

**PERANCANGAN DAN IMPLEMENTASI SISTEM KENDALI GERAK PADA
SKUTER DUA RODA SEIMBANG OTOMATIS MENGGUNAKAN METODE PID
BERBASIS MIKROKONTROLLER**

UNIVERSITAS TELKOM

**(DESIGN AND IMPLEMENTATION OF CONTROL SYSTEM DRIVING SELF
BALANCE SCOOTER ON TWO-WHEELS USING PID BASED ON
MICROCONTROLLER)**

TELKOM UNIVERSITY

Fikri Dzuraeka Putra¹, Mohammad Ramdhani,ST.MT.², Cahyantari Ekaputri,ST.MT.³

^{1,2}Prodi S1 Teknik Elektro, Fakultas Teknik Elektro, Universitas Telkom

¹fikridzuraeka@students.telkomuniversity.ac.id

²mohamadramdhani@telkomuniversity.ac.id,

³cahyantarie@telkomuniversity.ac.id

Abstrak

Teknologi pada bidang transportasi saat ini berkembang sangat pesat, ditunjukkan oleh munculnya sebuah kendaraan canggih dan praktis, yaitu *self balance scooter*. *Self balance scooter* disebut sebagai skuter seimbang karena kemampuan untuk berdiri seimbang dengan ditopang oleh dua buah roda di sisi kanan dan kiri. *Self balance scooter* dikendalikan menggunakan sistem kendali PID, sehingga *self balance scooter* dapat berjalan maju dan mundur. Pada tugas akhir ini, akan dirancang dari sebuah *Self balance Scooter*. Dengan menggunakan *mikrokontroler* sebagai pusat sistem kendali dengan pengolahan kecepatan motor menggunakan metode PID. Sensor pendukung yang digunakan antara lain sensor percepatan sudut menggunakan *accelerometer* dan sensor sudut menggunakan *gyroscope*, sehingga *Self balance Scooter* dapat berjalan dengan menggunakan massa tubuh sebagai acuan gerakannya. Setelah mendapat nilai pembacaan yang baik, data akan diproses dengan Kontrolir PID untuk mengatur arah dan kecepatan, sehingga sistem *self balance scooter* ini dapat berjalan maju dan mundur. Dengan optimal daya yang dikeluarkan sedikit, sehingga mampu bertahan hingga waktu yang lama. Dari hasil percobaan PID, diperoleh nilai yang optimal dari parameter PID, $K_p = 8.0$; $K_i = 3.0$ dan $K_d = 6.0$.

Kata kunci: *PID, mikrokontroler, accelerometer, gyroscope, self balance scooter.*

Abstract

The technology of transportation sector is currently growing very rapidly, indicated by the appearance of a sophisticated and practical vehicle, the self-balance scooter. Self balance scooter called balanced scooter because the ability to stand balanced, sustained by two wheels on the right and left. Self balance scooter controlled using PID control system, so that self-balance scooter can walk forward and backward. In this thesis, the authors designed a Self Balance Scooter using a microcontroller as the central control system with a processing speed of the motor using PID method. The support sensor used include an angular acceleration sensor using the accelerometer and gyroscope as angle sensor, Self balance scooter can run by using the body mass as a reference motion. After receiving the value of sensor readings, the data will be processed by the PID controller to set direction and speed, so the system of self-balance scooter can walk forward and backward. With optimal of power output, so it was able to survive until a long time. PID of the experimental results, obtained optimal values of PID parameters, $K_p = 8.0$; $K_i = 3.0$ and $K_d = 6.0$.

Keywords: *PID, mikrokontroler, accelerometer, gyroscope, self balance scooter.*

1. Pendahuluan

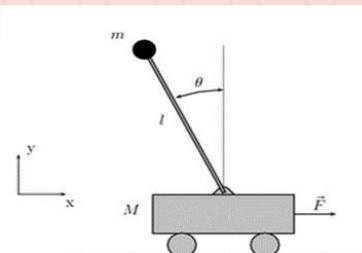
Self balance scooter adalah suatu alat yang memiliki dua roda yang tidak akan seimbang apabila dalam keadaan diam, yang merupakan pengembangan dari model pendulum terbalik. Sistem teknologi *self balance scooter* menggunakan mikroprosesor dan rangkaian sensor yang berguna untuk menjaga keseimbangan pengendara. *Self balance scooter* karena kemampuannya untuk berdiri seimbang walaupun hanya ditopang oleh dua buah roda di sisi kanan dan kirinya, dapat juga disebut robot karena kemampuannya untuk mempertahankan keadaan agar tetap seimbang.

Pada tugas akhir kali ini digunakan pengendali gerak menggunakan metode PID, dengan menggunakan catu daya baterai dan sel surya sehingga *self balance scooter* akan berjalan maju dan mundur terhadap jalan dan sensor Inertial Measurement Unit sebagai sensor keseimbangan saat berjalan.

2. Dasar Teori

2.1 Self Balance Scooter

Self Balance Scooter beroda dua merupakan suatu kendaraan yang memiliki dua roda di kedua sisinya dan tidak akan seimbang tanpa sebuah metode kontrol yang baik. Saat *Self Balance Scooter* beroda dua condong ke depan atau miring ke kanan, maka yang perlu dilakukan adalah motor akan memutar searah jarum jam sehingga *Self Balance Scooter* beroda dua akan berputar ke arah depan. Gaya yang digunakan untuk menyeimbangkan dihasilkan dari putaran roda.[2] Pada gambar 2.1 adalah gambar pemodelan *self balance scooter*.



Gambar 2.1 Pemodelan *self balance scooter*.

2.2 Sensor

Sensor merupakan piranti untuk mengkonversi energi yang berasal dari kejadian fisik menjadi arus listrik atau tegangan untuk keperluan pengukuran, kontrol, atau informasi. Sensor terbagi dalam beberapa macam seperti: sensor thermal, sensor optik, dan sensor mekanik.

2.2.1 Gyroscope

Gyroscope adalah sebuah perangkat yang digunakan untuk mengukur atau mempertahankan orientasi dari sebuah objek. *Gyroscope* merupakan sebuah roda berat yang berputar pada jari-jarinya. Sebuah giroskop mekanis terdiri dari sebuah roda yang diletakkan pada sebuah bingkai. Roda ini berada di sebuah batang besi yang disebut dengan poros roda (*spin axis*). Ketika *gyroscope* digerakkan, maka ia akan bergerak mengitari porosnya. Poros tersebut terhubung dengan lingkaran-lingkaran yang disebut gimbal. Gimbal tersebut juga terhubung dengan gimbal lainnya pada dasar lempengan. Jadi saat piringan itu berputar, unit *gyroscope* itu akan tetap menjaga posisinya seperti pada saat pertama kali *gyroscope* diputar.

2.2.2 Accelerometer

Accelerometer adalah perangkat yang berfungsi untuk mengukur akselerasi. Akselerasi yang diukur dengan *accelerometer* belum tentu memiliki laju perubahan velositas. Sebaliknya, *accelerometer* mendapatkan akselerasi yang dimaksud dengan fenomena berat yang dialami oleh uji massa pada kerangka acuan perangkat *accelerometer*. Sebagai contoh, *accelerometer* di permukaan bumi akan mengukur akselerasi $g = 9.81 \text{ m/s}^2$ lurus ke atas karena beratnya. Sebaliknya, *accelerometer* jatuh bebas ke bumi mengukur nol untuk akselerasinya.

2.2.3 Inertial Measurement Unit (IMU)

sensor MPU-6050. IMU digital ini memiliki 3-axis *accelerometer* dan 3-axis *gyroscope* yang mana sensor ini terhubung secara I2C, dan ketika dikombinasikan dengan filter, akan menampilkan pembacaan kemiringan yang sangat stabil.^[1]

2.3 Motor DC

Pada *self balance scooter* ini menggunakan motor DC kursi roda karena pemakaian daya rendah, serta efisiensi tinggi saat pemakaian motor DC. Motor DC berfungsi sebagai motor penggerak roda bagian kanan dan kiri pada *self balance scooter*.

2.4 Driver Motor

Fungsi *driver motor* yaitu untuk menjalankan motor sebagai mengatur arah putaran motor maupun kecepatan putaran motor dan digunakan *driver motor* karena arus yang keluar dari mikrokontroler tidak mampu memenuhi

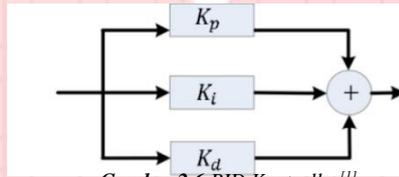
kebutuhan motor DC, serta mengubah tegangan yang dikeluarkan mikrokontroler agar sesuai dengan tegangan yang dibutuhkan motor tersebut.

2.5 Mikrokontroler

Mikrokontroler adalah IC yang dapat diprogram berulang kali, baik ditulis atau dihapus. Biasanya digunakan untuk pengontrolan otomatis dan manual pada perangkat elektronika. Rata-rata *mikrokontroler* memiliki instruksi manipulasi bit, akses ke I/O secara langsung dan mudah, dan proses *interrupt* yang cepat dan efisien.

2.6 PID

Sistem Kontrol PID (*Proportional-Integral-Derivative controller*) merupakan kontroler untuk menentukan presisi suatu sistem instrumentasi dengan karakteristik adanya umpan balik pada sistem tersebut (Feed back). Sistem kontrol PID terdiri dari tiga buah cara pengaturan yaitu kontrol P (*Proportional*), D (*Derivative*) dan I (*Integral*), dengan masing-masing memiliki kelebihan dan kekurangan.^[1]



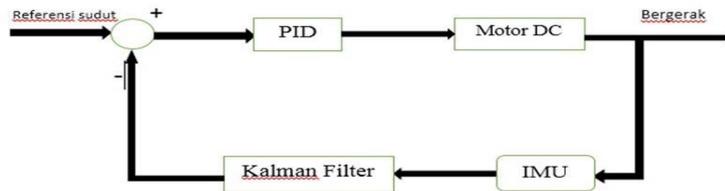
Gambar 2.6 PID Kontroler^[1]

2.7 Kalman Filter

Kalman filter merupakan filter digital rekursif yang dapat mengestimasi proses dengan sangat efektif. Kalman filter dapat mengurangi noise pada pengukuran sensor sebelum masuk ke dalam sistem kontrol. Oleh karena itu kalman filter sering digunakan pada sistem kontrol yang sensitif terhadap noise karena dapat meminimalkan square error.

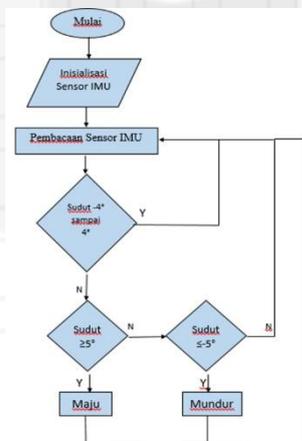
3. Perancangan Sistem

3.1 Perancangan Sistem Umum



Gambar 3.1 Diagram Blok Perancangan Umum Sistem

Berdasarkan Gambar 3.1, sistem yang telah direalisasikan menggunakan input berupa nilai sudut. Input nilai sudut *self balance scooter* yang digunakan adalah sudut dari satu sumbu yang tegak lurus dengan sumbu gravitasi bumi. Ketiga sumbu kemiringan tersebut dibaca oleh sensor *gyroscope*, *accelerometer*. Sensor *gyroscope* membaca nilai sudut dari perubahan nilai kemiringan yang dideteksi, sensor *accelerometer* membaca nilai percepatan dari kemiringan tersebut, serta sensor potensiometer membaca nilai simpangan pada stang *self balance scooter*. Nilai dari ketiga sensor ini dikirimkan ke *Mikrokontroler*. Flowchart dari sistem adalah sebagai berikut:



Gambar 3.2 Flowchart

3.2 Perancangan Perangkat Keras

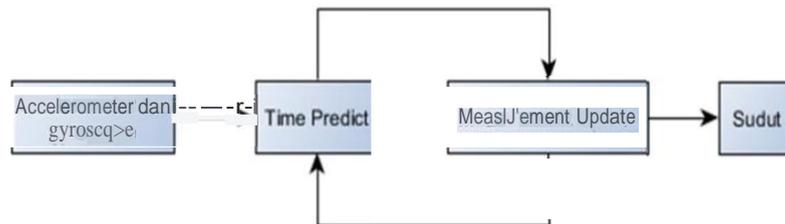


Gambar 3.3 Perancangan Perangkat Keras

Perancangan perangkat keras atau *hardware* yang telah dilakukan adalah pembuatan mekanik, merancang *plant* serta pemilihan komponen yang digunakan seperti potensiometer, motor DC, sensor IMU, dan Arduino mega.

3.3 Penggabungan sensor dengan kalman filter

Accelerometer dan gyroscope memiliki kelemahan masing-masing jika digunakan untuk pengukuran sudut. Oleh karena itu kedua sensor tersebut biasanya digabungkan dengan beberapa metode. Metode yang digunakan pada tugas akhir ini adalah kalman filter. Pada kasus ini kalman filter mendapat masukan nilai dari sensor accelerometer dan gyroscope yang kemudian akan diestimasi nilai sudut yang dibaca oleh kedua sensor tersebut.



Gambar 3.4 Diagram Blok Penggabungan Sensor.

3.4 Pemodelan Sistem

Pemodelan sistem *self balance scooter* ini sama dengan pemodelan inverter pendulum. Pada dasarnya sama pada *self balance scooter* pendulum disini adalah beban sedangkan kereta adalah beban bodi yang menyatu dengan roda. Setelah mendapatkan persamaan sistem, kemudian dihitung state space pada sistem menggunakan matlab dan didapat hasil fungsi transfer sistem adalah:

$$G(s) = \frac{0,01046s - 4,602}{s^2 + 0,001898s - 6,511} \tag{1}$$

3.5 Perhitungan PID

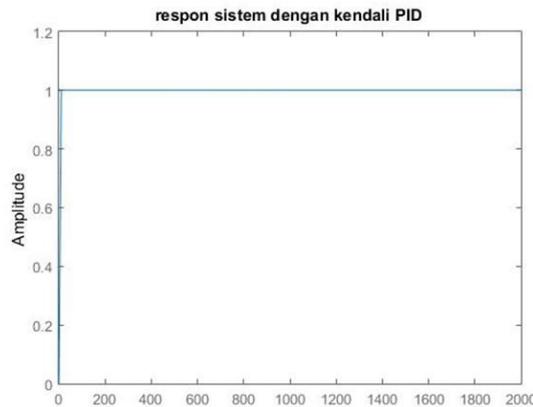
Setelah didapat fungsi transfer sistem pada *self balance scooter* kemudian dihitung. Untuk mendapatkan nilai PID pada perancangan sistem yang telah dibuat. Nilai parameter PID yang telah didapat adalah:

$$m(s) = 63,693s^2 + 47,2 \int_0^t ds + 21,5 \frac{d}{dt} \tag{2}$$

Parameter PID pada perancangan ini adalah Kp: 63,693, Ki: 47,2, dan Kd: 21,5.

3.6 Simulasi sistem dengan matlab

Dalam perancangan ini digunakan perangkat matlab untuk mensimulasikan dan mengamati karakter sistem apabila diberi masukan tertentu. Untuk dapat mensimulasikan sistem yang dibutuhkan pada proses PID seperti input, kostanta Kp, Ki, Kd, fungsi alih sistem dan output. Sistem diuji dengan berbagai skala nilai parameter Kp, Ki, dan Kd yang berbeda-beda sehingga dapat ditentukan sebagai dasar dalam proses tuning plant sebenarnya. Gambar (3.5) adalah beberapa grafik yang dapat menjadi acuan dalam penentuan parameter Kp, Ki, dan Kd.

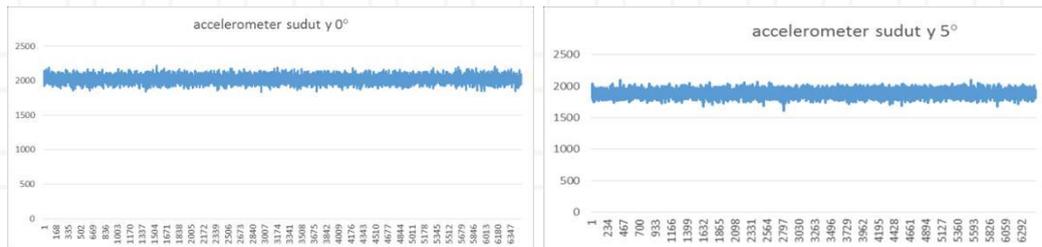


Gambar 3.5 Grafik respon dengan Kp:64, Ki:47, dan Kd:22

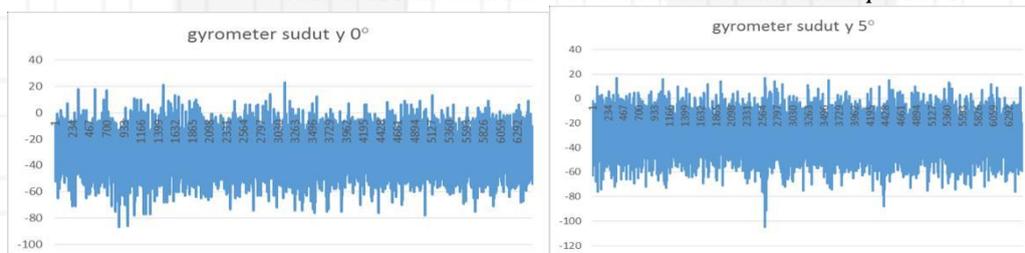
4. Pengujian Dan Analisa

4.1 Pengujian sensor MPU 6050

Hasil pengujian sensor MPU6050 menunjukkan bahwa sensor ini dapat bekerja dengan baik sesuai dengan gambar plot.



Gambar 4.1 Pembacaan nilai asli accelerometer 0° dan 5° terhadap sumbu Y



Gambar 4.2 Pembacaan nilai asli gyrometer 0° dan 5° terhadap sumbu Y

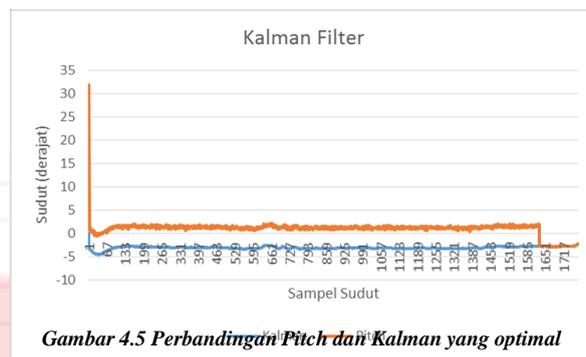
Dengan melihat grafik pada gambar 4.1, 4.2, 4.3 dan gambar 4.4 dapat berfungsi dengan baik. Penulis disini melakukan uji coba pada sudut y, dikarenakan self balance scooter hanya menggunakan sudut y untuk melakukan keseimbangan. Ketika sensor mendapat gangguan gerak dari luar nilai sensor berubah. Penulis juga hanya melakukan pengujian pada 0° dan 5°. Hal ini menunjukkan bahwa sensor dapat bekerja dengan baik dan bisa mengirimkan data ke mikrokontroler..

4.2 Pengujian Kalman

Pengujian kalman filter dilakukan dengan cara membandingkan nilai keluaran sudut dari accelerometer dengan nilai keluaran sudut dari kalman filter selama IMU dalam keadaan diam dan digerakan.

- Parameter kalman yang digunakan
- Q angle (Q accelerometer) = 0.001
- Q bias (Q gyroscope) = 0.003

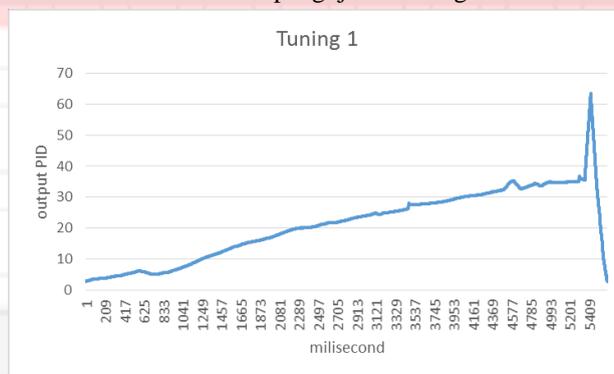
R measure = 0.03



Gambar 4.5 Perbandingan Pitch dan Kalman yang optimal

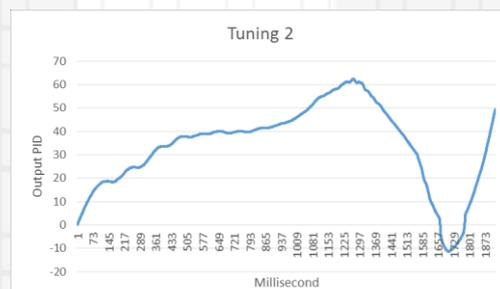
4.3 Pengujian PID

Tujuan utama dari penggunaan kontrolir PID adalah untuk bergerak maju dan mundur secara stabil. Kontroler ini terdiri dari kontrol proporsional, kontrol integral dan kontrol derivatif. Kontrol proporsional dapat mempercepat respon sistem, kontrol integral dapat memperbaiki error steady state dan kontrol derivatif dapat memperbaiki overshoot dan kestabilan sistem. Berikut adalah hasil pengujian tuning PID:



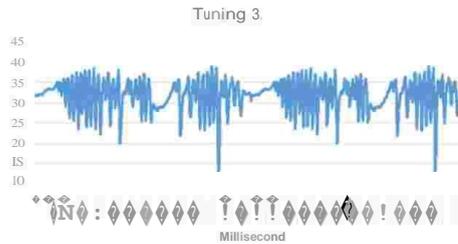
Gambar 4.6 Output PID dengan nilai Kp:1 Ki:0.1 Kd:1

Pada gambar 4.6 merupakan grafik respon output PID saat nilai Kp = 1, Ki = 0,1 dan Kd = 0,1. Hasil dari percobaan ini dapat dilihat respon dari sistem sangat kurang. Penggunaan nilai Kp yang kecil mengakibatkan respon masih kurang, sehingga self balance scooter tidak dapat bergerak kedepan. Karena respon kurang sehingga motor dc tidak dapat bergerak sesuai.



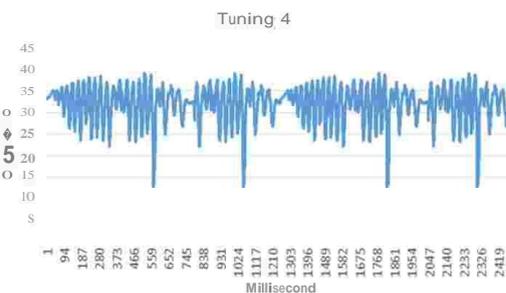
Gambar 4.7 Output PID dengan nilai Kp:5 Ki:0.2 Kd:3

Pada gambar 4.7 merupakan tuning PID dengan output PID Kp: 5, Ki 0.2 Kd: 3. Hasil dari percobaan ini dapat dilihat overshoot bertambah, kestabilan pada self balance scooter mulai ada walaupun sedikit, sehingga output PID sempat mencapai titik 60 akibat respon yang didapat masih kurang. Rise time pada sistem semakin berkurang akibat adanya nilai Kp, Ki, dan Kd



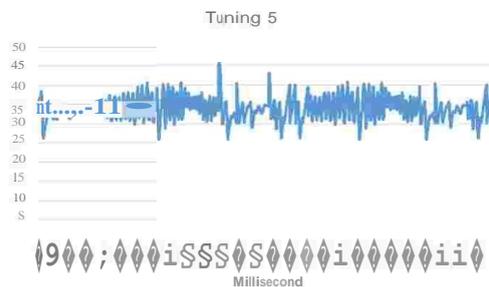
Gambar 4.8 Output PID dengan nilai Kp:8 Ki:1 Kd:3

Pada gambar 4.8 merupakan tuning PID dengan output PID Kp: 8, Ki 1 Kd: 3. Hasil dari percobaan ini dapat dilihat kestabilan self balance scooter meningkat. Ini disebabkan akibat overshoot meningkat, serta steady-state error menurun. Respon self balance scooter stabil akan tetapi masih terdapat time delay.



Gambar 4.9 Output PID dengan nilai Kp:8 Ki:3 Kd:4

Pada gambar 4.9 merupakan tuning PID dengan output PID Kp: 8, Ki: 3 Kd: 4. Hasil percobaan ini dapat dilihat overshoot semakin berkurang, kemudian kestabilan self balance scooter bertambah. Pada percobaan kali ini self balance scoote mampu berjalan seimbang, akan tetapi masih terdapat sedikit ketidak stabilan walaupun sedikit.



Gambar 4.10 Output PID dengan nilai Kp:8 Ki:3 Kd:6



Gambar 4.11 Output PID pada perancangan dengan nilai Kp:64 Ki:47 Kd:22

Pada gambar 4.10 dan 4.11 merupakan tuning PID dan nilai dari perhitngan perancangan PID. Pada perhitungan perancangan didapat nilai PID Kp: 64, Ki: 47, dan Kd: 22 menunjukkan self balance scooter tidak memiliki kestabilan dalam bergerak. Self balance scooter mempunya nilai setpoint untuk bergerak maju pada output PID menunjukkan nilai 35, ketika diberi parameter PID pada perhitungan perancangan self balance scooter tidak stabil untuk bergerak maju kedepan. Setelah melakukan tuning PID dengan metode trial and error maka nilai parameter maksimal didapat Kp: 8, Ki: 3, Kd: 6. Pada gambar 4.10 menunjukkan nilai tuning PID yang maksimal. Self balance scooter bergerak dengan stabil. Output PID selalu berada dekat dengan setpoint untuk bergerak, dapat disimpulkan

nilai parameter PID yang optimal adalah nilai tuning yang telah dilakukan. Perbedaan nilai parameter PID yang jauh, dikarenakan pada percobaan pada alat terdapat kendala. Sehingga nilai yang didapat pada parameter tuning dengan perhitungan terlihat jauh. Namun setelah dilakukan percobaan berulang-ulang didapat nilai parameter PID yang maksimal.

Nilai Parameter PID tuning		Nilai Parameter PID perhitungan	
KP	: 8.0	KP	: 64
KI	: 3.0	KI	: 47
KD	: 6.0	KD	: 22

5. Kesimpulan dan Saran

5.1.1 Kesimpulan

Dari hasil pengujian yang dilakukan pada perancangan self balance scooter menggunakan metode PID :

- Setelah melakukan perancangan mekanik self balance scooter didapat hasil mekanika yang belum sempurna. Pemakaian rantai pada mekanika self balance scooter kurang baik, sehingga menghambat percobaan yang telah dilakukan. Beban mekanika mencapai 44 kg sehingga motor dc yang dipakai harus mempunyai torsi yang besar.
- Berdasarkan percobaan yang telah dilakukan, didapat parameter PID adalah Kp:8,Ki:3,Kd:6.

5.2 Saran

Pada tugas akhir ini penulis menggunakan PID sebagai algoritma kontrolernya. Kekokohan dari sistem tidak seutuhnya teruji. Maka dari itu penulis sangat menyarankan agar metoda kontrol dapat dikembangkan lebih lanjut, entah mengembangkan metode tuning ataupun menggunakan metode yang lebih canggih seperti LQR.

Untuk saat ini self balance scooter hanya bisa diuji coba pada permukaan yang mendatar, kedepannya diharapkan akan ada pengujian self balance scooter pada permukaan bergelombang dan menanjak.

DAFTAR PUSTAKA

- [1] Royyan, M.. 2015. "Implementation of Kalman Filter and PID Controller for Inverted Pendulum Robot". Bandung. Telkom University.
- [2] Bobby, Grace.. 2015. "Desing and Implementision of Balance two-Wheeled Robot Based Microcontroller". Bandung. Telkom University.
- [3] Laksana, Andra, Iwan Setyawan dan Sumardi. 2011. "Balancing Robot Beroda Dua Menggunakan Metode Kendali Proporsional Integral". Semarang. Universitas Diponegoro.
- [4] Wiratran, Helmi, "Perancangan dan Implementasi Embedded Fuzzy Logic Controller untuk Pengaturan Kestabilan Gerak Robot Segway Mini". Surabaya. ITS.
- [5] Setyawan, Lukas B., Dedi Susilo dan Dede Irawan. " Sistem Kendali Gerak Segway Berbasis Mikrokontroller". Salatiga. Univeritas Kristen Satya Wacana.
- [6] Ruswanto, Sonie, Endah Suryawati Ningrum dan Irwan Ramli. 2011. "Pengaturan Gerak Dan Keseimbangan Robot Line Tracer Dua Roda Menggunakan PID Controller". Surabaya. Politeknik Elektronika Negeri Surabaya.
- [7] Fahruzi, Imam dan Emilio Santos Abdullah. "Integrasi Sensor Multifungsi Accelerometer untuk Mendeteksi Kekuatan Benturan". Batam. Politeknik Negeri Batam
- [8] Prasetyo, Arief Eko. 2009. "Perancangan Prototype Skuter Seimbang Menggunakan Pengendali PID Dan Pengendali Logika Fuzzy". Bandung. Institut Teknologi Bandung
- [9] Messner, Bill & Tilbury, Dawn., 2011. Control Tutorial For Matlab & Simulink,<http://ctms.engin.umich.edu/CTMS/index.php?example=InvertedPendulum§ion=SystemModeling>.