INVERTED PENDULUM PADA *PROTOTIPE* MOBIL DENGAN METODE KENDALI *PROPORSIONAL INTEGRATIF DERIVATIF*

SKRIPSI

KEMINATAN TEKNIK KOMPUTER

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Sigit Priyo Jatmiko
NIM: 125150301111010

PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2016

PENGESAHAN

INVERTED PENDULUM PADA *PROTOTIPE* MOBIL DENGAN METODE KENDALI *PROPORSIONAL INTEGRATIF DERIVATIF*

SKRIPSI

KEMINATAN TEKNIK KOMPUTER

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Sigit Priyo Jatmiko
NIM: 125150301111010

Skripsi ini telah diuji dan dinyatakan lulus pada
01 Desember 2016
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I	Dosen Pembimbing II
Wijaya Kurniawan, S.T, M.T	Barlian Henryranu Prasetio, S.T, M.T
NIK: 201201 820125 1 001	NIK: 201102 821024 1 001

Mengetahui
Ketua Jurusan Teknik Informatika

Tri Astoto Kurniawan, S.T., M.T. Ph.D.
NIP: 197105182003121001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 01 Desember 2016

Sigit Priyo Jatmiko

NIM: 125150301111010

KATA PENGANTAR

Puji syukur saya panjatkan kehadirat Allah Swt atas yang mana telah memberikan hidayah dan inayahnya sehingga, skripsi ini dapat terselesaikan. Skripsi yang berjudul "*Inverted Pendulum* Pada *Prototipe* Mobil dengan Metode Kendali *Proporsional Integratif Derivatif* " ini disusun untuk memenuhi sebagian persyaratan memperoleh gelar Sarjana Komputer.

Penulis menyadari bahwa dalam penyusunan skripsi ini tidak terlepas dari bantuan berbagai pihak. Oleh karena itu penulis mengucapkan terima kasih kepada berbagai pihak yang telah berkenan untuk memberikan bantuan demi kelancaran penyusunan skripsi ini diantaranya:

1. Bapak Yaimo dan Ibu Lasmini yang saya cintai, yang telah memberikan dukungan, semangat serta do'a.
2. Bapak **Wijaya Kurniawan, S.T, M.T** selaku dosen pembimbing I yang telah memberikan pengarahan dalam pembuatan proposal dan alat dalam skripsi ini.
3. Bapak **Barlian Henryranu Prasetio, S.T, M.T** selaku dosen pembimbing II yang telah memberikan pengarahan dalam penulisan skripsi yang sangat berguna dalam penyelesaian proposal skripsi ini.
4. Segenap dosen Fakultas Ilmu Komputer Universitas Brawijaya atas segenap ilmu pengetahuan dan perhatian yang diberikan Segenap staff dan pegawai Fakultas Ilmu Komputer Universitas Brawijaya atas segala bantuan yang bersifat administratif.
5. Seluruh teman-teman angkatan 2012, teman kontrakan, dan pihak yang telah membantu yang tidak dapat saya sebutkan satu per satu.

Penulis sangat mengharapkan kritik dan saran yang membangun, karana penulis menyadari banyak kekurangan dalam menyusun lapran skripsi ini. Harapan dari penulis agar penelitian ini dapat bermanfaat bagi penelitian selanjutnya dan perkembangan teknologi yang sedang dikembangkan.

Malang, 01 Desember 2016

Penulis

ABSTRAK

Penelitian tentang *pendulum* terbalik yang mempunyai karakteristik yang tidak stabil dan *nonlinier* banyak dilakukan diberbagai belahan dunia. Salah satu penelitian *pendulum* terbalik adalah *pendulum* terbalik pada model kereta, penelitian model kereta ini tidak bisa berpindah tempat hanya terpaku pada relnya saja, agar bisa berpindah – pindah supaya bisa dipergunakan untuk alat transportasi diperlukan perubahan dari model kereta ke penelitian Inverted *Pendulum* pada *prototipe* mobil.

Penelitian ini membuat seimbang bandul atau *pendulum* pada badan *prototipe* mobil dengan cara roda *prototipe* mobil maju atau mundur sesuai dengan arah jatuhnya bandul agar bandul dapat berdiri tegak, kontroller PID digunakan untuk mengatur pergerakan 4 roda terpasang pada *prototipe* mobil sehingga memungkinkan bandul atau *pendulum* dalam posisi seimbang atau tegak dengan cara bergerak maju atau mundur sesuai pembacaan sensor. c.

Dengan menggunakan metode proporsional integratif derivatif yang ditanamkan pada kontroller arduino uno dengan nilai Kp = 100, Ki = 2 dan Kd = 0 sistem dapat berjalan sesuai yang diinginkan yaitu bandul atau *pendulum* dapat seimbang dan motor mampu bergerak sesuai arah jatuhnya bandul atau *pendulum*.

Kata kunci – *pendulum* terbalik model kereta, *Inverted Pendulum*, Kontroller PID.

ABSTRACT

Research of inverted pendulum that has a characteristics nonlinear unstable and many carried out in various parts of the world. One of the studies inverted pendulum is a pendulum upside down on the model trains, model train, this research could not move just fixated on its rail only, in order to be moved – moved to be used for transportation of necessary change from the model train to the Inverted Pendulum research on prototype cars.

This research makes chemical equilibrium pendulum or pendulum on a prototype car by way of car prototype wheel forward or backward according to the direction of the fall of the pendulum so that the pendulum can stand upright, kontroller PID is used to regulate the movement of the 4 wheels mounted on a prototype of the car allowing the pendulum or pendulum in a position of equilibrium or upright by means of moving forward or backward according the reading of the sensor.

By using the method proportionate derivatives integratif implanted in kontroller arduino uno worth kp = 100, ki = 2 and kd = 0 system can go according to which they desire pendulum or pendulum can balances and motor able to move in the direction of pendulum or pendulum.

Keywords – Inverted pendulum model train, Inverted Pendulum, Kontroller PID.

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

DAFTAR LAMPIRAN

BAB 1 PENDAHULUAN

1.1 Latar belakang

Penelitian tentang *pendulum* terbalik sudah lama dilakukan diberbagai belahan dunia, dengan berbagai cara dan bentuk penelitian semisal penelitian tentang analisis kesetabilan dan kontrol optimal *double pendulum* terbalik pada kereta menggunakan metode *linear quadratic regulator* (LQR) dan *modelling and simulation for optimal control of nonlinier inverted pendulum dynamical system using pid control adn lqr*. Banyak metode digunakan dalam penelitian untuk mencari metode apa yang terbaik untuk menyeimbangkan *pendulum* tersebut. *Pendulum* terbalik mempunyai karateristik yang tidak stabil dan nonlinier, titik berat berada pada atas titik tumpunya sehingga perlu diseimbangkan (anyakrawati, 2015).

Prinsip kerja dari *pendulum* terbalik membuat banyak penelitian seperti *pendulum* terbalik pada model kereta yang merupakan salah satu *plant* nonlinier yang mempunyai karasteristik dari sistem non linier yang sederhana tapi sulit untuk dikontrol sehingga sering digunakan berbagai metode untuk mengkontrolnya (Agustinah, 2014), bagian utama dari sistem *pendulum* model kereta adalah kereta dan *pendulum*. Kereta dapat bergerak horizontal pada lintasan yang terbatas, sedangkan *pendulum* dapat bergerak bebas terhadap porosnya. Balancing robot beroda dua juga menggunakan prinsip kerja dari *pendulum* terbalik karena robot menyeimbangkan badan robot agar mampu berdiri tegak lurus terhadap permukaan bumi dibidang yang datar (Ketaren, 2015). Pengaplikasian prinsip kerja *pendulum* terbalik dapat berguna dalam menunjang aktifitas manusia dalam kesehariannya, seperti yang diberitakan di liputan6 (Liputan6.com, 2015) *one self-balancing electric unicycle* menjadi salah satu alat transportasi yang dipamerkan dalam ajang Consumer Electronic Show (CES) tahun 2015, alat ini mampu menjaga keseimbangan *pendulum* terhadap permukaan bumi dan mampu dikendarai oleh satu orang.

Beberapa hal yang disebutkan diatas membuktikan bahwa penelitian *pendulum* terbalik penting untuk dilakukan, tujuan dari penelitian tidak berhenti pada *pendulum* dengan model kereta, akan tetapi berlanjut pada penelitian *pendulum* terbalik sebagai alat transportasi yang mampu membawa manusia pindah dari satu tempat ke tempat yang lain. Penelitian *pendulum* dengan model kereta yang mempunyai titik berat terhadap gravitasi bumi terpaku pada rel kereta yang tidak mampu berpindah tempat, sedangkan pada alat transportasi harus mampu dipindah-pindah. Oleh karena itu, perlu dilakukan penelitian yang memodelkan sistem *pendulum* model kereta ke *pendulum* yang mempunyai roda agar selanjutnya bisa membuat alat transportasi yang mampu berpindah-pindah pada titik beratnya. Pada penelitian "*Inverted Pendulum* pada *prototipe* mobil dengan metode kendali *proporsional, integratif derivatif*" yaitu memodelkan sistem *pendulum* kereta ke model *prototipe* mobil yang mampu berpindah-pindah tempat, sehingga pada akhirnya bisa membuat alat transportasi dengan

prinsip *pendulum* terbalik yaitu *segway.* Penelitian ini menggunakan metode PID, kontroller arduino uno sebagai kontroller utama sistem, driver motor L298N sebagai driver motor, dan motor DC sebagai penggerak *prototipe* mobil agar bandul atau *pendulum* yang mempunyai sifat dinamis atau tidak menentu pergerakannya (maju atau mundur) yang terpasang di *body* mobil bagian atas mampu seimbang sesuai yang diharapkan.

1.2 Rumusan masalah

Berdasarkan latar belakang diatas dapat ditarik rumusan masalah sebagai berikut:

1. Bagaimana parameter PID agar sistem *Inverted Pendulum* dapat seimbang?
2. Berapa akurasi yang dihasilkan sensor MPU6050?
3. Bagaimana respon motor terhadap nilai pembacaan sensor?

1.3 Tujuan

Tujuan dari penelitian yang dilakukan adalah dapat menyeimbangkan bandul yang ada diatas *prototipe* mobil dengan menggunakan metode PID dan mengimplementasikan sistem *pendulum* kereta menjadi *pendulum* pada *prototipe* mobil.

1.4 Manfaat

Manfaat bagi penulis adalah menerapkan ilmu yang didapat selama berkuliah di Universitas Brawijaya, serta pembaca diharapkan bisa mengembangkan penelitian ini lebih lanjut untuk menemukan nilai PID yang paling *responsif* dengan metode PID.

1.5 Batasan masalah

Agar penelitian yang dilakukan ini lebih terfokus sesuai sistem, maka penelitian ini dibatasi dalam hal:

1. *Prototipe* mobil yang digunakan menggunakan empat roda dan diletakan pada bidang yang datar.
2. *Tunning* parameter kontrol Proporsional Integral Derivatif (PID) dilakukan dengan cara *trial dan error.*

1.6 Sistematika pembahasan

Penulisan skematik pada pembahasan dan penyusunan laporan tentang penelitian dapat diuraikan sebagai berikut:

BAB I PENDAHULUAN

Menjelaskan tentang latar belakang, rumusan masalah, tujuan, manfaat, batasan masalah, dan sistematiak pembahasan dari "*Inverted Pendulum* Pada *Prototipe* Mobil Dengan Metode Kendali *Proporsional Integratif Derivatif*".

BAB II LANDASAN KEPUSTAKAAN

Menjelaskan tentang landasan teori terkait dengan penelitian dan menjelaskan penelitian sebelumnya yang serupa dengan penelitian yang dilakukan.

BAB III METODE PENELITIAN

Membahas tentang langkah kerja yang dilakukan dalam penulisan diantaranya studi literatur, analisis kebutuhan system, implementasi dan pengujian dari *Inverted Pendulum* pada *Prototipe* Mobil.

BAB IV REKAYASA KEBUTUHAN

Membahas tentang kebutuhan yang arus dipenuhi dalam proses penelitian yang akan dilakukan. Terdapat beberapa kebutuhan dalam penelitian seperti kebutuhan fungsional, kebutuan non fungsional, kebutuhan user dan sebagainya.

BAB V PERANCANGAN DAN IMPLEMENTASI

Membahas tentang perancangan alat dan implementasi alat.

BAB VI PENGUJIAN DAN ANALISIS

Menampilkan hasil dari data yang didapat dari perancangan pengujian penggunaan *tunning PID* untuk system kendali *inverted pendulum* pada *prototipe* mobil. Kemudian menganalisis hasil apakah sesuai dengan yang diharapkan.

BAB VII PENUTUP

Bab ini Menguraikan kesimpulan yang didapatkan dari perancangan sistem ini. Kemudian memberi saran dari hasil pengujian yang terjadi serta pengembangan dari sistem yang perlu ditingkatkan.

BAB 2 LANDASAN KEPUSTAKAAN

Bab ini berisi landasan kepustakaan yang meliputi tinjauan pustaka yang diambil dari beberapa jurnal dan dasar teori yang diperlukan. Tinjauan pustaka membahas penelitian-penelitian yang telah ada dan berkaitan dengan *inverted pendulum* atau *pendulum* terbalik.

2.1 Tinjauan Pustaka

Dibawah ini ada beberapa penelitian yang serupa yang sudah pernah dilakukan.

Tabel 2.1 Tinjauan Pustaka

Judul	Penulis	Kelebihan	Kekurangan
Analisis Kesetabilan dan Kontrol Optimal *Double Pendulum* Terbalik pada Kereta Menggunakan Metode *Linear Quadratic Regulator* (LQR).	Chusnul Fathonah (2016)	• *Double pendulum* • Mampu stabil dalam waktu 30detik • Biaya rendah	• Berupa Analisa
Modelling and Simulation for Optimal Control of Nonlinier Inverted Pendulum Dynamical System Using PID Control adn LQR	Lal Bahadur Prasad,et al (2012)	• Biaya Rendah • Memakai dua metode PID dan LQR	• Berupa Simulasi
Stabilisasi pada Sistem *Pendulum*-Kereta dengan Mengunakan Metode *Fuzzy-Sliding Mode Control*	Trihastuti Agustinah, Niora Fatima Tanzinia (2014)	• Biaya rendah • Respon yang baik dan memiliki *undershoot* yang lebih kecil	• Berupa simulasi

Berdasarkan penelitian yang sebelumnya, pada sistem ini akan dirancang sebuah sistem *Inverted pendulum* pada *prototipe* mobil dengan menggunakan kontroller arduino uno, dilengkapi dengan sensor MPU6050 (*accelerometer* dan *gyroscope*) untuk mendeteksi kemiringan *pendulum* secara terus menerus agar ketika *pendulum* akan jatuh akan berdiri tegak kembali dengan cara driver motor

L298N menggerakan motor DC searah jatuhnya *pendulum*. Kelebihan sistem yang dibuat menggunakan sensor *accelerometer* dan *gyroscope* yang mampu membaca sudut dengan baik dan sistem dapat berpindah tempat.

2.2 Dasar Teori

Dasar teori akan di bahas di sub 2.2.1 sampai 2.2.7

2.2.1 *Inverted Pendulum*

Dasar untuk membuat robot beroda dua dapat seimbang, yaitu dengan cara mengendalikan roda searah dengan jatuhnya bagian atas sebuah robot. Jika semua proses berjalan dengan baik maka robot akan seimbang (Kataren *et al,*. 2015).

Gambar 2.1 Balancing Robot Beroda Dua Menyeimbangkan Diri

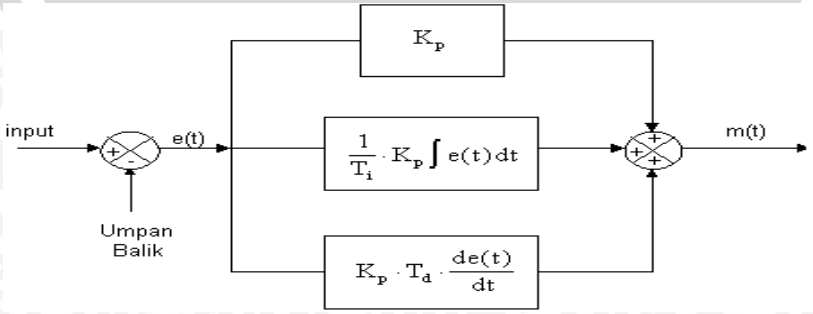
Sumber : Jurnal Elementer Vol.1

Saat balancing robot beroda dua condong kedepan atau miring ke kanan seperti Gambar 2.1 maka tindakan yang perlu dilaksanakan adalah motor bergerak searah dengan arah kemiringan yang terjadi, sehingga robot akan kembali tegak lurus dengan permukaan bodang datar, gaya yang digunakan untuk menyeimbangkan robot didapat dari putaran roda yang dihasilkan dari motor DC.

2.2.2 **Metode PID**

PID adalah (*proportional-integratif-derivative controller*) merupakan kontroler dalam menentukan presisi pada suatu sistem instrumentasi dengan karakteristik adanya umpan balik suatu sistem tersebut .

Komponen PID terdiri dari tiga jenis *Proportional*, *Integratif* dan *Derivatif* dengan masing-masing memiliki kelebihan dan kekurangan. Ketiganya dapat digunakan bersamaan maupun sendiri-sendiri bergantung respon yang kita inginkan pada suatu *plant*.



Gambar 2.2 Diagram blok kontrol PID

Kontrol Proporsional

kontrol proporsional memiliki fungsi dalam memperkuat sinyal kesalahan penggerak (sinyal error), sehingga akan mempercepat keluaran sistem dalam mencapai titik referensi. Hubungan antara input kontroler u(t) dengan sinyal *error* e(t) terlihat pada persamaan 1.

$$u(t) = K_p e(t) \dots\dots\dots\dots (1)$$

Penggunaan kontrol Proposional memiliki berbagai keterbatasan yang disebabkan oleh sifat kontrol yang tidak dinamik ini. Meskipun demikian dalam aplikasi-aplikasi dasar yang sederhana kontrol Proposional ini cukup mampu dalam memperbaiki respon transien terutama rise time dan settling time.

Kontrol Integratif

Kontrol integratif pada prinsipnya memiliki tujuan dalam menghilangkan kesalahan keadaan tunak (offset) yang sering dihasilkan oleh kontrol proporsional. Hubungan antara output kontrol integral u(t) dengan sinyal error e(t) terlihat pada persamaan 2.

$$u(t) = K_i \int_0^t e(t)\, dt \dots\dots\dots\dots (2)$$

Kontrol I dapat memperbaiki dan menghilangkan respon steady-state, namun pada pemilihan Ki yang tidak tepat akan menimbulkan respon transien yang tinggi dan menyebabkan ketidakstabilan sistem. Pemilihan Ki yang sangat tinggi dapat menyebabkan output berosilasi karena menambah orde sistem.

Kontrol Derivatif (turunan)

Kontrol derivatif dapat disebut juga sebagai pengendali laju, karena output kontroler sebanding dengan laju perubahan sinyal error. Hubungan antara output kontrol derivatif u(t) dengan sinyal error e(t) terlihat pada persamaan 3.

$$u(t) = K_d \frac{de(t)}{dt} \dots\dots\dots\dots (3)$$

Dengan sifat ini ia dapat digunakan dalam memperbaiki respon transien dengan cara memprediksi error yang kemungkinan akan terjadi. Kontrol Derivative hanya berubah ketika ada perubahan error, sehingga ketika error statis kontrol ini tidak akan bereaksi, hal ini juga yang menyebabkan kontroler Derivative tidak dapat digunakan sendiri. Sehingga mendapat persamaan

$$u(t) = K_p e(t) + K_i \int_0^t e(t)\, dt + K_d \frac{de(t)}{dt} \dots\dots\dots\dots (4)$$

Atau

$$u(t) = K_p \left(e(t) + \frac{1}{Ti} \int_0^t e(t) dt + T_d \frac{de(t)}{dt} \right) \dots\dots\dots\dots (5)$$

Keterangan :

u(t) = sinyal output pengendali PID

Kp = konstanta proporsional

Ti = waktu integral

Td = waktu derivatif

Ki = konstanta integral (Kp⁄Ti)

Kd = konstanta derivatif (Kp.Td)

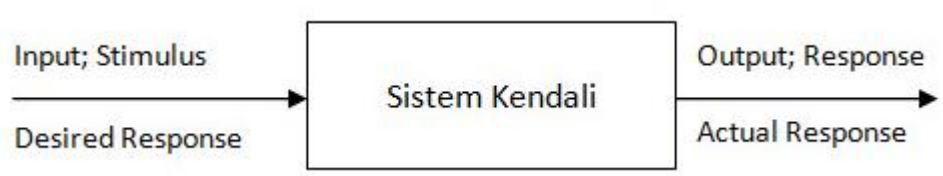
e(t) = sinyal error = referensi – keluaran plant = set point – nilai sensor

Pengontrol PID akan memberikan aksi kepada Control Valve berdasarkan besar error yang diperoleh. Karakteristik pengontrol PID dipengaruhi besar oleh kontribusi ketiga parameter P, I dan D. Penyetelan konstanta Kp, Ki dan Kd akan berakibat penonjolan sifat dari setiap elemen. Satu atau dua dari ketiga konstanta dapat disetel dengan lebih menonjol dibanding yang lain. Konstanta yang menonjol itulah yang akan memberikan kontribusi pengaruh untuk respon sistem secara keseluruhan.

2.2.3 **Sistem Kendali**

Sistem kendali adalah suatu susunan komponen fisik yang terhubung atau terkait sedemikian rupa sehinga dapat mengarahkan, memerintah, dan mengatur diri sendiri atau sistem lainnya. Di dalam dunia *engineering* dan *science* sistem kendali cenderung dimaksudkan sebagai sistem kendali dinamis (Agustian. 2013).

Sistem kendali terdiri dari sub-sistem dan proses (atau plants) yang disusun untuk mendapatkan keluaran (output) dan kinerja yang ingin dicapai dari input yang diberikan. Gambar 2.3 di bawah ini menununjukkan blok diagram untuk sistem kendali paling sederhana, sistem kendali membuat sistem dengan input yang diberikan menghasilkan output yang diinginkan.



Gambar 2.3 Deskripsi sederhana sistem kendali

Sumber : http://te.unib.ac.id/lecturer/indraagustian/2013/06/definisi-sistem-kendali/

2.2.4 **Motor DC**

Motor DC adalah jenis motor listrik yang bekerja dengan menggunakan sumber tegangan DC. Motor DC atau motor arus searah ini menggunakan arus langsung dan tidak langsung/*direct-unidirectional*. Motor DC digunakan pada

penggunaan khusus yang mana diperlukan penyalaan *torque* yang tinggi atau percepatan yang tetap sebagai kisaran kecepatan yang luas (Zonaelektro. 2013).

Gambar 2.4 Motor DC

Sumber : http://letsmakerobots.com/node/42224

2.2.5 **Arduino Uno**

Arduino Uno adalah papan sirkuit berbasis mikrokontroler ATmega328. IC (integrated circuit) ini memiliki 14 input/output digital (6 output untuk PWM), 6 analog input, resonator kristal keramik 16 MHz, Koneksi USB, soket adaptor, pin header ICSP, dan tombol reset. Hal tersebut digunakan untuk mensupport mikrokontrol secara mudah ketika terhubung dengan kabel power USB atau kabel power supply adaptor AC ke DC serta battery (Ihsan, 2015). Bentuk dari board arduino uno dapat dilihat pada gambar 2.5.

Gambar 2.5 *Board* arduino uno

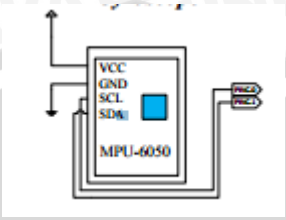
Sumber : http://www.caratekno.com/2015/07/pengertian-arduino-uno-mikrokontroler.html

2.2.6 Sensor Gyroscope dan Accelerometer

Gyroscope merupakan suatu alat elektronik yang berfungsi untuk mengukur keceptan sudut dengan satuan (°/s) yang dialami oleh suatu benda *pitch*, *roll* dan *yaw*. Sedangkan sensor *accelerometer* adalah suatu piranti elektronik yang digunakan sebagai alat ukur percepatan yang terjadi dalam keadaan tertentu. Sensor *accelerometer* dapat digunakan untuk mendapatkan posisi dari suatu benda dengan melakukan percepatan itu sendiri sebanyak dua kali terhadap waktu (Seifert, et al, 2007).

Sensor MPU-6050 membutuhkan tegangan kerja 3,3V. Tetapi modul ini telah dilengkapi dengan regulator tegangan 3,3V sehingga bisa langsung dihubungkan ke tegangan 5V. Sensor ini mempunyai dua buah keluaran yaitu SCL dan SDA masing-masing dihubungkan ke PC.0 dan PC.1.

Gambar 2.6 Sensor MPU-6050 3-Axis Accelerometer+ 3-Axis Gyroscope

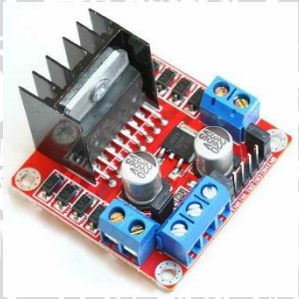


Gambar 2.7 Konfigurasi Sensor MPU-6050

Sumber : Tugas akhir Self-Balancing Scooter Menggunakan Metode Kendali Proporsional Integral Derivatif

2.2.7 Driver Motor L298N

L298N adalah contoh IC yang dapat digunakan sebagai driver motor DC. IC ini menggunakan prinsip kerja H-Bridge. Tiap H-Bridge dikontrol menggunakan level tegangan TTL yang berasal dari output mikrokontroler. L298N dapat mengontrol 2 buah motor DC. Tegangan yang dapat digunakan untuk mengendalikan robot bisa mencapai tegangan 46 Vdc dan arus mencapai 2 A untuk setiap kanalnya (Intructables.com. 2015).



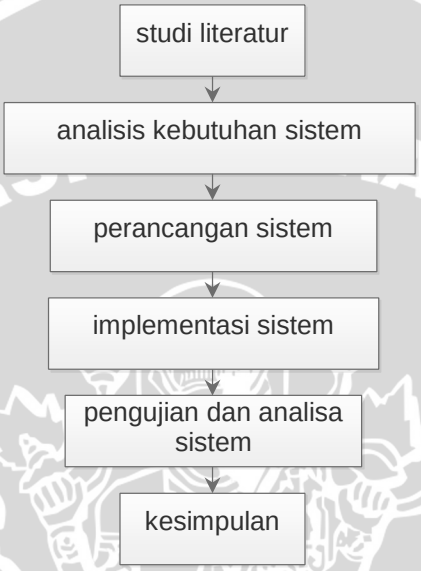
Gambar 2.8 Driver Motor L298N

Sumber : http://www.instructables.com/id/Control-DC-and-stepper-motors-with-L298N-Dual-Moto/

BAB 3 METODOLOGI

3.1 Metode Penelitian

Dalam bab ini akan dibahas mengenai metode yang digunakan dalam skripsi ini, meliputi : studi literatur, Analisa kebutuhan, Perancangan, Implementasi, dan kesimpulan. Berikut ini merupakan diagram alir yang digunakan dalam metodote ini.



Gambar 3.1 Diagram Alir Metode Penelitian

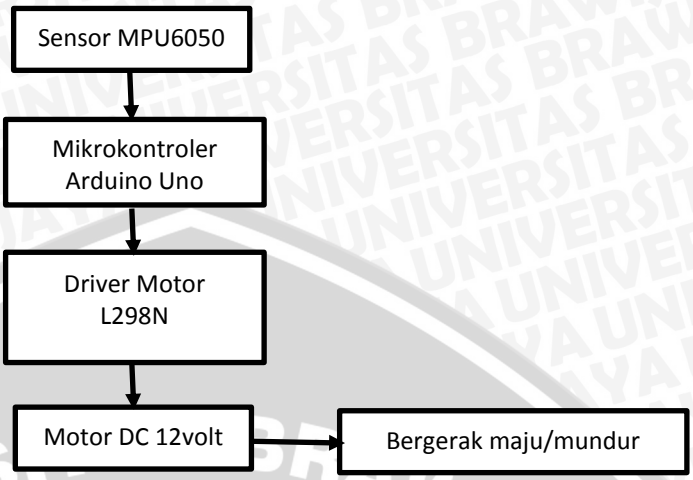
3.1.1 **Studi Literatur**

Studi literatur yang dilakukan oleh peneliti bertujuan untuk mempelajari penjelasan dasar teori yang digunakan untuk perancangan sistem. Pada tahap studi literature ini mempelajari teori-teori yang digunakan dalam pengerjaan skripsi. Teori-teori pendukung tersebut diperoleh dari buku, jurnal, e-book dan penelitian sebelumnya yang berkaitan dengan skripsi. Referensi utama yang diperlukan dalam penulisan ini adalah *forum Inverted Pendulum, control* pid dengan tunning *trial dan error.*

3.2 Perancangan Sistem

Perancangan sistem dalam penelitian ini ada pada *prototipe* mobil yang membawa bandul atau *pendulum*.

Diagram dari sistem yang dibuat dapat dilihat pada gambar 3.2



Gambar 3.2 Diagram sistem *inverted pendulum* pada *prototipe* mobil

Dalam sistem ini menggunakan sensor MPU6050 untuk mendeteksi kemiringan sensor, sensor dihubungkan pada kontroller arduino uno, hasil deteksi sensor difilter dengan kalman filter pada kontroller arduino uno ini dimaksudkan mehilangkan *noice* (gelombang yang tidak menentu) agar hasil deteksi lebih akurat, selanjutnya data diolah dengan kontroller PID untuk menentukan respon yang berupa nilai PWM. Nilai PWM dari kontroller arduino uno dikirim ke driver motor L298N dan driver motor menggerakan motor DC 12volt sesuai perintah yang dikirimkan kontroller arduino uno apakah motor DC bergerak maju atau mundur.

3.3 Implementasi

Implementasi sistem ini akan dilakukan sesuai dengan perancangan sistem yang telah dibuat sebelumnya. Pada bagian ini terdapat berbagai macam proses implementasi yaitu pertama sensor gyroscope dan accelerometer mendeteksi keseimbangan dari *pendulum*, setelah mendapatkan nilai derajat keseimbangan, nilai dikirim ke Arduino uno sebagai controller, arduino uno mengolah nilai tersebut untuk dikirim ke driver motor dan driver motor mengirim sinyal ke motor untuk bergerak maju atau mundur untuk mencapai keseimbangan *pendulum* terhadap *prototipe* mobil.

3.4 Pengujian

Pengujian pada skripsi ini dilakukan agar dapat menunjukkan bahwa sistem dapat bekerja secara baik dan sesuai yang diharapkan. Pengujian sistem dilakukan meliputi:

1. Pengujian pada sensor *gyroscope* dan *accelerometer* pada mikrokontroler arduino.
2. Pengujian keseluruhan *system inverted pendulum* pada *prototipe* mobil.

3.5 Analisa

Untuk mengukur kinerja dari *system inverted pendulum* pada *prototipe* mobil dengan menggunkan kontroller arduino uno, dilakukan analisis untuk mengetahui hasil yang digunakan untuk menarik kesimpulan dari penelitian yang dilakukan. Hasil dari analisa digunakan untuk mengetahui sistem berjalan sesuai dengan yang diingikan.

3.6 Kesimpulan

setelah semua tahap perancangan implementasi serta pengujian selesai dapat diambil kesimpulan. Kesimpulan di ambil untuk menjawab rumusan masalah yang dirumuskan sebelumnya. Tahapan terakhir dari penulis adalah saran yang dimaksudkan untuk memperbaiki kesalahan-kesalahan yang terjadi pada *system inverted pendulum* pada *prototipe* mobil dan menyempurnakan penulisan serta memberikan pertimbangan atas hasil yang telah dilakukan.

BAB 4 REKAYASA KEBUTUHAN

4.1 Gambaran Umum Sistem

Dalam perancangan yang sistem dilakukan terdapat tujuan, kegunaan, Karakteristik Pengguna, Batasan Perencanaan dan Implementasi.

4.1.1 **Tujuan**

Tujuan dari sistem yang di buat adalah melanjutkan penilitian *inverted pendulum* dengan model kereta yang sudah ada lintasan keretanya, pada penelitian yang dilakukan tidak digunakan model kereta melainkan dengan model *prototipe* mobil.

Sistem ini juga bertujuan untuk mencari nilai respon PID yang paling bagus agar bandul yang ada diatas *prototipe* mobil dapat tegak lurus keatas.

4.1.2 **Kegunaan**

Sistem yang dibuat berguna untuk belajar dan mengembangkan teori-teori *inverted pendulum* yang selanjutnya bisa digunakan untuk penelitian ke arah nilai guna seperti membuat alat transportasi dua roda atau pun dengan satu roda.

4.1.3 **Karasteristik Pengguna**

Pengguna dalam sistem ini bertindak sebagai user yang dapat memberikan intruksi seperti menjalankan sistem, memprogram sistem dengan nilai PID yang mempunyai respon lebih bagus. Pengguna tidak dapat mengendarai atau mengunakan sistem alat yang dibuat tetapi bisa melihat apa yang dilakukan oleh sistem yang buat.

4.1.4 **Batasan Perencanaan dan Implementasi**

Batasan perancangan dan implementasi dari alat ini dapat dilihat sebagai berikut:

a. Sistem yang digunakan menggunakan sensor Mpu6050 6dof, sensor mendeteksi kemiringan bandul.
b. Penggerak sistem ini menggunakan motor DC 12 volt.
c. Rangka alat menggunakan *prototipe* mobil 4 roda.

4.2 Kebutuhan Sistem

Kebutuhan sistem untuk menggambakan kebutuhan - kebutuhan sistem. Kebutuhan fungsional dan non-fungsional dianalisis sesuai dengan kebutuhan sistem sehingga mempermudah dalam mendesain dan mengimplementasikan sistem.

4.2.1 Kebutuhan Fungsional

Kebutuhan fungsional merupakan kebutuhan yang harus dipenuhi dalam suatu sistem. Berikut beberapa kebutuhan fungsional dari sistem ini yang harus bekerja agar dapat bekerja dengan baik yaitu :

a. Sensor MPU6050 dapat mendeteksi kemiringan bandul dengan baik.

   Jika bandul miring ke kanan atau kiri maka sensor MPU6050 akan medeteksi kemiringan akan mengirimkan output ke arduino uno yang akan digunakan untuk menggerakan motor supaya bandul tidak jatuh.

b. Respon sistem.

   Sistem menerima respon berupa hasil kemiringan dari sensor MPU6050 yang selanjutnya diolah dengan parameter PID yang telah dimasukan oleh *user* dan mengirimkan data respon maju atau mundur ke driver motor.

c. Bidang datar untuk meletakan mobil.

   Sistem dapat berjalan sesuai yang diingikan (bandul dapat berdiri tegak terhadap permukaan bumi) jika diletakan pada bidang yang datar.

d. Embeded sistem arduino uno

   Arduino uno adalah pusat semua kontroller sistem, jika arduino uno tidak ada maka sistem tidak berjalan.

e. Sistem mampu mengerakan aktuator.

   Sistem yang dibuat menggunakan aktuator motor DC 12volt dengan kecepatan 1200 RPM.

f. Catu daya

   Sumber listrik paling penting karena sistem bekerja jika adanya sumber listrik.

4.2.2 Kebutuhan Non-Fungsional

Kebutuhan non-fungsional meliputi beberapa kebutuhan yang harus dipenuhi agar sistem berjalan dengan baik. Pada sistem ini kebutuhan fungsionalnya antara lain:

a. LED

   Tidak atau adanya LED sistem tetap bekerja.

4.3 Spesifikasi Perangkat keras

a. Laptop

Laptop yang digunakan pada pembuatan sistem ini adalah Asus A46CB generasi ke 3, dengan prosesor i5 yang dilengkapi dengan Nvidia Geforce 740M untuk *video graphic* nya. Tampilan dari laptop asus a46cb bisa dilihat pada gambar 4.1.

Gambar 4.1 Tampilan *laptop* Asus A46CB

Sumber : http://id.priceprice.com/ASUS-A46CB-WX024D-6285/specs/#Specs

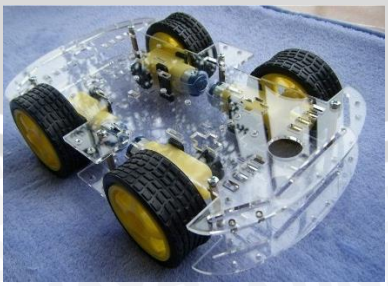
Untuk spesifikasi lengkapnya bisa dilihat pada tabel 4.3.1.

Tabel 4.1 Spesifikasi *Laptop* Asus A46CB generasi ke 3

Tipe Laptop	Notebook		
Spesifikasi Dasar			
CPU	Core i5	Model Prosesor	Core i53317U
Kecepatan Prosesor	1.7Ghz, Turbo Boost 2.6Ghz		
Mode GPU	nVidiaGeforce GT740M		
Memori & Penyimpanan			
RAM	8GB	Tipe Memori	DDR3
Slot Memori	2 DIMMs	Tipe Penyimpanan	HDD
HDD	500GB	Format HDD	
Drive Optikal	DVD/RW SuperMulti DL		
Layar			
Ukuran Layar	14 Inches LED	Resolusi	1366x768
Network			
Ethernet	10/ 100 / 100 Mbps	Wifi	802/11b/g/n
Konektifitas	HDMI, USB2.0, USB3.0		
Software			
OS	DOS	OS Ver	Free DOS
	Ukuran		
Dimensi		Berat	2 kg

b. *Prototipe* Mobil

Prototipe mobil memakai 4WD *robot car chasis,* menggunakan 4 motor DC sebagai penggerak. Untuk lebih jelasnya bisa dilihat pada gambar 4.2.



Gambar 4.2 4WD *robot car chasis*

Sumber : http://sfe-electronics.com/kategori/1704-4wd-robot-car-chasis.html

b. Motor DC 12volt

Motor yang digunakan dalam penelitian ini menggunakan motor DC, dengan input arus sebesar 12volt dengan kecepatan 1200 RPM.

c. Desain Bandul (*Pendulum*)

Desain bandul atau *pendulum* dibuat dari potongan akrilik. Panjang bandul atau *pendulum* 39 cm menyesuaikan dengan desain *prototipe* mobil, supaya bandul dapat bergerak maju atau mundur dengan bebas di

d. *Baterry Lithium Ion Rechargeable*

Baterray lithium yang digunakan adalah Merk Ultrafire dengan kapasitas arus 6800 mAH dan 3.7 volt persatuan batrei, untuk mencapai arus yang dibutuhkan sistem batrei dirangkai seri 4 batrei dengan hasil output arus 15volt. Bentuk dan tampilan baterai dapat dilihat pada gambar 4.3.

Gambar 4.3 *Baterry Lithium Ion Rechargeable*

Berikut ini spesifikasi *battery Lithium* ditunjukan pada tabel 4.2

Tabel 4.2 Spesifikasi *Battery Lithium*

Capacity	6800 mAH
Voltage	3.7 volt
Discharge	2C Constant
Weight	1g
Dimension	4.7 cm x 1.8 cm

e. Mikrokontroller

Mikrokontroller yang digunakan pada sistem ini adalah *Arduino Uno.* Tampilan dari mikrokontroller *Arduino Uno* bisa dilihat pada gambar 4.4.

Gambar 4.4 *Board* arduino uno

Sumber : http://www.caratekno.com/2015/07/pengertian-arduino-uno-mikrokontroler.html

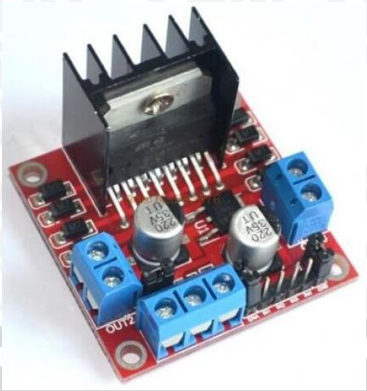
Berikut ini spesifikasi Mikrokontroler ditunjukan pada tabel 4.3

Tabel 4.3 Arduino Uno

Microcontroller	Atmega328P
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
PWM Digital I/O Pins	6
Analog Input Pins	6
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328P) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Clock Speed	16 MHz
Length	68.6 mm
Width	53.4 mm
Weight	25

f. Driver Motor

Driver motor yang digunakan adalah L298. L298 memiliki dua buah rangkaian H-Bridge di dalamnya, sehingga dapat digunakan untuk men-drive dua buah motor DC. Masing-masing dapat mengantarkan arus hingga 2A. Dalam sistem menggunakan 4 motor dirangkai menjadi 1 secara pararel. Pada gambar 4.5 dapat dilihat tampilan dari driver motor L298N.



Gambar 4.5 Motor Driver L298

Sumber : http://www.instructables.com/id/Control-DC-and-stepper-motors-with-L298N-Dual-Moto/

Berikut spesifikasi driver motor L298N dapat dilihat pada tabel 4.4.

Tabel 4.4 Driver Motor L298N

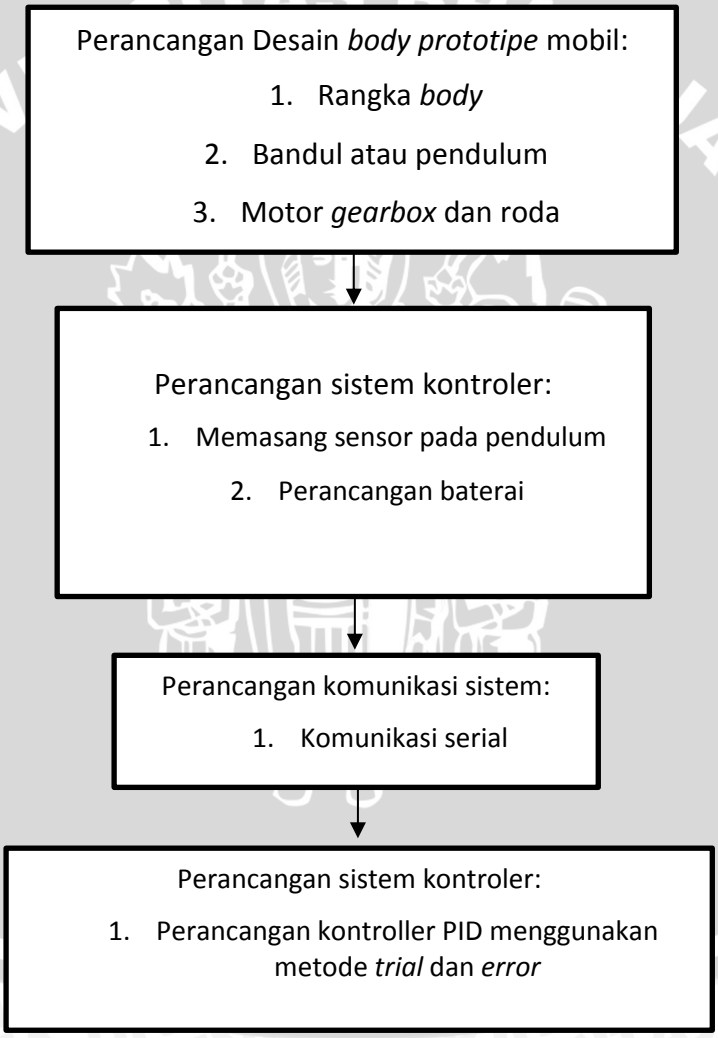
Tegangan	5 – 35 volt
Total Arus DC	4A
Package Pin I/O	12 pin

BAB 5 PERANCANGAN DAN IMPLEMENTASI

Bab ini membahas perancangan sistem yang meliputi tentang perancangan sistem, gambaran umum sistem, perancangan perangkat keras, perangkat lunak dan implementasi sistem.

5.1 Perancangan Sistem

Perancangan sistem dilakukan secara bertahap, adapun tahapan yang dilakukan untuk melakukan perancangan sistem adalah *body prototipe* mobil, perancangan elektronik, perancangan komunikasi dan perancangan *Tunning* PID, untuk lebih jelas langkah-langkahnya bisa dilihat pada diagram blok pada gambar 5.1.

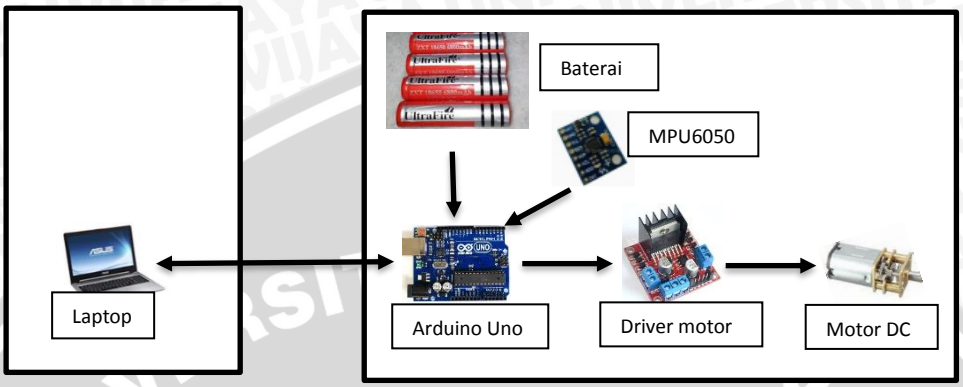


Gambar 5.1 Diagram Alir Perancangan *inverted pendulum*

5.1.1 Gambaran Umum Sistem

Sistem dirancang terbagai menjadi 2, yang pertama perancangan perangkat keras yang meliputi *body prototipe* mobil, bandul atau *pendulum*,

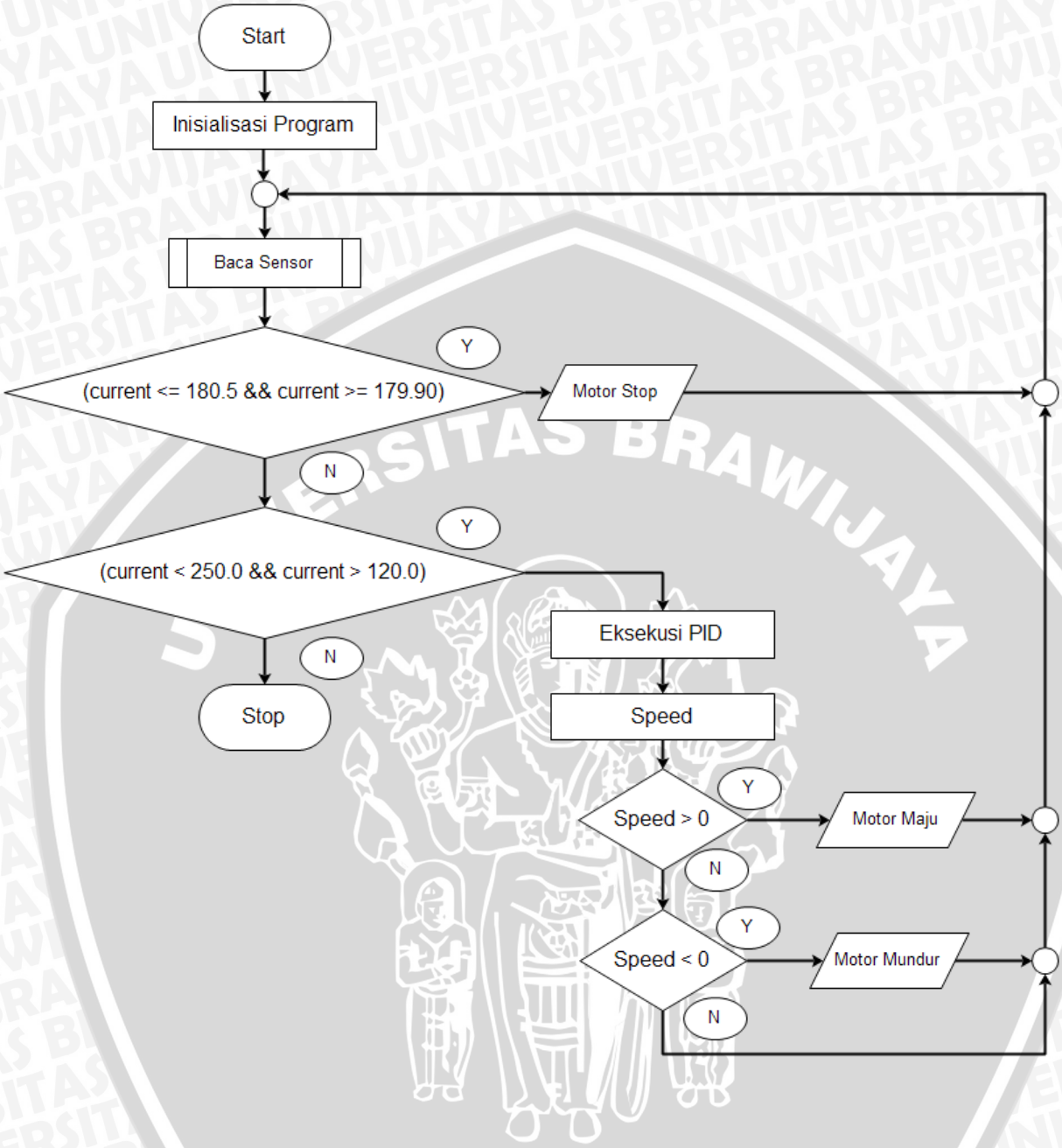
rangkaian mikrokontroler arduino uno, sensor MPU6050, driver motor L298N, dan motor DC. Yang kedua perancangan perangkat lunak yang didalamnya merancang sistem kontroller menggunakan pemograman bahasa C pada *interface* arduino uno. Untuk lebih jelasnya bisa dilihat pada gambar 5.2.



Gambar 5.2 Diagram rancangan sistem

Pada gambar 5.2 menjelaskan tentang diagram rancangan sistem yang dibuat. Pada diagram rancangan ada laptop sebagai *interface* sistem untuk melihat nilai kemiringan bandul dan melakukan pemograman pada *interface* arduino uno, selanjutnya ada baterai sebagai sumber daya sistem, MPU6050 sebagai sensor kemiringan, arduino sebagai kontroller sistem, motor driver L298N mengatur putaran motor DC dan motor DC N20 1200 RMP (*revolution per minut*) sebagai penggerak sistem yang dibuat.

Program utama dari sistem dimulai dengan inisialisasi program, setelah itu pembacaan sensor, nilai dari sensor didapat nilai sudut yang direpresentasikan oleh variabel dengan nama current, dan dibuat kondisi dimana (current <= 180.5 && current >= 179.90) jika bernilai benar maka motor stop jika bernilai tidak benar maka masuk kondisi selanjutnya (current < 250.0 && current > 120.0) jika bernilai tidak benar maka motor stop dan jika bernilai benar maka nilai akan dieksekusi pada metode PID yang menghasilkan nilai yang berupa PWM (speed), selanjutnya masuk pada kondisi speed >0 jika bernilai benar maka motor bergerak maju jika bernilai tidak masuk kondisi speed <0 jika nilai benar maka motor bergerak mundur dan jika bernilai tidak benar maka akan kembali kebaca sensor. Proses tersebut akan terus berulang sampai sistem dimatikan. Untuk lebih jelasnya dapat dilihat pada gambar 5.3.



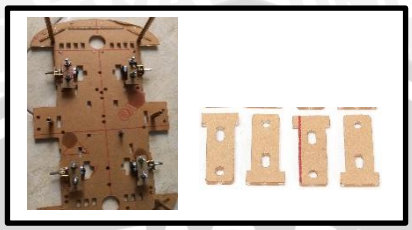
Gambar 5.3 Flowcart program utama

5.2 Perancangan Mekanik

Pada perancangan mekanik menjelaskan tentang perancangan perangkat keras seperti perancangan *body prototipe* mobil, tempat baterai, tempat motor dan *gearbox,* bandul atau *pendulum*. Untuk lebih jelasnya akan dijelaskan satu persatu pada sub-bab dibawah ini:

5.2.1 **Desain *body prototipe* mobil**

Desain *body prototipe* mobil keseluruan dibuat dari bahan akrilik, akrilik dipotong menjadi dua bagian dengan panjang 25,5 cm dan lebar 15,5 yang sama, kedua potongan tadi diberi lubang-lubang dengan diameter 1mm untuk tempat baut, dipotong untuk tempat roda. Pada gambar 5.3 bisa dilihat desain *prototipe* mobil.



Gambar 5.4 Desain *body prototipe* mobil

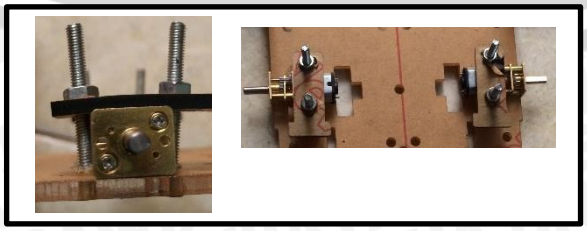
5.2.2 **Desain tempat baterai**

Desain tempat baterai dibuat dari kayu triplek yang dipotong membentuk sebuah kotak dengan panjang 13 cm dan lebar 6 cm menyesuaikan dengan bentuk baterai. Tempat baterai disini dibuat hanya untuk 4 baterai saja, untuk sambungan baterai atau konektor dibuat dari seng dan pir tempat batrei bekas, konektor tersebut disambung dengan 2 kabel (satu + dan satu -) dengan cara disolder. Untuk lebih jelas bentuk dan ukuran tempat baterai bisa dilihat pada gambar 5.5.

Gambar 5.5 Desain tempat baterai

5.2.3 **Desain motor dan *gearbox***

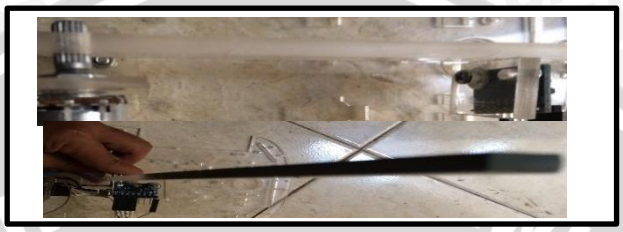
Desain motor dan *gearbox* menyesuaikan dengan bentuk motor, disini motor yang digunakan adalah motor DC N20 yang sudah ada *gearbox*nya. Untuk menempatkan motor dc ke *body* atau *chasis prototipe* mobil, pertama motor diletakan pada *chasis* diukur bentuknya, setelah itu dibuat 2 lubang baut disamping kiri dan kanan motor DC untuk membaut pangkon motor DC supaya motor DC tidak lepas saat motor berputar. Desain motor dan *gearbox* bisa dilihat pada gambar 5.6.



Gambar 5.6 Desain motor dan *gearbox*

5.2.4 **Desain bandul atau *pendulum***

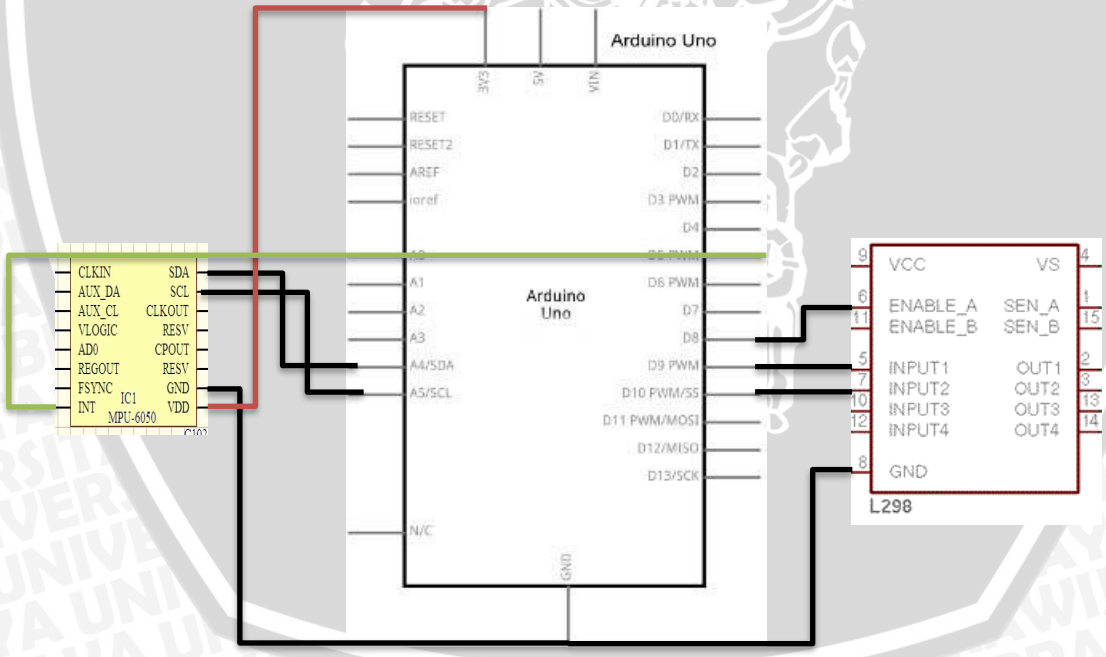
Desain bandul atau *pendulum* dibuat dari akrilik yang dipotong dengan panjang 30cm dan lebar 2cm. Bentuknya memanjang dengan ujung lebih kecil dari pangkal bandul atau *pendulum*. Pada pangkal bandul atau *pendulum* dibuat lubang untuk meletakan potensio meter, ini dimaksudkan agar bandul atau *pendulum* bergerak bebas kedepan atau kebelakang. Pada bandul atau *pendulum* juga dibuat tempat untuk meletakan sensor mpu6050. Untuk lebih jelasnya bisa dilihat pada gambar 5.7.



Gambar 5.7 Desain bandul atau *pendulum*

5.3 Perancangan *Elektrik*

Perancangan *elektrik* yang digunakan pada sistem ini yaitu sensor mpu6050 6dof, mikrokontroller arduino, dan driver motor L298N. Rangkaian dari masing alat akan dijelaskan pada gambar 5.8.



Gambar 5.8 skema rangkaian sistem

Pada gambar 5.8 dapat dilihat ada 3 buah komponen yang dirangkai menjadi satu sistem, pertama ada kontroller arduino uno yang membutuhkan catu daya 5 volt, arduino uno dihubungkan dengan sensor mpu6050 6dof melalui port analog 4 dan 5, port A5 disambungkan dengan port scl sensor dan port a4 disambungkan dengan port sda sensor, port catu daya 3,3 volt dihubungkan dengan port vdd pada sensor sebagai catu daya dan port ground pada arduino

juga dihubungkan pada port ground pada sensor. Untuk driver motor juga dihubungkan dengan kontroller arduino uno dengan port digital 9 dan 10 dihubungkan ke port 5 (input 1) dan 7 (input 2) driver motor, port digital 8 dihubungkan ke port 6 (enable_a) pada driver motor, port 2 dan 3 sebagai port output driver motor dihubungkan ke motor DC. Driver motor membutuhkan 12volt sebagai catu daya. Untuk lebih jelasnya sambungan atar port bisa dilihat pada tabel 5.1.

Tabel 5.1 Sambungan antara port

Port Kontroller		Port sensor MPU6050		Port driver motor
A5	Dihubungkan dengan	SCL	Dihubungkan dengan	
A4		SDA		
D5		IN		
3.3volt		VDD		
Ground		Ground		
D9				Input_1
D10				Input_2
D8				EnableA
Ground				Ground

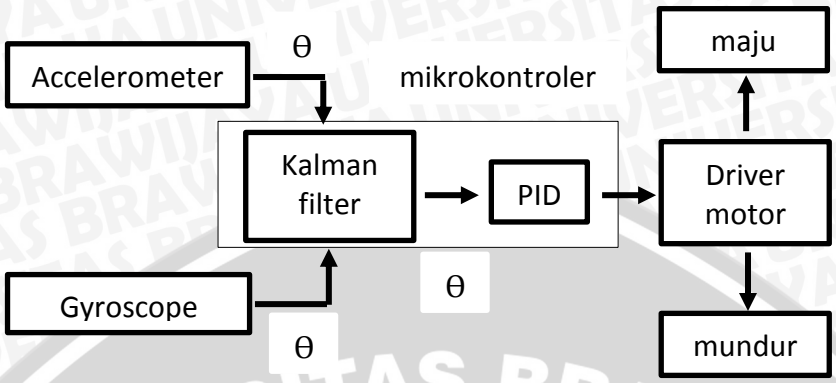
5.4 Perancangan Komunikasi

Perancangan komunikasi pada sistem ini adalah komunikasi atara kontroller arduino uno dengan laptop secara serial. Untuk menghubungkan laptop dengan arduino uno menggunakan USB A, setelah dihubungkan apabila lampu led yang ada pada arduino uno berkedip-kedip warna kuning maka arduino uno sudah terhubung dengan laptop. Gambar komuniksai arduino uno dengan laptop bisa dilihat pada gambar 5.9.

Gambar 5.9 Gambar komunikasi arduino uno dengan laptop

5.5 Perancangan Kontroller

5.5.1 Perancangan Kontroler Keseluruhan



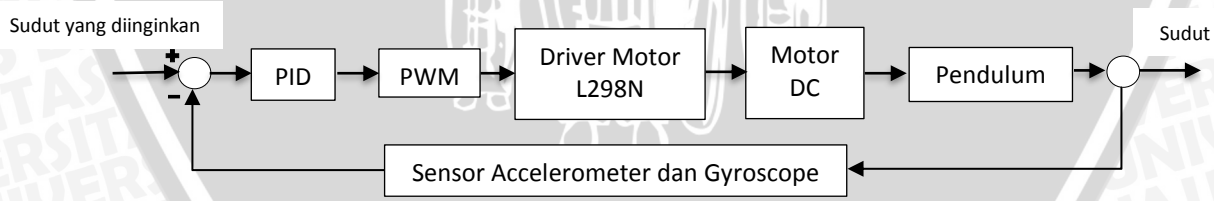
Gambar 5.10 Rancangan Sistem Kontrol pada *inverted pendulum*

System yang telah dirancangan ini dapat menyeimbangakan *pendulum* atau bandul. Hal ini dapat dilakukan dengan cara mendeteksi kemiringan dari *pendulum* atau bandul kemudian dari data kemiringan dikirim ke motor DC untuk menyeimbangkan *pendulum* atau bandul.

Kalman Filter diimplementasikan pada *pendulum*, dan dapat menemukan model yang akurat. Pemodelan didekati dengan metoda linear agar mudah diimplementasi. Kalman filter dapat memanfaatkan data accelerometer untuk menghilangkan drift dari keluaran gyroscope. Dalam proses, noise dari accelerometer juga akan diminimalkan bahkan dihilangkan.

Kalman filter sudah ada pada *library* arduino uno untuk pemfilteran sensor IMU6050 maka dari itu tidak dijelaskan pada laporan ini melainkan hanya menggunakan *library* dari arduino uno.

5.5.2 Perancangan Kontroller PID



Gambar 5.11 Diagram Blok Sistem

Keterangan gambar 5.11 akan dijabarkan dibawah ini:

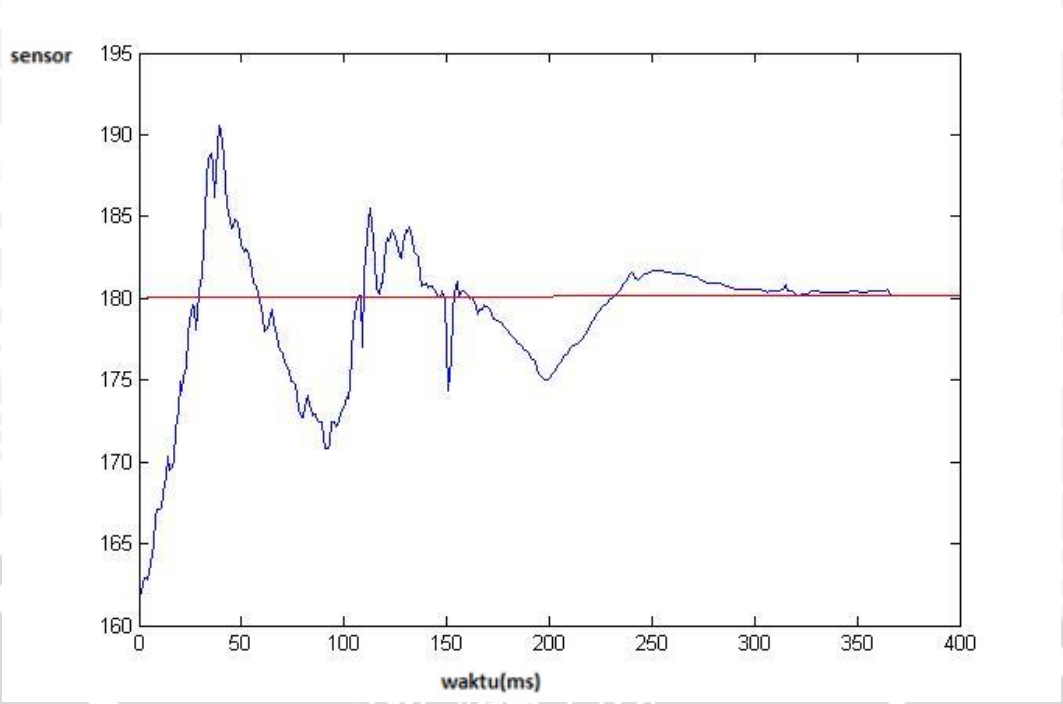
- Sudut yang diinginkan adalah nilai keseimbangan dan kecepatan posisi pada *pendulum* dan kecepatan motor yang dihasilkan oleh karakteristik driver motor dengan tegangan sebesar 12 volt.
- PID adalah kontroller untuk mengolah masukan dari sudut yang diingingkan untuk menyetabilkan *pendulum*.

- PWM dihasilkan dari pengolahan sudut menggunakan kontroller PID yang dilakukan pada arduino uno.
- Driver L298N adalah driver motor yang digunakan untuk menerima keluaran PWM dari arduino uno dan mengatur putaran motor DC.
- Aktuator menggunakan satu motor DC 12 volt yang berfungsi untuk menyeimbangkan *pendulum* yang mendapat sinyal masukan dari *output* PWM driver L298N.
- Nilai sudut didapat dari sensor yang dipasang pada *pendulum*.
- *Output* yang diingikan berupa sudut, jika sudut yang diinginkan tidak sesuai yang diinginkan, sensor akan mendapatkan nilai kemiringan *pendulum* kembali.
- Sensor IMU 6050 untuk menentukan kecepatan posisi dan keseimbangan pada *pendulum* dengan range yang sudah dilakukan saat penelitian.

Kontroller yang digunakan pada penelitian ini adalah kontroller Proporsional, Intregratif dan Derivatif atau disingkat PID. Penelitian ini menggunakan salah satu *tunning* dari PID yaitu *trial* dan *error* yaitu dengan cara memasukan nilai proporsional (Pi), integratif (Ki), dan derivatif (Kd). Setiap nilai mempunyai pengaruh yang berbeda seperti nilai Pi mengurangi *error steady state* (waktu awal), menurunkan *rise time* (waktu naik), dan meningkatkan *overshoot* (titik maksimum), selanjutnya nilai ki menghilangkan *error steady state* (waktu awal), menurunkan *rise time* (waktu naik) dan meningkatkan *overshoot* (titik maksimum), dan yang terakhir Kd menurunkan *overshoot* (titik maksimum) atau memberi efek redaman.

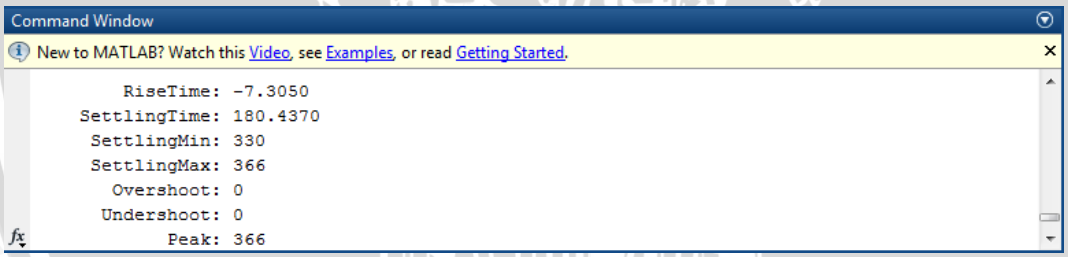
Pada penelitian ini terdapat beberapa percobaan memasukan nilai Pi, Ki, dan Kd sampai menemukan nilai yang dianggap paling baik untuk sistem yang dibuat.

1. Percobaan dengan Kp = 30, Ki = 15, dan Kd = 4 hasilnya ditunjukan gambar 5.12.



Gambar 5.12 Grafik Kp=30,Ki=15,Kd=4

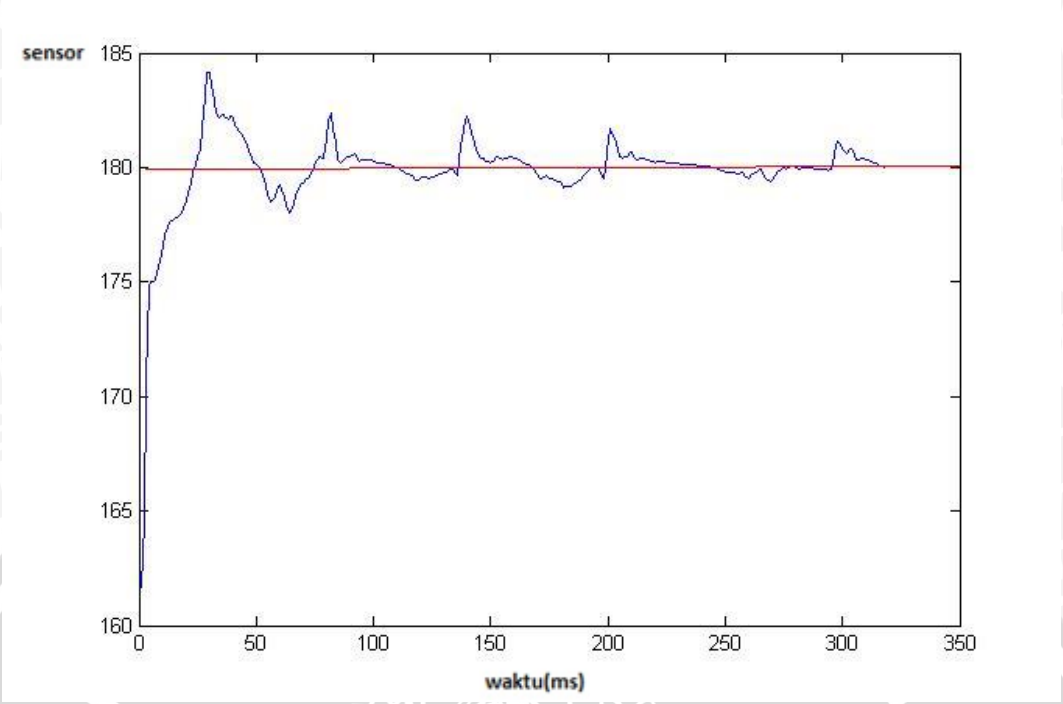
Dari percobaan yang sudah dilakukan dengan nilai derajat yang diinginkan pada nilai 180, grafik di gambar 5.12 menunjukan sistem dapat sesuai nilai yang diinginkan pada waktu ke 0,00290 detik.



Gambar 5.13 Step Info Kp=30,Ki=15,Kd=4

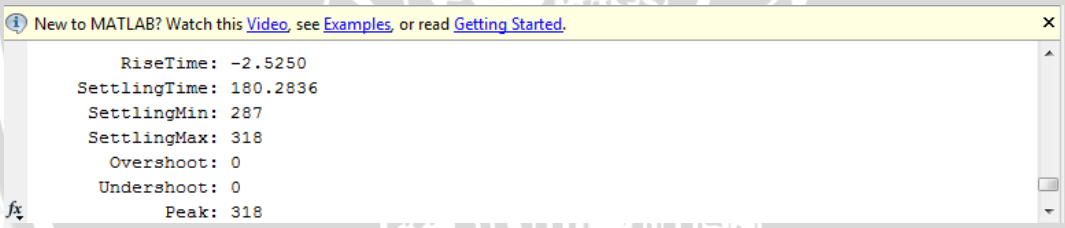
Dari gambar 5.13 didapat waktu naik -0,0073050 detik dan SettlingTime atau waktu tunak 0,001804370

2. Percobaan dengan Kp = 50, Ki = 10, dan Kd = 2 hasilnya ditunjukan gambar 5.14.



Gambar 5.14 Grafik Kp=50,Ki=10,Kd=2

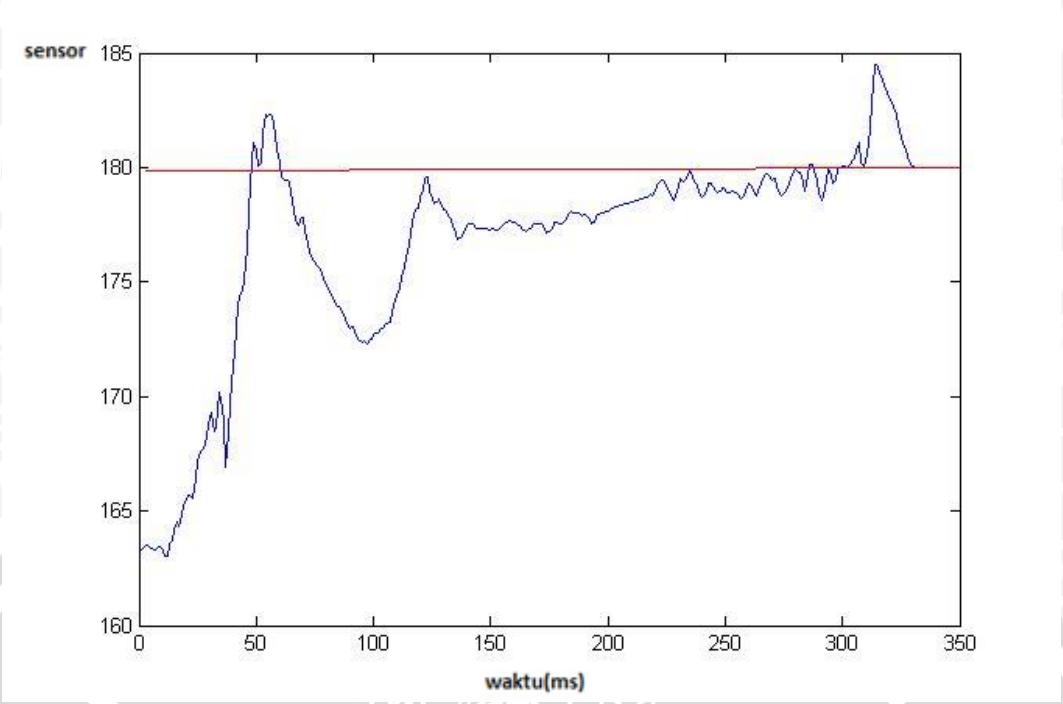
Dari percobaan yang sudah dilakukan dengan nilai derajat yang diinginkan pada nilai 180, grafik di gambar 5.14 menunjukan sistem dapat sesuai nilai yang diinginkan pada waktu ke 0,00220 detik sampai dengan 0,00290 setelah itu berosilasi kembali.



Gambar 5.15 Step Info Kp=50,Ki=10,Kd=2

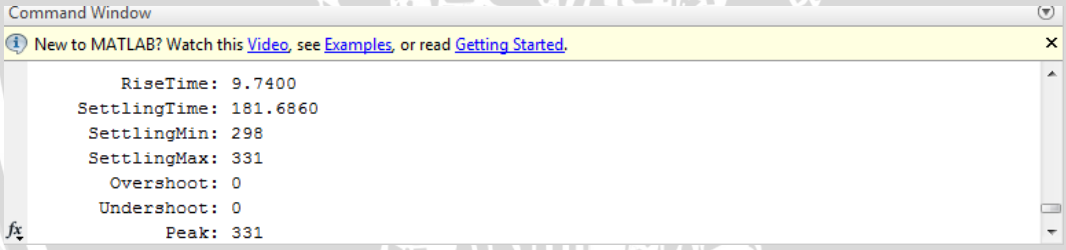
Dari gambar 5.15 didapat waktu naik -0,0025250 detik dan SettlingTime atau waktu tunak 0,001802836

3. Percobaan dengan Kp = 60, Ki = 10, dan Kd = 1 hasilnya ditunjukan gambar 5.16.



Gambar 5.16 Grafik Kp=60,Ki=10,Kd=1

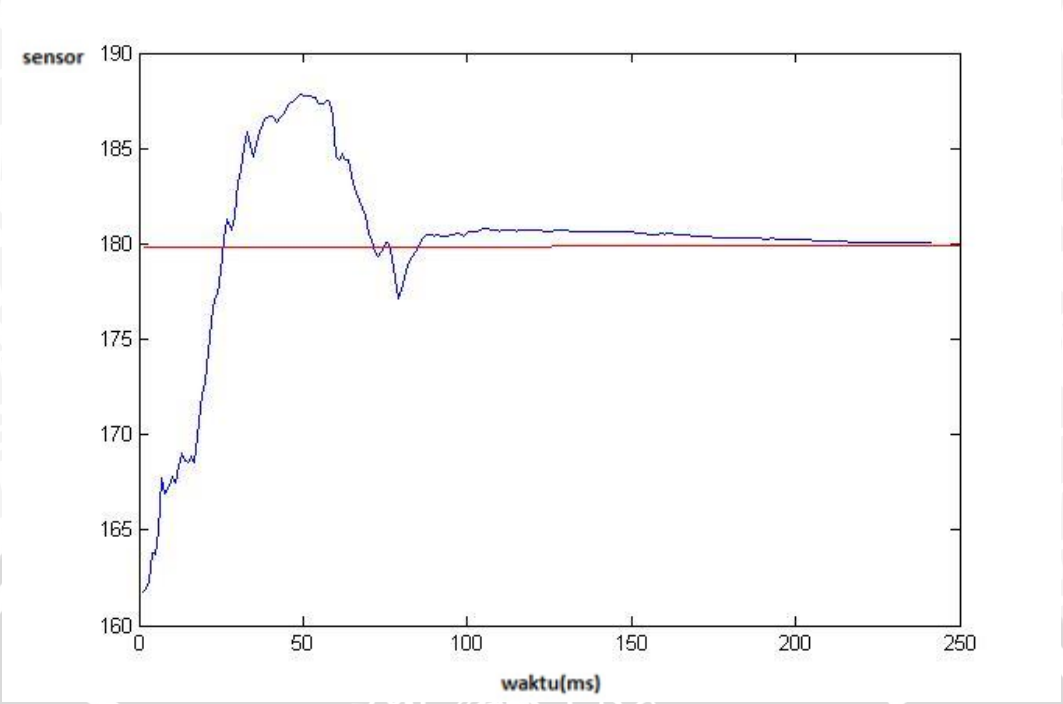
Dari percobaan yang sudah dilakukan dengan nilai derajat yang diinginkan pada nilai 180, grafik di gambar 5.16 menunjukan sistem berosilasi secara terus menerus.



Gambar 5.17 Step Info Kp=60,Ki=10,Kd=1

Dari gambar 5.17 didapat waktu naik 0,0097400 detik dan SettlingTime atau waktu tunak 0,001816860.

4. Percobaan dengan Kp = 100, Ki = 2, dan Kd = 0 hasilnya ditunjukan gambar 5.18.



Gambar 5.18 Grafik Kp=100,Ki=2,Kd=0

Dari percobaan yang sudah dilakukan dengan nilai derajat yang diinginkan pada nilai 180, grafik di gambar 5.18 menunjukan sistem dapat sesuai nilai yang diinginkan pada waktu ke 0,00165 detik dan seterusnya.

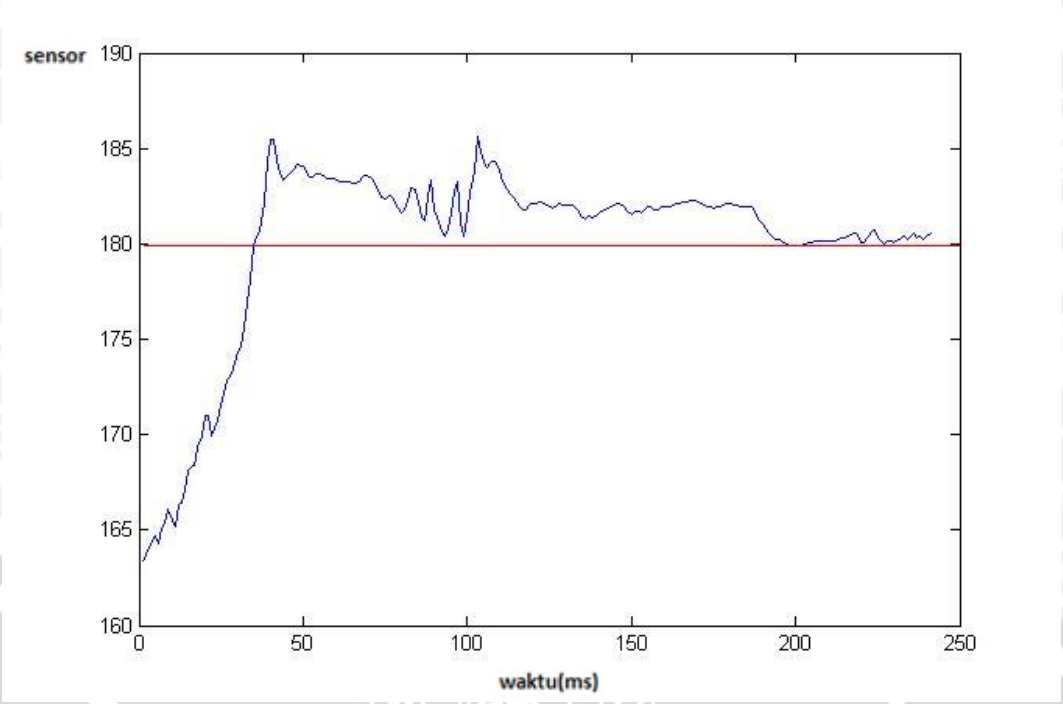
```
New to MATLAB? Watch this Video, see Examples, or read Getting Started.

        RiseTime: 1.4800
    SettlingTime: 180.0800
     SettlingMin: 217
     SettlingMax: 241
       Overshoot: 0
      Undershoot: 0
            Peak: 241
```

Gambar 5.19 Step Info Kp=100,Ki=2,Kd=0

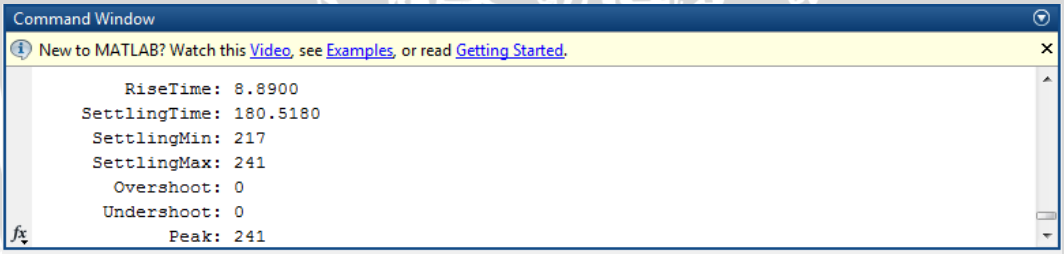
Dari gambar 5.19 didapat waktu naik 0,0014800 detik dan SettlingTime atau waktu tunak 0,001800800 detik.

5. Percobaan dengan Kp = 150, Ki = 30, dan Kd = 2 hasilnya ditunjukan gambar 5.20.



Gambar 5.20 Grafik Kp=150,Ki=30,Kd=2

Dari percobaan yang sudah dilakukan dengan nilai derajat yang diinginkan pada nilai 180, grafik di gambar 5.20 menunjukan sistem dapat sesuai nilai yang diinginkan pada waktu ke 0,00190 detik.



Gambar 5.21 Step Info Kp=150,Ki=30,Kd=2

Dari gambar 5.21 didapat waktu naik 0,0088900 detik dan SettlingTime atau waktu tunak 0,001805180 detik.

Setelah melakukan beberapa percobaan didapatkan waktu tercepat dan sesuai yang diinginkan pada 0,00165 detik dengan nilai Kp = 100, Ki = 2, dan Kd = 0.

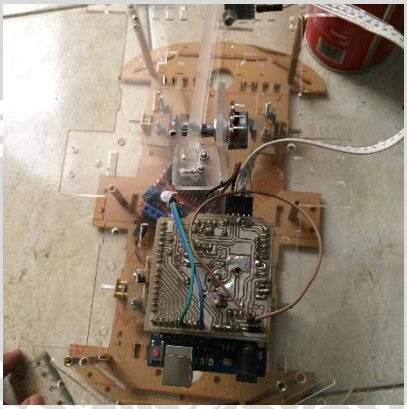
5.6 Implementasi Sistem

Implementasi sistem meliputi implementasi perangkat keras dan perangkat lunak, pada sub-bab 5.6.1 membahas implementasi perangkat keras dan 5.6.2 membahas implementasi perangkat lunak.

5.6.1 **Implementasi perangkat keras**

Implementasi perangkat keras pada *inverted pendulum* pada *prototipe* mobil yang digunakan adalah mikrokontroller arduino uno, driver motor menggunakan L298N, driver motor ini membutuhkan tegangan 12volt, sensor yang digunakan adalah sensor MPU6050 6dof.

Keseluruhan alat dihubungkan satu sama lain melalui port dimasing-masing alat agar terbentuk sebuah sistem yang dapat bekerja dengan baik. Pada gambar 5.22 akan ditunjukan implementasi perangkat keras yang sudah terhubung satu sama lain.



Gambar 5.22 Implementasi perangkat keras

Spesifikasi dari perangkat keras *inverted pendulum* pada *prototipe* mobil sebagai berikut:

- Perangkat komputer / laptop.
- Baterai.
- Arduino uno.
- Driver motor.
- Sensor IMU 6050.
- Motor DC 12volt.

5.6.2 **Implementasi perangkat lunak**

Perangkat lunak yang digunakan dalam penelitian ini yaitu interface Arduino uno, dibawah ini bisa dilihat potongan pogram dari *inverted pendulum.*

```
#include <Wire.h>

#include <Kalman.h>

#define RESTRICT_PITCH // Comment out to restrict roll to ±90deg instead - please read: http://www.freescale.com/files/sensors/doc/app_note/AN3461.pdf


Kalman kalmanX; // Create the Kalman instances
```

```
Kalman kalmanY;
int En = 8;
int dir = 9;
int rem = 10;

/* IMU Data */
int16_t accX, accY, accZ;
int16_t gyroX, gyroY, gyroZ;
int16_t tempRaw;

float gyroXangle, gyroYangle, accXangle; // Angle calculate using the gyro only
float compAngleX, compAngleY; // Calculated angle using a complementary filter
float kalAngleX, kalAngleY; // Calculated angle using a Kalman filter

uint32_t timer;
uint8_t i2cData[14]; // Buffer for I2C data
float current;
// masukan nilai PID
const float Kp = 100;
const float Ki = 2;
const float Kd = 0;
float pTerm, iTerm, dTerm, integrated_error, last_error, error;
const float K = 1.9*1.12; //set poin
#define   GUARD_GAIN   10.0
int speed;

#define runEvery(t) for (static typeof(t) _lasttime;(typeof(t))((typeof(t))millis() - _lasttime) > (t);_lasttime += (t))

// TODO: Make calibration routine
void analogWrite25k(int pin, int value)
{
    switch (pin) {
        case 9:
            OCR1A = value;
```

```
            break;
        case 10:
            OCR1B = value;
            break;
        default:
            // no other pin will work
            break;
    }
  }
  void setup() {
    Serial.begin(9600);

    // Configure Timer 1 for PWM @ 25 kHz.
     //TCCR1B = TCCR1B & B11111000 | B00000001; // Set PWM
frequency for D9 & D10 :
      TCCR1A = 0;           // undo the configuration done
by...
      TCCR1B = 0;           // ...the Arduino core library
      TCNT1  = 0;           // reset timer
      TCCR1A = _BV(COM1A1)  // non-inverted PWM on ch. A
             | _BV(COM1B1)  // same on ch; B
             | _BV(WGM11);  // mode 10: ph. correct PWM,
TOP = ICR1
      TCCR1B = _BV(WGM13)   // ditto
             | _BV(CS10);   // prescaler = 1
      ICR1   = 320;         // TOP = 320

    pinMode(dir,OUTPUT);
    pinMode(rem,OUTPUT);
    pinMode(En,OUTPUT);
    Wire.begin();
    imu_data();
   // Configure Timer 1 for PWM @ 25 kHz.
     //TCCR1B = TCCR1B & B11111000 | B00000001; // Set PWM
frequency for D9 & D10 :
      TCCR1A = 0;           // undo the configuration done
by...
      TCCR1B = 0;           // ...the Arduino core library
      TCNT1  = 0;           // reset timer
      TCCR1A = _BV(COM1A1)  // non-inverted PWM on ch. A
```

```
                | _BV(COM1B1)  // same on ch; B
                | _BV(WGM11);   // mode 10: ph. correct PWM,
  TOP = ICR1
        TCCR1B = _BV(WGM13)   // ditto
                | _BV(CS10);   // prescaler = 1
        ICR1   = 320;         // TOP = 320
    }

    void loop() {
    //runEvery(25)  // run code @ 40 Hz
    //  {
       filter();
       if (current <= 180.5 && current >= 179.90)
       {
         stop();
        Serial.print("niali derajat = ");
        Serial.print(current-90); Serial.print("\r\n");
        Serial.print("niali PWM = ");
        Serial.print(speed); Serial.print("\r\n");
        }
       else{
        filter();
        if (current < 250.0 && current > 120.0)
        {
        Pid();
        Motors();
        Serial.print("nilai derajat = ");
        Serial.print(current-90); Serial.print("\r\n");
        Serial.print("nilai PWM = ");
        Serial.print(speed); Serial.print("\r\n");

        }
        else
        {
          remm();
           Serial.print(speed); Serial.print("\r\n");
        }
      }
}
```

BAB 6 PENGUJIAN DAN ANALISA

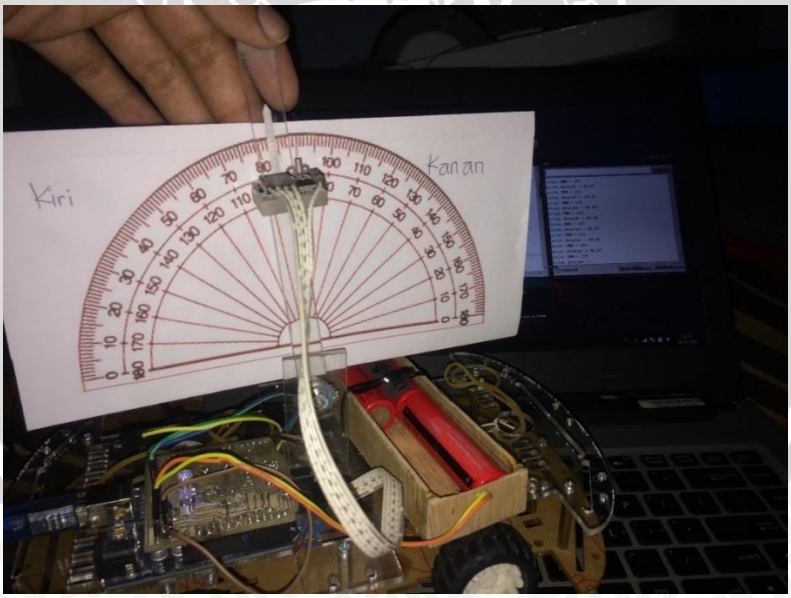
Pada bab ini akan dilakukan pengujian dan analisa dari sistem yang bertujuan untuk mengetahui akurasi sensor serta memastikan bahwa apakah sistem akan berjalan dengan baik. Pengujian dilakukan dengan dua tahap yaitu pengujian sensor dan pengujian keseluruhan alat.

6.1 Pengujian sensor

Dalam melakukan pengujian sensor yang diuji adalah perubahan sudut *gyroscope* dan *accelerometer* yang ada pada sensor MPU6050. Pengujian *gyroscope* dan *accelerometer* bertujuan untuk mengetahui tingkat keakuratan dari *gyroscope* dan *accelerometer* dalam membaca perubahan sudut bandul atau *pendulum*.

Sensor terpasang pada bandul atau *pendulum* dan terhubung dengan mikrokontroller arduino uno dan kemudian data dari keluaran sensor akan dibuat input dan keluarannya adalah nilai RPM. Kemiringan bandul atau *pendulum* dapat diubah – ubah sesuai dengan papan sudut yang sudah disediakan.

Perubahan sudut yang terjadi akan ditampilkan lewat serial monitor arduino uno pada komputer. Cara pengambilan data pada sensor MPU6050 dengan membandingkan nilai sudut yang didapatkan oleh sensor dan nilai sudut dari busur. Gambar 6.1 menunjukan cara pengambilan data pada sensor MPU6050 dan tabel 6.1 menunjukan pengamatan pada sensor MPU6050.



Gambar 6.1 Cara mendapatkan data sensor MPU6050

Tabel 6.1 Data pengamatan pada sensor MPU6050

No	DATA			
	Sudut sensor (^{0})	Sudut Aktual (^{0})	Akursi (0-100%)	Arah *pendulum*
1	40.40	40	99%	Miring kiri
2	50.30	50	99.40%	Miring kiri
3	60.41	60	99.32%	Miring kiri
4	70.34	70	99.51%	Miring kiri
5	80.61	80	99.24%	Miring kiri
6	90.07	90	99.92%	Tegak lurus
7	100.57	100	99.43%	Miring kanan
8	110.39	110	99.64%	Miring kanan
9	120.40	120	99.66%	Miring kanan
10	130.35	130	99.73%	Miring kanan
Rata-rata			99.48%	

Dari tabel 6.1 terlihat bahwa pembacaan sudut *gyroscope* dan *accelerometer* sudah akurat, terlihat dari selisih sudut sensor dengan sudut aktual yang kecil.

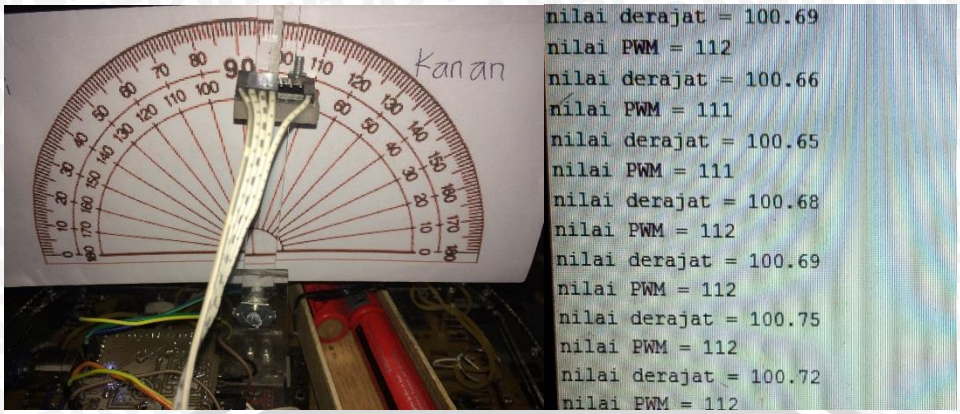
Akurasi sensor didapat dari perhitungan $akursi = \frac{sudut\ aktual}{sudut\ sensor}\ x\ 100\%$

Contohnya perhitungan akurasi sensor $akursi = \frac{90}{90.07}\ x\ 100\%$ = 99.92

6.2 Pengujian keseluruhan alat

Pengujian sistem secara keseluruhan memiliki tujuan untuk mengetahui cara kerja perangkat keras dan perangkat lunak setelah diintegerasikan dalam sebuah sistem yang terpadu.

Mula – mula *prototipe* mobil diletakan pada bidang datar (0^o) kemudian *pendulum* ditegakan ke atas terhadap posisi *prototipe* mobil kemudian *pendulum* dilihat berapa derajat dan berapa nilai PWM yang dihasilkan pada serial monitor, untuk melihat cara pengambilan datanya dapat dilihat pada gambar 6.2, gambar tersebut memperlihatkan pengukuran hasil derajat sensor dan nilai PWM dan posisi *pendulum* dalam ukuran derajat.



Gambar 6.2 Cara pengambilan data

Parameter yang diamati dalam penelitian ini adalah perubahan nilai derajat sensor dengan nilai PWM yang dihasilkan perhitungan sistem, sensor hanya bekerja atau mendeteksi kemiringan *pendulum* ketika nilai derajat sensor dalam *range* antara 250.0^0 sampai 120.0^0 selain itu sensor tidak mendeteksi kemiringan *pendulum*. Tabel 6.2 menunjukan data hasil pengujian keseluruhan sistem.

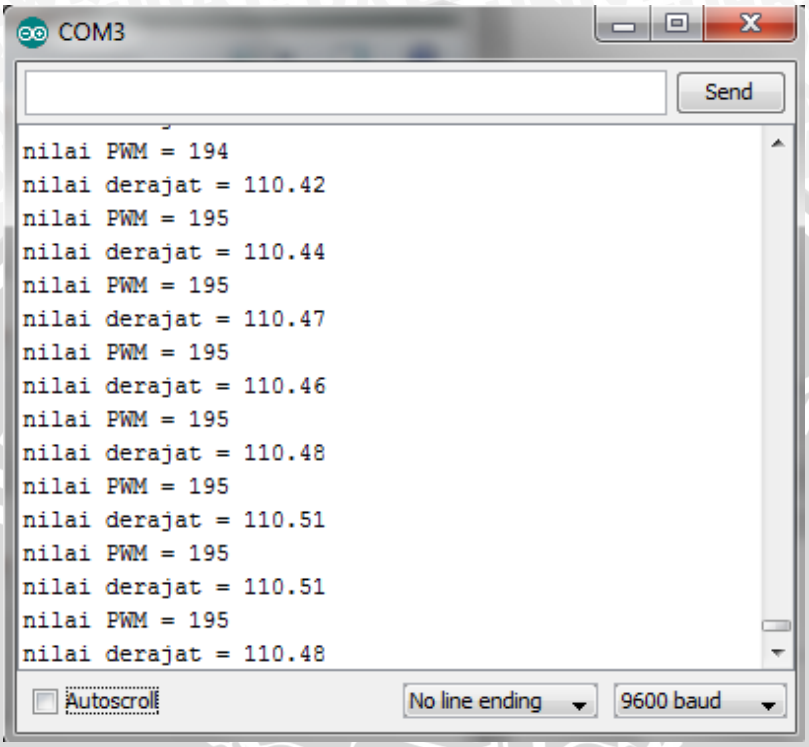
Tabel 6.2 Hasil pengujian nilai derajat sensor tehadap nilai PWM

No	Nilai derajat (^{0})	Nilai PWM	Posisi *pendulum* (derajat)	Gerak Motor DC (RPM)
1	87.61	0	Lurus (87^0)	Diam (0 RPM)
2	90.85	28	Lurus (90^0)	Diam (121 RPM)
3	105.60	154	miring ke kanan (105^0)	Bergerak pelan (664 RPM)
4	110.46	195	miring ke kanan (110^0)	Berputar cepat (841 RPM)
5	120.40	255	Miring ke kanan (120^0)	Berputar cepat (1100 RPM)
6	80.56	- 58	Miring ke kiri (80^0)	Berputar pelan (-250 RPM)
7	70.64	- 140	Miring ke kiri (70^0)	Berputar pelan (-603 RPM)
8	65.10	- 190	Miring ke kiri (65^0)	Berputar cepat (-820 RPM)
9	50.70	- 255	Miring ke kiri (50^0)	Berputar cepat (- 1100 RPM)

Nilai RPM didapat dari perhitungan $\boldsymbol{RPM = \frac{PWM}{255}\ x\ kecepatan\ motor}$

Contohnya $RPM = \frac{195}{255}\ x\ 1100 = 841\ RPM$

Gambar 6.3 contoh dari beberapa hasil pengujian keseluruhan sistem. Digambar tersebut dapat dilihat besar nilai derajat dan nilai PWM.



Gambar 6.3 Screenshot serial monitor

Dari tabel 6.2 dapat dilihat pengaruh perubahan derajat dengan nilai PWM, hasil pengujian membuktikan bahwa sistem berjalan sesuai yang diinginkan, ketika *pendulum* miring ke kanan maka motor akan bergerak kekanan, *pendulum* miring kekiri motor akan bergerak kekiri, dan *pendulum* tegak ke atas motor diam.

6.3 Analisa

Dari beberapa pengujian yang sudah dilakukan mendapatkan beberapa hasil. Perubahan nilai derajat sensor berpengaruh pada nilai PWM yang dihasilkan, pada pengujian sensor, sensor bekerja dengan baik ditunjukan dengan nilai akurasi 99.48%. Motor dapat bergerak maju atau mundur sesuai arah jatuhnya *pendulum*.

BAB 7 PENUTUP

7.1 Kesimpulan

Berdasarkan hasil perancangan, implementasi dan pengujian yang dilakukan, maka diambil kesimpulan sebagai berikut:

a. Dari pengujian yang sudah dilakukan mendapatkan hasil rata-rata keakuratan pengukuran sensor sebesar 99.48%, dari hasi pengamatan tersebut disimpulkan sensor yang dipakai akurat.
b. Dari pengujian keseluruhan sistem mendapatkan hasil, sensor mampu membaca sudut dengan baik, gerak motor DC sesuai dengan arah jatuhnya bandul atau *pendulum*.
c. Dengan menggunakan metode proporsional integratif derivatif yang ditanamkan pada kontroller arduino uno dengan nilai Kp = 100, Ki = 2 dan Kd = 0 sistem dapat berjalan sesuai yang diinginkan yaitu bandul atau *pendulum* dapat seimbang dan motor mampu bergerak sesuai arah jatuhnya bandul atau *pendulum*.

7.2 Saran

Pengembangan yang lebih lanjut diharapkan dapat menghasilkan sistem yang lebih baik dan kompleks dengan memperbaiki kinerja sistem yang telah dibuat, dari itu dapat ditarik beberapa saran sebagai berikut:

1. Sistem ini dapat dikembangkan dan di implementasikan pada alat transportasi dua roda atau satu roda.
2. Disarankan untuk menggunakan sensor selain MPU6050 dan metode yang lain agar menghasilkan sistem yang lebih baik pada penelitian selanjutnya.
3. Pada penelitian selanjutnya disarankan merubah bentuk body *prototipe* mobil menjadi alat transportasi sesungguhnya agar lebih bermanfaat.

DAFTAR PUSTAKA

Ketaren, Lio Prisko., et al. 2015. *Balancing* Robot Beroda Dua Menggunakan Kontrol Proposional Integral dan Derivatif. ELEMENTER 1:48-39

Tanzania, Niora Fatimah., Agustinah, Trihastuti. "Stabilisasi Pada Sistem *Pendulum*-Kereta dengan Menggunakan Metode Fuzzy-Sliding Mode Control". JURNAL TEKNIK POMITS. Vol. 3, No.1, 2014

Prasad, Lal Bahadur., et al. 2012. Modelling and simulation for optimal control of nonlinier inverted *pendulum* dynamical system using pid controller and LQR. ISBN: pp. 138-143.

Fathonah, Chusnul. 2016. Analisis Kesetabilan dan Kontrol Optimal *Double Pendulum* Terbalik pada Kereta Menggunakan Metode *Linear Quadratic Regulator* (LQR). Universitas Negeri Maulana Malik Ibrahim Malang.

Instructables. 2015. Control DC stepper motor with L298N Dual Motor Controller Moduler Arduino. [Teks online] Tersedia di: http://www.instructables.com/id/Control-DC-and-stepper-motors-with-L298N-Dual-Moto/ [diakses 30 Januari 2015]

Zonaelektro. 2013. Motor DC. [Teks online] Tersedia di: http://zonaelektro.net/motor-dc/ [diakses 5 Februari 2015]

Ihsan. 2013. Pengertian Arduino UNO Mikrokontroler Atmega328. [Teks dan image online] Tersedia di: http://www.caratekno.com/2015/07/pengertian-arduino-uno-mikrokontroler.html [diakses 5 Februari 2015]

Anyakrawati, Aretasiwi. 2015. "Implementasi Model Reference *Adaptive Systems* (Mras) Untuk Kestabilan Pada *Rotary Inverted Pendulum*," Jurnal Seminar 1:6-1

Agustian, Indra. M.Eng. 2013. Definisi Sistem Kendali. [Teks dan image online] Tersedia di: http://te.unib.ac.id/lecturer/indraagustian/2013/06/definisi-sistem-kendali/ [diakses 5 Februari 2015]

Mahardy, Denny. 2015. Walkherwheel 350, alat transportasi canggih masa depan. [teks dan image online] Tersedia di: http://tekno.liputan6.com/read/2157465/walkherwheel-350-alat-transportasi-canggih-masa-depan [Diakses 15 Agustus 2016]

Seifert, Kurt., Camacho Oscar. (2007). Implementing Positioning Algorithms Using Accelerometers, freescale Semiconductor, Rev 0.

Daftar Lampiran

Lampiran Program

1. Program PID

```
void Pid(){
  error = 180 - current; //nilai error pertama dr 180 - derajat sensor
  pTerm = Kp * error;// kp.e(t)
  integrated_error += error; //penambahan kelanjutan error
  iTerm = Ki * constrain(integrated_error, -GUARD_GAIN, GUARD_GAIN); // gain = set poin ki.integral e(t).dt
  dTerm = Kd * (error - last_error);
  last_error = error;
  speed = constrain(K*(pTerm + iTerm + dTerm), -255, 255); //dpt nilai pwm
  }//contrain adalah batasan nilai speed -255 sampai 255


void filter(){
  double dt = (double)(micros() - timer) / 1000000; // Calculate delta time
  while(i2cRead(0x3B,i2cData,14));
  accX = ((i2cData[0] << 8) | i2cData[1]);
  accY = ((i2cData[2] << 8) | i2cData[3]);
  accZ = ((i2cData[4] << 8) | i2cData[5]);
  tempRaw = ((i2cData[6] << 8) | i2cData[7]);
  gyroX = ((i2cData[8] << 8) | i2cData[9]);
  gyroY = ((i2cData[10] << 8) | i2cData[11]);
  gyroZ = ((i2cData[12] << 8) | i2cData[13]);
  accXangle = (atan2(accY,accZ)+PI)*RAD_TO_DEG;
  double gyroXrate = (double)gyroX/131.0;
  current = kalmanX.getAngle(accXangle, gyroXrate, dt);
  timer = micros();
  }
```

2. Program Motor

```
void Motors(){
  if (speed > 0)
  {
    //forward
```

```
    //speed = map(speed,0,-255,0,255);
    //speed=speed+100;
    digitalWrite(En,HIGH);
    analogWrite25k(rem, speed);
    analogWrite25k(dir, 0);
  }
  else
  {
    // backward
    // speed=speed-100;
    speed = map(speed,0,-255,0,255);
    digitalWrite(En,HIGH);
    analogWrite25k(rem, 0);
    analogWrite25k(dir, speed);
  }
}
void remm(){
  digitalWrite(En,HIGH);
  analogWrite25k(rem, 1);
  analogWrite25k(dir, 1);
  }
void stop(){
  digitalWrite(En,LOW);
  analogWrite25k(rem, 0);
  analogWrite25k(dir, 0);
}
```

3. Program MPU6050

```
void imu_data(){
  #if ARDUINO >= 157
  Wire.setClock(400000UL); // Set I2C frequency to 400kHz
#else
  TWBR = ((F_CPU / 400000UL) - 16) / 2; // Set I2C frequency to
400kHz
#endif

  i2cData[0] = 7; // Set the sample rate to 1000Hz - 8kHz/(7+1)
= 1000Hz
  i2cData[1] = 0x00; // Disable FSYNC and set 260 Hz Acc
```

```
filtering, 256 Hz Gyro filtering, 8 KHz sampling
  i2cData[2] = 0x00; // Set Gyro Full Scale Range to ±250deg/s
  i2cData[3] = 0x00; // Set Accelerometer Full Scale Range to
±2g
  while (i2cWrite(0x19, i2cData, 4, false)); // Write to all
four registers at once
  while (i2cWrite(0x6B, 0x01, true)); // PLL with X axis
gyroscope reference and disable sleep mode

  while (i2cRead(0x75, i2cData, 1));
  if (i2cData[0] != 0x68) {
    Serial.print(F("Error reading sensor"));
    while (1);
  }

  delay(100); // Wait for sensor to stabilize

  /* Set kalman and gyro starting angle */
  while (i2cRead(0x3B, i2cData, 6));
  accX = (i2cData[0] << 8) | i2cData[1];
  accY = (i2cData[2] << 8) | i2cData[3];
  accZ = (i2cData[4] << 8) | i2cData[5];

  // It is then converted from radians to degrees
#ifdef RESTRICT_PITCH // Eq. 25 and 26
  double roll  = atan2(accY, accZ) * RAD_TO_DEG;
  double pitch = atan(-accX / sqrt(accY * accY + accZ * accZ)) *
RAD_TO_DEG;
#else // Eq. 28 and 29
  double roll  = atan(accY / sqrt(accX * accX + accZ * accZ)) *
RAD_TO_DEG;
  double pitch = atan2(-accX, accZ) * RAD_TO_DEG;
#endif

  kalmanX.setAngle(roll); // Set starting angle
  kalmanY.setAngle(pitch);
  gyroXangle = roll;
  gyroYangle = pitch;
  compAngleX = roll;
  compAngleY = pitch;
```

```
  timer = micros();
}

void sensor(){
    /* Update all the values */
  while (i2cRead(0x3B, i2cData, 14));
  accX = ((i2cData[0] << 8) | i2cData[1]);
  accY = ((i2cData[2] << 8) | i2cData[3]);
  accZ = ((i2cData[4] << 8) | i2cData[5]);
  tempRaw = (i2cData[6] << 8) | i2cData[7];
  gyroX = (i2cData[8] << 8) | i2cData[9];
  gyroY = (i2cData[10] << 8) | i2cData[11];
  gyroZ = (i2cData[12] << 8) | i2cData[13];

  double dt = (double)(micros() - timer) / 1000000; // Calculate
delta time
  timer = micros();

  // It is then converted from radians to degrees
#ifdef RESTRICT_PITCH // Eq. 25 and 26
  double roll  = atan2(accY, accZ) * RAD_TO_DEG;
  double pitch = atan(-accX / sqrt(accY * accY + accZ * accZ)) *
RAD_TO_DEG;
#else // Eq. 28 and 29
  double roll  = atan(accY / sqrt(accX * accX + accZ * accZ)) *
RAD_TO_DEG;
  double pitch = atan2(-accX, accZ) * RAD_TO_DEG;
#endif

  double gyroXrate = gyroX / 131.0; // Convert to deg/s
  double gyroYrate = gyroY / 131.0; // Convert to deg/s

#ifdef RESTRICT_PITCH
  // This fixes the transition problem when the accelerometer
angle jumps between -180 and 180 degrees
  if ((roll < -90 && kalAngleX > 90) || (roll > 90 && kalAngleX
< -90)) {
    kalmanX.setAngle(roll);
    compAngleX = roll;
```

```
    kalAngleX = roll;
    gyroXangle = roll;
  } else
    kalAngleX = kalmanX.getAngle(roll, gyroXrate, dt); // Calculate the angle using a Kalman filter

  if (abs(kalAngleX) > 90)
    gyroYrate = -gyroYrate; // Invert rate, so it fits the restriced accelerometer reading
  kalAngleY = kalmanY.getAngle(pitch, gyroYrate, dt);
#else
  // This fixes the transition problem when the accelerometer angle jumps between -180 and 180 degrees
  if ((pitch < -90 && kalAngleY > 90) || (pitch > 90 && kalAngleY < -90)) {
    kalmanY.setAngle(pitch);
    compAngleY = pitch;
    kalAngleY = pitch;
    gyroYangle = pitch;
  } else
    kalAngleY = kalmanY.getAngle(pitch, gyroYrate, dt); // Calculate the angle using a Kalman filter

  if (abs(kalAngleY) > 90)
    gyroXrate = -gyroXrate; // Invert rate, so it fits the restriced accelerometer reading
  kalAngleX = kalmanX.getAngle(roll, gyroXrate, dt); // Calculate the angle using a Kalman filter
#endif

  gyroXangle += gyroXrate * dt; // Calculate gyro angle without any filter
  gyroYangle += gyroYrate * dt;
  //gyroXangle += kalmanX.getRate() * dt; // Calculate gyro angle using the unbiased rate
  //gyroYangle += kalmanY.getRate() * dt;

  compAngleX = 0.93 * (compAngleX + gyroXrate * dt) + 0.07 * roll; // Calculate the angle using a Complimentary filter
  compAngleY = 0.93 * (compAngleY + gyroYrate * dt) + 0.07 * pitch;
```

```
  // Reset the gyro angle when it has drifted too much
  if (gyroXangle < -180 || gyroXangle > 180)
    gyroXangle = kalAngleX;
  if (gyroYangle < -180 || gyroYangle > 180)
    gyroYangle = kalAngleY;

  /* Print Data */
#if 0 // Set to 1 to activate
  Serial.print(accX); Serial.print("\t");
  Serial.print(accY); Serial.print("\t");
  Serial.print(accZ); Serial.print("\t");

  Serial.print(gyroX); Serial.print("\t");
  Serial.print(gyroY); Serial.print("\t");
  Serial.print(gyroZ); Serial.print("\t");

  Serial.print("\t");
#endif

  Serial.print(roll); Serial.print("\t");
  Serial.print(gyroXangle); Serial.print("\t");
  Serial.print(compAngleX); Serial.print("\t");
  Serial.print(kalAngleX); Serial.print("\t");

  Serial.print("\t");

  Serial.print(pitch); Serial.print("\t");
  Serial.print(gyroYangle); Serial.print("\t");
  Serial.print(compAngleY); Serial.print("\t");
  Serial.print(kalAngleY); Serial.print("\t");

#if 0 // Set to 1 to print the temperature
  Serial.print("\t");

  double temperature = (double)tempRaw / 340.0 + 36.53;
  Serial.print(temperature); Serial.print("\t");
#endif
```

```
  Serial.print("\r\n");
  delay(2);
}
```