

## BAB II

### LANDASAN TEORI

#### 2.1 Robotino

Robotino adalah robot buatan Festo Didactic yang digunakan untuk edukasi dan penelitian serta kompetisi robot. Robotino memiliki fitur sistem gerak menggunakan *omni-directional drive*, *bumps sensor*, *infrared distance sensor*, dan *usb webcam*. Robotino didesain modular, sehingga dapat dengan mudah ditambahkan berbagai aksesoris pelengkap, seperti sensor *laser scanner*, *gyroscope*, dan *positioning system Northstar* dalam ruangan. (ROS, 2010).



Gambar 2.1 Robotino

Gambar 2.1 merupakan bentuk fisik dari Robotino. Robotino dapat bergerak maju, mundur dan menyamping ke segala arah, serta berputar di tempat, dengan menggunakan tiga roda. Robot ini dapat diintegrasikan dan digunakan

sebagai pilihan teknologi, misalnya untuk teknologi penggerak listrik, sensor, teknologi kontrol, pengolahan citra dan teknik pemrograman (Karras, 2009).

Robotino memiliki spesifikasi perangkat keras sebagai berikut : (Weber, dkk. 2010)

1. 1 buah *chasis*
2. 3 buah *drive unit*
3. 2 buah baterai
4. 1 buah *command bridge*
5. 1 buah kamera
6. 1 buah soket konektor VGA
7. 2 buah *USB ports*
8. 1 buah *ethernet*
9. 9 buah *sensor infrared*
10. 3 buah *incremental encoder*
11. 1 buah *anti-coollision sensor*
12. 1 buah *wireless LAN access point*
13. 1 buah *compact flash card*
14. 1 paket *I/O interface* tambahan



Gambar 2.2 Bagian-bagian Robotino (Weber, dkk. 2010).

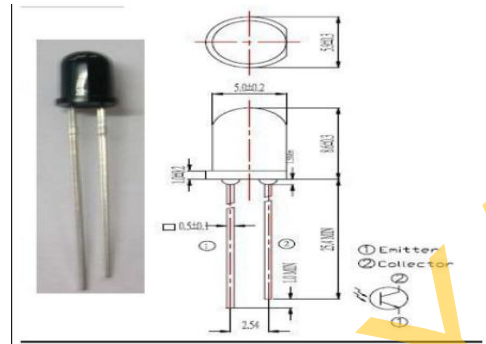
## 2.2 DF ROBOT (Flame Sensor)

Sensor dari DF ROBOT berikut ini dapat mendeteksi nyala api dengan panjang gelombang 760 ~ 1100 nm, sensor ini dapat mendeteksi suhu panas berkisar 25 C – 85 C. Sensitivitas dari produk ini sudah teruji dengan baik melalui beberapa percobaan yang telah dilewati membuat penulis memilih sensor dari DF ROBOT ini sebagai sensor yang akan diintegrasikan dengan robotino sebagai salah satu sarana untuk mendeteksi suhu dari api. Sensor ini dapat mendeteksi api dari jarak 100 cm dengan keluaran tegangan sebesar 0,5v, dan pada jarak 20 cm dengan objek sensor ini dapat mengeluarkan keluaran tegangan sebesar 5v.

Berikut adalah modul dan ukuran fisik dari DF ROBOT *flame* sensor :



(a)



(b)

Gambar 2.3 (a) Modul dan (b) bentuk dan ukuran fisik DF ROBOT flame sensor (DFRobot, 2011 ).

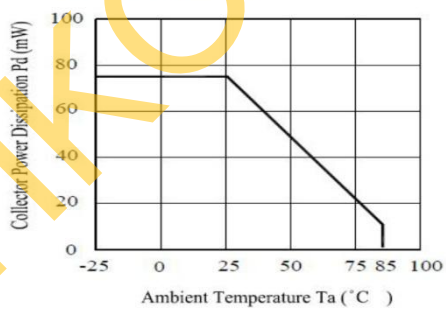
Tabel 2.2 Karakteristik Optikal – Elektro dari DF ROBOT flame sensor.

Parameter	Symbol	rating	units
Collector-Emitter Voltage	$V_{CE0}$	30	V
Emitter-Collector-Voltage	$V_{ECO}$	5	V
Collector Current	$I_c$	20	mA
Operating Temperature	$T_{opr}$	-25~+85°C	°C
Storage Temperature	$T_{stg}$	-40+85°C	°C
Lead Soldering Temperature	$T_{sol}$	260	°C
Power Dissipation at (or below) 25°C FreeAir Temperature	$P_c$	75	mW

Tabel 2.3 Satuan Simbol dan Unit pada DF ROBOT *flame* sensor.

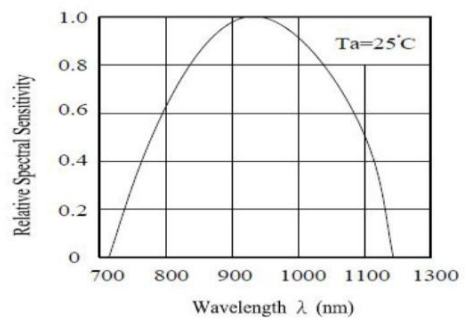
Parameter	Symbol	Condition	Min.	Typ.	Max.	Units
Collector-Emitter Breakdown Voltage	BV <sub>CEO</sub>	I <sub>c</sub> =100μA E <sub>e</sub> =0mW/C m <sup>2</sup>	30	---	---	V
Emitter-Collector Saturation Voltage	BV <sub>ECO</sub>	I <sub>E</sub> =100μA E <sub>e</sub> =0mW/C m <sup>2</sup>	5	---	---	V
Collector-Emitter Saturation Voltage	V <sub>CE(sat)</sub>	I <sub>c</sub> =2mA E <sub>e</sub> =1mW/C m <sup>2</sup>	---	---	0.4	V
Rise Time	t <sub>r</sub>	V <sub>CE</sub> =5V I <sub>c</sub> =1mA	---	15	---	μS
Fall Time	t <sub>f</sub>	R <sub>L</sub> =1000Ω	---	15	---	
Collector Dark Current	I <sub>CEO</sub>	E <sub>e</sub> =0mW/C m <sup>2</sup> V <sub>CE</sub> =20V	---	---	100	nA
On State Collector Current	I <sub>c(on)</sub>	E <sub>e</sub> =1mW/ C m <sup>2</sup> V <sub>CE</sub> =5V	1.77	---	7.07	mA
Wavelength of Peak Sensitivity	λ <sub>P</sub>	---	---	940	---	nm
Rang of Spectral Bandwidth	λ <sub>0.5</sub>	---	---	760-1100	---	nm

Fig.1 Collector Power Dissipation vs. Ambient Temperature



(a)

Fig.2 Spectral Sensitivity



(b)

Gambar 2.4 (a) Perbandingan kolektor power dengan *temperature* dan (b) panjang gelombang terhadap sensitivitas gelombang. (DF ROBOT.2011)

### 2.3 Webcam

*Webcam* adalah kamera video sederhana berukuran relatif kecil. sering digunakan untuk konferensi video jarak jauh atau sebagai kamera pemantau.

*Webcam* pada umumnya tidak membutuhkan kaset atau tempat penyimpanan data, data hasil perekaman yang didapat langsung ditransfer ke komputer.

(*Webcam, Inc.*2009).



Gambar 2.5 *Webcam*

(*Webcam, Inc.*2009)

Robotino dilengkapi dengan *webcam* yang dilengkapi oleh JPEG *compression*, Gambar hasil kompresi tersebut dapat di transmisikan ke PC via WLAN untuk evaluasi, pada ulasan kali ini *webcam* akan digunakan untuk mendeteksi lokasi dari api. *Webcam* akan melakukan proses pengolahan citra dengan mengolah RGB dari *image* yang ditangkap dan melakukan kalkulasi dengan warna api. (Weber, dkk.2010)

## 2.4 Pengolahan Citra

Pengolahan citra adalah salah satu cabang dari ilmu informatika. Pengolahan citra berkuat pada usaha untuk melakukan transformasi suatu citra/gambar menjadi citra lain dengan menggunakan teknik tertentu. (Munir, Rinaldi. 2004)

### 2.4.1 Citra

Citra adalah gambar dua dimensi yang dihasilkan dari gambar analog dua dimensi yang secara terus menerus menjadi gambar diskrit melalui proses *sampling*. Gambar analog dibagi menjadi  $x$  baris dan  $y$  kolom sehingga menjadi gambar diskrit. Persilangan antara baris dan kolom tertentu disebut dengan piksel  $[n,m]$ . (Munir, Rinaldi.2004)

### 2.4.2 Pengambangan (*Thresholding*)

Konversi dari citra hitam-putih ke citra biner dilakukan dengan operasi pengambangan (*thresholding*). Operasi pengambangan mengelompokkan nilai derajat keabuan setiap piksel kedalam 2 kelas, hitam dan putih.

Pengkonversian citra hitam-putih (*greyscale*) menjadi citra biner dilakukan untuk alasan-alasan sebagai berikut :

1. Untuk mengidentifikasi keberadaan objek, yang direpresentasikan sebagai daerah didalam citra.
2. Untuk lebih memfokuskan pada analisis bentuk morfologi, yang dalam hal ini intensitas piksel tidak terlalu penting dibandingkan bentuknya. Setelah

objek dipisahkan dari latar belakangnya, properti geometri dan morfologi/topologi objek dapat dihitung dari citra biner.

3. Mengkonversi citra yang telah di ditingkatkan kualitas tepinya (*edge enhancement*) ke penggambaran garis-garis tepi. Hal ini perlu dilakukan untuk membedakan tepi yang kuat yang berkoresponden dengan batas-batas objek dengan tepi lemah yang berkoresponden dengan perubahan *illumination*, bayangan, dll.

#### 2.4.3 Color Filtering

*Color filtering* adalah suatu teknik pengolahan citra yang yang dipakai untuk memanipulasi suatu citra berdasarkan warna spesifik. Cara kerjanya adalah dengan membandingkan komponen warna setiap *pixel* citra dengan warna spesifik. Apabila warnanya sesuai dengan warna spesifik komponen warna *pixel* tersebut dibiarkan saja. Namun, bila warnanya tidak sesuai dengan warna spesifik maka komponen warna *pixel* tersebut diubah menjadi warna *background*, biasanya menjadi warna hitam.

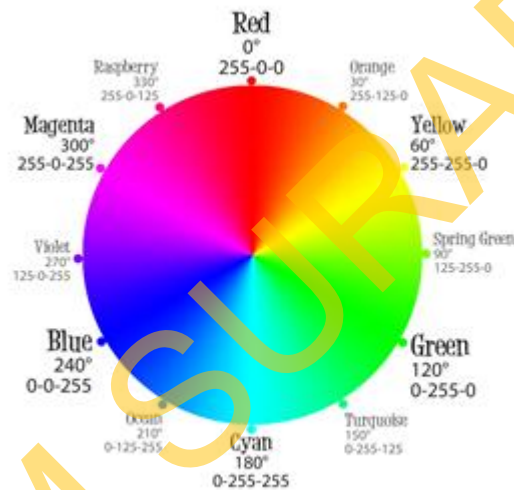
Warna yang digunakan dalam *color filtering* dapat direpresentasikan dalam berbagai ruang warna. Ada beberapa ruang warna yang dikenal, antara lain RGB (*Red, Green, Blue*), HSV (*Hue, Saturation, Value*), YCbCr, dsb. HSV merupakan ruang warna yang sangat cocok untuk mengidentifikasi warna-warna dasar, dimana warna dasar ini digunakan dalam penelitian sebagai warna identifikasi robot. Selain itu, HSV menoleransi terhadap perubahan intensitas cahaya. Inilah yang menjadi keunggulan HSV dibandingkan dengan ruang warna lainnya.



## 2.4.4 Color Space

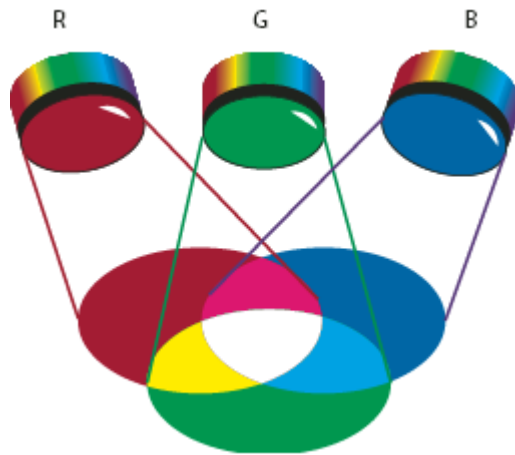
### 1. RGB

Model warna RGB adalah model warna berdasarkan konsep penambahan kuat cahaya primer yaitu *Red*, *Green* dan *Blue*. Dalam suatu ruang yang sama sekali tidak ada cahaya, maka ruangan tersebut adalah gelap total. Tidak ada signal gelombang cahaya yang diserap oleh mata kita atau RGB (0,0,0). Ruang warna RGB.dapat dilihat pada Gambar 2.6.



Gambar 2.6 RGB Colorwheels (Syuhadi,2010)

Apabila kita menambahkan cahaya merah pada ruangan tersebut, maka ruangan akan berubah warna menjadi merah misalnya RGB (255,0,0), semua benda dalam ruangan tersebut hanya dapat terlihat berwarna merah. Demikian apabila cahaya kita ganti dengan hijau atau biru (Permana,2010). Pencerahan serta pengaruh intensitas cahaya dalam ruang warna RGB dapat dilihat pada Gambar 2.7.



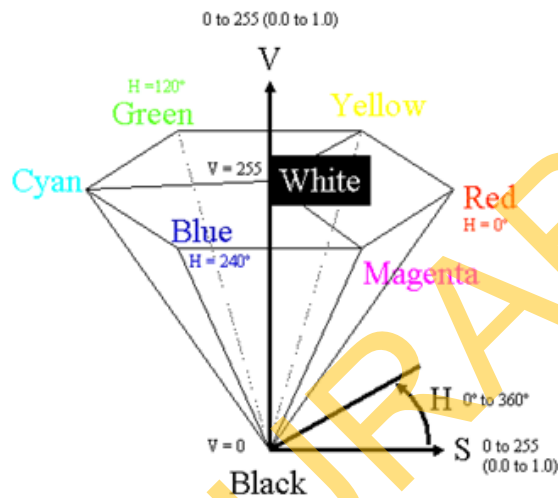
Gambar 2.7 RGB *Lightness* (Permana,2010)

## 2. HSV

Model HSV (*Hue Saturation Value*) menunjukkan ruang warna dalam bentuk tiga komponen utama, yaitu *hue*, *saturation* dan *value* (atau disebut juga *brightness*). *Hue* adalah sudut dari 0 sampai 360 derajat. Biasanya 0 adalah merah, 60 derajat adalah kuning, 120 derajat adalah hijau, 180 derajat adalah cyan, 240 derajat adalah biru dan 300 derajat adalah magenta.

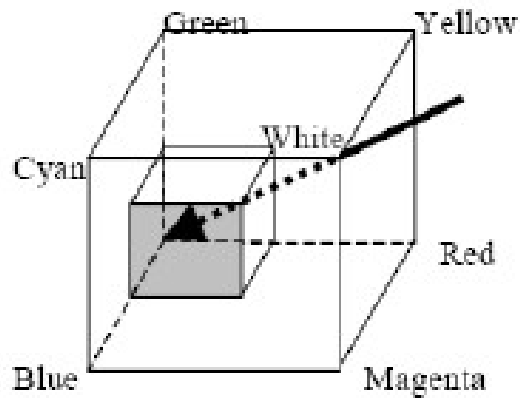
*Hue* menunjukkan jenis warna (seperti merah, biru atau kuning) atau corak warna, yaitu tempat warna tersebut ditemukan dalam spektrum warna. Merah, kuning dan ungu adalah kata-kata yang menunjukkan *hue*. Saturasi suatu warna adalah ukuran seberapa besar kemurnian dari warna tersebut. Sebagai contoh, suatu warna yang semuanya merah tanpa putih adalah saturasi penuh. Jika ditambahkan putih ke merah, hasilnya lebih berwarna-warni dan warna bergeser dari merah ke merah muda. *Hue* masih tetap merah tetapi nilai saturasinya berkurang. Saturasi biasanya bernilai 0 sampai 1 (atau 0% sampai 100%) dan menunjukkan nilai keabu-abuan warna dimana 0 menunjukkan abu-abu dan 1 menunjukkan warna primer murni. Komponen ketiga dari HSV adalah *value* atau

disebut juga intensitas, yaitu ukuran seberapa besar kecerahan suatu warna atau seberapa besar cahaya datang dari suatu warna. Nilai *value* dari 0% sampai 100%. Pemetaan ruang warna HSV dalam bentuk *Hexcone* dapat dilihat pada Gambar 2.8.



Gambar 2.8 HSV *Hexcone* (Permana,2010)

Suatu warna dengan nilai *value* 100% akan tampak sangat cerah dan suatu warna dengan nilai *value* 0 akan tampak sangat gelap. Sebagai contoh, jika *hue* adalah merah dan *value* bernilai tinggi maka warna akan terlihat cerah tetapi ketika nilai *value* bernilai rendah maka warna tersebut akan terlihat gelap. *Color space* dari HSV dapat dilihat pada Gambar 2.9.



Gambar 2.9 HSV Box Colorspace. (Permana,2010)

Dengan sistem koordinat HSV, beberapa pengamatan dapat dilakukan sehubungan dengan daerah warna kubus RGB. Yang pertama adalah vertek-vertek cyan, magenta dan kuning dari kubus yang menunjukkan warna yang lebih cerah dibanding dengan warna merah, hijau dan biru karena warna merah hijau dan biru diproyeksikan lebih rendah ke sumbu netral. Dengan cara yang sama, semua warna dalam piramid yang ditunjukkan vertek C,Y,M dan W berhubungan ke warna-warna lebih terang dan piramid yang ditunjukkan oleh titik pusat dan vertek R,G dan B berhubungan ke warna-warna yang lebih gelap. Warna dekat sumbu netral dalam kubus akan mempunyai banyak warna karena saturasinya kurang dan warna yang lebih jauh dari sumbu ini akan tampak lebih hidup (Permana,2010).

#### 2.4.5 *Smoothing*

Proses *smoothing* citra dilakukan untuk menekan *noise* pada citra. Gangguan tersebut biasa muncul akibat hasil dari alat pengambilan citra yang kurang bagus atau akibat saluran transmisi. Pada umumnya *noise* tersebut berupa

variasi intensitas suatu *pixel* yang tidak berkorelasi dengan *pixel-pixel* tetangganya. Secara visual, gangguan mudah dilihat oleh mata karena tampak berbeda dengan *pixel* tetangganya. *Gaussian Filtering* adalah filter ideal yang mampu mengurangi besarnya frekuensi spasial yang tinggi yang mampu mengurangi besarnya frekuensi yang lebih tinggi lagi dalam sebuah citra.

Ada beberapa cara yang berbeda untuk mengimplementasikan *Gaussian Filtering*. Salah satunya menggunakan *Spatial Filter*, konvolusi ini menggunakan *Operator Gaussian Smoothing 2-D* yang mirip dengan filter 1-D, tetapi menggunakan *kernel* yang berbeda yang mewakili bentuk Gaussian 1-D untuk nilai X dan nilai Gaussian 1-D untuk nilai Y. Sehingga konvolusi ini termasuk tipe *non linear kernel*. Dalam *Spatial Filter* digunakan persamaan 2.1.

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \dots\dots\dots(2.1)$$

Dimana x adalah jarak dari citra asli pada aksis horizontal, y adalah jarak dari citra asli dalam aksis vertikal, dan  $\sigma$  adalah standar deviasi dari distribusi *Gaussian*. Ketika diterapkan dalam gambar dua dimensi, rumus ini menghasilkan permukaan yang kontur, yaitu lingkaran konsentris dengan distribusi Gaussian dari titik pusat. Nilai dari distribusi ini digunakan untuk membangun sebuah konvolusi matriks yang diterapkan pada citra asli. Nilai baru pada setiap *pixel* diatur untuk rata-rata tertimbang dari *pixel* tetangga (*neighborhood*). Nilai *pixel* asli menerima bobot yang lebih berat (yang memiliki nilai tertinggi Gaussian) dan *pixel* tetangga menerima bobot yang lebih kecil sebagai jarak mereka meningkat *pixel* asli. Hal ini menghasilkan *blur* yang melindungi batas-batas dan tepi yang lebih baik daripada lainnya, *blurring filters* lebih seragam (Munir, 2004).

#### 2.4.6 *Computer Vision*

*Computer Vision* adalah pencitraan komputer dimana aplikasi tidak melibatkan manusia dalam proses pengulangan visual. Dengan kata lain, gambar yang diperiksa dan di olah oleh komputer. Meskipun orang yang terlibat dalam pengembangan sistem aplikasi, akhirnya membutuhkan komputer untuk mengambil informasi visual secara langsung (Umbaugh, 1998).

*Computer vision* merupakan sebuah proses otomatis yang menintegrasikan sejumlah besar proses persepsi visual, seperti pengolahan citra, klasifikasi citra, pengenalan citra dan akuisisi citra. *Computer vision* didefinisikan sebagai salah satu cabang ilmu pengetahuan yang mempelajari bagaimana komputer dapat mengenali obyek yang diamati atau diobservasi. Cabang ilmu ini bersama kecerdasan buatan (*Artificial Intelligence*) akan mampu menghasilkan sistem kecerdasan visual (*Visual Intelligence System*) (Munir, 2004).

$$Vision = Geometri + Measurement + Interpretatio.....(2.2)$$

Proses-proses dalam *computer vision* dapat dibagi menjadi tiga aktivitas:

- a. Memperoleh atau mengakuisisi citra digital.
- b. Melakukan teknik komputasi untuk memproses atau memodifikasi data citra.
- c. Menganalisis dan menginterpretasi citra dan menggunakan hasil pemrosesan untuk tujuan tertentu, misalnya memandu robot, mengontrol peralatan, memantau proses manufaktur, dan lain-lain.

## 2.4.7 OpenCV

OpenCV (*Open Computer Vision*) adalah sebuah API (*Application Programming Interface*) library yang sudah sangat familiar pada pengolahan citra *computer vision*. *Computer vision* itu sendiri adalah salah satu cabang dari bidang ilmu pengolahan citra (*Image Processing*) yang memungkinkan komputer dapat melihat seperti manusia. Dengan *computer vision* tersebut komputer dapat mengambil keputusan, melakukan aksi, dan mengenali terhadap suatu objek. Beberapa pengimplementasian dari *computer vision* adalah *face recognition*, *face detection*, *face/object tracking*, *road tracking*, dll.

OpenCV adalah *library open source* untuk *computer vision* untuk C/C++, OpenCV didesain untuk aplikasi *real-time*, memiliki fungsi-fungsi akuisisi yang baik untuk *image/video*. OpenCV juga menyediakan *interface* ke *Integrated Performance Primitives* (IPP) Intel sehingga jika anda bisa mengoptimasi aplikasi *computer vision* anda jika menggunakan prosesor Intel (Syafi'i,2011). Fitur yang dimiliki OpenCV antara lain :

1. Manipulasi data citra (*allocation, copying, setting, convert*).
2. Citra dan video I/O (*file dan kamera based input, image/video file output*).
3. Manipulasi Matriks dan Vektor beserta rutin-rutin aljabar linear (*products, solvers, eigenvalues, SVD*).
4. Data struktur dinamis (*lists, queues, sets, trees, graphs*).
5. Pemroses citra fundamental (*filtering, edge detection, corner detection, sampling and interpolation, color conversion, morphological operations, histograms, image pyramids*).

6. Analisis struktur (*connected components, contour processing, distance Transform, various moments, template matching, Hough Transform, polygonal approximation, line fitting, ellipse fitting, Delaunay triangulation*).
7. Kalibrasi kamera (*calibration patterns, estimasi fundamental matrix, estimasi homography, stereo correspondence*).
8. Analisis gerakan (*optical flow, segmentation, tracking*).
9. Pengenalan obyek (*eigen-methods, HMM*).
10. Graphical User Interface (*display image/video, penanganan keyboard dan mouse handling, scroll-bars*).

OpenCV terdiri dari 3 *library*, yaitu:

1. CV : Untuk algoritma *Image Processing* dan *Vision*
2. Highgui : Untuk GUI, *Image* dan *Video I/O*
3. CXCORE : Untuk struktur data, support XML dan fungsi-fungsi grafis.

#### 2.4.8 OpenRobotino API

OpenRobotinoAPI (*Application Programming Interface*) adalah *library* aplikasi *programming* yang dibuat khusus untuk Robotino yang diciptakan untuk mempermudah *user* dalam membuat program pada Robotino. *Library* ini memungkinkan akses penuh terhadap sensor dan *actors* pada Robotino. Komunikasi antara Robotino dengan PC melalui jaringan TCP dan UDP, dan semuanya sangat transparan, meskipun program yang berjalan sudah tertanam pada Robotino ataupun secara *remote* (Robotino,2010).