DETEKSI DAN TRACKING *WAJAH PADA ROBOT MANIPULATOR 2* DOF *BERBASIS KONTROL* SELF CONSTRUCTING FUZZY NEURAL NETWORK

Kusno Suryadi
Program Studi Teknik Elektro, Fakultas Teknik dan Informatika, Universitas Gajayana Malang
email: cegukks@yahoo.com

ABSTRAK

Self Constructing Fuzzy Neural Network (SCFNN) digunakan sebagai metode pengambilan keputusan dalam mengindentifikasi dan *tracking* obyek wajah yang diimplementasikan pada robot manipulator 2 derajat kebebasan, yaitu gerakan arah memutar pada sumbu X, perputaran terhadap sumbu Y dan berhenti. Keuntungan dari penggunaan metode SCFNN adalah tidak perlu menginisialisasi rule, konvergen secara cepat dan presisi. SCFNN diusulkan sebagai perbaikan hasil dari penggunaan ANN dan *Fuzzy Logic Control*, karena SCFNN merupakan penggabungan dari kelebihan kedua metode tersebut. Kontrol SCFNN digunakan sebagai tahap akhir dari sistem *tracking* obyek yang diambil berdasarkan proses citra wajah yaitu besar *error* yang terjadi berdasarkan posisi wajah yang terdeteksi terhadap titik tengah kamera. Berdasarkan hasil pengujian sistem *steady state error* rata-rata pada proses *tracking* dengan posisi obyek yang berbeda adalah sebesar 1,4 *pixel* untuk gerakan kearah sumbu x dan 1,2 *pixel* untuk arah sumbu y dan *overshoot* sebesar 0%, serta *settling time* sebesar 1,66 detik hingga 1,72 detik.

Kata Kunci: Control SCFNN, Deteksi wajah, robot 2 DOF

1. PENDAHULUAN

System *tracking* pada proses deteksi objek, terdapat beberapa komponen utama yang menjadi perhatian para peneliti, diantaranya adalah jenis dan algoritma system control yang digunakan, serta aplikasi sensor yang digunakan untuk proses deteksi objek.

Kontrol system pada proses *tracking* objek bertujuan untuk memperbaiki kinerja pergerakan robot, seperti halnya kecepatan pergerakan, *steadystate error, overshoot*, dll. Sedangkan sensor pada system ini adalah berfungsi sebagai sumber informasi yang digunakan untuk acuan *(setting point)* dan informasi target yang ingin dicapai pada system. Pada saat ini sensor yang banyak diminati oleh banyak peneliti adalah amplikasi sensor visual untuk proses identifikasi, deteksi dan bahkan dalam proses *recognition.*

Penggunaan sensor merupakan faktor utama yang perlu diperhatikan pada sistem pengaturan pergerakan robot, sebab sensor merupakan sumber informasi yang akan diperoses dalam pengaturan *tracking* dan pendeteksian sebuah obyek.

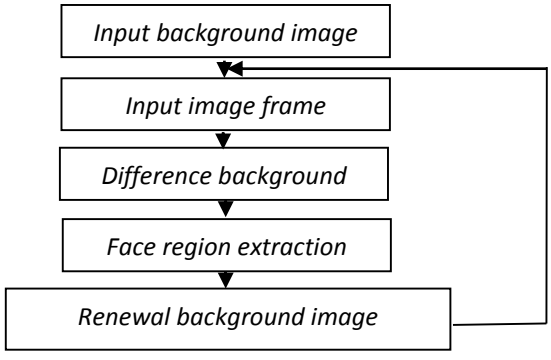
Deteksi dan *tracking* posisi obyek bergerak dengan menggunakan pengindraan visual saat ini telah banyak berkembang, seperti dalam bidang keamanan, biomedika, dan masih banyak lagi yang lainnya. Dalam tracking posisi dan deteksi obyek bergerak, posisi obyek, dan sensor yang dalam hal ini menggunakan kamera berada dalam area yang sama yang terdiri dari berbagai macam penghalang yaitu berupa obyek-obyek lain yang juga bergerak. Sehingga dibutuhkan suatu sistem yang mampu membedakan obyek yang menjadi target dengan obyek-obyek yang bukan target, serta dibutuhkan system pengaturan *(control)* yang handal sehingga mampu mengikuti gerakan obyek yang menjadi target dan sesuai dengan setting yang diinginkan.

Dalam penelitian ini, dibahas sistem *tracking* wajah dengan pengindraan visual pada robot manipulator dengan dua derajat kebebasan menggunakan kontrol *self constructing fuzzy neural network (SCFNN).*

2. Tinjauan Pustaka

2.1 Deteksi wajah

Proses deteksi wajah dilakukan dengan membedakan gambar *background* dengan gambar wajah, hal ini dapat dilakukan dengan menggunakan prosedur seperti yang ditunjukkan pada Gambar 1.



Gambar 1. Ekstraksi daerah wajah (*Takuma Funahashi, 2004*)

Proses penentuan daerah wajah dapat dilakukan dengan cara mengubah gambar input menjadi citra biner seperti daerah kepala. Setelah proses perubahan gambar ke bentuk biner maka langkah selanjutya adalah proses proyeksi. Proses proyeksi yang dilakukan adalah dalam arah x dan y sehingga ukuran horizontal wajah dapat diambil. Proses pembacaan atau *scaning* pada daerah kanan

dan daerah kiri yaitu pada sumbu x, dapat dipergunakan untuk menentukan nilai koordinat sebelah kanan dan sebelah kiri daerah wajah. Sedangkan pembacaan pada sumbu y dapat menentukan nilai koordinat puncak daerah wajah.

Proses pengurangan *background* sangat dipengaruhi oleh perubahan-perubahan seperti pencahayaan pada *background*, sehingga diperlukan sebuah metode untuk pembaharuan secara bersamaan latarbelakang gambar tersebut. Perubahan nilai rata-rata tiap-tiap *pixel* pada nilai ke *(n-1)* frame sebelumnya dilanjutkan.

Mendeteksi wajah dapat juga menggunakan metode *Haarcascade Classifier* dengan OpenCV yaitu *cvHaar*. *cvHaar* adalah teknik *modern* yang memverifikasi roman wajah atau obyek yang lainnya. *OpenCV* menggunakan tipe deteksi wajah yang disebut dengan *Haar Cascade Classifier*. Dengan memberikan gambar yang berasal dari *file* maupun *live video*, detektor ini menguji tiap lokasi gambar dan mengklasifikasi sebagai wajah atau bukan wajah.

2.2 Image Based Visual Servoing

Visual sevoing didefinisikan sebuah kegiatan untuk mengontrol *"pose*" lengan-lengan robot beserta *end-effector* dengan menggunakan informasi visual. Dengan semakin meningkatnya kecepatan mikroprosesor sangat dimungkinkan untuk memasukkan sensor visual dalam suatu *close-loop system*, sehingga dapat memanfaatkan keuntungan dari suatu *close-loop system*.

Image-based visual servoing feature point dari citra digital secara langsung digunakan sebagai *feedback (Djoko Purwanto, 2004)*. Hubungan antara *error* pada citra digital dan sinyal kontrol pada lengan robot didekati dengan transformasi linier yang biasa disebut dengan *image jacobian.*

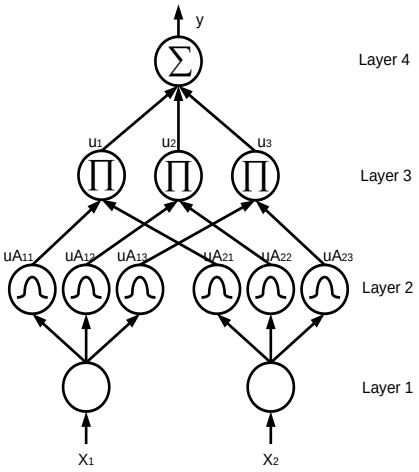
2.3 Kontrol Self Constructing Fuzzy Neural Network (SCFNN)

Kontrol *Self Construkcting Fuzzy Neural Network* merupakan control yang mengkombinasikan antara metode *Neural Nettwork* dengan *fuzzy logic*, dimana struktur *fuzzy* dibentuk berdasarkan hasil *learning* dari metode *Neural Network.* Cara kerja dari SCFNN dibentuk berdasarkan struktur yang terbentuk dari empat lapisan seperti yang ditunjukkan pada Gambar 6. Cara kerja dari masing-masing lapisan adalah sebagai berikut:

Lapis 1: Tidak ada perhitungan dilakukan pada lapisan ini. Masing-masing node pada lapisan ini adalah lapisan masukan, yang hanya mentransmisikan nilai masukan pada lapisan berikutnya secara langsung.

Lapis 2: Node pada lapisan ini menghubungkan label linguistik dari variabel masukan pada lapis 1, nilai membership menyatakan derajat sebuah nilai masukan terhadap fuzzy set yang dikalkulasi pada lapis 2. Fungsi membership Gaussian menyatakan perhitungan pada lapis 2 :

$$uA_{ji} = \exp\left(-\frac{[x_i - m_{ji}]^2}{\sigma_{ji}^{\ 2}}\right) \quad (1)$$



Gambar 2. Struktur Dasar Control SCFNN (Wan-De Weng, 2005)

dimana m_{ij} dan σ_{ij} menyatakan mean dan varian dari fungsi *membership Gaussian* untuk masukan ke-i dan rule ke-j.

Lapis 3: Node pada lapis ini menyatakan pengkondisian satu rule fuzzy logic. Menerima satu dimensional derajat membership yang menghubungkan rule dari node lapis 2 (disebut juga Firing Strength rule). Keluaran dari node lapis 3 adalah :

$$u_j = \left[\prod_i^M uA_{ji} \cdot x_i\right] \quad (2)$$

Lapis 4: Node pada lapis ini bertindak sebagai *defuzzifie*r. Node tunggal pada lapis ini diberi label sigma yang berarti menjumlahkan perkalian sinyal masukan dari lapis 3 dengan bobot ωj. y adalah keluaran dari SCFNN dimana bagian konsekuen dinyatakan sebagai ωj.

$$y = \sum_{j=1}^{M} u_j \cdot w_j \quad (3)$$

a. Algoritma Pembelajaran Untuk SCFNN

Terdapat dua tipe algoritma pembelajaran, pembelajaran struktur dan pembelajaran parameter yang dikembangkan untuk mengkonstruksi SCFNN. Pembelajaran struktur digunakan untuk menemukan pemisahan ruang masukan *fuzzy* yang sesuai dan *rule fuzzy logic* yang meliputi dua obyek :

1. Untuk meminimalisasi jumlah rule yang dibangkitkan
2. Untuk meminimalisasi jumlah fuzzy set pada *universe of discourse* untuk masing-masing masukan.

Pembelajaran parameter berbasis pada algoritma pembelajaran *supervised* (contohnya bobot penghubung pada bagian konsekuen dan parameter

fungsi membership di*update* dengan metode *backpropagation*). Pada awalnya, hanya terdapat node masukan dan keluaran saja pada struktur SCFNN tanpa *membership* dan *rule node*. Membership dan *rule node* dibangkitkan secara otomatis dan dinamis pada waktu proses pembelajaran berdasarkan pada data masukan dengan proses pembelajaran struktur dan parameter.

b. Fase Pembelajaran Struktur

Langkah awal pada fase pembelajaran struktur adalah mendefinisikan perlu tidaknya menambah rule untuk melatih data yang merupakan *fuzzy set universal of discourse* dari masing-masing masukan. Masing-masing kelompok pada masukan merupakan sebuah *rule fuzzy logic* yang potensial dengan mij dan σij menyatakan mean dan varian dari kelompok tersebut. Untuk masing-masing pola *xi*, kekuatan *rule* dapat diinterpretasikan sebagai derajat dari pola yang datang terhadap kelompok tersebut. Untuk efisiensi perhitungan, dapat dihitung firing strength yang dinyatakan sebagai $\prod u_{ji}$ sebagai pengukuran derajatnya :

$$F_j = \prod_i u_{ji} \tag{4}$$

dimana $F_j \in [0,1]$. Menggunakan pengukuran derajat, dapat menyatakan kriteria dari generasi *fuzzy rule* yang baru dari setiap data yang datang. Cara menemukan derajat maksimum :

$$F_{\max} = \max_{1 \le j \le u_{t+1}} F_j \tag{5}$$

dimana u(t) adalah jumlah rule yang ada pada waktu t. Jika $F_{\max} \le \overline{F}$ maka sebuah rule baru dibangkitkan dimana $\overline{F} \in [0,1]$ adalah batas yang digunakan selama proses pembelajaran. Saat satu rule dibangkitkan, langkah selanjutnya memberi tanda untuk mean dan varian dari fungsi membership yang baru. Tujuannya adalah meminimalkan fungsi obyektif dan mean dan varian semuanya di adjust pada saat fase pembelajaran parameter. Mean dan deviasi varian dari fungsi membership yang baru diset sebagai berikut :

$$m_{ji(R(t+1))} = x_i$$

$$\sigma_{ji(R(t+1))} = \sigma_{init}$$

dimana x_i adalah data baru dan σ_{init} adalah konstanta.

Saat pembangkitan generasi fungsi membership yang berhubungan dengan rule fuzzy, bobot $w_{j(u(t+1))}$ untuk rule fuzzy yang baru dibuat. Pada umumnya, nilai $w_{j(u(t+1))}$ dipilih secara acak.

Keseluruhan algoritma untuk generasi rule fuzzy yang baru adalah sebagai berikut :
Dalam system ini keseluruhan algoritma untuk generasi *rule fuzzy* baru adalah sebagai berikut :

1. Jika x_i adalah pola data pertama, maka bangkitkan rule baru dengan menentukan $mean\ m_{i1} = x_i$, $varian\ \sigma_{i1} = \sigma_{init}$, dan *bobot* w_1 ditentukan secara random.
2. Untuk masing-masing data baru x_I , Temukan $\mu_{\max} = \max_{1 \le j \le u_{t+1}} \mu_j$. Jika $\mu_{\max} \ge \overline{\mu}$ maka tidak ada proses yang dilakukan. Selain itu, jika $u_{(t+1)} = u_{(t)} + 1$, maka proses yang dilakukan adalah membangkitkan rule baru dengan $mean\ m_{ji(u(t+1))} = x_i$, $varian\ \sigma_{ji(u(t+1))} = \sigma_{init}$ dan *bobot* $w_{j(u(t+1))} = random$. Dimana σ_{init} adalah konstanta

c. Fase Pembelajaran Parameter

Setelah struktur jaringan terbentuk berdasarkan pola pelatihan, jaringan kemudian memasuki fase pembelajaran parameter untuk meng-*adjust* parameter jaringan secara optimum berdasarkan pola data yang sama. Proses pembelajaran adalah untuk mereduksi fungsi kesalahan dimana derajatnya kesalahan di adjust dengan menggunakan derajat negatif. Ide *backpropagasi* digunakan untuk pembelajaran *supervised* dengan menggunakan metode *gradient descent*. Dengan mengambil contoh satu keluaran untuk lebih jelasnya, tujuannya adalah untuk mereduksi fungsi kesalahan sebagai berikut :

$$E = \frac{1}{2}[y^d - y]^2 \tag{6}$$

dimana y^d adalah keluaran yang diinginkan dan y adalah keluaran sistem. Kemudian algoritma pembelajaran parameter berbasis pada backpropagasi yang dinyatakan sebagai berikut :
Lapis 4 : Error dipropagasikan dengan perhitungan

$$\delta^{(4)} = -\frac{\delta E}{\delta y} = y^d - y \tag{7}$$

dan bobot di-update dengan persamaan :

$$\Delta w_j = -\frac{\delta E}{\delta w_j} = [-\frac{\delta E}{\delta y}][\frac{\delta y}{\delta w_j}] = \delta^{(4)} \cdot u_j \tag{8}$$

Bobot pada lapis 4 di-update dengan persamaan berikut :

$$w_{j(t+1)} = w_{j(t)} + \eta \Delta w_j$$

$$w_{j(t+1)} = w_{j(t)} + \eta \delta^{(4)} u_j \tag{9}$$

faktor η adalah parameter kecepatan pembelajaran (learning rate).
Lapis 3 : Pada lapisan ini error dihitung dan dipropagasikan sebagai berikut:

$$\delta^{(3)} = -\frac{\delta E}{\delta u_j} = [-\frac{\delta E}{\delta y}][\frac{\delta y}{\delta u_j}] = \delta^{(4)} \cdot w_j \tag{10}$$

Lapis 2 :Error dihitung sebagai beriku

$$\delta^{(2)} = -\frac{\delta E}{\delta uA_{ji}} = [-\frac{\delta E}{\delta y}][\frac{\delta y}{\delta u_j}][\frac{\delta u_j}{\delta uA_{ji}}] \quad (11)$$
$$= \delta^{(3)}$$

Lapis 1 : Meng-update mean dan varian :

$$\Delta m_{ji} = -\frac{\delta E}{\delta m_{ji}} = [-\frac{\delta E}{\delta uA_{ji}}][\frac{\delta uA_{ji}}{\delta m_{ji}}] = \delta^{(2)} uA_{ji}[\frac{2\cdot(x_i - m_{ji})}{\sigma_{ji}^{\ 2}}] \quad (12)$$

$$\Delta\sigma_{ji} = -\frac{\delta E}{\delta\sigma_{ji}} = [-\frac{\delta E}{\delta uA_{ji}}][\frac{\delta uA_{ji}}{\delta\sigma_{ji}}] = \delta^{(2)} uA_{ji}[\frac{2\cdot(x_i - m_{ji})^2}{\sigma_{ji}^{\ 3}}] \quad (13)$$

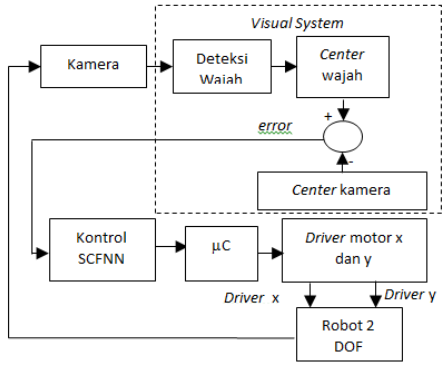
mean dan varian dari fungsi membership di-*update* dengan persamaan :

$$m_{ji(t+1)} = m_{ji(t)} + \eta\Delta m_{ji}$$
$$\sigma_{ji(t+1)} = \sigma_{ji(t)} + \eta\Delta\sigma_{ji} \quad (14)$$

3. METODOLOGI

3.1 Perencanaan Sistem

Sistem pengaturan pergerakan robot untuk deteksi dan *tracking* wajah direalisasikan berdasarkan diagram blok Gambar 4. Referensi posisi wajah adalah seting posisi yang menjadi target system, referensi titik wajah tersebut berada ditengah-tengah *field of view* dari kamera. Algoritma pergerakan robot dalam proses *tracking* wajah menggunakan sensor visual dapat dijelaskan sebagai berikut, kamera digital diletakkan pada *end effector* yang digunakan untuk menangkap gambar wajah. Setelah gambar wajah didapat, maka langkah selanjutnya adalah menentukan titik tengah wajah.



Gambar 3. Diagram blok sistem pengaturan pergerakan robot untuk deteksi dan tracking wajah

Setelah diketahui titik tengah wajah, kemudian menentukan selisih antara posisi titik tengah wajah dengan posisi titik tengah kamera. Penentuan selisih ini bertujuan untuk menentukan besar *error* yang kemudian digunakan sebagai masukan kontrol. Pada sistem ini keluaran kontrol masih bekerja dalam ranah *feature space* sehingga harus ditransformasikan terlebih dahulu kedalam ranah *joint space* sebelum dikeluarkan ke driver motor.

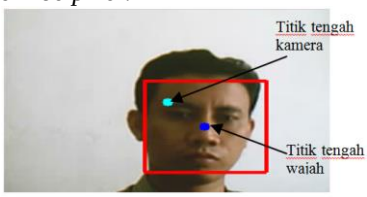
3.2 Deteksi Wajah

Untuk mendapatkan image wajah dilakukan dengan metode *Haar Cascade Classifier*, yaitu ketika *image* yang ditangkap oleh kamera sudah melalui *preprocessing*, maka langkah selanjutnya dilakukan *scanning* pada *image* tersebut yang bertujuan untuk menguji tiap lokasi gambar dan mengklasifikasi sebagai wajah atau bukan wajah. Klasifikasi mengasumsikan skala yang tetap untuk wajah. Ketika wajah lebih besar maupun lebih kecil daripada skala ini, maka klasifikasi bekerja pada gambar beberapa kali, untuk mencari wajah yang melewati range dari skala yang ada.

Klasifikasi dilakukan dengan menggunakan file data *haarcascade_profileface*, yang berfungsi untuk memutuskan klasifikasi tiap lokasi gambar. *Haarcascade_profileface* merupakan file yang berisikan kumpulan data posisi wajah. Prosedur pembentukan *rectangular* dan penentuan titik tengah wajah adalah sebagai berikut,

```
for(int i=0;i<(pFaceRectSeq? pFaceRectSeq->total:0); i++)
{
    CvRect* rect = (CvRect*)cvGetSeqElem(pFaceRectSeq, i);
result.push_back(cvRect((int)(rect->x * scale),
(int)(rect->y * scale),(int)(rect->width *scale),
(int)(rect->height * scale)));
 (int)center_x = cvRound((rect->x + rect-width*0.5)*scale);
 (int)center_y = cvRound((rect->y + rect-height*0.5)*scale);
}
```

Hasil deteksi wajah ditunjukkan pada Gambar 4, berdasarkan gambar tersebut dihasilkan titik tengah wajah. Sedangkan untuk titik tengah kamera ditentukan berdasarkan resolusi kamera. Pada system ini digunakan kamera dengan resolusi 240x340 *pixel*, sehingga titik tengah kamera berada dititik 120x160 *pixel*.

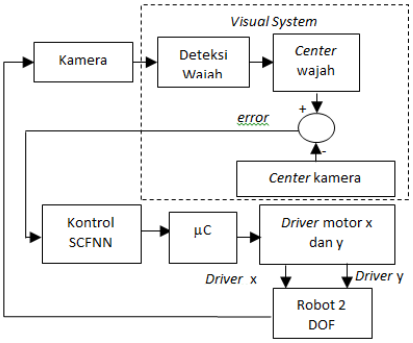


Gambar 4. Hasil Deteksi wajah

3.3 Algoritma Pergerakan

Kamera digital yang terletak pada *end effector* akan menangkap gambar wajah. Proses pertama yang dilakukan adalah proses *image preprocessing*. Proses ini dilakukan sebagai proses awal yang bertujuan untuk mendapatkan posisi wajah terhadap titik tengah kamera. Posisi wajah yang telah dideteksi kemudian digunakan untuk menentukan selisih titik tengah obyek terhadap titik tengah kamera *(error)*. Besar *error* ini kemudian digunakan sebagai *Feature setting* kontrol SCFNN. *Feature*

setting merupakan *setting point* dimana referensi titik tengah wajah berada ditengah-tengah *field of view* dari kamera. Algoritma dari sistem ini dapat dilustrasikan dalam diagram blok Gambar 5.



Gambar 5. Pengaturan pergerakan robot 2 DOF berbasis SCFNN

Jika data objek yang ditangkap kamera dikenali sebagai objek target, maka system akan mengunci data objek tersebut dan melakukan tracking yang dilakukan secara *real time*.

Pengaturan kecepatan pada system ini, ditentukan oleh selisih antara titik tengah kamera terhadap posisi obyek. Selisih antara titik tengah kamera dan titik tengah wajah (disebut dengan *error)*, sinyal *error* ini kemudian dijadikan sebagai sinyal masukan pada kontrol SCFNN. Kontroller SCFNN akan melakukan proses kontroller dan kemudian mengirimkan data pada rangkaian minimum sistem untuk menentukan arah gerakan dan prosentase PWM yang dibangkitkan untuk pengaturan kecepatan.

Perubahan posisi objek pada pergerakan ke arah x dan arah y terhadap titik referensi dihitung:

$\Delta x = m\_kcx - m\_ocx$

$\Delta y = m\_kcy - m\_ocy$

m_kc adalah titik center kamera (referensi)
m_ocx adalah titik center obyek pada sumbu x
m_ocy adalah titik center obyek pada sumbu y

3.4 Robot 2 Derajat kebebasan

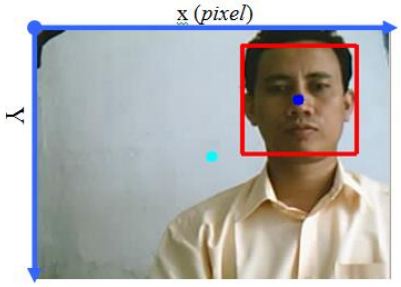
Pada tahap ini dibangun robot 2 derajat kebebasan yang digunakan untuk proses *tracking* obyek. Robot terdiri dari unit penggerak berupa dua buah motor, seperti pada Gambar 6. Untuk proses *tracking*, robot dikontrol oleh unit pengontrol menggunakan personal komputer. Pada ujung lengan robot dipasang sensor visual yang berfungsi untuk pengindraan visual.

Gambar 6. Robot 2 Derajat Kebebasan

4. HASIL DAN PEMBAHASAN

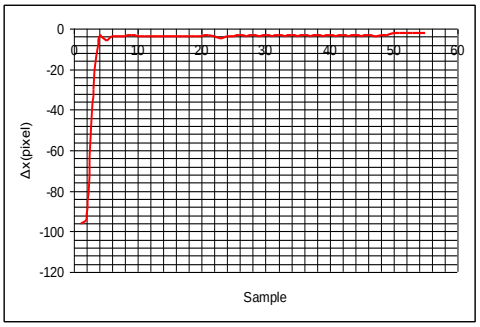
Pada pengujian dilakukan beberapa percobaan dengan posisi obyek berbeda, kemudian dilakukan analisis data dengan mengamati perubahan pergerakan obyek terhadap titik referensi. Data hasil pengujian adalah sebagai berikut:

1. Response sistem terhadap perubahan pada arah sumbu (x,y) dengan posisi awal obyek terdeteksi dititik (256,68) *pixel.* Gambar 7 menunjukkan hasil deteksi objek dan Gambar 8 menunjukkan hasil response pergerakan robot.

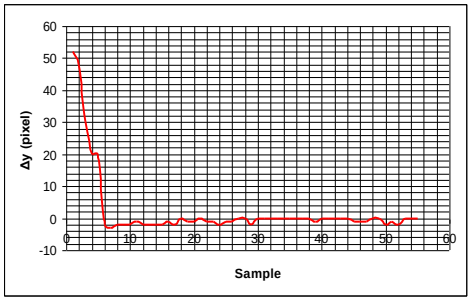


Gambar 7. Hasil deteksi objek di titik (256, 68) *pixel*

Berdasarkan hasil pengujian besar *overshoot* pada response tersebut adalah 0% dengan kesalahan hingga mencapai target adalah 2 *pixel* pada arah sumbu x dan 0 *pixel* pada arah sumbu y



(a)



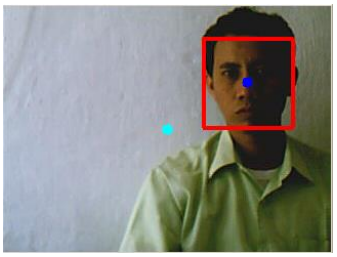
(b)

Gambar 8 (a) response pergerakan pada sumbu x ,
(b) response pergerakan pada sumbu y

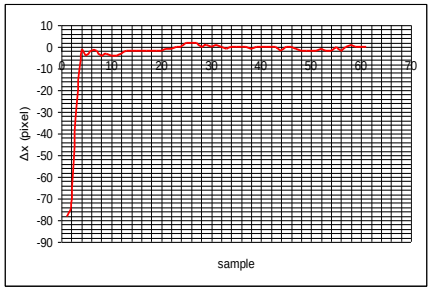
2. Response sistem terhadap perubahan pada arah sumbu (x,y) dengan posisi awal obyek berada pada titik (238,78) *pixel.* Pada pengujian ini obyek berada pada posisi awal (238,78) *pixel*, sehingga sistem bergerak dalam arah sumbu x dan sumbu y. Gambar 11 menunjukkan hasil deteksi objek dititik (238, 78) *pixel*, sedangkan Gambar 12 menunjukkan

No.	Posisi obyek (*pixel*)		*Steady state error (pixel)*		*Overshoot* (%)		Kecepatan (detik)	
	Sumbu		Sumbu		Sumbu		Sumbu	
	x	y	x	y	x	y	x	y
1	256	68	2	0	0	0	1,4	1,6
2	238	78	0	2	0	0	1,8	1,4
3	65	170	2	2	0	0	2	2,2
4	256	186	2	0	0	0	1,8	1,7
5	186	152	1	2	0	0	1,6	1,4

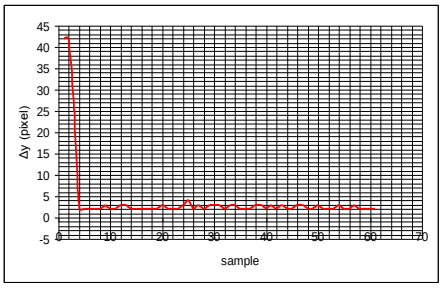
hasil pengujian response pergerakan robot. Dari hasil pengujian tersebut didapatkan *overshoot* pada response adalah 0% dengan kesalahan hingga mencapai target adalah 0 *pixel* pada arah sumbu x dan 2 *pixel* pada arah sumbu y.



Gambar 9. Deteksi objek dititk (238, 78) *pixel*



(a)



(b)

Gambar 10 (a) response pergerakan pada sumbu x, (b) response pergerakan pada sumbu y

Tabel 1. menunjukkan hasil pengujian dengan deteksi wajah berada pada posisi yang berbeda.

Tabel 1. Hasil Pengujian *tracking* wajah berbasis SCFNN.

Berdasarkan hasil percobaan, kesalahan rata-rata pada sumbu x adalah sebesar 1,4 *pixel* dan error rata-rata pada sumbu y adalah sebesar 1,2 *pixel* dengan *overshoot* sebesar 0%. Sedangkan kecepatan rata-rata adalah 1,72 detik untuk pergerakan kearah sumbu x dan 1,66 detik untuk pergerakan kearah sumbu y.

5. KESIMPULAN

Berdasarkan beberapa pengujian yang dilakukan pada kontrol robot berbasis *self constructing fuzzy neural network* dapat disimpulkan sebagai berikut:

1. Kontrol SCFNN mampu menginilialisasi jumlah rule yang sesuai berdasarkan proses learning dengan parameter-parameter dibentuk juga berdasarkan proses learning.
2. Berdasarkan pengujian kesalahan rata-rata pada proses tracking menggunakan kontrol SCFNN adalah sebesar 1,4 *pixel* pada arah sumbu x dan 1,2 *pixel* pada arah sumbu y. Sedangkan *overshoot* untuk pergerakan kearah sumbu x dan sumbu y adalah 0%.
3. Kecepatan rata-rata dari pergerakan robot menggunakan kontrol SCFNN adalah 1,72 detik untuk gerakan arah sumbu x dan 1,66 detik untuk gerakan kearah sumbu y.

DAFTAR PUSTAKA

Boyoon Jun and Gauray S. Sukhatme. *Detecting moveing object using a single camera on a mobile robot in outdor environtment*. Proceeding of the 8th Conference on intelegent autonomous system. Hal 980-987. Amsterdam, Belanda 2004.

C.H. Messom, G. Sen Gupta, S. Demidenko and Yuen Sion, Improving Predictive control of mobile robot, Instrumentation and measuremen technology conference vail, CO, USA, 20-22 may 2003.

Calos Perez, Oscar Reinoso, M.Asuncio Vicente, *Robot hand tracking using an adaptif fuzzy logic controller*, WSCG 2004, WSCG Poster proceedings.

Dongbing Gu, dkk. *Learning fuzzy logic controller for reactive robot behaviours*, Department of Computer Science, University of Essex Wivenahoe Park, Colchester CO4 3SQ,UK. Proceedings of IEEE/ASME International Conference on Advanced Intelligent Mechatronics, kobe, Japan, 20-24 juli 2003.

Sutedjo, dkk. *Penerapan self constructing fuzzy neural network sebagai observer fluksi pada motor induksi tiga fasa*, Institut teknologi sepuluh November Surabaya, Seminar nasional aplikasi teknologi informasi, UII yokyakarta, Juni 2005.

Wan-Dewang,dkk. *the design SCFNN based nonlinier channel equalizer*, Gradued school of engineering science and technology, Department of Electrical Engineering National Yunlin University of Science and Technology Yulin ,640 Taiwan, January 2005.