BAB III

METODE PENELITIAN

3.1 Metode Penelitian

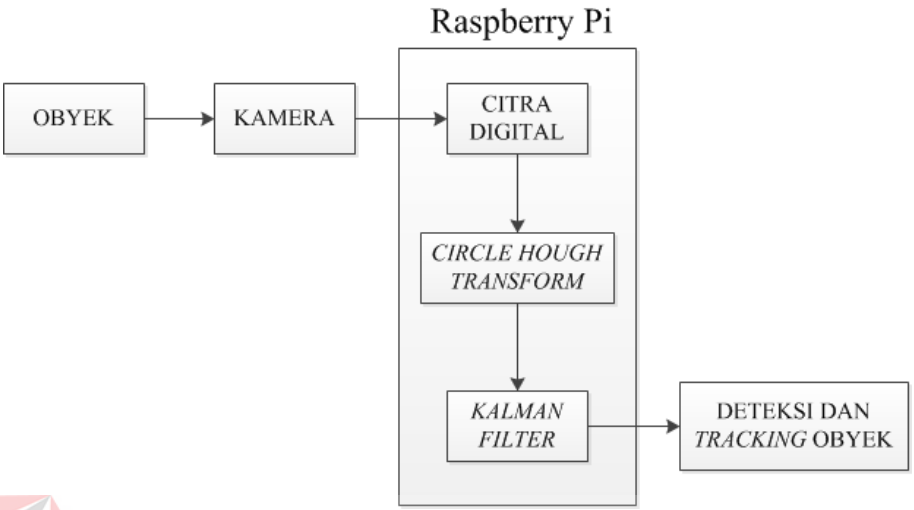
Metode penelitian yang digunakan dalam pengerjaan Tugas Akhir ini adalah studi literatur, pembuatan program serta melakukan deteksi dan *tracking* obyek. Pada penelitian tugas akhir ini, terdapat obyek berupa bola tenis berwarna hijau.. Blok diagram penelitian ditunjukkan pada Gambar 3.1.

Gambar 3.1. Blok Diagram Penelitian

Penelitian diawali dengan melakukan pendeteksian obyek menggunakan teknik *image processing* dari kamera. Teknik *image processing* menggunakan *library* OpenCV.Setelah obyek dapat dideteksi oleh kamera, langkah selanjutnya adalah melakukan *tracking* terhadap perpindahan posisi dari obyek. *Tracking* akan menghasilkan output berupa koordinat dari titik tengah obyek.

3.2 Prosedur Penelitian

Prosedur penelitian yang digunakan pada pengerjaan Tugas Akhir ini ditunjukkan oleh Gambar 3.2.

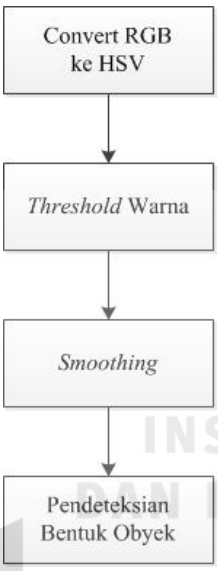


Gambar 3.2. Prosedur Penelitian

Kamera akan menghasilkan citra digital yang *real-time* dari obyek. Kemudian pada citra digital tersebut dilakukan proses pendeteksian obyek dengan menggunakan *Circle Hough Transform*.Didalam pendeteksian obyek menggunakan metode *Circle Hough Transform* terdapat proses pendeteksian warna dan bentuk obyek berupa lingkaran. Obyek yang telah terdeteksi baik warna dan bentuknya, diproses menggunakan metode *Kalman Filter* untuk dilakukan proses *tracking*obyek.Pada proses *tracking* obyek terdapat sebuah frame *detector* berupa sebuah *circle*untuk mengikuti pergerakan obyek. Pada *Kalman Filter* terdapat 2 tahapan yaitu *update* waktu dan *update* pengukuran.

3.3 Pendeteksian Obyek menggunakan CHT

Pada proses pendeteksian obyek menggunakan metode *Circle Hough Transform*terdapat tahapan-tahapan dalam pendeteksian obyek seperti yang ditunjukkan oleh Gambar 3.3.



Gambar 3.3. Tahapan *Circle Hough Transform*

Proses tahapan *Circle Hough Transform*, dijelaskan pada sub bab berikut:

3.3.1 *Convert* RGB ke HSV

Pada tahap ini dilakukan konversi model warna pada citra atau frame, yang semula frame mempunyai model warna RGB (*Red, Green, Blue*) diubah ke model warna HSV (*Hue, Saturation, Value*). Proses pengkonversian model warna ini menggunakan bantuan dari *library* OpenCV. Setelah proses pengkonversian pada model warna citra,

selanjutnya dilakukan proses pencarian nilai minimum dan maksimum dari setiap elemen HSV yang mempresentasikan warna dari obyek. Pencarian nilai tersebut bertujuan untuk proses *thresholding* warna pada citra. Berikut merupakan program untuk pengkonversian model warna RGB ke HSV:

```
cvCvtColor(imgOri, imgHSV, CV_BGR2HSV);
```

Program lengkap secara keseluruhan dapat dilihat pada Lampiran 1.

3.3.2 *Threshold* Warna

Pada tahap ini dilakukan *thresholding* warna pada citra atau frame. Fungsi dari *thresholding*pada penelitian ini adalah mengubah citra yang semula mempunyai 3 layer, yaitu HSV (*Hue, Saturation, Value*) menjadi citra biner atau hitam putih yang mempunyai 1 layer. Sehingga dapat diketahui daerah mana yang termasuk obyek dan *background* dari citra secara jelas.Nilai yang digunakan untuk *threshold* warna adalah nilai minimum dan maksimum dari setiap elemen HSV. Nilai pixel yang mempunyai nilai diluar dari *range* nilai minimum dan maksimum tersebut, nilai pixelnya diubah nilainya menjadi sama dengan 0. Sedangkan nilai pixel yang berada pada *range* nilai minimum dan maksimum tersebut, nilai pixelnya diubah nilainya menjadi sama dengan 255. Dimana pixel yang mempunyai nilai 0, pixelnya berwarna hitam. Sedangkan, pixel yang mempunyai nilai 255, pixelnya berwarna putih. Tahap ini bertujuan untuk menyeleksi warna obyek dengan

warna benda yang tertangkap oleh kamera.Berikut merupakan program *threshold* warna pada citra:

```
cvInRangeS(imgHSV, cvScalar(H_MIN, S_MIN,
V_MIN),cvScalar(H_MAX, S_MAX, V_MAX), imgProses);
```

Pada penelitian ini untuk mendeteksi warna dari obyek yaitu hijau, nilai dari HSV sebagai berikut:

H_MIN : 33

H_MAX : 60

S_MIN : 77

S_MAX : 256

V_MIN : 65

V_MAX : 256

Program lengkap secara keseluruhan dapat dilihat pada Lampiran 1.

3.3.3 *Smoothing*

Pada tahap ini dilakukan *smoothing* pada citra atau frame. *Smoothing* dilakukan dengan filter Gaussian. Filter Gaussian digunakan untuk menghilangkan *noise-noise* kecil yang ada disekitar obyek sehingga citra menjadi lebih *smooth*.Berikut merupakan program*smoothing* pada citra:

```
cvSmooth(imgProses, imgProses, CV_GAUSSIAN, 9, 9);
```

Program lengkap secara keseluruhan dapat dilihat pada Lampiran 1.

3.3.4 Pendeteksian Bentuk Obyek

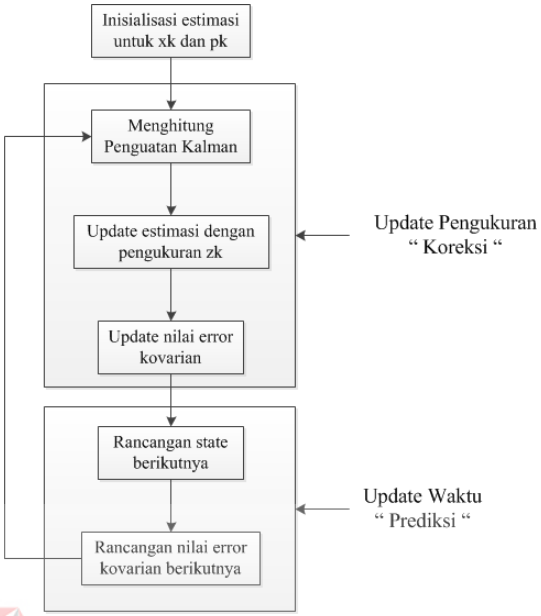
Pada tahap ini dilakukan pendeteksian bentuk obyek. Proses pendeteksian obyek dilakukan dengan menggunakan metode *Circle Hough Transform. Circle Hough Transform* mempunyai cara kerja yaitu membaca nilai pixel dari setiap baris dan kolom pada *grayscale image*.Bentuk obyek yang dideteksi didapat dari pixel-pixel yang mempunyai nilai 255. Dimana pixel-pixel yang mempunyai nilai 255 akan digolongkan menjadi satu. Pada matriks*grayscale image*akan terlihat pixel-pixel yang bernilai 255 akan membentuk sebuah bentuk yaitu lingkaran. Dengan terdeteksinya bentuk lingkaran tersebut, maka obyek yang dideteksi dapat terdeteksi.Setelah obyek berhasil dideteksi oleh kamera, maka selanjutnya dilakukan proses pengambilan nilai koordinat titik tengah obyek. Proses pengambilan nilai koordinat menggunakan bantuan dari *library* OpenCV. Berikut merupakan program pendeteksian bentuk obyek menggunakan metode *Circle Hough Transfrom*dan pengambilan nilai koordinat titik tengah obyek:

```
//Program deteksi obyek
lingkaran = cvHoughCircles(imgProses,
strStorage,CV_HOUGH_GRADIENT, 2,imgProses→height / 4, 150,
50, 5, 200);
//Program pengambilan nilai koordinat titik tengah obyek
koordinat = (float*)cvGetSeqElem(lingkaran, 1);
```

Program lengkap secara keseluruhan dapat dilihat pada Lampiran 1.

3.4 *Tracking* Obyek Menggunakan *Kalman Filter*

Pada peneletian ini proses *tracking* obyek menggunakan metode *Kalman Filter*. Input dari *Kalman Filter*berupa nilai koordinat yang masih mempunyai noise yang merupakan hasil dari pengambilan nilai koordinat titik tengah obyek. *Kalman Filter* digunakan untuk memfilter *noise* yang ada nilai koordinat tersebut.Pada *tracking* obyek menggunakan *Kalman Filter* terdapat beberapa proses sebelum mendapatkan output berupa nilai koordinat tanpa *noise*. Pertama, inisialisasi estimasi untuk kondisi dan nilai *error* kovarian. Kemudian masuk kedalam proses *update* pengukuran yang juga disebut proses koreksi. Selanjutnya, menghitung penguatan kalman dengan menggunakan nilai *error* kovarian pada waktu sebelumnya. Penguatan kalman ini bertujuan untuk meminimumkan nilai *error* kovarian. Setelah didapatkan nilai penguatan kalman, mengitung nilai ukur aktual yang digunakan untuk mengetahui estimasi kondisi saat ini. Proses berikutnya adalah meng*update* nilai *error* kovarian.. Setelah mendapatkan kondisi dan nilai error kovarian yang baru, kemudian dilanjutkan ke dalam proses *update* waktu yang juga disebut sebagai proses prediksi. Pada proses ini melakukan perhitungan pada estimasi kondisi dan nilai error kovarian untu waktu selanjutnya. Setelah dilakukan proses *update* waktu atau proses prediksi, selanjutnya masuk kembali kedalam proses *update* pengukuran atau koreksi. Operasi *Kalman Filter* dapat dilihat pada Gambar3.4.



Gambar 3.4.Operasi *Kalman Filter*

Berikut merupakan program *Kalman Filter* untuk melakukan *tracking* pada obyek:

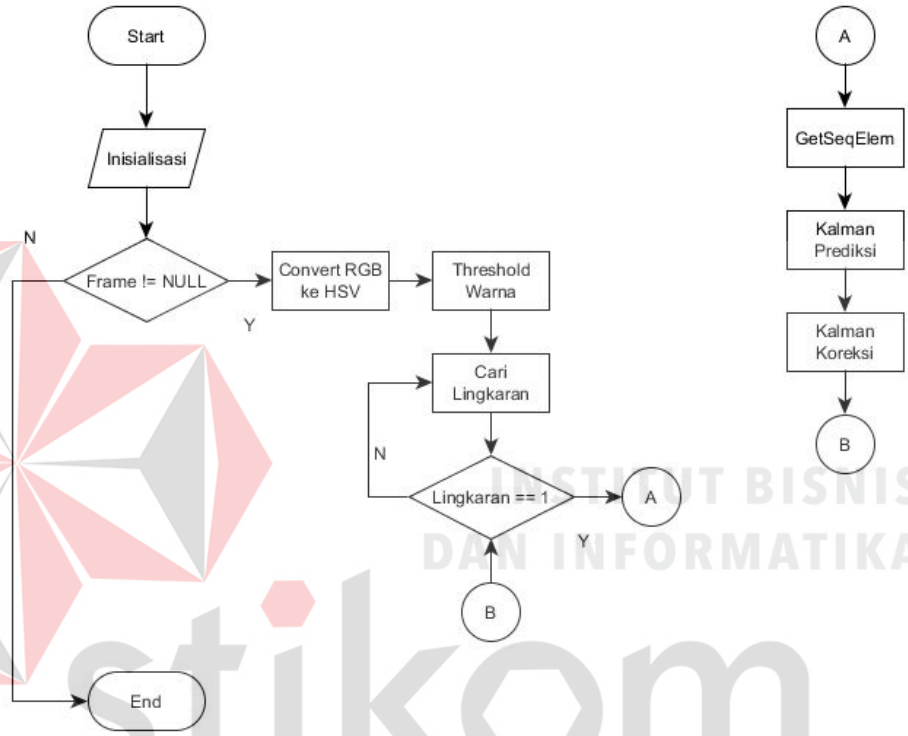
```
//Pembuatan matriks A dan H
const float A[] = {1, 0, 1, 0,
        0, 1, 0, 1,
            0, 0, 1, 0,
                0, 0, 0, 1 };
const float H[] = {1, 0, 0, 0,
                0, 1, 0, 0 };
```

```
//Inisialisasi matriks kovarian untuk state
cvSetIdentity(kalman→error_cov_post, cvRealScalar(1e+10));
//Mengasumsikan matriks kovarian Q_k dan R_k untuk pengukuran
cvSetIdentity(kalman→process_noise_cov,   cvRealScalar(1e-6));   // Q_k
cvSetIdentity(kalman→measurement_noise_cov, cvRealScalar(1e-2)); //
R_k
// Proses Kalman Prediksi (update waktu): (x_k)p = A x_(k-1)
cvKalmanPredict(kalman, 0);
kf_prediksi = cvPoint((int)kalman→state_pre→data.fl[0],
(int)kalman→state_pre→data.fl[1]);
// Matrix pengukuran (z_k) mempunyai hubungan linier ke x_k: z_k =
H_k x_k + v_k ~ N(0, R_k)
measurement→data.fl[0] = (float)pt_Measurement.x;
measurement→data.fl[1] = (float)pt_Measurement.y;
// Proses Kalman Koreksi (update pengukuran): x_k = (x_k)p + K_k(z_k -
H_k *(x_k)p)
cvKalmanCorrect(kalman, measurement);
// Menerima state koreksi
Kf_koreksi = cvPoint((int)kalman→state_post→data.fl[0],
(int)kalman→state_post→data.fl[1]);
```

Program lengkap secara keseluruhan dapat dilihat pada Lampiran 1.

3.5 *Flowchart* Program

Pada penelitian ini mempunyai *flowchart* program seperti yang ditunjukkan pada Gambar 3.5.



Gambar 3.5. *Flowchart* Program

Penjelasan dari *flowchart* pada Gambar 3.5 adalah sebagai berikut :

1. Inisialisasi merupakan proses deklarasi variabel-variabel yang digunakan didalam program. Pada penelitian ini variabel-variabel seperti variabelyang digunakan untuk menampung gambar tangkapan kamera, dll.

2. Menampilkan status pada frame. Ketika frame bernilai NULL, maka program akan berhenti. Sedangkan, ketika frame bernilai tidak NULL, maka program akan menjalankan perintah berikutnya.

3. *Convert* RGB ke HSV merupakan pengkonversian model warna pada frame. Proses ini bertujuan untuk mengubah model warna frame yang semula mempunyai model warna RGB ke HSV.

4. *Threshold* warna merupakan proses penyeleksian warna obyek. Proses ini bertujuan untuk mencari nilai *threshold* minimal dan maksimal dari model warna HSV pada frame.

5. Proses cari lingkaran bertujuan untuk pendeteksian bentuk obyek yang berupa lingkaran didalam frame.

6. Ketika obyek terdeteksi, maka nilai dari lingkaran menjadi 1.

7. Ketika nilai lingkaran sama dengan 1, maka program akan melanjutkan ke perintah berikutnya.

8. Proses selanjutnya yaitu GetSeqElem. GetSeqElem merupakan proses pengambilan nilai koordinat titik tengah obyek. Koordinat yang keluar berupa nilai koordinat x dan y.

9. Setelah didapatkan nilai koordinat x dan y. Nilai tersebut dimasukkan ke dalam proses kalman prediksi dan koreksi.

3.6 Metode Pengujian dan Evaluasi Sistem

Dalam pengujian sistem ini akan dilakukan pengujian terhadap keberhasilan sistem dalam mendeteksi dan *tracking* terhadap obyek. Pengujian yang telah dilakukan dimulai dari pengujian program deteksi warna obyek, pengujian program deteksi bentuk obyek, pengujian terhadap jarak obyek dengan kamera, pengujian program tracking obyek, dan pengujian titik tengah obyek berdasarkan posisi pada matriks *image*.

3.6.1 Pengujian dan Evaluasi Program Deteksi Warna Obyek

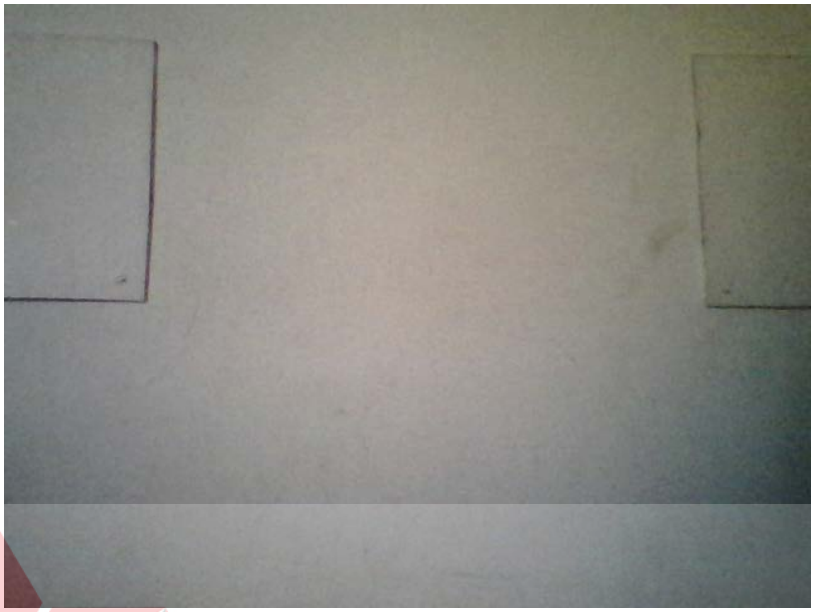
Faktor pendeteksian warna sangat berpengaruh pada tingkat keberhasilan deteksi. Deteksi warna dilakukan dengan melakukan penyeleksian warna terhadap obyek yang tertangkap oleh kamera. Pengujian dilakukan dengan menjalankan program, kemudianmenempatkan obyek didepan kamera agar obyek dapat ditangkap oleh kamera. Selanjutnya, dilakukan *threshold* warna dengan melakukan kalibrasi pada nilai pada model warna HSV. Hasil pendeteksian warna obyek dari program yang telah dibuat ditunjukkan oleh Gambar 3.6 dan 3.7. Gambar 3.6 menunjukkan frame proses yang belum dikalibrasi, sedangkan frame proses yang telah dikalibrasi ditunjukkan oleh Gambar 3.7.

Gambar 3.6 Frame Proses yang Belum Dikalibrasi

Gambar 3.7Frame Proses yang Telah Dikalibrasi

3.6.2 Pengujian dan Evaluasi Program Deteksi Bentuk Obyek

Pengujian ini dilakukan untuk mendeteksi bentuk obyek pada penelitian ini. Pendeteksian bentuk obyek juga sangat penting dalam proses *tracking* obyek. Sebelum melakukan *tracking* obyek, harus dilakukan terlebih dahulu proses pendeteksian obyek. Pendeteksian obyek bertujuan untuk menentukan titik tengah dari obyek.Keberhasilan pendeteksian obyek dapat dilihat adanya frame *detector* berupa titik tengah yang ditandai dengan warna hijau dan garis tepi yang melingkar yang ditandai dengan warna merah pada obyek. Pengujian dilakukan dengan menjalankan program, kemudianmenempatkan obyek didepan kamera agar obyek dapat ditangkap oleh kamera. Kemudian, dilakukan pendeteksian pada bentuk obyek. Hasil pendeteksian bentuk obyek ditunjukkan oleh Gambar 3.8 dan 3.9. Gambar 3.8 menunjukkan frame aslisebelum obyek terdeteksi, sedangkan frame asli setelah obyek terdeteksi ditunjukkan oleh Gambar 3.9. Dapat dilihat pada Gambar 3.9, ketika obyek terdeteksi oleh kamera muncul frame *detector*berupa lingkaran berwarna merah yang menunjukkan bahwa obyek tersebut merupakan obyek yang dideteksi, yaitu bola tenis.



Gambar 3.8 Frame Asli Sebelum Obyek Terdeteksi

Gambar 3.9 Frame Asli Setelah Obyek Terdeteksi

3.6.3 Pengujian dan Evaluasi Terhadap Jarak Obyek Dengan Kamera

Pengujian ini dilakukan untuk mengetahui seberapa jauh kamera dapat mendeteksi obyek. Dimana jarak sangat berpengaruh terhadap pendeteksan obyek. Setelah melakukan beberapa pengujian, diharapkan dapat memperoleh jarak yang efektif antara obyek dengan kamera sehingga proses pendeteksian dapat dilakukan dengan tepat. Pengujian dilakukan cara dengan menempatkan obyek di depan kamera dengan jarak yang telah ditentukan. Pengujian dilakukan mulai dari jarak 5 cm sampai dengan 180 cm.

3.6.4 Pengujian dan Evaluasi Program Tracking Obyek

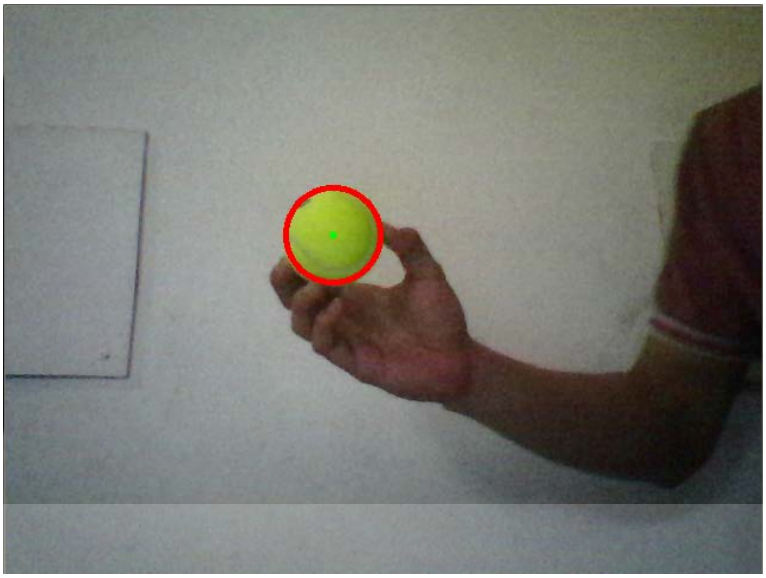
Pengujian ini dilakukan untuk mengetahui terjadinya perubahan nilai koordinat obyek ketika dilakukan pergerakan pada obyek. *Tracking* pada penelitian ini menggunakan *Kalman Filter*. *Tracking* obyek dilakukan setelah mengetahui titik tengah dari obyek. *Tracking* akan menghasilkan titik koordinat dari posisi titik tengah obyek. Pada *Kalman Filter* terdapat dua nilai koordinat yaitu nilai koordinat pada proses estimasi dan pengukuran. Pada pengujian ini akan didapatkan perbandingan pada nilai koordinat dan *error* real, estimasi dan pengukuran. Selain dilakukan perbandingan pada nilai koordinat dan *error*, dilakukan juga pengujian untuk melihat perubahan nilai x dan y pada pergerakan obyek ke atas, bawah, kanan dan kiri. Keberhasilan *tracking* dapat terlihat ketika terjadi perubahan posisi koordinat dari titik tengah obyek, maka nilai koordinat harus berganti ke posisi

koordinat.Pengujian ini dilakukan dengan cara menjalankan program, kemudian dilanjutkan dengan mengarahkan bola ke kamera agar kamera dapat mendeteksi obyek. Setelah obyek dapat terdeteksi oleh kamera, dilakukan pergerakan obyek dengan memindahkan posisi obyek ke atas, bawah, kanan dan kiri.Hasil *tracking* obyek dari program yang telah dibuat ditunjukkan oleh Gambar 3.10 sampai 3.13. Gambar 3.10 dan 3.11 menunjukkan posisi dan nilai koordinat titik tengah obyek sebelum dilakukan *tracking*, sedangkan posisi dan nilai koordinat titik tengah obyek setelah dilakukan *tracking* ditunjukkan oleh Gambar 3.12 dan 3.13.

Gambar 3.10 Frame Asli Sebelum Dilakukan *Tracking*

```
ball position x = 271.000000, y = 327.000000
ball position x_p = 269.368317, y_p = 295.652252
ball position x_c = 270.641602, y_c = 298.738770
```

Gambar 3.11 Nilai Koordinat Sebelum *Tracking*



Gambar 3.12 Frame Asli Setelah Dilakukan *Tracking*

```
ball position x = 267.000000, y = 195.000000
ball position x_p = 266.744781, y_p = 193.785889
ball position x_c = 268.100861, y_c = 193.681961
```

Gambar 3.13 Nilai Koordinat Setelah *Tracking*

3.6.5 Pengujian dan Evaluasi Titik Tengah Obyek Berdasarkan Posisi Pada Matriks *Image*

Pengujian ini dilakukan untuk mengetahui ketepatan koordinat titik tengah obyek dari hasil program yang dibandingkan dengan hasil analisis yang dilakukan pada matriks *image*. Pengujian dilakukan dengan mengambil frame dari hasil tangkapan kamera yang berupa *grayscale image*. Setelah itu, dilakukan pengkonversian dari *grayscale image* yang berupa frame kedalam matriks *image*. Selanjutnya dilakukan pencarian titik tengah pada matriks *image* yang kemudian dicocokkan dengan nilai koordinat titik tengah obyek

hasil dari program. Perpindahan titik tengah obyek dilakukan secara konstan dan terdapat 2 pengujian. Pada pengujian pertama dilakukan pengujian dengan memindahkan posisi titik tengah obyek yang menyebabkan nilai koordinat x berubah sedangkan nilai koordinat y tetap. Pada pengujian kedua dilakukan pengujian dengan memindahkan posisi titik tengah obyek yang menyebabkan nilai koordinat y berubah sedangkan nilai koordinat x tetap.