

PENGENDALIAN *MOBILE ROBOT* BERBASIS *WEBCAM* MENGGUNAKAN PERINTAH ISYARAT TANGAN

¹⁾Daniel Richard Andriessen ²⁾Harianto ³⁾Madha Christian Wibowo

- 1) S1 / Jurusan Sistem Komputer, STIKOM Surabaya. Email: ricky.richard7@ymail.com
- 2) S1 / Jurusan Sistem Komputer, STIKOM Surabaya. Email: hari@stikom.edu
- 3) S1 / Jurusan Sistem Komputer, STIKOM Surabaya. Email: madha@stikom.edu

Abstract

Robotino is an Omni-Directional robot created by Festo Didactic that already integrated webcam with a USB interface and a system using omni-directional movement of the drive is capable of moving in all directions and has a variety of sensors that can be programmed as needed.

It background in the development of control robot technology, the research is “application mobile robot control webcam-based using a hand gesture commands”, which uses image processing template matching methods.

Robotino can do the movement in accordance with the detected hand gesture commands from the camera on the Personal Computer (PC) which integrated with Robotino. The application can detect this type of human hand gesture commands using HSV color space with 100% success rate of 20 times the distance between the test by making a hand gesture with a PC camera between 110 up to 130 cm.

Keywords: Robotino, HSV Color Filtering, Metode Template Matching

1. PENDAHULUAN

Perkembangan teknologi robot dari hari ke hari terus semakin maju pesat. Di era awal teknologi robot, pergerakan robot tidak bisa dikontrol oleh manusia dari jarak jauh. Tahun 1926, teknologi robot berkembang lagi. Robot dapat dikontrol oleh manusia dari jarak jauh menggunakan *joystick* atau *wire remote control*, dengan jarak kontrol adalah sejauh panjang kabel. Sampai kemudian di tahun yang sama, Nikola Tesla dengan perahu boat ciptaannya, membuat perkembangan pada *console remote control*, tidak lagi menggunakan kabel tetapi menggunakan frekuensi radio yang memunculkan teknologi kontrol.

Kemudian perkembangan teknologi pengendalian robot yang signifikan terjadi di era milenium ini. Yaitu robot tidak lagi dikontrol menggunakan *console remote control*, tetapi menggunakan perintah manusia secara langsung. Salah satunya dalam penelitian ini adalah robot berjalan dan bergerak sesuai dengan isyarat tangan. Input isyarat tangan diambil dari kamera PC yang kemudian diproses pengolahan citra dan *template matching*. Kemudian robot akan bergerak sesuai dengan perintah isyarat tangan yang diberikan. Ada lima macam gerakan robot, yaitu maju, mundur, geser kanan, geser kiri, serta berhenti.

Pengolahan data input citra tersebut dilakukan menggunakan metode *template matching* dengan *pre-processing* menggunakan segmentasi warna HSV (*Hue Saturation Value*) dan *thresholding*. Metode *template matching* digunakan karena metode ini memiliki prinsip kerja yang sederhana, yaitu dengan mencocokkan tiap bagian citra yang didapatkan oleh kamera dengan citra sampel (*template*).

2. LANDASAN TEORI

4.1 Omni-Directional Robot

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*

sensors, infrared distance sensors, 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 1. Robotino (ROS, 2010).

4.2 Citra Digital

Citra digital adalah citra dua dimensi yang dapat ditampilkan pada layar monitor komputer sebagai himpunan berhingga (diskrit) nilai digital yang disebut *pixel (picture elements)*. *Pixel* adalah elemen citra yang memiliki nilai yang menunjukkan intensitas warna. Berdasarkan cara penyimpanan atau pembentukannya, citra digital dapat dibagi menjadi dua jenis. Jenis pertama adalah citra digital yang dibentuk oleh kumpulan *pixel* dalam array dua dimensi. Citra jenis ini disebut citra *bitmap* atau citra *raster*. Jenis citra yang kedua adalah citra yang dibentuk oleh fungsi-fungsi geometri dan matematika. Jenis citra ini disebut grafik vektor. Citra digital (diskrit) dihasilkan dari citra analog (*kontinu*) melalui digitalisasi. Digitalisasi citra analog terdiri atas penerokan (*sampling*) dan kuantisasi (*quantization*). Penerokan adalah pembagian citra ke dalam elemen-elemen diskrit (*pixel*), sedangkan kuantisasi adalah pemberian nilai intensitas warna pada setiap *pixel* dengan nilai yang berupa bilangan bulat (G.W. Awcock, 1996).

4.3 Pengolahan Citra

Pengolahan citra merupakan teknik manipulasi citra secara digital yang khususnya menggunakan komputer, menjadi citra lain yang sesuai untuk digunakan dalam aplikasi tertentu. Agar mudah diinterpretasi oleh manusia atau komputer, pengolahan citra harus dilakukan dengan berbagai macam metode untuk mencapai citra sesuai yang diinginkan.

Operasi pengolahan citra digital umumnya dilakukan dengan tujuan memperbaiki kualitas suatu gambar sehingga dapat dengan mudah diinterpretasikan oleh mata manusia dan untuk mengolah informasi yang ada pada suatu gambar untuk kebutuhan identifikasi objek secara otomatis. (Murinto, 2009)

A. Thresholding

Thresholding merupakan konversi citra berwarna ke citra biner yang dilakukan dengan cara mengelompokkan nilai derajat keabuan setiap *pixel* kedalam 2 kelas, hitam dan putih. Pada citra hitam putih terdapat 256 level, artinya mempunyai skala “0” sampai “255” atau [0,255], dalam hal ini nilai intensitas 0 menyatakan hitam, dan nilai intensitas 255 menyatakan putih, dan nilai antara 0 sampai 255 menyatakan warna keabuan yang terletak antara hitam dan putih.

Contoh operasi titik berdasarkan intensitas adalah operasi pengambangan (*thresholding*). Pada operasi pengambangan, nilai intensitas *pixel* dipetakan ke salah satu dari dua nilai, α_1 atau α_2 , berdasarkan nilai ambang (*threshold*) T dapat ditunjukkan seperti pada persamaan berikut: (Munir, 2004)

$$f_{x,y}' = \alpha_1, \quad f_{x,y} < T \quad \alpha_2, \quad f_{x,y} \geq T$$

B. 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. (Dhiemas, 2011)

C. Metode *template matching*

Template matching adalah salah satu teknik dalam pengolahan citra digital yang berfungsi untuk mencocokkan tiap-tiap bagian dari suatu citra dengan citra yang menjadi sampel (*template*). Teknik ini banyak digunakan dalam bidang industri sebagai bagian dari *quality control*.

Prinsip metode ini adalah membandingkan antara citra *streaming* yang akan dikenali dengan citra sampel (*template*). Citra objek yang akan dikenali mempunyai tingkat kemiripan sendiri terhadap masing-masing citra sampel (*template*). Pengenalan dilakukan dengan melihat nilai tingkat kemiripan tertinggi dan nilai batas ambang pengenalan dari citra objek tersebut. Bila nilai tingkat kemiripan berada di bawah nilai batas ambang maka citra objek tersebut dikategorikan sebagai objek tidak dikenal.

Padaprinsipnya metode *template matching* memiliki karakteristik antara lain:

1. Relatif mudah untuk diaplikasikan dalam teknik pengolahan citra digital.

2. Hasilnya relatif sangat akurat karena mendeteksi kesalahan hingga ukuran *pixel*.
3. Metode ini cukup rentan terhadap perbedaan orientasi antara citra sampel(*template*) dengan citra yang akan diidentifikasi, yang meliputi: ukuran, posisi dan kualitas citra.
4. Untuk mendapatkan hasil yang maksimal maka metode ini sangat tergantung pada teknik pengolahan citra digital yang lain seperti *enhacement*, *color filtering*, dll.

(Wardhana, 2008).

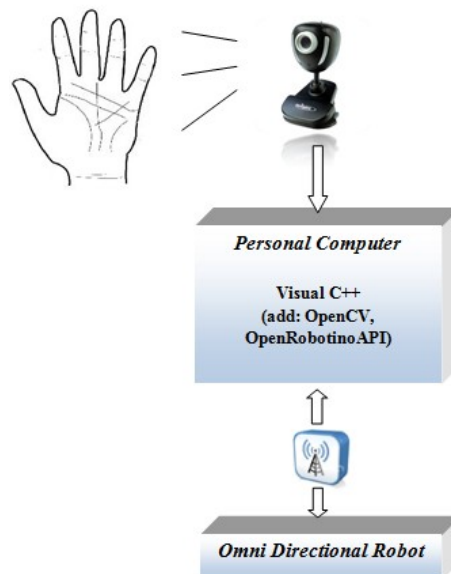
4.4 OpenCV

OpenCV (*Open Computer Vision*) adalah sebuah API (*Application Programming Interface*) library yang sudah sangat familiar pada pengolahan citra menggunakan *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 *vision* tersebut komputer dapat mengambil keputusan, melakukan aksi, dan mengenali terhadap suatu objek. Beberapa pengimplementasiandari *Computer Vision* adalah *Face Recognition*, *Face Detection*, *Face/Object Tracking*, *Road Tracking*, dll.

OpenCV adalah *libraryOpen Source* untuk *Computer Vision* untuk C/C++, OpenCV didesain untuk aplikasi *real-time*, memiliki fungsi-fungsi akuisisi yang baik untuk *image/video*.(Syafi'i,2011)

3. METODE PENELITIAN

3.1 Blok Diagram Sistem



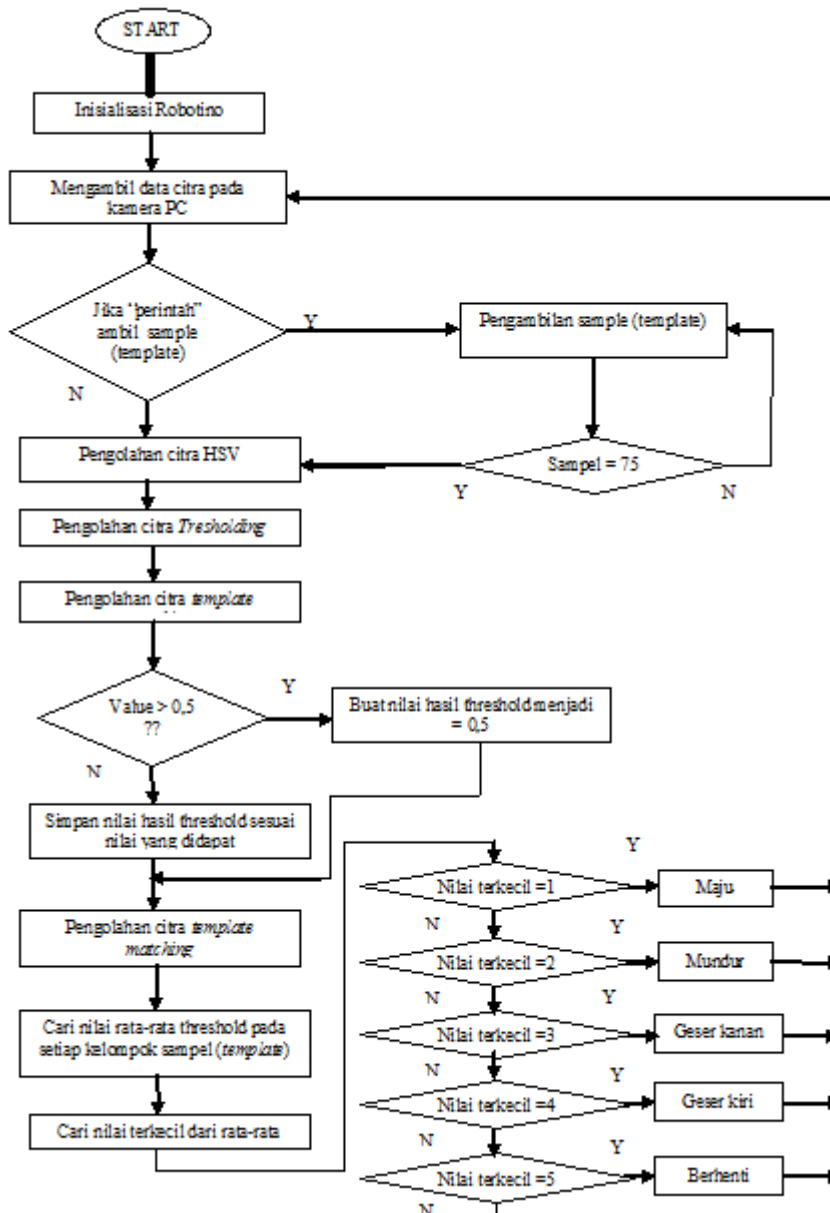
Gambar 2. Blok Diagram Sistem

Pertama sebagai input, citra didapatkan dari kamerapada PC user. Kemudian diproses menggunakan *console application* Visual C++ 2008 yang sudah terhubung dengan *library* OpenRobotinoAPI. Citra tersebut akan diolah dengan memanfaatkan *library* OpenCV. Citra yang diperoleh adalah citra isyarat tangan. Setelah mendapatkan citra isyarat tangan, maka selanjutnya citra tersebut akan diproses untuk kemudian dibandingkan dengan citra sampel (*template*). Kemudian robot akan melakukan gerakan sesuai dengan perintah isyarat tangan tersebut yaitu maju, mundur, belok

kanan, belok kiri, dan berhenti. Robot juga memberikan informasi pada PC berupa citra yang diambil secara *streaming* dan ditampilkan pada layar PC.

3.2 Algoritma Program

Untuk algoritma program yang digunakan akan dijelaskan melalui *flowchart* berikut:



Gambar 3. *Flowchart* Aplikasi

1. Kendali Robot

Untuk kendali Robotino digunakan OpenRobotinoAPI yaitu *library* aplikasi *programming* yang dibuat khusus untuk Robotino. *Library* ini memungkinkan akses penuh terhadap sensor dan *actors* pada Robotino. Tahap penelitian untuk kendali robot dimulai dari menyambungkan koneksi dari PC ke Robotino melalui media *wireless*, kemudian

mengatur pergerakan dari Robotino, mengakses *infrared distance sensor* pada Robotino, dan penerimaan data gambar secara *streaming*.

2. Pengolahan Citra

Pada tahap pengolahan citra dilakukan penelitian tentang cara konversi dari ruang warna RGB ke ruang warna HSV, kemudian hasil dari konversi citra digunakan untuk proses *color filtering* gambar dan diubah menjadi citra biner melalui proses *thresholding*, setelah mendapatkan citra biner hasil *color filtering* maka dilakukan proses *smoothing*, yang digunakan untuk menghilangkan *noise* pada citra tersebut.

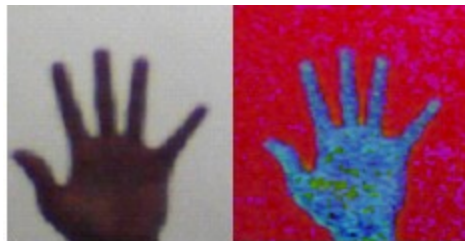
Setelah proses *color filtering* selesai, maka yang terakhir proses untuk mendeteksi jenis perintah isyarat tangan dengan metode *template matching*.

4. HASIL DAN PEMBAHASAN

4.1. Pengujian *Pre-Processing*

Pengujian berikut adalah pengujian data citra pada hasil pengolahan citra, dengan melihat tampilan data citra yang sudah dikonversikan ke ruang warna HSV. Kemudian dilakukan *color filtering* untuk mendeteksi warna kulit tangan menggunakan *thresholding*.

Untuk tahap proses *pre-processing* diperlukan konversi ruang warna RGB ke HSV, berikut adalah hasil konversi :


















Gambar 3. Hasil konversi RGB ke HSV

4.2. Pengujian Metode *Template Matching*





Pada tabel berikut adalah lima kelompok citra sampel yang dijadikan *template* untuk pembandingan dengan citra-citra *streaming* pada pengujian :

Tabel 1. kelompok citra sampel

	Sampel Value1	Sampel Value2	Sampel Value3	Sampel Value4	Sampel Value5
a					
b					
c					

Berikut seperti pada tabel2 sampai pada tabel7 adalah hasil perbandingan beberapa citra sampel (*template*) value1 sampai dengan value5 dengan beberapa citra *streaming* :

Tabel 2. Hasil pengujian sampel value1-5 dengan model beberapa citra streaming

Sampel Value	Coba1	Coba2	Coba3	Coba4
				
1a	0,114113	0,454561	0,795432	0,790518
1b	0,301332	0,513142	0,571363	0,751228
1c	0,345635	0,888943	0,733373	0,740587
2a	0,734525	0,832323	0,785233	0,914378
2b	0,765522	0,852228	0,801255	0,924342
2c	0,778266	0,812441	0,745129	0,915674
3a	0,867433	0,910078	0,672771	0,882079
3b	0,861343	0,887433	0,767432	0,828006
3c	0,852249	0,838722	0,719643	0,876597
4a	0,894005	0,924566	0,891233	0,701926
4b	0,914659	0,894821	0,834819	0,684608
4c	0,894034	0,927908	0,883441	0,670085
5a	0,984732	0,934842	0,923331	0,672308
5b	0,978822	0,959374	0,954652	0,672907
5c	0,979331	0,923731	0,984732	0,628474

Sampel Value	Coba5	Coba6	Coba7	Coba8
1a	0,623321	0,562308	0,664408	0,770518
1b	0,650045	0,537249	0,620786	0,751228
1c	0,578943	0,512354	0,664823	0,740587
2a	0,065443	0,393249	0,774565	0,914378
2b	0,241327	0,513142	0,417364	0,924342
2b	0,319835	0,378972	0,425885	0,915674
3a	0,583109	0,633247	0,628831	0,882079
3b	0,556261	0,618946	0,601287	0,828006
3c	0,574012	0,629801	0,640719	0,876597
4a	0,643551	0,654012	0,644012	0,701926
4b	0,616557	0,629644	0,624231	0,684608
4c	0,612387	0,607890	0,610089	0,670085
5a	0,845341	0,895646	0,886352	0,672308
5b	0,862989	0,890065	0,824971	0,672907
5c	0,979331	0,923731	0,984732	0,628474

Sampel Value	Coba9	Coba10	Coba11	Coba12
1a	0,867433	0,823321	0,872301	0,770518
1b	0,867433	0,826466	0,851535	0,751228
1c	0,860089	0,823321	0,842371	0,740587
2a	0,745341	0,746317	0,741224	0,914378
2b	0,723367	0,66164	0,713982	0,924342
2c	0,645341	0,706631	0,631004	0,915674
3a	0,063386	0,221864	0,435615	0,882079
3b	0,232126	0,453142	0,401237	0,828006
3c	0,369831	0,392972	0,442885	0,876597
4a	0,572412	0,552391	0,620004	0,671926
4b	0,584155	0,552391	0,645904	0,684608
4c	0,572841	0,534539	0,681344	0,670085
5a	0,610256	0,630056	0,657293	0,692308
5b	0,685673	0,630056	0,654338	0,672907
5c	0,624675	0,672361	0,657271	0,628474

Sampel Value	Coba13	Coba14	Coba15	Coba16
1a	0,894005	0,895630	0,898305	0,770518
1b	0,921469	0,917759	0,924239	0,751228
1c	0,856814	0,824034	0,871134	0,740587
2a	0,643411	0,643551	0,743551	0,914378
2b	0,616557	0,610057	0,716557	0,924342
2c	0,666087	0,685201	0,712727	0,915674
3a	0,572455	0,552319	0,620094	0,882079
3b	0,584155	0,552691	0,642904	0,828006
3c	0,572418	0,534039	0,681347	0,876597
4a	0,023311	0,151864	0,412281	0,651926
4b	0,312126	0,308112	0,371293	0,644608
4c	0,209422	0,370032	0,473449	0,670085
5a	0,512329	0,508882	0,512934	0,692308
5b	0,542938	0,519002	0,620001	0,672907
5c	0,523221	0,551346	0,629381	0,628474

Sampel Value	Coba17	Coba18	Coba19	Coba20
1a	0,988652	0,984732	0,912732	0,790518
1b	0,975422	0,961622	0,934822	0,751228
1c	0,970011	0,976131	0,953331	0,740587
2a	0,845341	0,834187	0,845354	0,914378
2b	0,862989	0,862211	0,862007	0,924342
2c	0,879331	0,979331	0,879767	0,915674
3a	0,685273	0,730056	0,754342	0,882079
3b	0,685653	0,710056	0,754312	0,828006
3c	0,785673	0,730767	0,696338	0,876597
4a	0,545673	0,520056	0,614338	0,701926
4b	0,519239	0,49518882	0,528029	0,684608
4c	0,519912	0,519002	0,513391	0,670085
5a	0,067999	0,218647	0,478366	0,672308
5b	0,172126	0,208112	0,472123	0,672907
5c	0,289193	0,270992	0,4934118	0,628474

Dengan melihat hasil pada tabel di atas, dapat disimpulkan bahwa pengujian *template matching* sudah berjalan dengan baik. Dari citra sampel (*template*) yang telah tersimpan, kemudian dibandingkan dengan citra *streaming*, maka akan didapatkan nilai masing-masing *threshold* rata-rata dari setiap citra *streaming*. Nilai yang berwarna merah adalah warna yang ikut diproses karena paling mendekati kemiripan. Kemudian *streaming* yang memiliki nilai rata-rata terkecil, maka itulah citra yang paling mirip dengan citra sampel (*template*).

4.3. Pengujian Sistem Keseluruhan

Pengujian Pengujian terakhir adalah pengujian sistem secara keseluruhan dari awal hingga akhir, dimana pengujian dilakukan dengan menjalankan aplikasi secara keseluruhan. Memberikan input isyarat tangan pada kamera PC yang diambil dari jarak kurang lebih 110 cm sampai dengan 130 cm, maka Robotino bergerak sesuai perintah isyarat tersebut. Isyarat maju, mundur, bergeser ke kiri, bergeser ke kanan, ataupun berhenti.

5. KESIMPULAN

1. Aplikasi dapat berjalan sesuai keinginan, yaitu mendeteksi citra pola isyarat tangan untuk kemudian diproses melalui *template matching*, yang dijadikan input untuk pergerakan Robotino.
2. Aplikasi dapat menggerakkan Robotino sesuai dengan jenis perintah isyarat tangan dengan tingkat akurasi 100% melalui 60 kali percobaan, diambil dengan jarak antara tangan dengan kamera PC kurang lebih antara 110-130 cm.

6. DAFTAR PUSTAKA

- Awcock, G.W. 1996. *Applied Image Processing*. Singapore: McGraw-Hill
- Dhiemas, R.Y.S. 2011. *Pencarian Posisi Robot: Studi Kasus Pencarian Sumber Kebocoran Gas*. (Online). (<http://jiki.cs.ui.ac.id/index.php/jiki/article/download/39/44>). Diakses Tanggal 27 Oktober 2011

Munir, Rinaldi. 2004. *Pengolahan Citra Digital dengan Pendekatan Algoritmik*. Bandung : Informatika Bandung

Murinto, M. 2009. Analisis Perbandingan Metode Intensity Filtering Dengan Metode Frequency Filtering Sebagai Reduksi Noise Citra Digital.(Online).

(<http://journal.uii.ac.id/index.php/Snati/article/view/1695/1477>). Diakses 15 September 2011.

3100010041280/14629). Diakses 9 September 2011

ROS. 2010. Robots Using ROS Robotino.(Online). (<http://www.ros.org/news/2010/06/robots-using-ros-Robotino.html>). Diakses 29 September 2011

Syafi'i, Slamet Imam.2011. Open Computer Vision(OpenCV). (Online).

(<http://slametux.blogdetik.com/2011/10/19/open-computer-vision-opencv>). Diakses 30 September 2011

Wardhana, Aditya Wisnu, Yudi Prayudi. (2008). Penggunaan metode templete matching untuk identifikasi kecacatan pada PCB. Jogjakarta : Informatika.