

## Balancing Robot Menggunakan Metode Kendali Proporsional Integral Derivatif

Rizka Bimarta<sup>\*1</sup>, Agfianto Eko Putra<sup>2</sup>, Andi Dharmawan<sup>3</sup>

<sup>1</sup> Prodi Elektronika dan Instrumentasi, FMIPA, UGM Yogyakarta

<sup>2,3</sup> Jurusan Ilmu Komputer dan Elektronika, FMIPA UGM, Yogyakarta

e-mail: <sup>\*1</sup>[rizkabimarta@gmail.com](mailto:rizkabimarta@gmail.com), <sup>2</sup>[agfi68@gmail.com](mailto:agfi68@gmail.com), <sup>3</sup>[dharmawan.andi@gmail.com](mailto:dharmawan.andi@gmail.com)

### Abstrak

Pendulum terbalik memiliki pusat gravitasi yang berada di atas poros putar sehingga menyebabkan pendulum terbalik tidak seimbang. Suatu kendali khusus dibutuhkan agar pendulum seimbang dengan cara menggerakkan kereta beroda yang menjadi tumpuan dari pendulum. Penerapan pendulum terbalik dapat ditemui pada balancing robot. Tujuan dari penelitian ini adalah merancang bangun sebuah sistem pengendalian robot dengan dua roda menggunakan sistem kendali untuk membuat robot yang seimbang (balancing robot).

Sistem ini mempunyai masukan akselerometer yang digunakan untuk mengukur percepatan sudut ( $m/s^2$ ) dan giroskop untuk mengukur kecepatan sudut ( $rad/s$ ). Luaran dari akselerometer dan giroskop digabungkan dengan metode complementary filter untuk mendapatkan nilai sudut. Sudut yang diperoleh kemudian dibandingkan dengan set point yang nilainya  $0^\circ$ . Nilai selisih dari set point dan sudut complementary filter diolah menggunakan metode kendali Proporsional Integral Derivatif. Proses kendali PID ini diprogram pada Arduino IDE yang hasilnya diumpankan ke motor DC untuk mengatur kecepatan putar motor DC. Untuk arah putar motor DC ditentukan apabila sudut complementary filter kurang dari nol, maka motor akan berputar mundur. Sedangkan jika sudut complementary filter lebih dari nol, maka motor akan berputar maju.

Nilai konstanta PID berdasarkan hasil tuning dengan metode Ziegler-Nichols metode osilasi adalah  $K_p=1.5$ ,  $K_i=0.75$ ,  $K_d=1.85$  dan nilai koefisien pada algoritma complementary filter adalah  $a=0.96$ .

**Kata kunci**—inverted pendulum, balancing robot, kendali PID, IMU, complementary filter

### Abstract

Center of gravity's inverted pendulum is located above its pivot point therefore inverted pendulum is unstable. Specific control is needed so that inverted pendulum stable which is by move the cart where the pendulum is mounted. Inverted pendulum application can be found in balancing robot. The purpose of this research is to design a system to control a two wheeled robot using the control system to balance it.

The inputs are accelerometer to measure angular acceleration ( $m/s^2$ ) and gyroskop to measure angular velocity ( $rad/s$ ). The output's of accelerometer and gyroscope are fused by complementary filter algorithm method to get the actual angle. The actual angle is then compared to set point which is  $0^\circ$ . The differences between set point and actual angle are processed using Proportional Integral Derivative control method. The process of PID control is programmed using Arduino IDE which its result is fed to DC motors. The direction of DC motors are determined by two conditions, if actual angle less than zero then DC motors will spin backwards. Whereas if actual angle more than zero then DC motors will spin forward.

The PID control's constans value based on Ziegler-Nichols Oscillation tuning method are  $K_p=1.5$ ,  $K_i=0.75$ ,  $K_d=1.875$  and complementary filter's coefficient is  $a=0.96$ .

**Keywords**—inverted pendulum, balancing robot, PID control, IMU, complementary filter

## 1. PENDAHULUAN

*Inverted pendulum* atau pendulum terbalik adalah pendulum yang terengsel ke kereta beroda yang dapat bergerak maju dan mundur pada bidang horisontal di sepanjang lintasan. Mempelajari dan merancang kendali untuk suatu pendulum adalah cara untuk membuktikan performa suatu kendali. Alasan dibalik pembelajaran mengenai pendulum adalah berdasarkan pada fakta bahwa banyak rekayasa sistem yang dimodelkan sebagai pendulum. Misalnya dalam vektor kendali pendorong roket, dinamika lapangan roket dapat didekati dengan pendulum sederhana. Dalam dunia biomekanik, pendulum digunakan untuk memodelkan bipedal yang berjalan dinamis dan juga digunakan dalam studi gerak roda dan mekanisme penyeimbangan. Selain itu pada tahun 2001, perusahaan Segway Inc. memperkenalkan alat transportasi berdasarkan pada *inverted pendulum* yang diberi nama Segway [1].

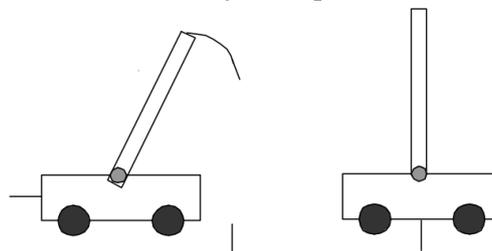
Penerapan konsep *inverted pendulum* dalam dunia robotika bisa dilihat pada *balancing mobile robot*, yaitu *mobile robot* dengan dua roda yang roda tersebut diasumsikan sebagai kereta beroda dan badan robot diasumsikan sebagai pendulum. Sistem ini tidak stabil karena ketika kereta beroda diberi gangguan dari luar maka pendulum akan jatuh. Ketika pendulum normal akan seimbang ketika digantung ke bawah, maka pada *inverted pendulum* atau *balancing robot*, untuk mempertahankan agar pendulum atau badan robot tidak jatuh dibutuhkan suatu kendali khusus [2].

Pusat massa dari pendulum terbalik berada di atas poros putar (kereta beroda) sehingga menyebabkan pendulum terbalik tidak seimbang. Ketika pusat massa pendulum terbalik secara langsung berada di atas poros putar maka poros putar akan tetap diam. Jika poros putar sedang diam dan pusat massa sedikit bergeser dari posisi vertikal, maka pendulum tidak akan kembali ke posisi semula tetapi cenderung mencari posisi keseimbangan baru sehingga pusat massanya berada pada posisi terendah mungkin. Sistem kendali umpan balik yang mengatur poros putar dapat digunakan untuk menyeimbangkan pendulum terbalik [3]. Kendali yang baik akan membuat pendulum tetap seimbang dengan cara mengendalikan pergerakan poros putar atau kereta beroda dimana pendulum tersebut terpasang.

## 2. METODE PENELITIAN

### 2.1. Balancing Robot

*Balancing robot* beroda dua merupakan suatu *mobile robot* yang memiliki dua buah roda yang tidak akan seimbang apabila tanpa adanya kontroler. *Balancing robot* ini merupakan pengembangan dari model pendulum terbalik (*inverted pendulum*) yang diletakkan di atas kereta beroda [4]. Prinsip dari *balancing robot* ditunjukkan pada Gambar 1.



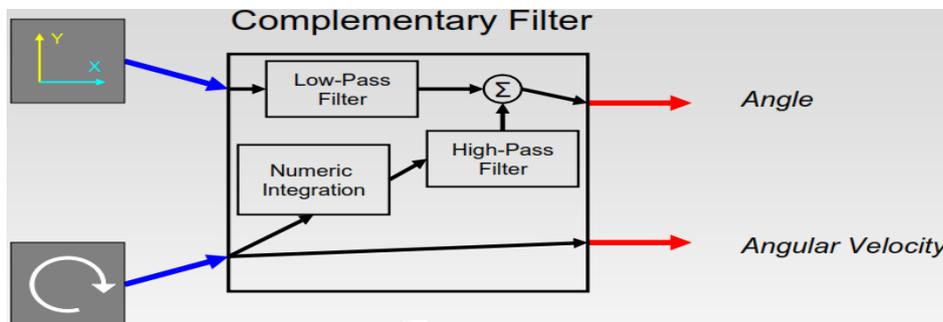
Gambar 1 Prinsip *balancing robot* [4]

Berdasarkan Gambar 1, permasalahannya ada pada kereta beroda yang bisa bergerak maju dan mundur dan pendulum yang ujungnya menyatu pada kereta beroda sehingga ketika kereta beroda bergerak maka pendulum akan jatuh. Oleh karena itu dibutuhkan kendali sehingga pendulum tetap seimbang dan berdiri tegak [5].

## 2.2. Complementary Filter

*Complementary filter* adalah tapis yang menggunakan dua atau lebih fungsi alih yang jika dijumlahkan akan menghasilkan fungsi alih dengan nilai satu. Yang dimaksud fungsi alih disini adalah proporsi perbandingan besarnya prosentase nilai akselerometer (1-a) dan prosentase nilai giroskop (a) yang digunakan pada proses *sensor fusion* yang jika dijumlahkan prosentase tersebut bernilai 100% yang ditunjukkan pada persamaan (1). Prinsip kerja dari *Complementary filter* ditunjukkan pada Gambar 2, yaitu untuk menggabungkan nilai sudut sekaligus untuk menghilangkan derau. Sinyal masukan pertama yaitu nilai sudut hasil pengukuran oleh akselerometer memiliki derau pada frekuensi tinggi sehingga difilter dengan *low pass filter*. Sinyal masukan kedua yaitu kecepatan sudut hasil pengukuran oleh giroskop memiliki derau pada frekuensi rendah sehingga difilter dengan *high pass filter*. Hasil penjumlahan dari kedua sinyal yang telah difilter tersebut berupa sudut [6].

$$\text{angle} = (a) * (\text{angle} + \text{gyroRate} * dt) + (1 - a) * (\text{accAngle}) \quad (1)$$



Gambar 2 Prinsip kerja *complementary filter* [6]

## 2.3. Kendali PID

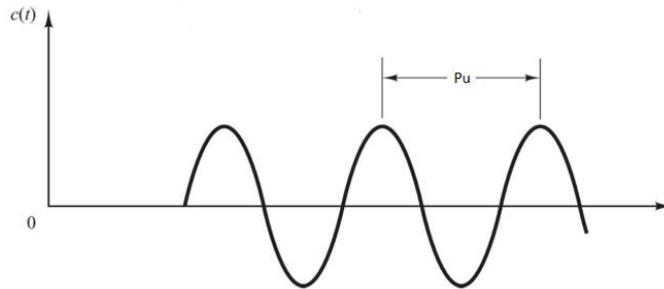
Skema kontrol PID diberi nama sesuai dengan tiga komponen pengkoreksi utama (Proporsional, Integral, Derivatif), dimana jumlah ketiganya merupakan jumlah dari variabel termanipulasi (*Manipulated Variable/ MV*). Proporsional, Integral, dan Derivatif dijumlahkan untuk menghitung nilai keluran dari kontroler PID tersebut [7]. Dengan mendefinisikan  $u(t)$  sebagai luaran kontroler, bentuk akhir dari algoritma PID ditunjukkan pada persamaan (2).

$$u(t) = MV(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{de(t)}{dt} \quad (2)$$

## 2.4 Metode Tuning Ziegler-Nichols Metode Osilasi

Metode ini digunakan pada sistem *close loop* yang merupakan metode tuning kendali PID untuk menentukan nilai penguatan proporsional  $K_p$ , *integral time*  $T_i$ , dan *derivative time*  $T_d$ . Metode ini akan memberikan nilai *overshoot* sebesar 25% pada *step response* [8]. Langkah-langkah untuk melakukan tuning konstanta PID menggunakan metode osilasi Ziegler-Nichols adalah sebagai berikut :

- $K_i$  dan  $K_d$  diberi nilai 0
- Nilai  $K_p$  dinaikkan terus sampai sistem mengalami osilasi yang stabil. Nilai  $K_p$  yang diperoleh tersebut ditulis sebagai  $K_u$
- Nilai  $P_u$  merupakan periode osilasi seperti yang ditunjukkan pada Gambar 3
- Nilai  $K_p$ ,  $K_i$ ,  $K_d$  dihitung berdasarkan Tabel 1



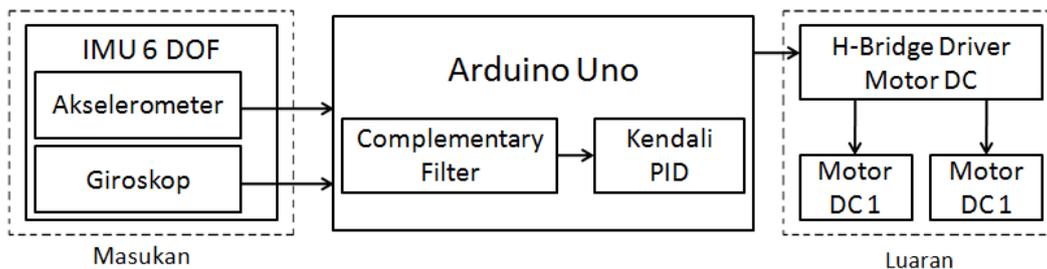
Gambar 3 Tuning Ziegler-Nichols metode osilasi [9]

Tabel 1 Model penalaan konstanta PID dengan metode Ziegler-Nichols metode osilasi [9]

Tipe Pengendali	Kp	Ti	Td
P	0,5 Ku	0	0
PI	0,45 Ku	Pu / 1,2	-
PID	0,6 Ku	0,5 Pu	0,125 Pu

2.4. Rancangan Sistem Keseluruhan

Rancangan sistem meliputi rancangan secara keseluruhan dimana ada rancangan mekanik, rancangan perangkat keras, dan rancangan perangkat lunak. Rancangan mekanik menjelaskan mengenai pembuatan mekanik dan sistem penunjang yang berada di dalamnya. Rancangan perangkat keras berisi penjelasan tentang perancangan rangkaian elektronik dari sistem yang digunakan. Rancangan perangkat lunak berisi perancangan program dan algoritma yang digunakan dalam sistem. Rancangan sistem secara keseluruhan ditunjukkan pada Gambar 4.



Gambar 4 Diagram blok sistem keseluruhan

Rancangan Penentuan Center of Gravity

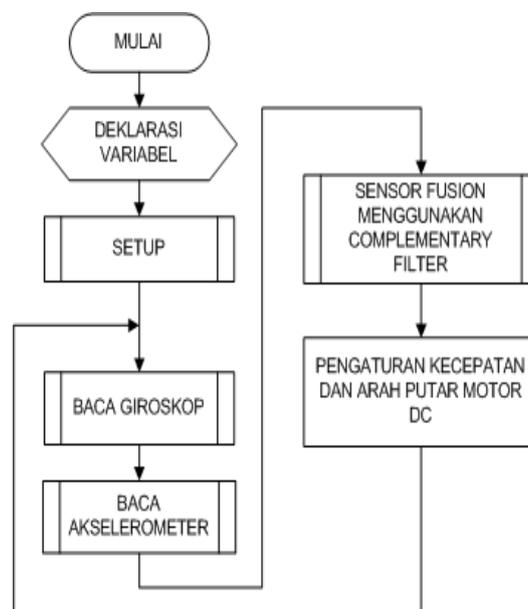
Center of Gravity atau pusat gravitasi pada suatu objek untuk bisa stabil idealnya adalah 1/3 dari tinggi objek. Penentuan letak CoG dilakukan dengan cara memegang robot dengan satu tangan pada masing-masing sisinya sampai robot seimbang. Kemudian garis imajiner digambar diantara kedua tangan. Selanjutnya robot dipegang pada dua sisi yang berlawanan yang berbeda dari sebelumnya (axis kedua) dan ditentukan titik seimbangnya dengan menggambar garis imajiner diantara titik tersebut. Ulangi ketiga kalinya dalam orientasi terakhir, titik garis berpotongan terhubung adalah pusat robot Anda gravitasi. Tetapi sebelum menentukan letak CoG menggunakan tiga garis imajiner tersebut terlebih dahulu dilakukan analisis agar letak COG berada pada 1/3 dari tinggi robot.

Rancangan Perangkat Lunak

Perangkat lunak yang dimaksud disini adalah program yang dimasukkan ke dalam mikrokontroler. Pemrograman tersebut dibuat dengan bahasa pemrogram khusus oleh Arduino

pada Arduino IDE (*Integrated Development Environment*) versi 1.0.2. Arduino sendiri merupakan bahasa turunan dari C++ sehingga fungsi-fungsi C++ dan C dapat berjalan di Arduino. Pemilihan bahasa Arduino ini dipilih karena untuk memadai perhitungan matematis dalam sistem *balancing robot* dan bersifat *open source*. Karena sifatnya yang *open source* ini, maka banyak *library* yang dikembangkan oleh personal atau komunitas di luar *developer* Arduino itu sendiri.

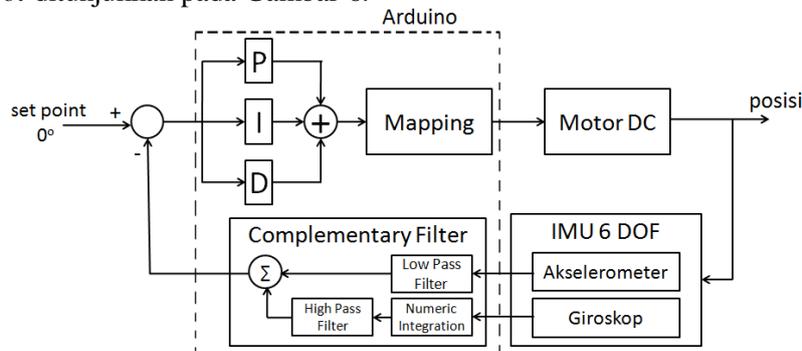
Garis besar jalannya program adalah membaca nilai kemiringan dari *balancing robot* dan membandingkannya dengan nilai referensi (*set point*), kemudian menentukan kecepatan putar pada masing-masing motor DC. Pengontrol PID digunakan untuk mengontrol kecepatan putar motor DC melalui PWM berdasarkan perbandingan dari nilai sudut referensi yang dikehendaki tersebut dengan nilai sudut yang terukur. Pemrograman dari Arduino akan mencakup deklarasi sensor, penerapan algoritma *complementary filter*, kendali PID, kendali untuk motor DC. Untuk garis besar proses jalannya program ditunjukkan melalui diagram alir pada Gambar 5.



Gambar 5 Diagram alir sistem pada *balancing robot*

### Rancangan Sistem Kendali

Kendali PID adalah kendali utama yang digunakan pada sistem ini. Parameter masukan utama yang digunakan dalam kendali PID adalah nilai *error* atau selisih nilai luaran sistem (*variable proses*) dengan nilai yang diharapkan (*set point*). Diagram blok kendali PID pada *balancing robot* ditunjukkan pada Gambar 6.

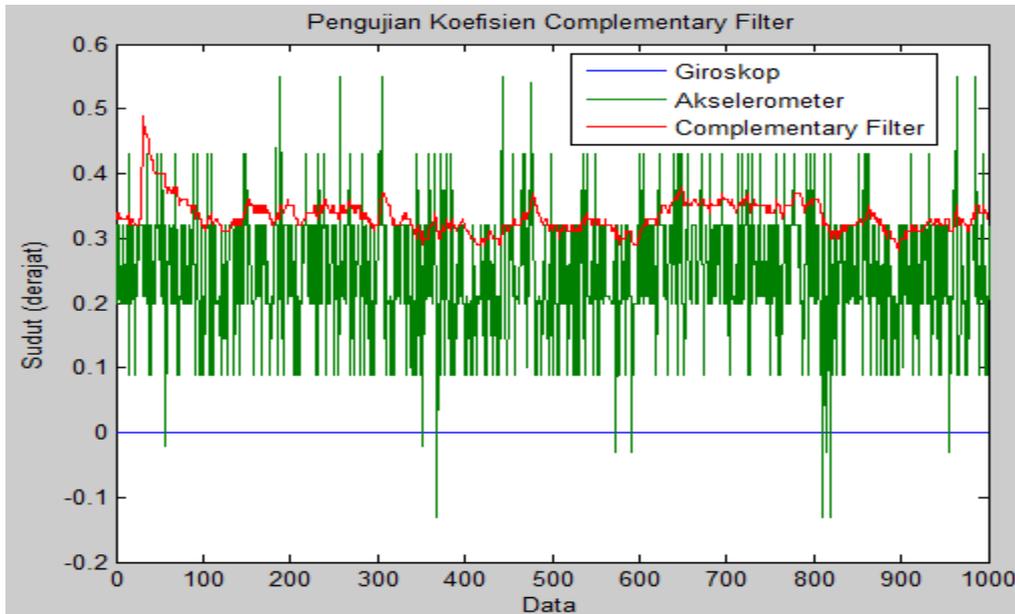


Gambar 6 Diagram blok sistem kendali *balancing robot*

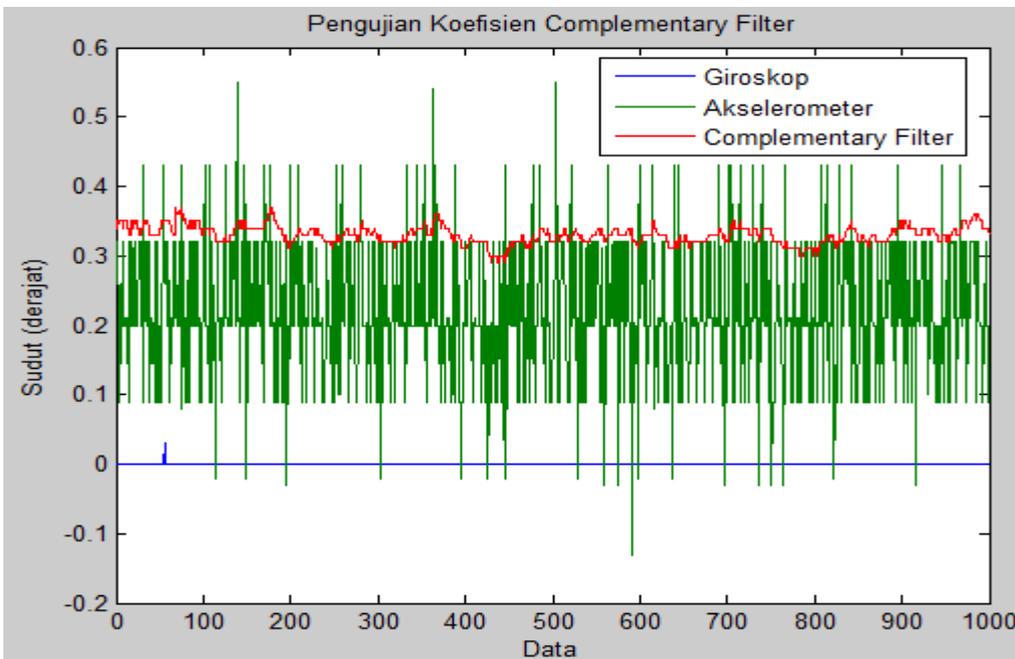
### 3. HASIL DAN PEMBAHASAN

#### 3.1. Pengujian Sudut Menggunakan Metode Algoritma Complementary filter

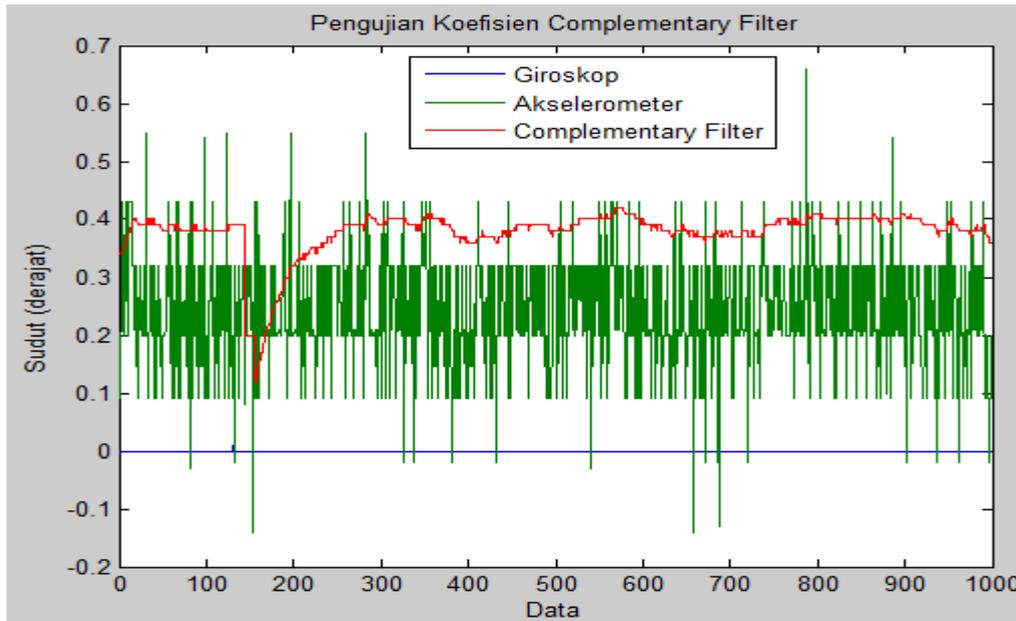
Pengujian ini dilakukan dengan mengimplementasikan algoritma *Complementary Filter* dengan cara melakukan variasi nilai koefisien filter  $a$  untuk dilihat pengaruhnya pada nilai sudut yang dihasilkan seperti pada Gambar 7 - 9.



Gambar 7 Pengujian algoritma *Complementary filter* pada sudut  $0^\circ$  dengan koefisien  $a = 0.95$



Gambar 8 Pengujian algoritma *Complementary filter* pada sudut  $0^\circ$  dengan koefisien  $a = 0.96$



Gambar 9 Penguian algoritma *Complementary filter* pada sudut  $0^\circ$  dengan koefisien  $a = 0.97$

Dari Gambar 7 sampai Gambar 9 dapat disimpulkan bahwa ketika nilai koefisien filter (a) masih rendah maka luaran filter terdapat lebih banyak *noise* (Gambar 7), tetapi apabila nilai koefisien filter terlalu tinggi maka luaran filter akan melebihi dari pembacaan sensor itu sendiri (Gambar 9). Hal tersebut terjadi karena semakin besar nilai koefisien filter maka akan memperlama *time constant* yang mengakibatkan waktu update sinyal semakin lama. Maka, pada penelitian ini koefisien filter yang digunakan adalah 0.96 (Gambar 8). Sehingga berdasarkan persamaan (1) bahwa jumlah koefisien filter harus 1. Sehingga nilai  $a=0.96$  dikalikan dengan hasil pengukuran sudut oleh giroskop dan  $1-a=0.04$  dikalikan dengan hasil pengukuran sudut oleh akselerometer.

Pengukuran sudut pada akselerometer baik pada frekuensi rendah sebab akselerometer mampu mengukur sudut lebih baik apabila sedang diam sehingga pada algoritma *complementary filter*, nilai akselerometer ditapis dengan *low pass filter* untuk meloloskan frekuensi rendah. Untuk giroskop, ketika sedang diam maka nilai kecepatan sudut adalah nol. Sedangkan ketika giroskop bergerak maka giroskop mampu mengukur kecepatan sudut sehingga giroskop baik digunakan pada frekuensi tinggi. Oleh karena itu, pada algoritma *complementary filter* giroskop ditapis dengan *high pass filter* agar dapat meloloskan frekuensi tinggi.

### 3.2 Penguian Sistem Kendali Balancing Robot

*Balancing robot* mempunyai 1 sudut dinamis yang harus selalu dikontrol agar dapat seimbang. Pada sistem ini, robot berotasi terhadap sumbu Y sehingga untuk akselerometer dibutuhkan 2 axis yaitu X dan Z sedangkan untuk giroskop dibutuhkan 1 axis yaitu sumbu Y. Nilai sudut dari algoritma *complementary filter* dibandingkan dengan nilai *set point*, selisih nilai tersebut berupa nilai *error* yang digunakan pada algoritma PID. Semakin besar nilai *error* maka semakin besar pula hasil dari algoritma PID yang menyebabkan semakin besarnya nilai PWM yang dihasilkan. Karena semakin besar simpangan sudut maka dibutuhkan semakin besar pula nilai PWM untuk mengembalikan agar pendulum dapat kembali berdiri tegak lurus.

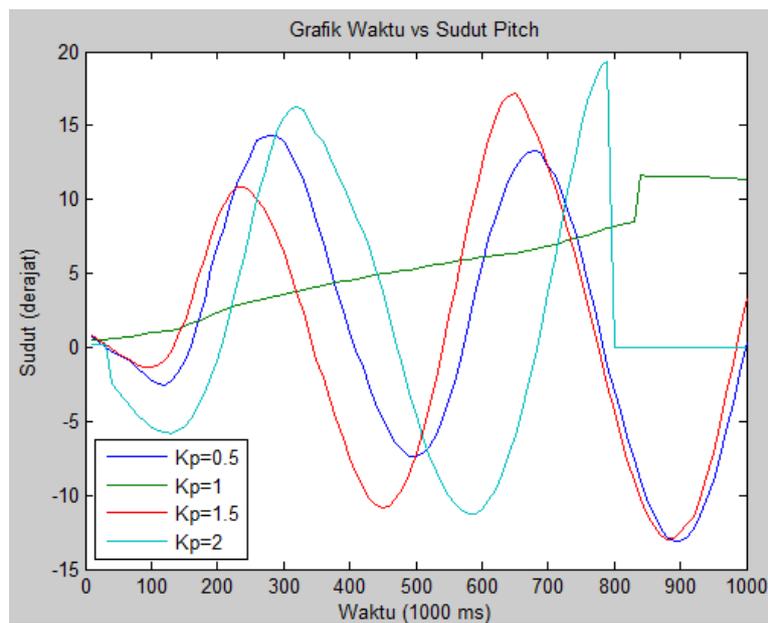
### 3.3. Penguian Kendali PID Menggunakan Metode Tuning Ziegler-Nichols Metode Osilasi Penguian Kendali P

Nilai *error* yang akan diberikan kepada kontrol proporsional didapat dari nilai *set point* dikurangi dengan nilai sudut *pitch* yang dideteksi oleh sensor. Nilai *error* ini dapat berupa *error*

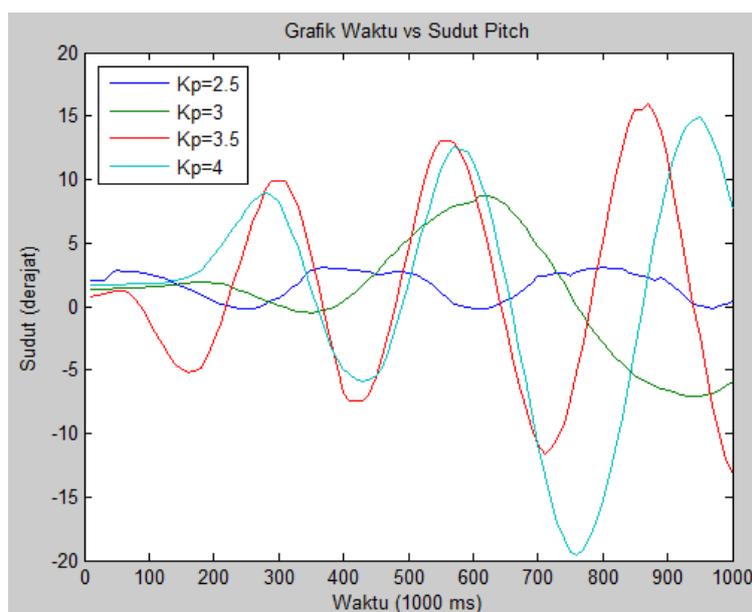
negatif maupun *error* positif sehingga luarnya dapat membuat sistem beresilasi pada daerah sekitar *set point*.

Ketika kendali P digunakan, dibutuhkan  $K_p$  (*proportional gain*) yang besar untuk memperbaiki *steady state error*. Sistem yang stabil tidak akan masalah apabila  $K_p$  yang digunakan besar. Jika konstan *steady state error* bisa diterima, maka kendali P bisa digunakan. Tetapi ketika nilai  $K_u$  terlalu besar maka variable proses akan beresilasi. Jika nilai  $K_p$  terus dinaikkan maka osilasi menjadi semakin besar dan sistem akan menjadi tidak stabil. Tetapi jika P terlalu kecil maka respon sistem terhadap perubahan yang terjadi menjadi lambat.

Pertama yang dilakukan adalah memberikan variasi kenaikan nilai  $K_p$  yaitu 0.5 sampai 4 dengan variasi kenaikan nilai  $K_p = 0.5$  yang hasilnya dapat dilihat pada Gambar 10 dan 11. Ketika sistem sudah beresilasi maka nilai  $K_p$  tersebut disimpan sebagai nilai  $K_u$ .



Gambar 10 Grafik sudut *pitch* vs waktu ( $K_p=0.5 - 2$ )



Gambar 11 Grafik sudut *pitch* vs waktu ( $K_p=2.5 - 4$ )

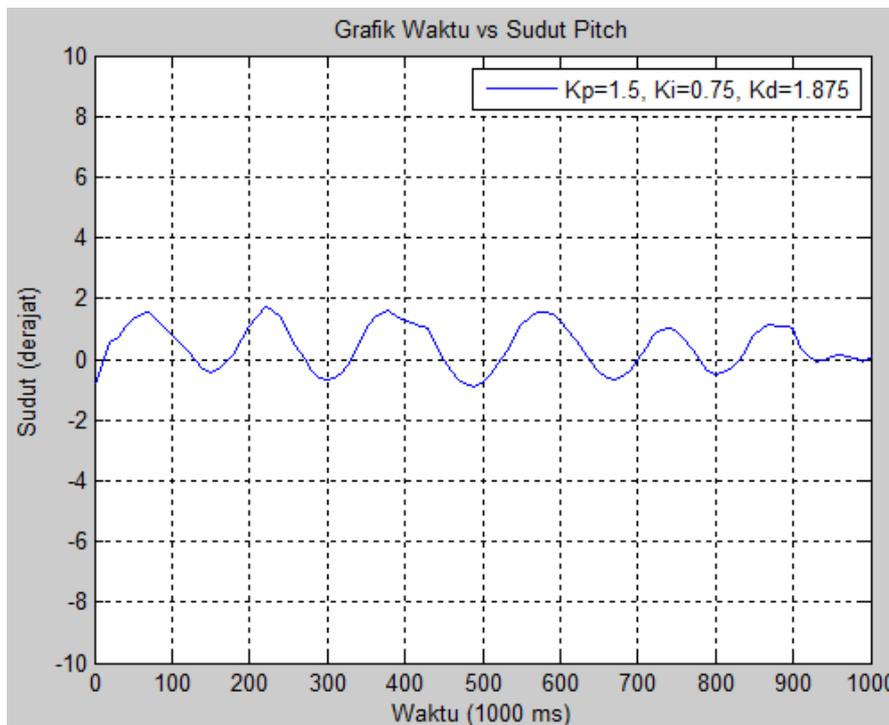
Berdasarkan grafik pada Gambar 10 nilai  $K_p=0.5$  menuju arah tak hingga,  $K_p=1$ ,  $K_p=1.5$ ,  $K_p=2$  osilasi yang terjadi masih belum stabil. Untuk Gambar 11 ketika  $K_p=3$ ,  $K_p=3.5$ ,  $K_p=4$  osilasi yang dihasilkan semakin lama semakin besar, sehingga nilai  $K_p$  yang mengalami osilasi yang cukup stabil adalah  $K_p=2.5$ .

Pengaruh konstanta proporsional dalam sistem adalah memperbesar nilai *overshoot* (suatu keadaan yang melebihi nilai batas atau lonjakan), semakin besar konstanta proporsional yang dimasukkan ke dalam sistem, maka nilai *overshoot* akan semakin besar pula dan sistem menjadi tidak stabil (jauh dari *setpoint*). Sehingga konstanta proporsional tidak dapat dipakai sendirian atau harus diimbangi dengan konstanta lain sebagai peredamannya [9]. Konstanta yang dimaksud adalah konstanta derivatif.

Setelah diketahui nilai  $K_p$ , maka dilakukan perhitungan nilai  $K_p$ ,  $K_i$ ,  $K_d$  berdasarkan metode tuning Ziegler-Nichols II yaitu Metode Osilasi pada Tabel 1, maka nilai  $K_p=2.5$  yang diperoleh pada Gambar 11 disimpan sebagai  $K_u$ , sehingga nilai  $K_p = 1.5$ . Untuk nilai periode osilasi  $P_u$  adalah 15 ms, sehingga nilai  $K_i = 0.75$  dan nilai  $K_d = 1.875$ .

#### Pengujian Kendali PID

Pengujian dilakukan dengan memberikan nilai  $K_p$ ,  $K_i$ , dan  $K_d$  yang diperoleh dari dengan menggunakan metode tuning Ziegler-Nichols sebelumnya.



Gambar 12 Grafik sudut *pitch* vs waktu ( $K_p=1.5$ ,  $K_i=0.75$ ,  $K_d=1.875$ )

Berdasarkan grafik pada Gambar 12, untuk analisis *response transient* nya dapat dilihat bahwa sistem mengalami osilasi diawal kemudian mempunyai *settling time* pada 1000 ms. *Rise time* dari sistem sekitar 100 ms. *Overshoot* yang dihasilkan cenderung kecil karena nilai  $K_d$  berfungsi untuk mengurangi *overshoot*. Lalu untuk osilasi yang masih terjadi diawal dikarenakan efek dari kontanta  $K_i$ . *Steady state error* yang dihasilkan sudah cukup kecil, tidak sampai  $1^\circ$ , sedangkan *set point* sendiri adalah  $0^\circ$ .

#### 4. KESIMPULAN

Berdasarkan penelitian yang dilakukan maka dapat diambil kesimpulan bahwa telah berhasil dibuat *balancing robot* dengan menggunakan sistem kendali PID untuk sudut *pitch* dengan nilai konstanta  $K_p = 1.5$ ,  $K_i = 0.75$ ,  $K_d = 1.875$  dan koefisien filter ( $\alpha$ ) pada algoritma *complementary filter* adalah 0.96.

#### 5. SARAN

1. Metode tuning konstanta kendali PID dilakukan dengan metode *auto tuning* agar hasil yang diperoleh lebih optimal.
2. Metode *fusion sensor* menggunakan algoritma lain seperti Kalman Filter atau DCM (*Discrete Cosine Matrix*) sebagai perbandingan dengan metode *complementary filter*.

#### DAFTAR PUSTAKA

- [1] Miller, P., 2008, Building a Two Wheeled Balancing Robot, *Disertasi*, Faculty of Engineering and Surveying, University of Southern Queensland, Queensland.
- [2] Stang, J., 2005, The Inverted Pendulum, *Tesis*, Teknik, Cornell University, Cornell.
- [3] Krakow, K.I., 2005, *System-specific PI Control Theory for Fluid and Motion Systems*, Universal Publishers, United States.
- [4] Hutama, I., 2011. Kendali Pendulum Terbalik Dinamis, *Skripsi*, Teknik, Universitas Gadjah Mada, Yogyakarta.
- [5] Sultan, K., 2003, Inverted Pendulum Analysis Design and Implementation, *Tesis*, Industrial Electronics, Institute of Industrial Electronics Engineering, Karachi.
- [6] Colton, S., 2007. *The Balance Filter: A Simple Solution for Integrating Accelerometer and Gyroscope Measurements for a Balancing Platform*, <http://web.mit.edu/scolton/www/filter.pdf>, diakses pada 20 Februari 2013.
- [7] Wicaksono, E.G., 2012, Balancing Robot Beroda Dua Menggunakan Metode Kendali Proporsional Derivatif, *Skripsi*, Teknik, Universitas Diponegoro, Semarang.
- [8] Ogata, K. 1997. *Teknik Kontrol Automatik* – terjemahan: Ir. Edi Laksono. Erlangga: Jakarta.
- [9] Dorf, R. C., dan H. Bishop, R., 2010, *Modern Control System Twelfth Edition*, Prentice Hall International, United Kingdom.