Perancangan Sistem Navigasi Menggunakan Kamera pada Quadcopter untuk Estimasi Posisi dengan Metode Neural Network

Redi Kharisman,Katjuk Astrowulan, dan Rusdhianto Effendie AK
Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember (ITS)
Jl. Arief Rahman Hakim, Surabaya 60111
E-mail: redikharisman11@gmail.com, ditto@ee.its.ac.id, katjuk@ee.its.ac.id

***Abstrak*—Sistem navigasi banyak digunakan sebagai pengenal objek dalam radius tertentu, penunjuk arah tujuan, penunjuk posisi objek berada dan sebagainya. Adakalanya sistem navigasi juga memiliki *noise* atau ketidakakuratan informasi dalam implementasinya sehingga membuat interpretasi terhadap posisi objek menjadi berbeda-beda pula. Dengan menganggap *quadcopter* sebagai suatu objek terbang dengan warna dominan, penentuan posisi dapat dilakukan dengan menggunakan dua buah kamera. Dengan kamera, setiap perubahan posisi objek akan diketahui langsung melalui titik tengah objek dengan proses pencitraan dan *tracking*nya. Namun kamera memiliki *noise* yang dapat menyebabkan perubahan data yang cukup sering dalam setiap cuplikan datanya. Metode *Backpropagation Neural Network* digunakan sebagai estimator yang ditujukan agar dapat mengikuti cuplikan data sekaligus juga memberikan posisi yang lebih akurat dengan mengeliminasi *noise* yang terdapat dalam cuplikan data yang terekam.**

***Kata Kunci*—kamera, sistem navigasi, estimasi, *Backpropagation Neural Network*.**

I. PENDAHULUAN

Peenelitian tentang UAV telah berkembang pesat karena kegunaan UAV yang telah meluas, dari keperluan akademik hinggal keperluan militer, dari keperluan proyek penelitian kecil, hingga megaproyek. Salah satunya digunakan sebagai pesawat pengintai, yang pengendalian dan monitoringnya dilakukan jarak jauh.

Umumnya, pada UAV memiliki sistem penerbangan yang sudah canggih, namun memiliki sistem navigasi yang minimalis. Sebab pengembangan UAV difokuskan pada teknik manuver, kestabilan gerakan, ketahanan terhadap turbulensi dan lainnya. Namun hanya sedikit yang mengembangkan sistem navigasinya. Sistem navigasi mempunyai manfaat yang luas dalam bidang teknologi. Sistem navigasi dapat berfungsi sebagai pengenal objek dalam radius tertentu, penunjuk arah tujuan yang akurat, sebagai estimator posisi dan sebagainya.

Sistem navigasi kadang juga memiliki *noise* atau ketidakakuratan informasi yang diterima dalam implementasinya yang diseKetidakakuratan ini membuat estimasi posisi objek yang berbeda-beda pula. Ketidak akuratan informasi juga disebabkan oleh *delay* yang terjadi selama proses *transmitting* dan *receiving* data.

Penentuan posisi objek dilakukan dengan menggunakan kamera, *Global Positioning System* (GPS), radar dan sebagainya. Dalam penggunaannya, GPS digunakan pada objek yang besar, yang memiliki radius delapan meter. Sebab GPS mendeteksi perubahan dalam setiap delapan meternya. Kamera merupakan alat yang tepat dalam menentukan posisi objek. Dengan kamera, setiap perubahan posisi objek akan diketahui langsung melalui pencitraannya. Namun kamera memiliki *noise* berupa pengaruh intensitas cahaya, kesamaan warna lingkungan, ragam warna objek dan lainnya yang dapat menyebabkan perubahan data dalam setiap cuplikan datanya.

Penggunaan estimator dengan *Backpropagation Neural Network* pada penelitian ini ditujukan untuk memberikan posisi yang lebih akurat dengan mengeliminasi *noise* yang terdapat dalam cuplikan data yang terekam sekaligus memberikan hasil yang juga dapat mengikuti pola data.

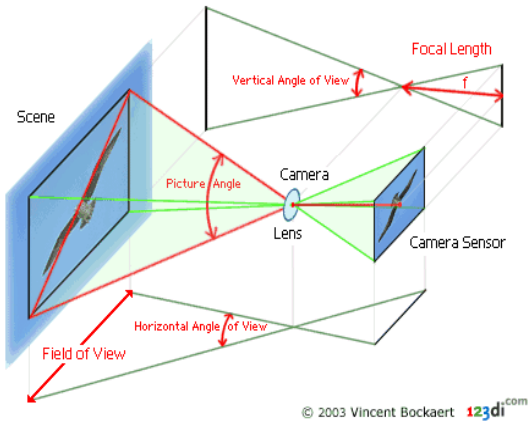
II. TEORI DASAR

Pada penelitian ini, pengenalan objek dilakukan berdasarkan ciri intensitas warna. Warna yang akan dikenali adalah warna dominan dengan wana *background* yang dominan pula namun dengan warna yang berbeda.

Perhitungan jarak dilakukan dengan metode *Stereoscopy*. Metode ini menggunakan dua buah kamera yang diletakkan pada jarak dan posisi tertentu. Untuk memudahkan perhitungan, sedapat mungkin posisi kedua kamera diatur sama dengan jarak pemisah yang diketahui. Kedua kamera ini akan merekam gambar dan menghasilkan data posisi dari koordinat pusat dari area objek. Objek yang berdimensi tiga dimensi akan diproyeksikan menjadi gambar dua dimensi. Koordinat posisi objek dikenali sebagai koordinat satuan pixel. Dengan metode *Stereoscopy* inilah koordinat pusat dari area objek yang tercuplik akan dihasilkan koordinat ruang relatif terhadap titik tengah dari dua buah kamera. Pemrosesan gambar, hingga mendapatkan titik tengah, dilakukan dengan memanfaatkan *library* gratis OpenCV yang berbasis bahasa pemrograman C++.

Backpropagation Neural Network digunakan karena algoritma ini juga banyak dipakai pada aplikasi pengaturan. Proses pelatihannya didasarkan pada hubungan yang sederhana, yaitu : Jika output memberikan hasil yang salah, maka pembobot (Weight) dikoreksi supaya erornya dapat diperkecil dan respon jaringan selanjutnya diharapkan akan lebih mendekati harga yang benar. BPNN yang merupakan singkatan *Backpropagation Neural Network* juga berkemampuan untuk memperbaiki pembobot pada lapisan tersembunyi (hidden layer).

Secara garis besar, mengapa algoritma ini disebut sebagai propagasi balik, dapat dideskripsikan sebagai berikut: Ketika Jaringan diberikan pola input sebagai pola pelatihan maka pola tersebut menuju ke unit-unit pada lapisan tersembunyi untuk diteruskan ke unit-unit lapisan output. Kemudian unit-unit lapisan output memberikan tanggapan yang disebut sebagai output jaringan. Saat output jaringan



Gambar 1 Sudut-sudut yang dibentuk oleh lensa dalam penangkapan objek

tidak sama dengan output yang diharapkan maka output akan menyebar mundur (backward) pada lapisan tersembunyi diteruskan ke unit pada lapisan input. Oleh karenanya maka mekanisme pelatihan tersebut dinamakan *Backpropagation*/propagasi balik. Pada penelitian ini, BPNN digunakan untuk mengeliminasi *noise* yang terjadi selama proses pencuplikan, sekaligus sebagai penunjuk bahwa hasil estimasi dapat mengikuti data yang tercuplik

III. PERANCANGAN SISTEM

A. Kalibrasi Kamera

Proses pencitraan pada penelitian ini dilakukan melalui program Microsoft Visual C++. Sebelum masuk ke proses pencitraan, ada beberapa kalibrasi yang perlu dilakukan untuk mendapatkan nilai batas yang diperlukan. Nilai itu berupa sudut jangkauan maksimum kamera dan persamaan titik temu garis pada objek dengan memanfaatkan metode *Stereoscopy*. Sudut jangkauan maksimum dapat diketahui melalui spesifikasi kamera, yaitu *Diagonal Field of View.* Dengan memanfaatkan gambar 1, maka sudut maksimum jangkauan kamera dengan gambar 2.

$$\text{Tg}\ \beta = \frac{\frac{1}{2}Ypix}{X} \qquad \text{Tg}\ \frac{1}{2}\alpha = \frac{\frac{1}{2}Xpix}{X} \qquad \text{Tg}\ \frac{1}{2}\theta = \frac{\frac{1}{2}S}{X}$$

$$\frac{1}{X} = \frac{\text{Tg}\ \frac{1}{2}\beta}{\frac{1}{2}Ypix} \qquad \frac{1}{X} = \frac{\text{Tg}\ \frac{1}{2}\alpha}{\frac{1}{2}Xpix} \qquad \theta = 2\ Tg^{-1}\frac{\frac{1}{2}S}{X}$$

Dengan subtitusi persamaan, maka :

$$\theta = 2\ Tg^{-1}\left(\frac{S}{Ypix}\ \text{tg}\ \frac{1}{2}\beta\right)$$

atau

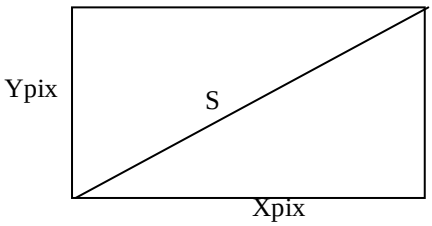
$$\theta = 2\ Tg^{-1}\left(\frac{S}{Xpix}\ \text{tg}\ \frac{1}{2}\alpha\right)$$

resolusi kamera adalah 640x480 pixels, maka :

$$\beta\text{max} = 2\ Tg^{-1}\left(\frac{Ypix}{S}\ \text{tg}\ \frac{1}{2}\theta\right)$$
$$= 36.7927^{\circ}$$

Dan

$$\alpha\text{max} = 2\ Tg^{-1}\left(\frac{Xpix}{S}\ \text{tg}\ \frac{1}{2}\theta\right)$$
$$= 47.8295^{\circ}$$

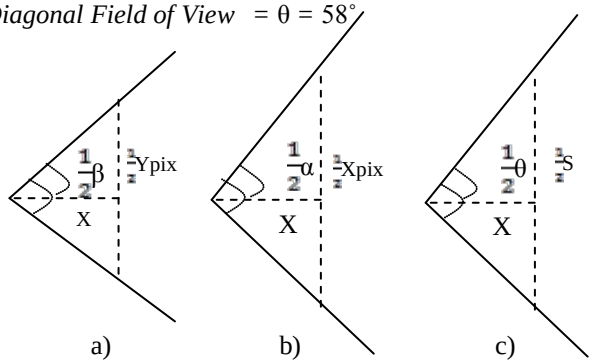


$S = \sqrt{Xpix^2 + Ypix^2}$

Vertical Field of View = β

Horizontal Field of View = α

Diagonal Field of View = θ = 58˚



Gambar 2 a) Vertical Field of View, b) Horizontal Field of View, c) Diagonal Field of View

Sebelum melakukan proses *tracking*, citra objek terlebih dahulu diolah dengan mendekomposisi warna yang tidak diinginkan baik pada objek maupun lingkungannya dan menampilkan warna yang dikehendaki. Proses dekomposisi warna ini dilakukan pada program Microsoft Visual C++.

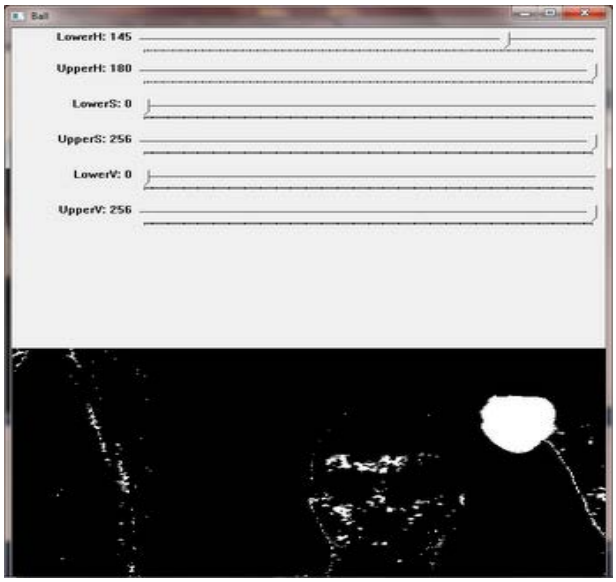
Proses dekomposisi warna dapat dilakukan dengan konversi warna citra dari format RGB ke format HSV dan mencari interval warna objek. Hal demikian dilakukan karena faktor pertimbangan, bahwa lebih mudah untuk melakukan dekomposisi warna dengan format HSV dibanding dengan format RGB.

HSV fokus pada pencerahan dan penggelapan warna. H adalah *hue*, merepresentasikan warna dengan nilai 0-180. S adalah *saturation* yang merepresentasikan tingkat kecerahan warna H yang diinginkan dengan nilai 0-255. V adalah *value* yang merepresntasikan tingkat kegelapan warna H yang diinginkan dengan nilai 0-255.

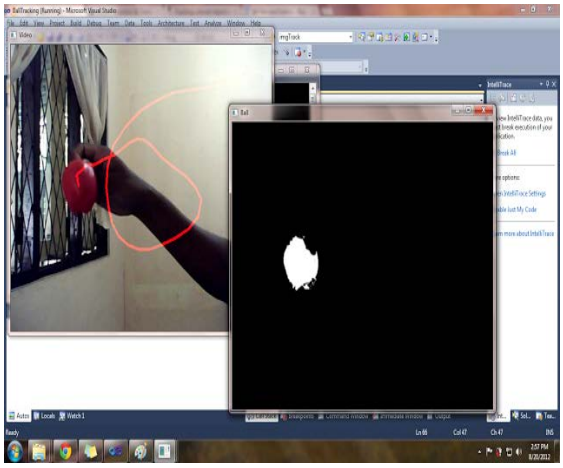
Dengan mendekomposisi warna-warna yang tidak diinginkan seperti gambar di atas, maka nilai-nilai internal HSV telah didapat. Nilai-nilai ini kemudian akan digunakan sebagai pengenalan warna objek selama proses *tracking.* Proses dekomposisi dilakukan dengan memanfaatkan fungsi-fungsi yang disediakan oleh library OpenCV. Begitupun juga pada saat proses *tracking*, seperti fungsi untuk mencari besar area hasil pencitraan objek, fungsi untuk mencari titik tengah area, memberi garis *tracking* sebagai penanda gerakan-gerakan objek. Dengan fungsi untuk mencari titik tengah area, maka koordinat objek hasil pencitraan dapat diketahui.

B. Perhitungan Jarak

Hasil *tracking*, akan menunjukkan koordinat pusat objek dalam satuan pixel. Sudut maksimum horizontal dan vertical juga telah didapat. Dengan memanfaatkan titik temu objek pada dua kamera, maka proses perhitungan dapat dilakukan. Proses perhitungan tersebut dilaksanakan pada program.



Gambar 3 Proses dekomposisi warna dengan masing-masing interval HSV-nya



Gambar 4 Pengolahan citra dengan proses tracking objek

Gambar di atas merupakan ilustrasi objek yang berada pada suatu jarak, relatif terhadap d, yang merupakan titik tengah jarak antar kamera. Ilustrasi di atas menunjukkan objek berada dalam sumbu X dan Y.

Pada proses pengolahan citra akan ditampilkan posisi dari titik tengah objek dalam bidang proyeksi. Posisi objek dimasukkan dalam variabel Xpix1 dan Ypix1 pada kamera1, sedangkan Xpix2 dan Ypix2 pada kamera2.

$\Delta a1$ = Xpix1 - 320 $\Delta a2$ = Xpix2 - 320

$\Delta b1$ = Ypix1 – 240 $\Delta b2$ = Ypix2 – 240

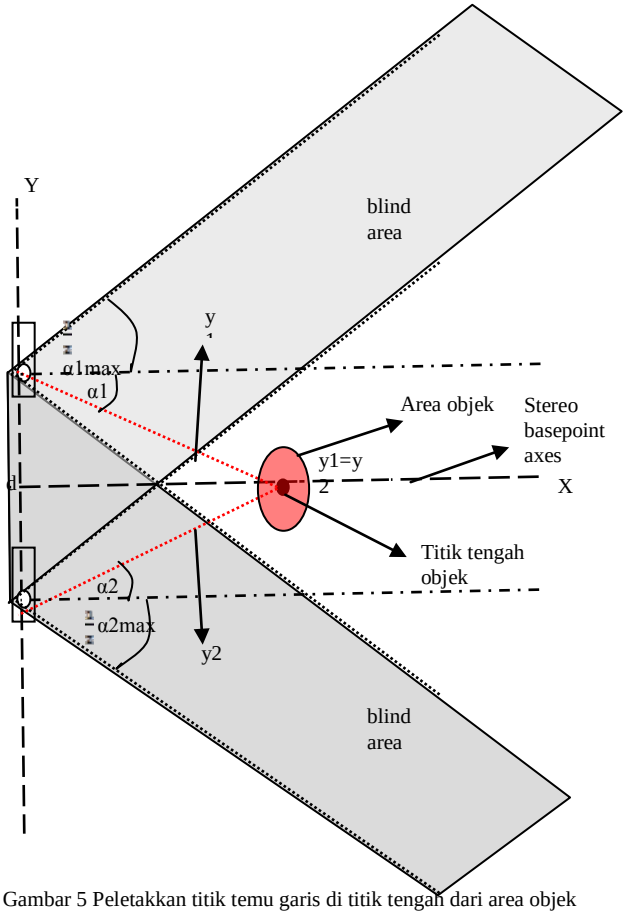
$$\alpha 1 = \frac{\Delta a1}{320}\frac{1}{2}\alpha max \qquad \alpha 2 = \frac{\Delta a2}{320}\frac{1}{2}\alpha max$$

$$= \frac{\Delta a1}{320}\frac{1}{2}47.8295^{\circ} \qquad = \frac{\Delta a2}{320}\frac{1}{2}47.8295^{\circ}$$

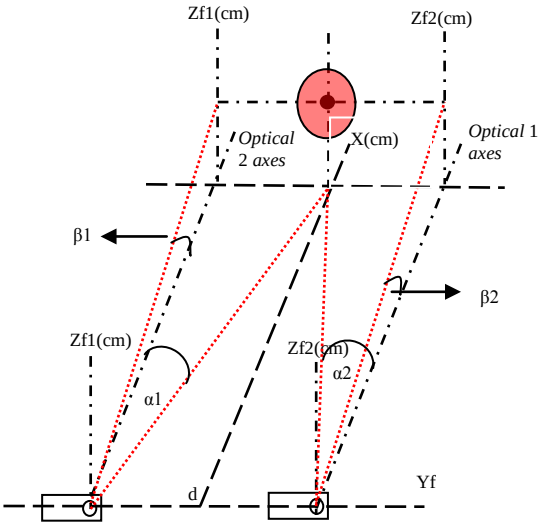
$$\beta 1 = \frac{\Delta b1}{240}\frac{1}{2}\beta max \qquad \beta 2 = \frac{\Delta b2}{240}\frac{1}{2}\beta max$$

$$= \frac{\Delta b1}{240}\frac{1}{2}36.7927^{\circ} \qquad = \frac{\Delta b2}{240}\frac{1}{2}36.7927^{\circ}$$

Oleh karena dua buah kamera berada pada koordinat Zf yang sama, maka sudut β1 dan β2 tidak akan jauh berbeda. Maka pencarian β dilakukan dengan mencari rata-rata β1 dan β2.



Gambar 5 Peletakkan titik temu garis di titik tengah dari area objek



Gambar 6 Titik temu garis pada objek dan sudut ketinggian yang terbentuk

$$\beta = \frac{\beta 1 + \beta 2}{2}$$

dengan demikian, perhitungan persamaan titik temu garis dapat dilakukan.

y1 = tg (α1) X – d

y2 = tg (α2) X + d

Tg (α1) X – d = tg (α2) X + d

$$X = \left| \frac{2d}{Tg(\alpha 1) - tg(\alpha 2)} \right|$$

$$Y = tg(\alpha 1)\frac{2d}{Tg(\alpha 1) - tg(\alpha 2)} - d$$

Atau

$$Y = tg(\alpha 2)\frac{2d}{Tg(\alpha 1) - tg(\alpha 2)} + d$$

$$Z = X\, tg(\beta)$$

$$= \frac{2d}{Tg(\alpha 1) - tg(\alpha 2)} Tg(\beta)$$

C. Simulasi dengan library OpenCV

Proses pencitraan dapat dilakukan dengan memanfaatkan *library* OpenCV. Proses dekomposisi warna, proses *tracking* hingga perhitungan jarak dilaksanakan pada satu *project* menggunakan Microsoft Visual C++ dengan komponen *include* yang terdapat pada *library* yang disediakan OpenCV. Dengan OpenCV, banyak hal yang dapat dilakukan untuk mengolah citra gambar. Salah satunya adalah capture frame yang didapat dari kamera yang aktif. Dengan fungsi-fungsi OpenCV pemrograman dapat dibuat untuk mengakses kamera, file image, file video untuk diproses citra digitalnya sesuai kebutuhan.

```
#include <cv.h>
#include <highgui.h>
.....
```

Penggunaan fungsi-fungsi memerlukan library. Oleh karena itu pemanggilan library melalui **#include** perlu dilakukan pada awal program.

```
.....
int main(){
CvCapture* capture =0;

capture = cvCaptureFromCAM(1);
if(!capture){
printf("Capture failure\n");
return -1;
}
......
```

Program di atas merupakan program dasar untuk mengakses port device CAM. Dengan program tersebut, maka port device kamera seperti kamera pada laptop, kamera yang dikenali akan diakses dan kemudian menjalankan device tersebut.

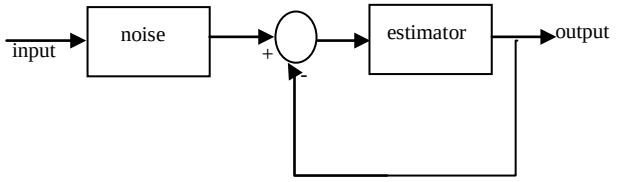
D. Perancangan Estimator

Pada penelitian ini metode *Backpropagation* digunakan sebagai estimator dari posisi objek yang telah terekam sebelumnya. Dalam perekaman posisi objek, terdapat *noise-noise* yang menyebabkan posisi objek kurang presisi. Metode ini juga dapat dikatakan sebagai filter untuk menghilangkan *noise* yang terjadi pada perekamannya. Secara umum, blok diagram keseluruhan dapat digambarkan pada gambar 7.

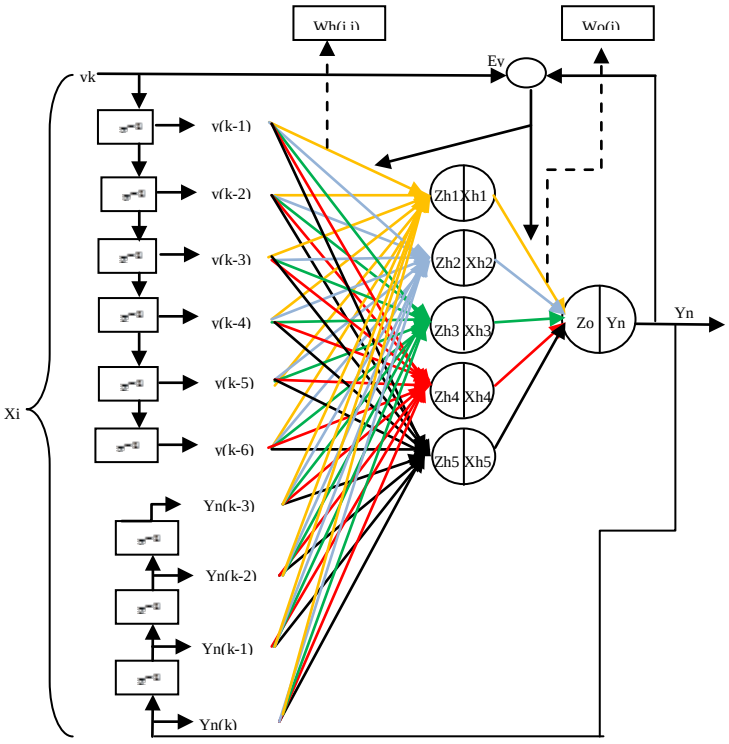
Estimator yang dimaksud adalah Backpropagation. Backpropagation merupakan metode pelatihan terawasi (supervised learning), dalam artian mempunyai target yang akan dicari. Ciri dari Backpropagation adalah meminimalkan error pada output yang dihasilkan oleh jaringan. dalam metode *Backpropagation*, biasanya digunakan jaringan multilayer. Jaringan multilayer yang dimaksud adalah layer yang terdiri dari input layer (layer masukan), hidden layer (layer tersembunyi), output layer (layer keluaran). Dalam pengembangannya, hidden layer dapat terdiri dari satu atau lebih unit hidden layer. Dengan menjabarkan estimator, maka blok diagram menjadi seperti gambar 8.

Diagram Backpropagation di atas dapat dijelaskan sebagai berikut :

yk = vektor data input yang berisikan data posisi



Gambar 7 Diagram blok sistem keseluruhan



Gambar 8 Diagram Backpropagation

Xi = input unit. Total input unit berjumlah 10. Masing-masing input unit akan diisi oleh vektor data input. Dan mendapat *delay*.

Xh = *hidden* unit dengan fungsi aktifasinya.

Yn = output unit, dengan fungsi aktifasinya.

error = selisih antara yk dan Yn yang hasilnya kemudian dirambatkan ke lapisan tersembunyi untuk merevisi bobot Wo(j) kemudian ke lapisan input untuk merevisi bobot Wh(i,j).

Secara matematis, diagram Backpropagation di atas dapat ditulis sebagai berikut :

Feed forward

Zh1 = X1 Wh1,1 + X2 Wh2,1 + X3 Wh3,1 + X4 Wh4,1 + X5 Wh5,1 + X6 Wh6,1 + X7 Wh7,1 + X8 Wh8,1 + X9 Wh9,1 + X10 Wh10,1

Xh1 = λ Zh1

Zh2 = X1 Wh1,2 + X2 Wh2,2 + X3 Wh3,2 + X4 Wh4,2 + X5 Wh5,2 + X6 Wh6,2 + X7 Wh7,2 + X8 Wh8,2 + X9 Wh9,2 + X10 Wh10,2

Xh2 = λ Zh2

Zh3 = X1 Wh1,3 + X2 Wh2,3 + X3 Wh3,3 + X4 Wh4,3 + X5 Wh5,3 + X6 Wh6,3 + X7 Wh7,3 + X8 Wh8,3 + X9 Wh9,3 + X10 Wh10,3

Xh3 = λ Zh3

Zh4 = X1 Wh1,4 + X2 Wh2,4 + X3 Wh3,4 + X4 Wh4,4 + X5 Wh5,4 + X6 Wh6,4 + X7 Wh7,4 + X8 Wh8,4 + X9 Wh9,4 + X10 Wh10,4

Xh4 = λ Zh4

Zh5 = X1 Wh1,5 + X2 Wh2,5 + X3 Wh3,5 + X4 Wh4,5 + X5 Wh5,5 + X6 Wh6,5 + X7 Wh7,5 + X8 Wh8,5 + X9 Wh9,5 + X10 Wh10,5

Xh5 = λ Zh5

Perhitungan di atas adalah perhitungan pada lapisan pertama, yaitu lapisan tersembunyi. Selanjutnya dilakukan perhitungan pada lapisan output.

Zo = Xh1 Wo1 + Xh2 Wo2 + Xh3 Wo3 + Xh4 Wo4 + Xh5 Wo5 + Xh6 Wo6

Yn = λ Zo

Ey = yk - Yn

Setelah output Yn dan error didapatkan, maka perambatan error dapat dilakukan untuk merevisi bobot pada lapisan output.

Backpropagation of Error

Wbaru = Wlama + α e λ X

α = learning rate, 0< α<1

e = error yang terjadi pada output

λ = fungsi gain

Dengan,

Wo1(baru) = Wo1(lama) + α Ey λ Xh1

Wo2(baru) = Wo2(lama) + α Ey λ Xh2

Wo3(baru) = Wo3(lama) + α Ey λ Xh3

Wo4(baru) = Wo4(lama) + α Ey λ Xh4

Wo5(baru) = Wo5(lama) + α Ey λ Xh5

Untuk menghitung error pada *hidden* unit, digunakan suatu persamaan agar merambat secara proporsional terhadap input.

$$Eh(j) = \frac{Wo(j)Xh(j)}{Zo} Ey$$

$$Eh1 = \frac{Wo1\,Xh1}{Zo} Ey$$

$$Eh2 = \frac{Wo2\,Xh2}{Zo} Ey$$

$$Eh3 = \frac{Wo3\,Xh3}{Zo} Ey$$

$$Eh4 = \frac{Wo4\,Xh4}{Zo} Ey$$

$$Eh5 = \frac{Wo5\,Xh5}{Zo} Ey$$

Masing-masing error pada *hidden* unit akan digunakan untuk merevisi bobot pada lapisan tersembunyi. Untuk mempersingkat perhitungan, maka digunakan variabel untuk mewakili index bobot.

Wh(i,1)baru = Wh(i,1)lama + α Eh1 λ X(i)

Wh(i,2)baru = Wh(i,2)lama + α Eh2 λ X(i)

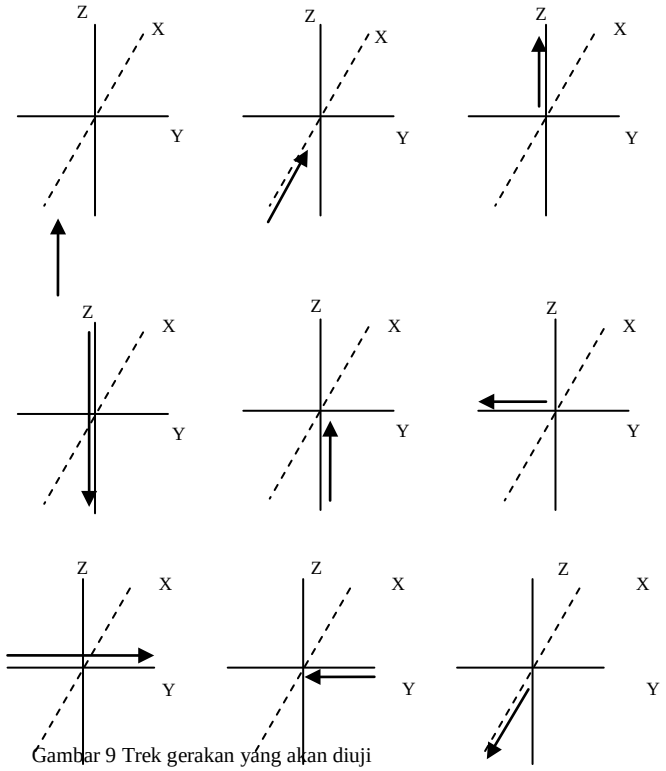
Wh(i,3)baru = Wh(i,3)lama + α Eh3 λ X(i)

Wh(i,4)baru = Wh(i,4)lama + α Eh4 λ X(i)

Wh(i,5)baru = Wh(i,5)lama + α Eh5 λ X(i)

Perlu untuk diketahui, bahwa inisialisasi bobot yang diberikan pada saat awal perhitungan adalah dengan memberi nilai random pada setiap bobotnya, baik pada lapisan tersembunyi maupun pada lapisan output. Yn(k), Yn(k-1), Yn(k-2), Yn(k-3) yang merupakan output, pada pertama kali proses, diinisialisasi sama dengan nol. Yang artinya bahwa pertama kali proses, estimator belum memiliki output. Sehingga nilainya diset sama dengan nol. Dan pada proses selanjutnya, estimator akan memiliki output dan seterusnya.

Hal demikian akan terjadi hingga akhir dari data input yang diberikan. Dengan berakhirnya pemrosesan seluruh data input, berarti telah selesai satu kali siklus pelatihan. Hal demikian akan terjadi selama banyak pelatihan yang



Gambar 9 Trek gerakan yang akan diuji

dikehendaki atau terjadi selama belum memenuhi kondisi yang diinginkan.Banyaknya siklus pelatihan dapat bervariasi tergantung dari keinginan pemrogram. Umumnya semakin banyak siklus pelatihan, maka hasil yang didapat semakin mendekati target dengan tanpa noise.

IV. PENGUJIAN SISTEM

A. *Pengujian Data*

Pengujian ini dilakukan untuk memastikan data yang dihasilkan oleh kamera satu dan kamera dua sinkron. Artinya, dengan penggabungan 2 kamera ini, diharapkan menghasilkan data posisi yang sesuai dengan gerakan objek. Gerakan yang di uji adalah sebagai pada gambar 9.

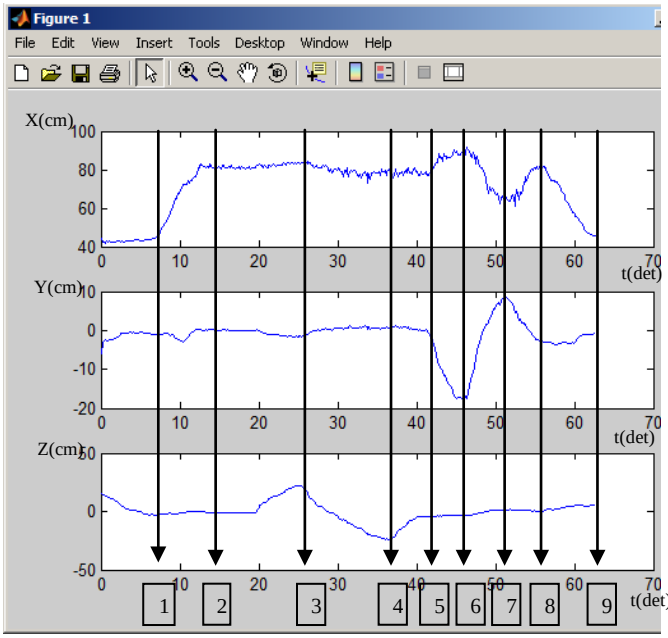
Gambar diatas adalah perubahan-perubahan gerakan objek yang diinginkan, kemudian akan dibandingkan dengan hasil data yang diperoleh gambar 11.

Dari gambar di atas, tampak grafik pergerakan mulai dari langkah ke-1 hingga ke-9. Gerakan yang terjadi merupakan gerakan non konstan dan manual. Oleh karenanya jeda pada langkah 2 ke langkah 3 merupakan gerakan yang paling lama. Sampling data pada pengujian ini adalah 7 data per detik.

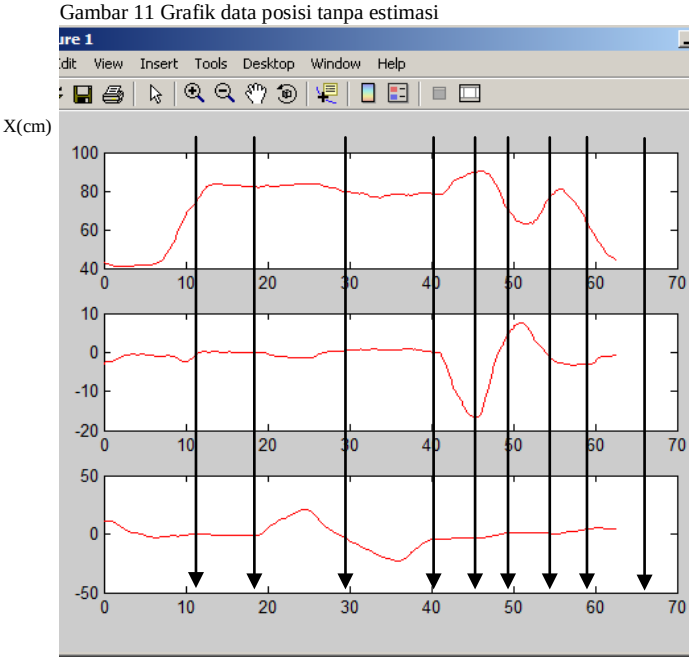
Pada grafik X terlihat jelas noise-noise yang terjadi pada detik ke 25 hingga detik ke 55. Noise ini berupa luas area yang sedikit berubah sehingga menyebabkan titik tengah citra akan berubah pula. Untuk diketahui bahwa data titik tengah dari citra ini lah yang digunakan untuk menghitung jarak objek. Titik tengah dari citra ini berupa koordinat dalam pixel. Noise juga terjadi karena warna objek dengan warna yang menggerakkan objek terdapat sedikit kesamaan. Sehingga pada detik ke 42 hingga ke-55 seolah-olah ada gerakan dalam sumbu X. Padahal gerakan yang terjadi dijaga untuk hanya bergerak pada sumbu Z kemudian Y.

B. *Pengujian Estimator*

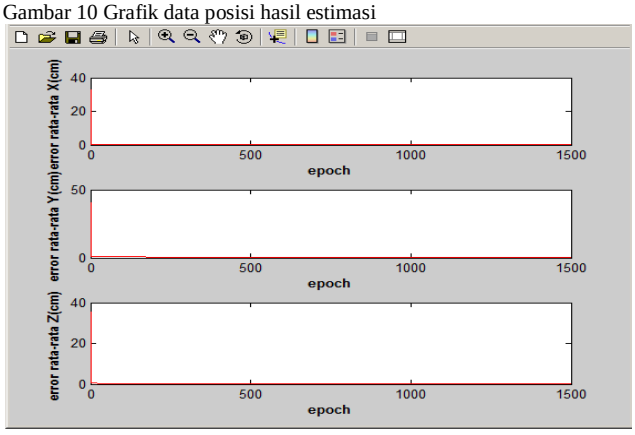
Pada pengujian estimator, akan ditampakkan hasil estimasi dari data yang telah tersimpan sebelumnya.



Gambar 11 Grafik data posisi tanpa estimasi



Gambar 10 Grafik data posisi hasil estimasi



Gambar 12 Grafik error rata-rata data dalam setiap pelatihan

Sejumlah data posisi yang tercuplik, secara otomatis akan tersimpan dan membentuk file dengan format *.txt yang kemudian data tersebut akan dipanggil melalui program MATLAB untuk digunakan sebagai input estimatornya.

Dapat dilihat bahwa noise yang terjadi banyak berkurang dibanding dengan grafik sebelumnya. Hasil estimasi yang didapat mengurangi sebagian noise kecil yang terjadi. Sehingga grafik hasil estimasi terlihat lebih halus jika dibandingkan dengan grafik tanpa estimasi.

Rata-rata error yang terjadi pada tiap data dalam 1 kali pelatihan dapat dilihat pada grafik 12.

Gambar 4.5 merupakan rata-rata *error* kuadrat yang terjadi pada tiap pelatihan dengan jumlah total 1500 kali pelatihan.

$$Err(k) = \frac{\sqrt{\sum_{i=1}^{n} t(i) - o(i)}}{n}$$

n = jumlah data
t = nilai target yang akan dicapai dicapai
o = nilai *output*
Err = Error pada satu siklus pelatihan
k = jumlah siklus pelatihan

Hasil negatif menunjukkan bahwa hasil estimasi masih lebih besar daripada target. Sebaliknya, hasil positif menunjukkan bahwa target lebih besar daripada hasil estimasi. Semua *error* yang dihasilkan mempunyai rata-rata nilai kurang dari 0,5 yang kemudian akan menuju ke nilai nol dengan arti bahwa setiap pelatihan yang dilakukan bertujuan untuk meminimalkan selisih yang terjadi antara target dengan hasil estimasi.

V. KESIMPULAN

Dari hasil pengujian keseluruhan sistem dapat disimpulkan :

1. *noise* yang terjadi seringkali terjadi karena faktor eksternal. Faktor eksternal seperti adanya kesamaan warna di sekitar lingkungan objek, intensitas cahaya yang memengaruhi pencitraan objek. Kurang presisinya peletakkan kamera webcam menyebabkan kurang akuratnya data yang tercuplik, sehingga data menjadi kurang presisi jika dibandingkan dengan pengukuran nyata.
2. Estimator mampu mengikuti cuplikan pola data sekaligus juga memberikan hasil yang lebih halus dibanding dengan pola data sumber.
3. Proses pelatihan yang diberikan pada estimator bervariasi, tergantung dari bentuk data hasil estimasi.
4. Terdapat kesalahan dalam proses estimasi berupa tidak munculnya data hasil estimasi, yang disebabkan oleh inisialisasi acak pembobot

DAFTAR PUSTAKA

[1] Fausett, Laurene, ”Fundamentals of Neural Network”, Prentice Hall International Inc, 2004
[2] Shermal Fernando, 2010.” Color Detection & Object Tracking”, <http://opencv-srf.blogspot.com/2010/09/object-detection-using-color-seperation.html>
[3] Jonathan Blow, 2010. “Graphics Tech: Shadow Maps (part 1)” <http://the-witness.net/news/2010/03/graphics-tech-shadow-maps-part-1/>
[4] Abdia Away, Gunaidi, “The Shortcut of MATHLAB Programming”, Informatika,Bandung,2006
[5] Kadir,Abdul,Oktober 2009. “Mudah Menjadi Programmer C++”, Andi, Yogyakarta
[6] Vincent Bockaert, “Focal Length”, <http://www.dpreview.com/glossary/optical/focal-length>