

SISTEM PENJEJAKAN BOLA MENGGUNAKAN WEBCAM BERBASIS PROSESOR ARM11

Ad'han Yulyandri^{#1}, Fernando Ardilla^{#2}, Setiawardhana^{#3}
[#] Jurusan Teknik Komputer, Politeknik Elektronika Negeri Surabaya
 Jl. Raya ITS, Sukolilo, Surabaya
adhan@student.eepis-its.edu
nando@eepis-its.edu
setia@eepis-its.edu

Abstract

Sistem penjejak bola merupakan sistem yang mampu mengenali dan mendeteksi keberadaan bola. Sistem dilengkapi dengan sebuah webcam sebagai sensor untuk meng-capture objek bola. Sistem dibangun pada arsitektur prosesor MCIMX31 dengan core prosesor ARM11 yang fleksibel dan efisien untuk menerapkan sistem *embedded*. Prosesor ini terdapat pada modul *phyCORE-i.MX31* yang menyediakan beberapa *device* yang sama dengan *device* yang disediakan oleh PC, sehingga pengguna bisa mengembangkan sistem pada modul sama halnya pada PC. Pada sistem akan dibuat suatu aplikasi yang menampilkan hasil pengolahan citra dari objek bola yang tertangkap webcam dan akan ditampilkan pada layar LCD modul. Agar aplikasi dapat berjalan pada modul, maka dibutuhkan sistem operasi yang kompatibel dengan hardware pada modul. Untuk itu diperlukan optimasi sistem operasi pada modul agar aplikasi dapat berjalan dengan semestinya. Sistem penjejak bola dapat berjalan dengan baik pada arsitektur *phyCORE-i.MX31* dengan sistem operasi *linux embedded* yang telah dibuat. Sistem dapat mendeteksi bola dengan prosentase keberhasilan 83.3 % untuk kondisi cahaya yang stabil. Sistem penjejak bola tidak stabil jika kondisi cahaya tidak sesuai, dengan prosentase keberhasilan 32.5 % pada malam hari dan 35 % pada siang hari.

Keywords : ARM11, *phyCORE-i.MX31*, *linux embedded*, sistem penjejak bola..

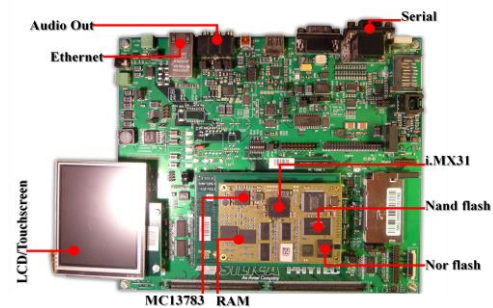
1 Pendahuluan

Salah satu keluarga prosesor ARM adalah ARM11 dengan inti prosesor ARM1136JF-S. Banyak sekali vendor khususnya pada bidang prosesor yang menggunakan *Central Processing Unit* berbasis prosesor ARM11, salah satunya adalah MCIMX31 produksi *Freescale Semiconductor*. MCIMX31 merupakan prosesor multimedia yang memanfaatkan keunggulan fitur dari keluarga prosesor ARM11. MCIMX31 ini terdapat pada *Rapid Development Kit phyCORE-i.MX31* yang akan digunakan pada penelitian ini.

Pada penelitian ini akan dibangun sebuah sistem penjejak bola pada modul *phyCORE-i.MX31* yang lebih fleksibel dan efisien untuk menerapkan

sistem *embedded*. *phyCORE-i.MX31* menyediakan beberapa *device* yang sama dengan PC (*Personal Computer*), sehingga pengguna bisa mengembangkan sistem pada *phyCORE-i.MX31* sama halnya pada PC. Webcam diakses melalui USB modul *phyCORE-i.MX31*, hasil *capture webcam* akan diproses pada *phyCORE-i.MX31* menggunakan *library* OpenCV-1.0 dan hasilnya akan ditampilkan pada LCD modul menggunakan fitur GUI (*Graphic User Interface*) dari *QtopiaCore*.

2 *phyCORE-i.MX31 Development Kit*



Gambar 1. *phyCORE-i.MX31*

Gambar 1 menunjukkan arsitektur *development kit* PHYTEC *phyCORE-i.MX31 board*. Pada *phyCORE-i.MX31* memiliki karakteristik prosesor yang berbasis arsitektur ARM11 sehingga untuk memasukkan sebuah sistem operasi diharuskan melewati beberapa tahap dikarenakan pada setiap arsitektur prosesor yang berbeda memiliki perlakuan yang berbeda pada setiap sistem operasi.

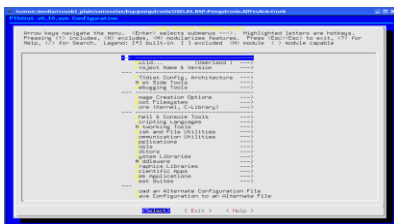
Ada beberapa tahapan supaya sistem operasi dapat digunakan oleh *user*, langkah yang pertama dilakukan adalah melakukan *download bootloader, filesystem, kernel* pada *phyCORE-i.MX31*. Proses tersebut dilakukan supaya *phyCORE-i.MX31* dapat melakukan *booting* dan dapat memasuki sistem operasi yang telah dibangun. Secara *default phyCORE-i.MX31* akan melakukan *booting* melalui media *serial*. Jika kita ingin melakukan *booting* menggunakan media *SD Card* atau media lainnya maka terlebih dahulu diharuskan melakukan konfigurasi pada *environment setting*. Sehingga *phyCORE-i.MX31* dapat melakukan *booting* dari media yang berbeda.

3 BSP (Board Support Package)

BSP merupakan seperangkat *tools* atau paket *software*, *source*, maupun *library-library* yang memungkinkan untuk ditempatkan atau dimasukkan pada fungsi sebuah *development board*. BSP umumnya digunakan untuk membangun elemen-elemen seperti : *root-filesystem (rootfs)*, *kernel linux*, *bootloader*, atau sebuah *toolchain* untuk kebutuhan *cross compiling*.

4 PTXdist

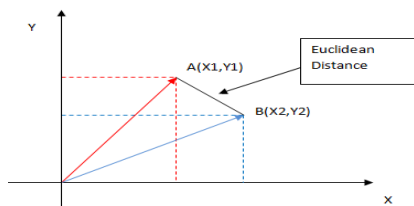
PTXdist adalah sebuah *tools* atau *software* untuk membangun sistem (*build system*) yang menyediakan penciptaan ulang sebuah sistem linux *embedded* khusus yang telah dibuat, umumnya *PTXdist* digunakan untuk mengontrol secara Kconfig GUI pembuatan *bootloader*, *kernel linux embedded*, *filesystem*, serta *toolchain* yang telah disediakan oleh BSP. *PTXdist* memungkinkan pengguna untuk membuat distribusi linux sendiri pada target *board (phyCORE-i.MX31)*. Hasil akhir dari *PTXdist* adalah : file *image linux kernel*, *image filesystem (library-library, utilitas, modul kernel, dll)*, set paket *.ipk*, direktori lengkap dengan isi dari *filesystem* target, dll. Dari sudut pandang programmer, *PTXdist* merupakan seperangkat *makefiles*, *script bash*, *script config* dan *patch*, *source codes* dari paket-paket akan didownload secara otomatis dari internet atau pengguna bisa memasukkan paket *source* sendiri yang telah di download secara manual ke direktori *src* dari *PTXdist*.



Gambar 2. Menu konfigurasi PTXdist

5 Euclidean Distance

Euclidean vector atau sering disebut dengan vektor adalah obyek geometri yang memiliki panjang (*magnitude*) dan arah (*direction*). Sedangkan ruang vektor adalah sebuah struktur matematika yang dibentuk oleh sekumpulan vektor. Vektor-vektor tersebut dapat ditambahkan, dikalikan dengan bilangan real dan lain-lain.



Gambar 3. Ruang vektor dua dimensi

Gambar 4 adalah contoh dari ruang vektor dua dimensi, pada ruang vektor tersebut terdapat 2 vektor yaitu vektor A dan vektor B. Untuk menghitung jarak antara vektor A dan vektor B digunakan persamaan *euclidean distance*.

Berikut merupakan penyelesaian dalam menghitung jarak antara vektor A dan vektor B. Panjang vektor A dan B dapat didefinisikan sebagai berikut:

$$\| \vec{A} \| = \sqrt{X1^2 + Y1^2}$$

$$\| \vec{B} \| = \sqrt{X2^2 + Y2^2}$$

Dengan demikian, untuk menghitung jarak antara kedua vektor tersebut menggunakan persamaan sebagai berikut :

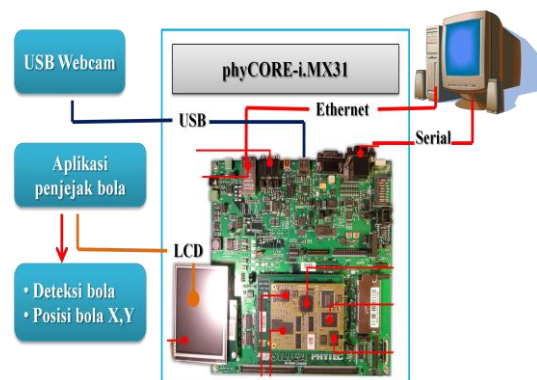
$$d(\vec{A}, \vec{B}) = \sqrt{(X1 - X2)^2 + (Y1 - Y2)^2}$$

Sedangkan untuk n dimensi ruang vektor, jarak *euclidean distance* ditentukan dengan menggunakan persamaan sebagai berikut :

$$d(\vec{u}, \vec{v}) = \sqrt{(u_1 - v_1)^2 + (u_2 - v_2)^2 + \dots + (u_n - v_n)^2}$$

Nilai *euclidean distance* merupakan nilai kemiripan citra digital. Semakin dekat (mendekati nilai 0) semakin mirip citra digital tersebut. Pada tugas akhir ini metode ini digunakan untuk mengubah gambar dengan format warna RGB menjadi gambar biner (*thresholding*). Dengan cara mencari nilai RGB terdekat sesuai warna target yang diinginkan dan mengubahnya ke gambar biner, pada penelitian ini adalah bola berwarna orange.

6 Rancangan Umum



Gambar 4. Blok diagram sistem

Gambar 5 menunjukkan blok diagram sistem secara keseluruhan, komputer *host* dan target (*phyCORE-i.MX31*) terhubung secara jaringan dan serial. Komunikasi serial digunakan untuk mengontrol seluruh proses pada *phyCORE-i.MX31* melalui program konsol serial (*microcom*). Sedangkan komunikasi secara jaringan digunakan

untuk keperluan transfer data, mulai dari perbaruan *kernel* maupun *filesystem*, transfer data dari *filesystem host* ke *filesystem target* atau sebaliknya. Aplikasi penjejak bola dibuat pada komputer *host* dan dijalankan pada target. *USB webcam* akan meng-capture bola dan hasilnya ditampilkan pada LCD dengan aplikasi penjejak bola yang dibuat dengan *OpenCV-1.0* dan *QtopiaCore* sebagai pembangun GUI-nya. Pada aplikasi, bola akan dikenali dan dideteksi, serta dicari posisinya yaitu berupa titik tengah dari bola (piksel x,y)

7 Membangun Kernel

Kernel disesuaikan dengan *device* yang akan dipakai untuk sistem penjejak bola yang akan dibuat. *Device* utama yang harus ditambahkan pada *kernel* untuk sistem penjejak bola adalah sebuah *USB webcam*. Tentu saja sebuah *device* agar bisa dikenali oleh suatu sistem operasi, *device* harus memiliki *driver*. *Driver* harus kompatibel dengan *hardware* yang digunakan. Segala sesuatu yang berhubungan dengan *hardware* dan sistem operasi diatur oleh sebuah *kernel*. *Kernel* default yang disediakan oleh *phyCORE-i.MX31* tidak memiliki konfigurasi *driver* untuk sebuah *USB webcam*, jadi pengguna harus membuat *kernel* sendiri dan mengaktifkan *driver* tersebut. Salah satu *driver* yang paling efektif untuk mengenali *device* berupa *USB webcam* pada sistem operasi linux adalah *UVC (USB Video Class)*. *UVC* mampu bekerja pada bermacam-macam keluarga prosesor, salah satunya adalah *ARM11* yang digunakan pada penelitian ini. Untuk membangun *kernel* tersebut, penulis menggunakan fitur dari *PTXdist* dan paket dari *OSELAS.BSP()*.

Install *PTXdist* :

```
~ tar -zxf ptxdist-1.0.1.tgz
~ tar -zxf ptxdist-1.0.1-patches.tgz
~ cd ptxdist-1.0.1
~/ptxdist-1.0.1> ./configure
~/ptxdist-1.0.1> sudo make
~/ptxdist-1.0.1> make install
```

Ekstrak *OSELAS.BSP()* :

```
~# tar -zxf OSELAS.BSP-Phytec-phyCORE-
i.MX31-PD09.1.0.tar.gz
```

Konfigurasi *kernel* :

```
~> cd OSELAS.BSP-Phytec-phyCORE-i.MX31-
PD09.1.0
~/OSELAS.BSP-Phytec-phyCORE-i.MX31-
PD09.1.0> ptxdist kernelconfig
```

Build *kernel* :

```
~/OSELAS.BSP-Phytec-phyCORE-i.MX31-
PD09.1.0# ptxdist go
```

Update *kernel* ke *phyCORE-i.MX31* :

```
> update_kernel nor uImage
```

8 Membangun Root-filesystem

Root-filesystem merupakan satu set folder yang berisi *executable*, *library-library*, dan berbagai macam konfigurasi file, seperti halnya sistem operasi pada umumnya. *Filesystem* digunakan untuk mengorganisasi, menyimpan, memperoleh dan manajemen informasi pada media penyimpanan permanen pada *phyCORE-i.MX31*.

Konfigurasi *library-library filesystem* :

```
~> cd local/OSELAS.BSP-Phytec-phyCORE-
i.MX31-PD09.1.0
~/OSELAS.BSP-Phytec-phyCORE-i.MX31-
PD09.1.0> ptxdist menuconfig
```

Build *filesystem* :

```
~/OSELAS.BSP-Phytec-phyCORE-i.MX31-
PD09.1.0# ptxdist go
```

Update *filesystem* ke *phyCORE-i.MX31* :

```
> update_rootfs nor root.jffs2
```

9 Instalasi QtopiaCore

QtopiaCore merupakan sebuah *software* untuk membuat aplikasi pada *mobile phone*. *QtopiaCore* menyediakan fitur untuk membuat aplikasi berbasis GUI. Karena beberapa *library-library* pendukung GUI pada *phyCORE-i.MX31* tidak dibangun pada *filesystem* karena minimnya kapasitas media penyimpanan, maka untuk membuat aplikasi penjejak bola menggunakan fitur GUI yang disediakan oleh *QtopiaCore*. Untuk instalasi *QtopiaCore* sebagai berikut :

Ekstrak *QtopiaCore* :

```
~# tar xzf QtopiaCore.tgz
```

Install *library QtopiaCore* pada target :

```
~# cd QtopiaCore
~/QtopiaCore# ./install-target.sh 192.168.3.130
```

Install *library QtopiaCore* pada *host* :

```
~# cp -a /home/andrey/QtopiaCore/ Trolltech
/usr/local
~# cp -a /home/andrey/QtopiaCore/ tslib-i.MX31
/usr/local
~/QtopiaCore# ./install-host.sh
```

Patching library :

```
PATH=/usr/local/Trolltech/QtopiaCore-4.3.3-
arm/bin:$PATH
export PATH
```

10 Instalasi OpenCV-1.0

OpenCV-1.0 digunakan untuk mengolah citra digital pada sistem penjejak bola yang akan dibuat. Agar fungsi-fungsi OpenCV dapat berjalan dengan baik pada arsitektur ARM11 phyCORE-i.MX31, maka dibutuhkan konfigurasi khusus saat kompilasi source OpenCV untuk menjadikannya library-library.

Ekstrak OpenCV-1.0 :

```
~# tar -zxf opencv-1.0.0.tar.gz
```

Instalasi OpenCV-1.0 :

```
~# cd /home/andrey/opencv-1.0.0
~/opencv-1.0.0# ./configure \
--prefix=/usr/local/opencv-1-0.0 \
--host=arm-1136jfs-linux-gnueabi \
--target=arm-1136jfs-linux-gnueabi \
--without-swig \
--without-python \
--without-xine \
--without-ffmpeg \
--without-1394libs \
--without-quicktime \
--without-carbon \
--with-gtk \
CXX=arm-1136jfs-linux-gnueabi-g++ \
CPP=arm-1136jfs-linux-gnueabi-cpp \
CC=arm-1136jfs-linux-gnueabi-gcc

~/opencv-1.0.0# make
~/opencv-1.0.0# make install
```

Patching library OpenCV-1.0 pada host :

```
LD_LIBRARY_PATH=/usr/local/opencv-1-
0.0/lib:$LD_LIBRARY_PATH
LD_LIBRARY_PATH

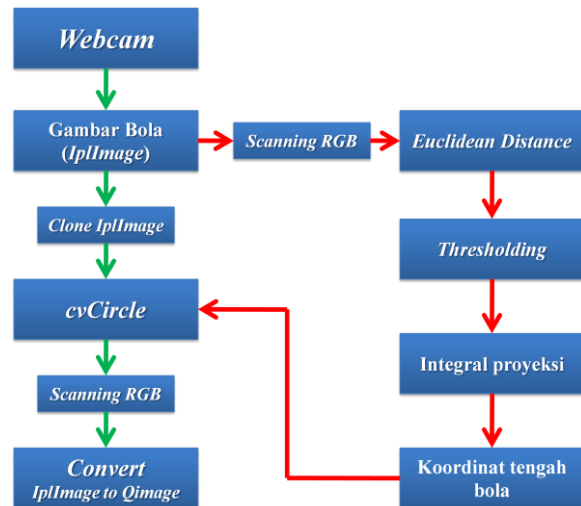
PKG_CONFIG_PATH=/usr/local/opencv-1-
0.0/lib/pkgconfig:$PKG_CONFIG_PATH
export PKG_CONFIG_PATH
```

Copy library OpenCV-1.0 ke filesystem target /media/nand, dan buat path :

```
export
LD_LIBRARY_PATH=/media/nand/opencv:$
LD_LIBRARY_PATH
```

11 Pembuatan Sistem

Untuk membuat software atau aplikasi penjejak bola, penulis menggunakan fitur dari OpenCV-1.0 untuk pengolahan citra digital dan QtopiaCore untuk GUI (Graphic User Interface). Bola yang digunakan adalah bola dengan warna orange. Bola dikenali berdasarkan warna dan bentuknya. Berdasarkan blok diagram perancangan software, pertama-tama dilakukan scanning nilai RGB per piksel dari gambar yang diambil oleh webcam yang telah dideklarasikan dengan format gambar OpenCV yaitu IplImage. Gambar 5 adalah blok diagram kerja aplikasi sistem penjejak bola pada phyCORE-i.MX31 menggunakan library-library OpenCV-1.0 dan QtopiaCore.



Gambar 5. Rancangan sistem penjejak bola

Dari proses scanning RGB, maka akan didapatkan nilai-nilai RGB per piksel dari gambar yang diambil oleh webcam. Selanjutnya dari nilai RGB yang didapat, dilakukan penyeleksian warna orange menggunakan Euclidean distance :

$$d = \sqrt{(R - r)^2 + (G - g)^2 + (B - b)^2}$$

d = nilai Euclidean distance

R = nilai merah dari gambar yang diambil webcam

G = nilai hijau dari gambar yang diambil webcam

B = nilai biru dari gambar yang diambil webcam

r = nilai rata-rata merah

g = nilai rata-rata hijau

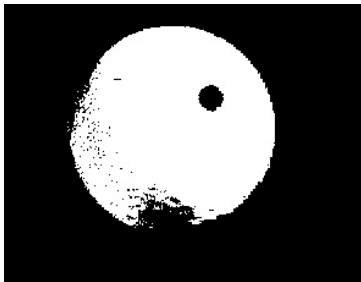
b = nilai rata-rata biru

--- } orange

Warna orange ditentukan dengan mengambil nilai masing-masing R, G, dan B pada sebuah gambar bola berwarna orange, kemudian diambil rata-rata nilainya dari selisih nilai RGB tertinggi dan terendah kemudian dibagi dua. Nilai RGB ini kemudian dimasukkan ke dalam persamaan Euclidean distance

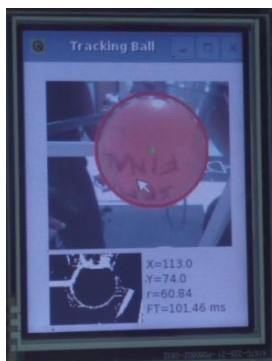
sebagai pengurang untuk nilai RGB dari gambar yang diambil oleh webcam. Selisih dari nilai RGB gambar dari webcam dan RGB dengan nilai orange yang telah ditentukan masing-masing dikuadratkan dan hasilnya dijumlah, kemudian dengan akar pangkat dua, maka nilai Euclidean Distance (d) atau nilai yang mendekati orange akan ditemukan.

Dari nilai d yang didapat, kemudian dibuat kondisi untuk membuat gambar biner (*thresholding*), misalnya jika $d < 90$ adalah warna *orange*, maka nilai R , G , dan B dirubah nilainya menjadi masing-masing 255 dan selain warna *orange* atau $d \geq 90$ maka nilai R , G , dan B masing-masing adalah 0, maka gambar bola (*orange*) akan menjadi putih, dan latar belakangnya hitam, seperti pada gambar 3.2.



Gambar 6. Hasil *thresholding* bola berwarna *orange*

Dari gambar hasil *thresholding* tersebut kemudian dilakukan deteksi bentuk lingkaran menggunakan salah satu fungsi *library* OpenCV yaitu *cvHoughCircles*. Fungsi ini akan mendeteksi lingkaran berdasarkan gambar hasil *thresholding* tersebut dari pendeteksian tepi obyek. Dari fungsi ini juga dapat ditemukan titik tengah dari obyek bola tersebut berdasarkan piksel dari obyek yang dimaksud. Tidak hanya titik tengah, namun radius lingkaran juga dapat ditemukan dengan fungsi ini. Nilai tengah dan radius tersebut kemudian dijadikan parameter untuk membuat sebuah garis melingkar untuk melingkari objek bola menggunakan fungsi dari OpenCV yaitu *cvCircles*. Gambar 3.26 merupakan hasil deteksi bola *orange* menggunakan *cvHoughCircles*.



Gambar 7. *cvHoughCircles* dan *cvCircles*

12 Pengujian Sistem

Uji coba awal sistem dilakukan dengan memindah-mindahkan posisi bola untuk mengetahui apakah sistem dapat menjalankan fungsinya dengan baik yaitu mendeteksi keberadaan bola. Jika bola terdeteksi maka bola akan dilingkari dan sistem akan menampilkan koordinat tengah bola berdasarkan piksel.



Gambar 8. Sistem penjejak bola phyCORE-i.MX31



Gambar 9. Hasil pengujian awal (bola terdeteksi dan dapat menampilkan titik tengah bola)

Selanjutnya pengujian kestabilan sistem dengan mengubah-ubah jarak antara kamera dengan bola, serta tingkat pencahayaan yang diberikan. Pengujian dilakukan pada malam dan siang hari dengan mengubah-ubah kombinasi pencahayaan dari lampu neon pada ruangan tempat pengujian. Hal ini dilakukan untuk mengetahui jarak maksimum dan minimum, serta tingkat pencahayaan yang sesuai untuk sistem penjejakan bola.



Gambar 9. Kondisi pencahayaan tempat pengujian

Tabel 4. Prosentase keberhasilan pendeteksi bola

Malam hari		Siang hari	
Jarak (cm)	Prosentase (%)	Jarak (cm)	Prosentase (%)
2	25	2	0
10	62.5	10	75
30	25	30	25
40	37.5	40	37.5
50	12.5	50	37.5
Jumlah	162.5	Jumlah	175
Rata-rata (%)	$162.5/5 = 32.5$	Rata-rata (%)	$175/5 = 35$

Jika dihitung secara rata-rata, maka tingkat keberhasilan sistem dalam mendeteksi bola dengan intensitas cahaya yang berbeda-beda atau tidak stabil, pada malam hari adalah 32.5% dan siang hari tingkat keberhasilan sistem adalah 35 %. Jika sistem mendeteksi bola dengan intensitas cahaya yang stabil atau ditentukan, maka keberhasilannya adalah 83.3 %, dapat dilihat dari tabel 4.4, dari enam kali pengujian, hanya satu kali bola tidak dapat dideteksi oleh system, yaitu pada jarak 50 cm, maka prosentasenya adalah $(5/6) \times 100\% = 83.3\%$. Jadi sistem penjejak bola dapat berjalan dengan baik pada arsitektur prosesor ARM11 pada modul *phyCORE-i.MX31*.

13 Kesimpulan

1. *USB Webcam* dapat dikenali oleh *phyCORE-i.MX31* dengan mengaktifkan *UVC (USB Video Class)* pada kernel *linux-2.6.30-rc7*
2. Konfigurasi *kernel* default untuk *phyCORE-i.MX31* adalah *PCM037_defconfig*
3. *OpenCV-1.0* dapat berjalan dengan baik pada *phyCORE-i.MX31*, namun tidak mendukung fitur *GUI (Graphic User Interface)*
4. Terbatasnya memori yang dimiliki modul menyebabkan beberapa *library* untuk *GUI* tidak dapat ditambahkan pada *phyCORE-i.mx31*.
5. Sistem penjejak bola dapat berjalan dengan baik pada *phyCORE-i.MX31* menggunakan fitur *GUI QtopiaCore*
6. Sistem penjejak bola pada *phyCORE-i.MX31* mampu mendeteksi bola dengan jarak kamera lebih dari 2 cm dan kurang dari 50 cm dengan pencahayaan yang sesuai.
7. Dari pengujian sistem penjejak bola yang dilakukan, sistem dapat mendeteksi bola dengan prosentase keberhasilan 83.3 % untuk kondisi cahaya yang stabil.
8. Sistem penjejak bola tidak stabil jika kondisi cahaya tidak sesuai, dengan prosentase keberhasilan 32.5 % pada malam hari dan 35 % pada siang hari.

DAFTAR PUSTAKA

- [1] Dokumentasi L-700e_2 "Hardware Manual *phyCORE-i.MX31*" pada <http://www.phytec.com>
- [2] Dokumentasi L-703e_4 "QuickStart Instructions Linux-Kit *phyCORE-i.MX31*" pada <http://www.phytec.com>
- [3] Dokumentasi OSELAS.BSP() "QuickStart Manual OSELAS.BSP() *phytec phyCORE-i.MX31*" pada <http://www.pengutronix.com>
- [4] Gilles Curchod, "Multimedia Application Using *i.MX31*", heig-vd, REDS, Diploma Project, 2008
- [5] Doug Abbott, "Linux for Embedded and Real-time Application", Newnes, U.S Amerika, 2003
- [6] P. Raghavan, Amol Lad, Sriram Neelakandan, "Embedded Linux System Design and Development", Auerbach Publications, U.S Amerika, 2006
- [8] Intel Reference Manual, "Open Source Computer Vision Library", U.S Amerika, 2001
- [9] Dokumentasi TrollTech "Reference Manual *QtopiaCore-4.1*", pada <http://www.doc.trolltech.com>
- [10] Liwei Wang, Yan Zhang, Jufu Feng, "On the Euclidean Distance of Images", School of Electronics Engineering and Computer Sciences, Peking University Beijing, China
- [11] Andre Treptow, Andreas Masselli, Andreas Zell, "Real-Time Object Tracking for Soccer-Robots without Color Information", University