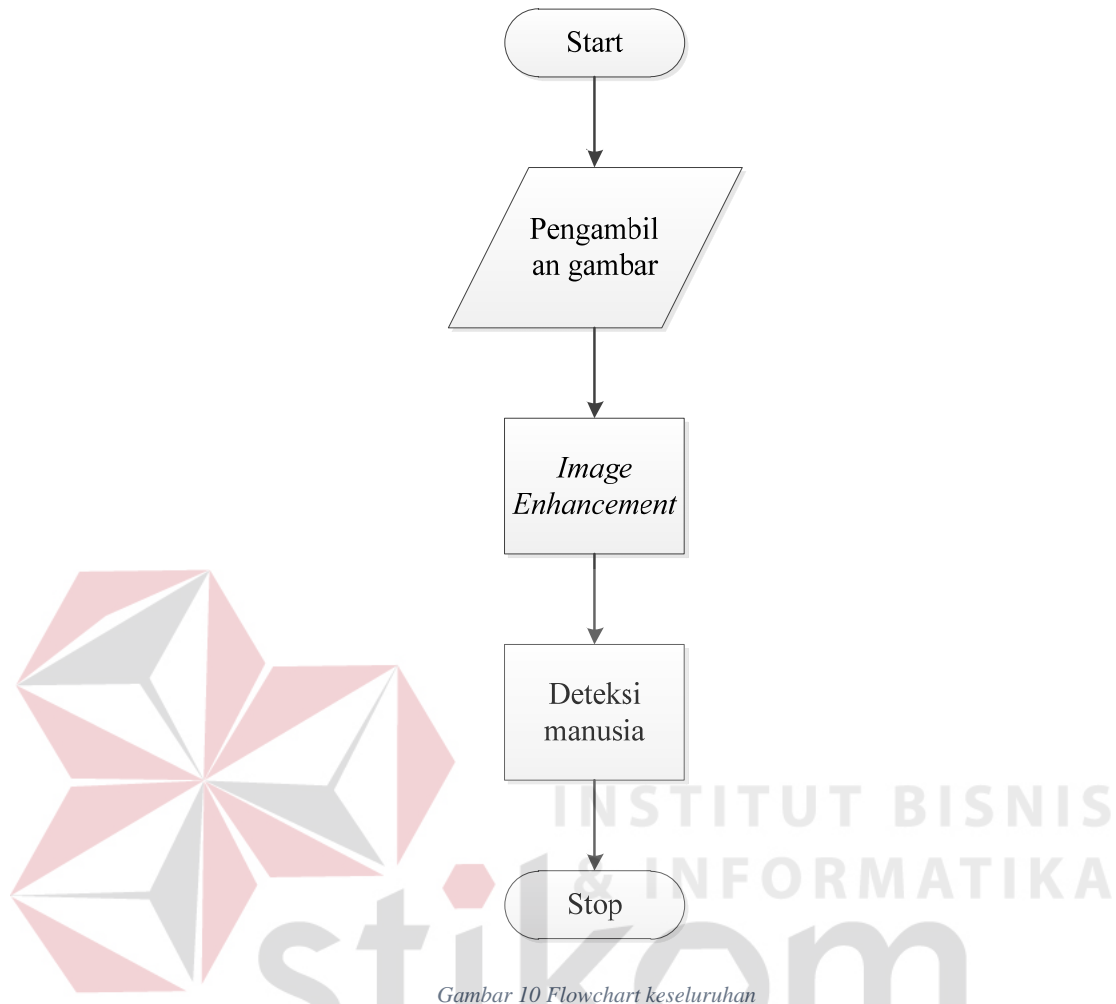


Gambar 9 diagram blok

### 3.4. Perancangan Perangkat Lunak

Dalam perancangan perangkat lunak, *compiler* yang digunakan adalah Microsoft Visual C++. Dan untuk *library* yang digunakan pada pengolahan citra yaitu *library* OpenCV.

Kemudian dalam penulisannya atau dalam pembuatan program, akan meliputi bagian-bagian penting dalam setiap langkah-langkah perbagian sesuai dengan algoritma atau logika sekuensial dari awal sampai output. Berikut adalah algoritma program secara global.



Gambar 10 Flowchart keseluruhan

Berikut penjelasan dari diagram alur proses deteksi badan manusia beserta kode pemrograman menggunakan metode deteksi Haar-like feature:

1. Proses mengaktifkan *webcam* dan melakukan konversi ke berbagai ruang warna.
2. *Webcam* akan melakukan proses *scanning* objek pada *frame* gambar yang didapatkan.
3. Mengambil data *training Haar cascade classifier* berupa file berformat xml pada program dengan fungsi kode pemrograman `CascadeClassifier::load`, nama file yang dipanggil berupa file

\*.xml yang berada pada direktori *project* yang disimpan. Berikut penggalan kode pemrograman untuk mengambil data training :

```
String full_cascade_name = "haarcascade_upperbody.xml";
```

4. Proses selanjutnya adalah membandingkan nilai *pixel threshold* dan *frame* objek yang tertangkap kamera dengan nilai *pixel haar training*. *Haar-like feature* memproses gambar dalam wilayah kotak-kotak yang berisi beberapa *pixel* dari sebuah bagian gambar. Kemudian piksel-piksel dalam satu wilayah tersebut dijumlahkan dan dilakukan proses perhitungan sehingga didapatkan perbedaan dalam setiap wilayah kotak-kotak tersebut. Perbedaan inilah yang dapat dijadikan sebuah kode untuk menandai wilayah tersebut sehingga dapat memproses bagian gambar. Untuk dapat mendeteksi objek yang diinginkan digunakan kode fungsi pemrograman:

```
detectMultiScale(const Mat& image,  
CvHaarClassifierCascade* cascade, double scaleFactor=1.1,  
int minNeighbors=3, int flags=0, Size minSize=Size(),  
Size maxSize=Size()).
```

**Keterangan:**

- Image** : Frame gambar yang telah dikonversi ke dalam ruang warna grayscale
- Cascade** : Haar training yang digunakan
- Scale** : Ukuran gambar yang akan dikurangi pada setiap skala
- minNeighbors** : Banyaknya objek yang akan terdeteksi oleh metode HAAR

`minSize` : ukuran minimal objek yang akan dideteksi

`maxSize` : ukuran maksimal objek yang akan dideteksi

Berikut penggalan kode pemrograman untuk deteksi badan manusia bagian atas:

```
Full_cascade.detectMultiScale(thresholded, full,
1.1,1,0|CV_HAAR_SCALE_IMAGE, Size(50,50);
```

### 3.5. Metode Pengujian dan Evaluasi Sistem

Untuk mengetahui apakah aplikasi yang dibuat dapat berjalan sesuai yang diharapkan, maka akan dilakukan pengujian dan evaluasi sistem untuk setiap tahapan-tahapan dalam pembuatan aplikasi. Dimulai dari pengambilan citra, *color filtering* menggunakan ruang warna RGB, BGR dan *thresholding*, deteksi badan manusia menggunakan metode *Haar-like feature* dan penjejakan badan manusia bagian atas sesuai dengan pergerakan manusia.

#### 3.5.1 Pengujian *Streaming* Citra Melalui Kamera *Webcam*

Untuk mengetahui apakah data citra sudah dapat diakses langsung melalui kamera *webcam*, maka dilakukan pengujian dengan cara menjalankan program (*running*) pemanggil kamera di Microsoft Visual Studio, yaitu untuk mengakses *console* kamera *webcam* secara langsung dari program. Kemudian citra yang tampil akan diuji apakah dapat menampilkan data citra secara *streaming*.

#### 3.5.2. Perhitungan *Histogram Equalization*

Contoh tahap-tahap perhitungan manual untuk ekualisasi histogram adalah sebagai berikut:

1. Misalkan terdapat sebuah citra keabuan dengan nilai-nilai piksel sebagai berikut:

(i,j)	0	1	2	3	4	5	6	7
0	29	40	44	39	111	116	81	108
1	40	44	62	90	111	111	108	58
2	186	132	132	154	154	154	150	229
3	136	184	175	155	171	171	171	148
4	254	133	133	136	154	151	151	157
5	140	133	133	131	136	138	149	149
6	115	128	133	136	136	136	138	117
7	122	128	138	143	133	119	139	128

Tabel 5 Contoh Value dari Sebuah Gambar

2. Dari nilai piksel pada matriks citra tersebut, dihitung frekuensi dan distribusi kumulatif dari nilai skala keabuannya. Daftar frekuensi dan perhitungan distribusi kumulatif dapat dilihat pada tabel berikut:

Skala Keabuan	Frekuensi	Distribusi Kumulatif	Skala Keabuan	Frekuensi	Distribusi Kumulatif
29	1	1	136	6	38
39	1	2	138	3	41
40	2	4	139	1	42
44	2	6	140	1	43
58	1	7	143	1	44
62	1	8	148	1	45
81	1	9	149	2	47
90	1	10	150	1	48
108	2	12	151	2	50
111	3	15	154	4	54
115	1	16	155	1	55
116	1	17	157	1	56
117	1	18	171	3	59
119	1	19	175	1	60
122	1	20	184	1	61

Skala Keabuan	Frekuensi	Distribusi Kumulatif
128	3	23
131	1	24
132	2	26
133	6	32

Skala Keabuan	Frekuensi	Distribusi Kumulatif
186	1	62
229	1	63
254	1	64

Tabel 6 Daftar Frekuensi dan Distribusi Kumulatif

3. Menghitung nilai keabuan dari hasil perhitungan distribusi kumulatif menggunakan rumus yang sebelumnya telah dituliskan di atas. Berikut ini merupakan contoh perhitungan untuk skala keabuan 2 dan 30:

$$K_o = \text{round} \left( \frac{1x(2^8 - 1)}{8x8} \right) = \text{round} \left( \frac{255}{64} \right) = 4$$

$$K_o = \text{round} \left( \frac{2x(2^8 - 1)}{8x8} \right) = \text{round} \left( \frac{510}{64} \right) = 8$$

Hasil perhitungan untuk seluruh nilai skala keabuan dapat dilihat pada tabel berikut:

Keabuan Awal	Frekuensi	Keabuan Hasil
29	1	4
39	1	8
40	2	16
44	2	24
58	1	28
62	1	32
81	1	36
90	1	40
108	2	48
111	3	60
115	1	64
116	1	68
117	1	72
119	1	76
122	1	80

Keabuan Awal	Frekuensi	Keabuan Hasil
136	6	151
138	3	163
139	1	167
140	1	171
143	1	175
148	1	179
149	2	187
150	1	191
151	2	199
154	4	215
155	1	219
157	1	223
171	3	235
175	1	239
184	1	243



Keabuan Awal	Frekuensi	Keabuan Hasil	Keabuan Awal	Frekuensi	Keabuan Hasil
128	3	92	186	1	247
131	1	96	229	1	251
132	2	104	254	1	255
133	6	127			

*Tabel 7 Hasil Perhitungan nilai skala keabuan*

Hasil matriks citra setelah proses histogram equalization adalah sebagai berikut:

(i,j)	0	1	2	3	4	5	6	7
0	4	16	24	8	60	68	36	48
1	16	24	32	40	60	60	48	28
2	247	104	104	215	215	215	191	251
3	151	243	239	219	235	235	235	179
4	255	127	127	151	215	199	199	223
5	171	127	127	96	151	163	187	187
6	64	92	127	151	151	151	163	72
7	80	92	163	175	127	76	167	92

*Tabel 8 Hasil Matriks Citra Setelah Proses Histogram Equalization*

### 3.5.3. Pengujian Deteksi Badan Manusia

Untuk mengetahui apakah aplikasi dapat mendeteksi badan manusia bagian atas. Pengujian ini akan dilakukan berdasarkan jarak manusia dengan kamera, intensitas cahaya dan banyaknya badan manusia yang tertangkap oleh kamera. Pengujian ini dinyatakan berhasil jika aplikasi sudah dapat mendeteksi badan manusia bagian atas yang tertangkap oleh kamera. Setiap badan yang tertangkap

oleh kamera dan berhasil dideteksi akan ditandai dengan sebuah lingkaran yang berada disekeliling badan.

#### **3.5.4. Pengujian Deteksi Badan Manusia dengan Berbagai Intensitas Cahaya**

Setelah mengetahui bahwa aplikasi dapat mendeteksi badan manusia bagian atas dengan baik, maka dilakukan pengujian deteksi pada berbagai intensitas cahaya. Untuk menguji kemampuan *image enhancement* yang ada pada program tersebut. Kemampuan *image enhancement* diharapkan mampu membantu deteksi badan manusia untuk berhasil mendeteksi pada intensitas cahaya yang lebih gelap atau lebih terang dari yang sebelumnya.

#### **3.5.5. Pengujian Delay Image Enhancement**

Setelah diterapkan kemampuan *image enhancement* pada gambar yang ditangkap oleh kamera, akan terjadi *delay* atau keterlambatan gambar dibandingkan tanpa menggunakan kemampuan *image enhancement*, oleh karena itu dilakukan pengujian selisih delay yang terjadi.