SISTEM KONTROL KESEIMBANGAN PADA *ELECTRIC UNICYCLE* KETIKA KONDISI PENGEREMAN MENGGUNAKAN METODE KALMAN FILTER

Balancing Control System For Electric Unicycle On Braking State Using Kalman Filter Method

Faisal Pakpahan[1], Porman Pangaribuan[2], Ramdhan Nugraha[3]

[1,2,3]Prodi S1 Teknik Elektro, Fakultas Teknik Elektro, Universitas Telkom

[1]pakpahan@students.telkomuniversity.ac.id, [2]pormanpangaribuan@telkomuniversity.ac.id, [3]ramdhan@telkomuniversity.ac.id

Abstrak

Electric Unicycle **merupakan kendaraan sepeda motor roda satu yang menggunakan energi listrik sebagai sumber tenaganya. Keunggulan dari kendaraan ini adalah menggunakan teknologi *self-balancing* yang memudahkan saat dikendarai karena sistem kendali yang otomatis. Teknologi *self-balancing* ini juga dipilih karena memiliki tingkat keamanan yang tinggi sehingga pengendara tidak perlu khawatir terjatuh.**

Teknologi *self-balancing* pada kendaraan ini menggunakan kombinasi sensor *gyroscope* dan *accelerometer* yang tercakup pada modul GY521 MPU6050. Modul ini menghasilkan *output* berupa informasi kemiringan sudut kendaraan dengan tolak ukur garis normal gravitasi terhadap bumi. *Output* ini selanjutnya difilter, dengan metode Kalman Filter, menggunakan Arduino Mega. Setelah data diolah pada Arduino Mega, data tersebut dijadikan sebagai akan langsung mempengaruhi kecepatan motor *Brushless DC.* Metode yang dipakai untuk mengolah kecepatan motor BLDC adalah metode *Fuzzy Logic Control.*

Dengan mencondongkan badan kedepan, maka kendaraan akan otomatis bergerak maju. Demikian pula untuk pengereman, pengendara hanya perlu menggerakkan badan ke arah yang dituju. Parameter Kalman filter yang diimplementasikan pada tugas akhir ini adalah $Q_{accelerometer}$ = 0.001, Q_{bias} = 0.003 dan R = 0.03. Sementara parameter *Fuzzy Logic* yang digunakan adalah dengan *output* PWM dengan rentang nilai 60 sampai 80.

Kata kunci: ***Accelerometer, Gyroscope,*** **MPU6050,** ***Electric Unicycle,*** **Motor** ***Brushless DC, Kalman Filter***

Abstract

Electric Unicycle is a motorcycle wheel one that uses electrical energy as its power source. The advantages of this vehicle is using technologies that facilitate self-balancing while driving because the automatic control system. Self-balancing technology was also chosen because it has a high level of security so that the rider does not have to worry about falling.

Self-balancing technology on these vehicles use a combination of gyroscope and accelerometer sensors are included in the module MPU6050 GY521. These modules generate output information tilt angle of the vehicle with a benchmark of the normal line to the earth's gravity. The output is then filtered by Kalman Filter, using the Arduino Mega. Once the data is processed on the Arduino Mega, which will serve as the data will directly affect Brushless DC motor speed. The method used to process the BLDC motor speed is Fuzzy Logic Control.

By leaned forward, then the vehicle will automatically move forward. Similarly to braking, the driver only needs to move the body to the intended direction. Parameter Kalman filter is implemented in this thesis is Qaccelerometer = 0.001, Qbias = 0.003 and R = 0:03. While the parameters of Fuzzy Logic is used with PWM output with a range of values from 60 to 80.

Keywords: Accelerometer, Gyroscope, MPU6050, Electric Unicycle, Brushless DC Motor, Kalman Filter

1. Pendahuluan

Perkembangan teknologi yang sangat pesat menyebabkan kebutuhan manusia semakin meningkat. Menurut Jacques Ellul, teknologi adalah suatu metode menyeluruh yang secara rasional mengarah dan memiliki ciri efisiensi dalam setiap bidang kegiatan manusia (Ellul, 1964, p. xxv). Dalam hal mobilisasi, manusia akan terus berinovasi untuk menciptakan suatu kendaraan yang sangat fungsional, memiliki fleksibilitas yang tinggi, ramah lingkungan, serta memiliki pergerakan yang efisien. Beberapa negara maju sudah mulai mengembangkan suatu teknologi berupa alat/kendaraan yang diciptakan untuk efisiensi waktu. Kendaraan ini akan mempersingkat waktu jarak tempuh yang diperlukan saat melakukan perpindahan dari suatu tempat ke tempat lainnya,

Electric Unicycle merupakan kendaraan sepeda motor roda satu yang menggunakan energi listrik sebagai sumber tenaganya. Keunggulan dari kendaraan ini adalah menggunakan teknologi *self-balancing* yang memudahkan saat dikendarai karena sistem kendali yang otomatis. Teknologi *self-balancing* ini juga dipilih karena memiliki keamanan yang tinggi sehingga pengendara tidak perlu khawatir terjatuh saat mengendarainya.

Kendaraan ini memiliki banyak manfaat dalam hal mobilisasi perorangan. Di berbagai negara maju kendaraan ini sudah dioperasikan di bandar udara, tidak menutup kemungkinan kendaraan ini bisa dioperasikan di rumah sakit, gedung perkantoran, bahkan lingkungan kampus. Keunggulan lain dari kendaraan ini adalah bisa beroperasi di jalan yang sempit (gang) karena ukurannya yang kecil, serta bisa dioperasikan di supermarket dan trotoar dimana ada *sign* bahwa *scooter* diperbolehkan beroperasi.

2. Dasar Teori dan Perancangan

2.1. Perancangan Sistem

Gambar 1. Diagram Blok Sistem

Modul GY521 MPU6050, yang terdapat pada *Electric Unicycle*, terdiri dari 3 aksis *gyroscope* dan 3 aksis *accelerometer*. *Output* modul ini berupa informasi perubahan sudut yang akan mempengaruhi kecepatan kendaraan. Perubahan sudut kemiringan kendaraan diukur berdasarkan tolak ukur garis normal gravitasi terhadap bumi.

Data *output* MPU6050 akan dikirim dan diproses oleh Arduino UNO. Data tersebut terlebih dahulu akan mengalami proses filterisasi agar menghasilkan data *output* yang minim *noise*. Metode yang digunakan untuk filterisasi adalah Kalman Filter. Begitu selesai proses filterisasi, data tersebut akan menjadi nilai masukan *fuzzy*.

Setelah melalu tahap fuzzifikasi, *fuzzy rule interference*, dan defuzzifukasi, maka *output* dari proses *fuzzy* yang berupa nilai PWM (Pulse Width Module) akan dijadikan sebagai pengatur kecepatan motor BLDC. Nilai PWM yang merupakan sinyal *digital* harus dikonversi menjadi sinyal *analog* menggunakan rangkaian DAC (*Digital to Analog Converter*). Sebab *controller* BLDC hanya bisa menerima masukan berupa sinyal *analog*. Nilai *analog* berupa tegangan (*voltage*) inilah yang masuk ke *controller* BLDC untuk mengatur kecepatan motor BLDC.

Gambar 2. Diagram Alir Sistem

2.2. MPU6050 dengan Kalman Filter

Kalman Filter adalah filter digital rekursif yang dapat memprediksi dengan sangat efektif untuk proses apapun. Kalman Filter dapat digunakan sebagai *state* estimator. Kalman Filter juga dapat digunakan sebagai metode yang handal untuk mengestimasi keluaran suatu proses. Kegunaan lain dari Kalman Filter ini adalah mampu meminimalkan *mean of square error* dan membantu *least square method*.

Filter ini bertugas mengurangi *noise* dari sensor sebelum data *sensor* tersebut dikirim ke sebuah sistem kontrol. Hasil pengukuran sensor yang memiliki *noise* membuat sistem susah stabil dan memiliki respon yang buruk. Seperti yang dikatakan Rich Ooi, Kalman Filter tidak butuh data untuk disimpan dan data tersebut di proses ulang setiap pengukuran.

Kalman filter dibagi kedalam 2 bagian utama yaitu *time update* dan *measurement update*. *Time update* bisa disebut juga sebagai proses *predict*, yaitu menggunakan *estimation state* dari waktu sebelumnya untuk mendapatkan sebuah *estimation state* saat ini. *Measurement update* bisa disebut juga sebagai proses *correct*, yaitu menggunakan informasi pengukuran saat ini untuk memperbaiki prediksi dengan tujuan untuk mendapatkan *estimation state* yang akurat. Kalman filter akan terus menerus melakukan iterasi *predict – correct* seperti pada gambar 3.

Gambar 3. Kalman Flter

Persamaan – persamaan yang terdapat dalam blok predict dan correct adalah sebagai berikut :

Time update (Predict)

Predicted state $\hat{x}_{k|k-1} = A\hat{x}_{k-1|k-1} + Bu_k$ (1)

Predicted error covariance $P_{k|k-1} = AP_{k-1|k-1}A^T + Q$ (2)

Measurements update (Correct)

Update the estimate via z_k $\hat{x}_k = \hat{x}_{k|k-1} + K_k\tilde{y}_k$ (3)

Kalman gain $K_k = P_{k|k-1}H^T S_k^{-1}$ (4)

Update the error covariance $P_{k|k} = (I - K_k H)P_{k|k-1}$ (5)

$\hat{x}$ *Estimated State* — *R = Measurement variance matrix*
B = Control matrix — *z = measurement variabels*
P = Estimation error covariance — *u = control variabels*
Q = Process variance matrix — *H = measurement matrix*
K = Kalman gain

Kalman Filter diimplementasikan untuk proses penyaringan *noise* data *output* dari MPU6050 sebelum data tersebut diolah oleh mikrokontroler. Sebelum Kalman Filter dapat digunakan untuk menghilangkan *noise* pada hasil pembacaan MPU6050. Hal yang harus dilakukan pertama kali adalah memodelkan sistem persamaan Kalamn Filter tersebut.

$$\hat{x}_{k+1} = A\hat{x}_{k+1|k+1} + Bu_k \quad (6)$$

$$\begin{bmatrix}\theta \\ \dot{\theta}\end{bmatrix}_{k|k-1} = \begin{bmatrix}1 & -\Delta t \\ 0 & 1\end{bmatrix}\begin{bmatrix}\theta \\ \dot{\theta}\end{bmatrix}_{k-1|k-1} + \begin{bmatrix}\Delta t \\ 0\end{bmatrix}\dot{\theta}_k$$

$$\begin{bmatrix}\theta \\ \dot{\theta}\end{bmatrix}_{k|k-1} = \begin{bmatrix}\theta + \Delta t(\theta - \dot{\theta}_k) \\ \dot{\theta}_k\end{bmatrix} \quad (7)$$

Nilai *prior estimate* pembacaan sudut ($\hat{\theta}_{k+1}$) diperoleh berdasarkan estimasi persamaan sebelumnya ($\hat{\theta}_{k-1|k-1}$) dijumlahkan dengan bias dari persamaan tersebut (Δt). Langkah selanjutnya adalah memperbaharui nilai *error covariance* berdasarkan nila *error covariance* pada persamaan sebelumnya.

$$P_{k|k-1} = AP_{k-1|k-1}A^T + Q \quad (8)$$

$$\begin{bmatrix}P_{00} & P_{01} \\ P_{10} & P_{11}\end{bmatrix}_{k|k-1} = \begin{bmatrix}1 & -\Delta t \\ 0 & 1\end{bmatrix}\begin{bmatrix}P_{00} & P_{01} \\ P_{10} & P_{11}\end{bmatrix}_{k-1|k-1}\begin{bmatrix}1 & 0 \\ -\Delta t & 1\end{bmatrix} + \begin{bmatrix}Q_\theta & 0 \\ 0 & Q_{\dot{\theta}}\end{bmatrix}\Delta t$$

$$\begin{bmatrix}P_{00} & P_{01} \\ \end{bmatrix} = \begin{bmatrix}P_{00} + \Delta t(\Delta t P_{11} - P_{01} - P_{10} + Q_\theta) & P_{01} - \Delta t P_{11}\end{bmatrix}$$

$$[\,\theta_{10} \quad \theta_{11}\,]_{i-1} = [\,\theta_{10} - \Delta\theta_{11} \qquad \theta_{11} + \Delta\theta\,] \tag{9}$$

Setelah mendapatkan hasil perhitungan *error covariance*, Kalman Filter akan melakukan koreksi terhadap nilai nilai yang sudah diperbaharui. Koreksi tersebut dilakukan dengan cara menghitung *Kalman Gain*. Namun, sebelum menghitung nilai *Kalman Gain*, harus dihitung terlebih dahulu nilai eror estimasi.

$$P_k = C P_{k|k-1} C^T + R \tag{10}$$

$$= \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} P_{00} & P_{01} \\ P_{10} & P_{11} \end{bmatrix}_{k|k-1} \begin{bmatrix} 1 \\ 0 \end{bmatrix} + R$$

$$= P_{00\,k|k-1} + R$$

$$P_k = P_{00\,k|k-1} + var(w_k) \tag{11}$$

Salah satu faktor yang mempengaruhi hasil perhitungan hasil *noise* (w_k) adalah nilai *covariance* (R). Pada persamaan diatas, nilai R diubah dengan nilai $var(w_k)$ karena *covariance* sama besar nilainya dengan nilai *variance* ($var(w_k)$). Nilai $var(w_k) = var(w_k)$ karena tidak dipengaruhi oleh waktu. Jika nilai masukan $var(w_k)$ terlalu tinggi, maka respon filter akan sangat lambat. Jika nilai $var(w_k)$ terlalu kecil, nilainya akan *overshoot* dan banyak *noise* karena akan terlalu banyak data hasil pengukuran dari *accelerometer* yang masuk ke sistem. Hasil perhitungan dari persamaan 11 berupa nilai *error estimate*.

$$K_k = P_{k|k-1} C^T S_k^{-1} \tag{12}$$

$$\begin{bmatrix} K_0 \\ K_1 \end{bmatrix}_k = \begin{bmatrix} P_{00} & P_{01} \\ P_{10} & P_{11} \end{bmatrix}_{k|k-1} \begin{bmatrix} 1 \\ 0 \end{bmatrix} S_k^{-1} \tag{13}$$

$$= \begin{bmatrix} P_{00} \\ P_{10} \end{bmatrix}_{k|k-1} S_k$$

Persamaan 12 merupakan persamaan yang digunakan untuk menghitung nilai *Kalman Gain*. Jika nilai matrix *error covariance* (P), dan nilai R juga kecil, maka nilai *Kalman Gain* juga akan mengecil.

$$\hat{P}_k = P_k - K_k C P_{k|k-1} \tag{14}$$

$$= P_k - \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} P_{00} \\ P_{10} \end{bmatrix}_{k|k-1}$$

$$= P_k - P_{00\,k|k-1}$$

Pada persamaan 14, estimasi persamaan diperbaharui dengan perhitungan nilai sudut terbaru dan biasnya.

$$\hat{x}_k = \hat{x}_{k|k-1} + K_k \tilde{y}_k \tag{15}$$

$$\begin{bmatrix} \theta \\ b \end{bmatrix}_{k|k} = \begin{bmatrix} \theta \\ b \end{bmatrix}_{k|k-1} + \begin{bmatrix} K_0 \\ K_1 \end{bmatrix}_k \tilde{y}_k \tag{16}$$

$$= \begin{bmatrix} \theta \\ b \end{bmatrix}_{k|k-1} + \begin{bmatrix} K_0 \tilde{y}_k \\ K_1 \tilde{y}_k \end{bmatrix}$$

Pada persamaan 16, *posterior state* merupakan matrix 2$\times$ yang dengan komposisi hasil pembacaan *accelerometer* (θ) dan pembacaan *gyroscope* (b) diperoleh dari *prior state*, *Kalman Gain*, serta persamaan $\hat{P}_k$.

$$P_{k|k} = (I - K_k C) P_{k|k-1} \tag{17}$$

$$\begin{bmatrix} P_{00} & P_{01} \\ P_{10} & P_{11} \end{bmatrix}_k = \left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} K_0 \\ K_1 \end{bmatrix}_k \begin{bmatrix} 1 & 0 \end{bmatrix} \right) \begin{bmatrix} P_{00} & P_{01} \\ P_{10} & P_{11} \end{bmatrix}_{k|k-1} \tag{18}$$

$$= \begin{bmatrix} P_{00} & P_{01} \\ P_{10} & P_{11} \end{bmatrix}_{k|k-1} - \begin{bmatrix} K_0 P_{00} & K_0 P_{01} \\ K_1 P_{00} & K_1 P_{01} \end{bmatrix}$$

Tahap terakhir adalah menghitung nilai *error covariance* dan memperbaharui niali tersebut.

Setelah melalui tahap-tahap diatas, sistem akan melakukan *looping* dan mengkoreksi sendiri nilai *error covariance* yang dihasilkan berdasarkan hasil perhitungan dari tahap-tahap diatas.

2.3. Perancangan *Fuzzy Logic Control*

Fuzzy Logic adalah suatu cara yang tepat untuk memetakan suatu ruang *input* ke dalam suatu ruang *output*, dan mempunyai nilai yang kontinyu. Fuzzy dinyatakan dalam derajat dari suatu keanggotaan dan derajat dari kebenaran. Oleh sebab itu sesuatu dapat dikatakan sebagian benar dan sebagian salah pada waktu yang sama (Purnomo & Kusumadewi, 2004). Logika fuzzy merupakan representasi suatu pengetahuan yang dikonstruksikan dengan perintah *if-then rules*. Karakteristik dari metode *fuzzy logic* adalah sebagai berikut :

- Pemecahan masalah pada sistem bukan lewat angka-angka, melainkan dengan cara linguistik, atau melalui variabel-variabel yang mengandung unsur ketidakpastian.
- Penggunaan perintah *if-then rules* untuk menjelaskan keterkaitan antar variabel.

Tahapan *Fuzzy Logic* pada penelitian ini adalah sebagai berikut:

a. Fuzzifikasi

Fuzzifikasi merupakan proses awal dari perancangan *Fuzzy Logic Control*. Pada tahapan ini akan dibuat suatu fungsi keanggotaan dari nilai masukan serta menentukan banyaknya nilai linguistik dari fungsi keanggotaan tersebut.

Pada penelitian ini, proses fuzifikasi memiliki dua buah *input* dengan masing-masing *input* memiliki nilai linguistik sebanyak tiga buah. *Input* pada fuzzifikasi adalah berupa nilai eror pembacaan sudut (*Err_Angle*) dan nilai *delta error* pembacaan sudut (*DeltaErr_Angle*). Sedangkan *output*-nya berupa nilai PWM (Kec). Berikut adalah fungsi keanggotaan dari setiap *input* dan *output*.

Gambar 4. Fungsi Keanggotaan *ErrorAngle*

Gambar 5. Fungsi Keanggotaan *DeltaErrorAngle*

Gambar 6. Fungsi Keanggotaan *Output* (PWM)

b. Rule Inference

Pada proses ini terjadi pengolahan data masukan dengan hasil keluaran yang diinginkan berdasarkan aturan fuzzy. Aturan fuzzy tersebut akan di tuliskan pada tabel di bawah ini.

Tabel 1. *Rule Fuzzy*

Input		***Error Angle***				
		R.E5	**R.E4**	**R.E3**	**R.E2**	**R.E1**
Delta Error Angle	**R.De5**	*Brake5*	*Brake5*	*Brake5*	*Brake4*	*Brake3*
	R.De4	*Brake5*	*Brake4*	*Brake4*	*Brake3*	*Brake2*
	R.De3	*Brake5*	*Brake4*	*Brake3*	*Brake2*	*Brake2*
	R.De2	*Brake4*	*Brake4*	*Brake3*	*Brake2*	*Brake1*
	R.De1	*Brake4*	*Brake3*	*Brake2*	*Brake1*	*Brake1*

Berdasarkan tabel diatas, terdapat sebanyak sembilan aturan fuzzy, yaitu:

1. *IF* ErrorAngle = R.E1 *AND* DeltaErrorAngle = R.De1 *THEN* Kec = *Brake1*
2. *IF* ErrorAngle = R.E1 *AND* DeltaErrorAng = R.De2 *THEN* Kec = *Brake1*

...

24. *IF* ErrorAngle = R.E5 *AND* DeltaErrorAng = R.De4 *THEN* Kec = *Brake5*
25. *IF* ErrorAngle = R.E5 *AND* DeltaErrorAng = R.De5 *THEN* Kec = *Brake5*

Nilai *output* yang dihasilkan adalah nilai PWM. Nilai PWM yang dipakai untuk mengendalikan putaran balik motor BLDC adalah sebesar 64 sampai dengan 80. Kondisi *Brake1* memiliki nilai PWM sebesar 64, pwm *Brake2* sebesar 68, *Brake3* memiliki nilai pwm sebesar 72, *Brake4* sebesar 76, dan *Brake5* sebesar 80.

c. Defuzzifikasi

Proses defuzzifikasi merupakan suatu proses untuk mengubah nilai output fuzzy yang telah didapat dari prosesn inferensi menjadi suatu nilai yang tegas (*crisp*). Pada pengerjaan penelitian ini proses defuzzyfikasi menggunakan metode *Center of Gravity*. Proses penghitungan COG seluruh keluaran fuzzy untuk variable keluaran yang diberikan untuk menentukan besarnya nilai keluaran *crisp*.

3. Pengujian dan Analisis

3.1. Pengujian MPU6050 dengan Kalman Filter

Pengujian MPU6050 dengan Kalman Filter bertujuan untuk mengetahui apakah metode Kalman Filter berfungsi dengan baik pada hasil pembacaan MPU6050. Serta membandingkan hasil pembacaan MPU6050 tanpa filter dengan hasil pembacaan MPU6050 yang *output* nya sudah difilter dengan Kalman Filter.

Gambar 7. Pembacaan MPU6050

Berdasarkan pada penelitian yang dilakukan oleh Lauszus (2012), nilai [illegible] dan [illegible] yang paling efektif diimplementasikan pada persamaan Kalman Filter secara berturut turut adalah 0.001, 0.003, dan 0.03. Nilai-nilai tersebut akan menghasilkan *output* dari Kalman Filter yang paling efektif. Dapat dilihat dalam grafik di atas, keluaran hasil filter Kalman menunjukkan bahwa *noise* pembacaan *accelerometer* dan *gyroscope* berhasil diminimalisir dengan *sensor fusion*, tanpa membuat pembacaan sudut menjadi bias.

Berdasarkan hasil pengujian, Kalman Filter sangat efektif diimplementasikan pada sistem yang sudah dirancang. Jika dibandingkan dengan hasil pembacaan MPU6050 tanpa filter, *output* yang dihasilkan Kalman Filter sangat baik, karena pemodelan untuk prediksi sudut yang akan datang sangat membantu untuk meminimalisir *noise* pembacaan yang dihasilkan MPU6050.

3.2. Respon Pembacaan MPU6050 Terhadap Kemiringan Kendaraan

Pengujian ini dilakukan untuk mengetahui respon MPU6050 terhadap waktu. Berdasarkan pengujian ini, akan terlihat apakah ada data yang hilang atau tidak terbaca oleh sensor saat kendaraan diubah posisi nya dengan lama waktu perubahan kemiringan kendaraan yang berbeda-beda. Pengujian dilakukan dengan cara melihat respon MPU6050 berdasarkan kemiringan *Electric Unicycle*. Pengujian dilakukan dengan mengubah posisi *Electric Unicycle* yang awalnya berada pada kemiringan 15^0, diubah sampai berada pada kemiringan -15^0 dengan rentang waktu tertentu. Rentang waktu perpindahan *Electric Unicycle* pada pengujian adalah dengan waktu 10 detik, dan 5 detik.

Gambar 8. Respon MPU6050 dengan Waktu 10 Detik

Gambar 9. Respon MPU6050 dengan Waktu 5 Detik

3.3. Respon Pengereman dengan Beban Pengendara

Pengujian dilakukan dengan cara memberikan kecepatan awal kendaraan yang berbeda-beda. Ada tiga kecepatan awal yang diberikan, yaitu 9.2 m/s, 7.2 m/s, dan 6.28 m/s. Respon pengereman pada pengujian ini dilihat dari waktu tempuh kendaraan untuk benar-benar berhenti setelah memasuki sudut pengereman (sudut negatif), dan jarak yang diukur saat pengereman berlangsung.

Tabel 2. Respon Pengereman

Pengujian	PWM	Kecepatan Awal	Lama Pengereman	Jarak Pengereman
1	228	9.2 ⁄	2.2 detik	61 cm
2	127	7.2 ⁄	1.7 detik	45 cm
3	117	6.28 ⁄	1 detik	28 cm

Berdasarkan tabel 4.14, respon pengereman dengan adanya beban pengendara sangat berbeda jauh dengan respon pengereman tanpa ada beban pengendara.

4. Kesimpulan

Berdasarkan hasil pengujian dan analisa kinerja sistem yang dirancang, diperoleh beberapa kesimpulan sebagai berikut:

1. Pengimplementasian modul MPU6050 sangat tepat, karena modul ini sudah mencakup *accelerometer* dan *gyroscope*. Kedua jenis IMU ini saling bekerja sama. Dengan menggunakan modul MPU6050 pada penelitian ini, tugas *accelerometer* ialah memberikan pengukuran sudut saat sistem berada pada kondisi diam. Sedangkan pada saat sistem berotasi, *accelerometer* tidak bisa bekerja secara maksimal karena memiliki respon yang lambat. Kelemahan inilah yang dapat diatasi oleh *gyroscope* karena dapat membaca kecepatan sudut yang dinamis. Namun *gyroscope* juga memiliki kelemahan yaitu proses perpindahan kecepatan sudut dalam jangka waktu yang panjang menjadi tidak akurat karena ada efek *bias* yang dihasilkan oleh *gyroscope* itu sendiri.
2. Penggunaan metode Kalman Filter sangat efektif. Karena sifat pembacaan MPU6050 yang sensitif terhadap pergerakan dan *noise* hasil pembacaan MPU6050 sangat banyak. Dengan Kalman Filter, *noise* tersebut diminimalkan dan prediksi pembacaan MPU6050 berdasarkan persamaan Kalman Filter sangat membantu sehingga hasil *output* MPU6050 sesuai keinginan.
3. Berdasarkan data pengujian *Fuzzy Logic Control*, pemberian PWM untuk menghasilkan putaran balik motor BLDC kurang tepat. Alasannya adalah karakteristik motor BLDC yang dipakai tidak bisa menghasilkan putaran balik secara mendadak. Jadi saat kondisi motor BLDC dalam keadaan maju dan akan mundur, motor BLDC harus benar-benar tidak memiliki tegangan sisa dari tegangan PWM pada motor BLDC saat kondisi maju. Pengosongan tersebut harus dilalui, dan setelah benar-benar kosong, maka motor BLDC bisa menjalankan PWM mundur motor BLDC. Lama pengosongan tegangan sisa tersebut tergantung kecepatan awal motor BLDC. Semakin besar kecepatan motor BLDC, maka semakin lama waktu yang dibutuhkan untuk pengosongan tegangan sisa tersebut.

Daftar Pustaka :

[1] Ellul, J. 1964. *The Technological Society,* Translated by John Wilkinson. Random House, Note to the Reader, p. xxv.

[2] Eliezer, I. P. G., 2013. *Accelerometer dan Gyroscope.* [Online], http://www.geyosoft.com/2013/accelerometer-dan-gyroscope, diakses pada tanggal 29 November 2015.

[3] Hoffman, C., 2014. *TAKING A SPIN ON RYNO'S ONE-WHEELED, SELF-BALANCING ELECTRIC 'MICROCYCLE'* [Interview] (21 January 2014).

[4] Ernie, 2009. *Logika Fuzzy.* [Online], http://ndoware.com/fuzzy-logic-logika-fuzzy-bagian-i.html, diakses pada tanggal 7 Desember 2015.

[5] Kalman, R. E., 1960. A New Approach to Linear Filtering and Prediction Problems. *ASME - Journal Of Basic Engineering,* pp. 35-45.

[6] Lauzuz, Kristian Sloth, 2012. *A Pratical Approach to Kalman Filter and How to Implement It*. [Online] http://blog.tkjelectronics.dk/2012/09/a-practical-approach-to-kalman-filter-and-how-to-implement-it/, diakses pada tanggal 17 April 2016.

[7] Setiawan, W., 2014. *Cara kerja Accelerometer.* [Online], https://wirasetiawan29.wordpress.com/2014/04/06/accelerometer/, diakses pada tanggal 28 November 2015.